Material Property Prediction with Transformers

University of Turku
Department of Computing
Master of Science (Tech) Thesis
TurkuNLP
December 2024
Parisa Piran

UNIVERSITY OF TURKU
Department of Computing

Parisa Piran: Material Property Prediction with Transformers

Master of Science (Tech) Thesis, 53 p.
TurkuNLP
December 2024

The Transformer neural network architecture has had a profound impact on the state of the art in machine learning in numerous disciplines, well beyond its origins in Natural Language Processing. Nevertheless, the application of Transformer models to the material field remains a relatively underexplored avenue. Therefore, we evaluated the Transformer model's capability in utilizing Many-Body Tensor Representation (MBTR) method in prediction of materials' Highest Occupied Molecular Orbital (HOMO) energy. The dataset selected for this investigation was QM9, a popular dataset that enabled us to conduct comparative analyses of our model's efficacy against a broad spectrum of prior studies. In this study, we pursued two principal approaches. Initially, we evaluated the performance of the original MBTR representation and the Transformer on the dataset, implementing only minimal modifications to both the model and the representation. Subsequently, we explored a refined MBTR variant, more suitable for the variable sequence length input of the model, which encompasses the distances between atom pairs within a molecule, alongside a reconfigured Transformer designed to integrate encoded chemical symbols of atom pairs as inputs and utilize their distances for positional embeddings. Using the two approaches, we reached the MAE of 0.123 and 0.071, respectively. We find that the Transformer model, designed to process sequential input, is capable of learning to predict from molecular representations of variable length. It outperforms the most effective kernel-based methodologies and is comparable to other recently studied deep neural networks. In conclusion, we illustrate that, with only slight adaptations, Transformers are able to make comparably accurate predictions of materials' properties.

Keywords: Transformers neural network, Many-Body Tensor Representation, Natural Language Processing, Material datasets, Material properties

Contents

List of Figures

List of Tables

1 Introduction

Transformer neural networks achieved remarkable performance in the field of sequential modeling and Natural Language Processing (NLP). Their ability to effectively model long-range dependencies makes them highly applicable across diverse areas, from speech recognition to analysing DNA sequences. Although vanilla Transformer architecture was originally introduced for language translation tasks, its variants (e.g., Vision Transformers) have since been widely adopted in various other fields [1] [2].

The Transformer architecture is particularly designed to process variable-length sequences with long-distance dependencies. Unlike conventional machine learning methods, Transformers can effectively analyse unstructured data (e.g., text input of variable length). Another common model in processing of sequential data is Recurrent Neural Network (RNN) which relies on recurrent cells to capture information. In contrast, Transformers benefit from the self-attention mechanism that captures relationships and dependencies between elements, independent of their positions in a sequence. Additionally, Transformers are able to process inputs in parallel, enabling an efficient, high-speed training. These advantages, along with their superior performance, have motivated researchers to adopt Transformer architecture to tackle a wide range of complex problems.

A recent systematic study [3] was conducted to showcase the contribution of Transformers across different fields, categorising 650 Transformer-based models in-

troduced between 2017 and 2022. The five primary categories identified were NLP-focused studies, computer vision (CV), multi-modality, audio and speech processing, and IoT and signal processing. Additionally, the study highlights Transformer applications in reinforcement learning, cloud computing, and wireless networks, proving their versatility. The paper also outlines specific considerations for adapting Transformers to model non-textual data. For instance, in image classification using a Transformer-based model, different feature representation techniques significantly impact model performance. Adjusting the Transformer architecture to address task-specific challenges is also critical. For example, in medical signal processing, Tree-Tower Transformer Network is used to predict epileptic seizures. This Transformer based model is tailored to include three distinct encoders, each capturing specifically unique signal features. In conclusion, Transformers hold significant potential for novel applications while considering the domain-specific challenges.

Material informatics, an emerging field that leverages data-driven approaches to analyse and study materials for complex, multiscale insights, faces several challenges. Techniques such as clustering similar experimental observations, training predictive models to forecast material properties, and using statistical association analysis to identify intricate patterns in material behavior are instances of machine learning and data mining application in material science [4].

Designing materials with targeted properties has been a primary goal in materials science for a long time. Traditionally, synthesizing novel materials involves experimental methods or computational simulations, both of which demand significant time and resources. However, machine learning models offer a more cost-effective and sustainable alternative, enabling the discovery of new materials by analyzing chemical structures and accurately predicting their properties [5].

In the context of this thesis, we focus on predicting the Highest Occupied Molecular Orbital (HOMO) energy level for a dataset of organic molecules. The HOMO

energy level, along with Lowest Unoccupied Molecular Orbital (LUMO), explains charge transport and indicates materials' electrical conductivity, light absorption and chemical reactivity. Accurate determination of this property is crucial in developing devices such as organic photovoltaics, organic light-emitting diodes, organic field-effect transistors, perovskite photovoltaics, and perovskite LEDs. Density Functional Theory (DFT) is a quantum mechanical method commonly used to describe the molecular electronic structure and estimate properties (e.g., HOMO energy) with reasonable precision [6]. However, data-driven approaches offer faster and more cost-effective alternatives.

In the paper [7], the authors investigated predicting HOMO energy using a Kernel Ridge Regression (KRR) model. They compared prediction accuracy across two different molecular representation techniques. Representation methods extract valuable chemical and structural information by encoding the atomic structure [8]. Therefore, their implementation is a necessary data preprocessing step in computational analysis as illustrated in Figure 1.1. Various representation approaches exist, such as Simplified Molecular Input Line Entry System (SMILES), Bag of Bonds, Coulomb matrix (CM), and Many-Body Tensor Representation (MBTR). Since selecting a suitable representation significantly impacts model performance, it is necessary to implement a representation method that describes a set of informative molecular features for the ML model. The findings in [7] demonstrated that MBTR outperformed CM when applied with KRR, yielding more accurate predictions.

MBTR is a numerical representation method that describes atomistic systems invariant to rotation, translation, and permutation of atomic indices [9]. In our thesis, it represents the spatial structure of atomic combinations in a molecule. MBTR has the advantage of converting molecules of varying sizes into a fixed-size representation, which is essential for machine learning models that require constant-length inputs [8].

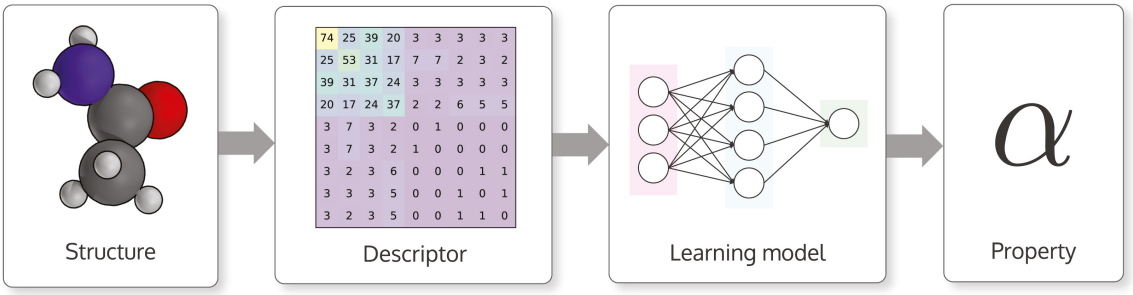


Figure 1.1: The general workflow of machine learning projects aimed at predicting material properties. The figure is reprinted from [8].

Furthermore, there have been some recent attempts to explore deployment of Large Language Models (LLMs) to advance research in material informatics [10] [11]. In these studies LLMs, particularly Transformers, are trained over extensive corpora and fine-tuned on task-specific datasets, becoming efficient tools for specialised applications. These applications include knowledge acquisition and summarisation, especially from unfamiliar or interdisciplinary fields, dataset extraction and structuring from unstructured text, feature extraction, and the development of automated laboratories with operating agents, among others. Therefore, such studies highlight the potential of LLMs as versatile, assisting tools contributing to exploration and standardisation of cross-disciplinary subjects. Nevertheless, research on adapting this advanced machine learning architecture to analyse the 3D structure of any chemical systems remains comparatively limited within this field.

In conclusion, this thesis aims to evaluate the performance of a state-of-the-art model, specifically Transformers, on prediction of quantum chemical properties. We examine the implementation of a well-known representation method, such as MBTR, when utilised with the Transformer model. A novel approach of tailoring Transformer architecture along with a modified version of MBTR is proposed, to analyse the 3D atomic structures of materials and predict their HOMO energy levels. Essentially, we address the following research questions:

1. How to implement and process the commonly used MBTR with Transformer neural network model?

2. How can the MBTR representation and Transformer architecture be adapted to ensure optimal compatibility and effectiveness in predicting materials' quantum chemical properties?

3. How effectively is the Transformer neural network able to learn from a molecular dataset, especially compared to the state-of-the-art?

4. Which hyperparameters influence the performance of Transformer models in predicting materials' quantum chemical properties?

In order to study these research questions, this thesis was structured into two major stages. In the initial stage, MBTR forms a structured set of continuous numerical features from our molecular dataset (i.e., QM9) which are then discretized and further analysed by the Transformer model. However, during the second phase, we introduced modifications to MBTR, leading to the development of a novel discrete representation called Many-Body Atom Distance (MBAD). We also implemented suitable adjustments to the Transformer architecture to evaluate its performance alongside the new representation method. Throughout both stages, hyperparameters were carefully optimised to achieve optimal results with each technique. Trained models from both stages are further compared for their property prediction accuracy on the test set. Our research framework is outlined in the Flowchart 1.2.

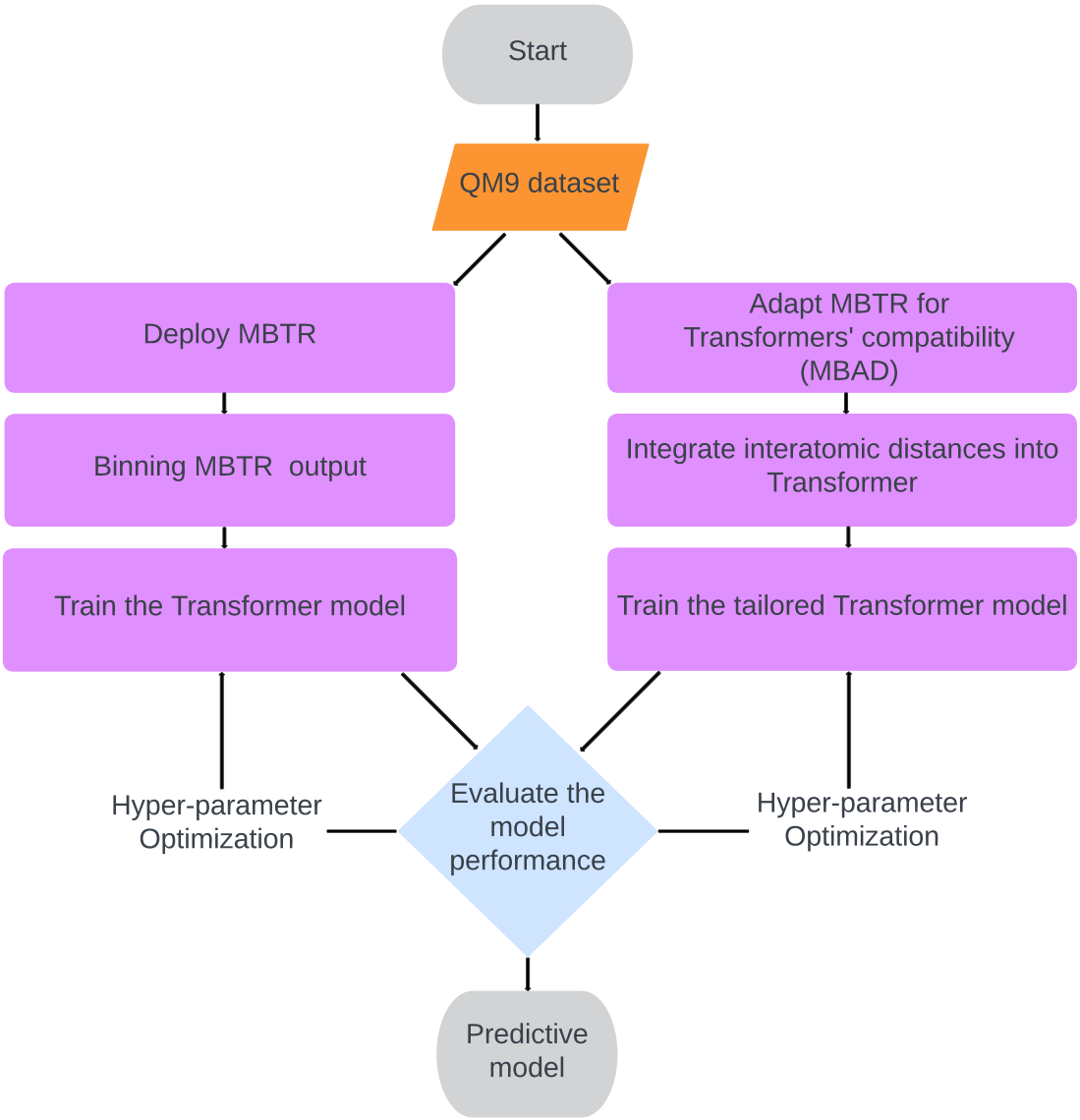


Figure 1.2: A step-by-step illustration of our research framework to achieve our research goals.

2 Methodology

2.1 Dataset

In this study, we employed Quantum Machines 9 (QM9), a well-known dataset within the realm of materials informatics. It is characterized by its extensive applications in scientific research and is particularly used for the prediction of quantum chemical properties. It provides a framework for benchmarking our findings against a substantial number of studies. This dataset consists of 133,814 stable small organic molecules with an average of approximately 18 consisting atoms, as reported in Figure 2.1. Figure 2.2 also demonstrates that the QM9 dataset is composed of 5 elements (Carbon, Hydrogen, Oxygen, Nitrogen, and Fluorine) with Hydrogen being the most common one. This dataset reports geometric, thermodynamic, electronic and energetic properties such as HOMO energy, computed at B3LYP/6-31G(2df,p) level of theory [12]. Although the QM9 dataset is considered a benchmark in the field of materials informatics, it exhibits certain limitations; notably, the reliability of the B3LYP functional in producing high-accuracy results has not been conclusively verified [13].

QM9 dataset reports molecular HOMO energy level computed using Density Functional Theory (DFT) method, as distributed in Figure 2.3. Despite the prevalent application of DFT in quantum mechanical simulations, this method is both computationally intensive and costly, which poses challenges when applied to large-

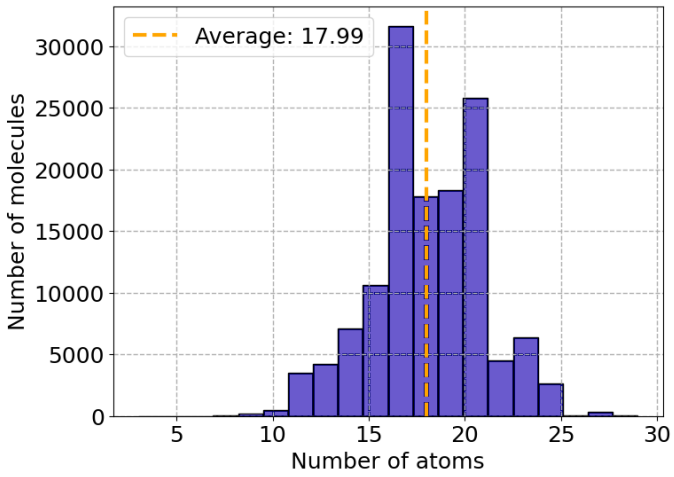


Figure 2.1: Distribution of molecule's sizes in QM9 dataset, highlighting the average size.

scale data. Therefore, data-driven alternatives are suggested presenting a more accurate estimation of quantum chemical properties [14].

We partition the dataset into segments of 80% for training, 10% testing and 10% validation. Transformer models, akin to other deep neural networks, exhibit enhanced performance when trained over large datasets; therefore, we dedicate the majority of our records to the model's training to maximise its efficiency. The validation set is used to evaluate and refine model performance by tuning the hyperparameters. Lastly, the test set is reserved for examining the model's prediction accuracy on unseen data, thereby indicating its generalization capabilities.

2.2 Molecular representation

The QM9 dataset includes atomic coordinates that define the 3D molecular structure, making it necessary to employ a suitable representation method to convert the dataset into an appropriate input for computational modelling. The transformation of spatial coordinates to relevant new features is broadly referred to as "Feature Engineering" or "Featurization" [8]. Given the range of available molecular represen-

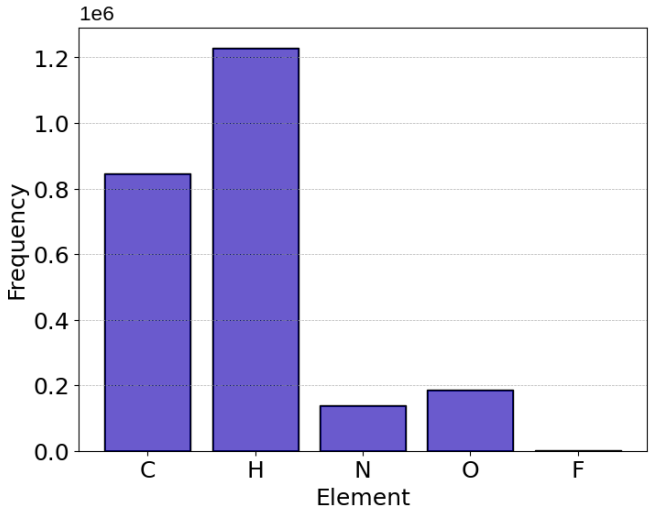


Figure 2.2: Distribution of elements among all molecules in QM9 dataset.

tations for different computational models, it is crucial to identify the most effective and contextually appropriate representation technique for each study. According to Himanen et al. [8], an effective representation should meet certain criteria, which are as follows:

1. Spatial Translation Invariance: The representation maintains consistency regardless of any translation of the coordinate system, ensuring that physical properties are not influenced by arbitrary modifications of the spatial system.

2. Rotational Invariance: The representation is isotropic, meaning any rotation of the molecular coordinates produces a constant representation.

3. Permutation Invariance of Atomic Indices: The representation is not affected by alterations in the order of atomic indices, as such changes are uninfluential on the molecular structural properties.

4. Uniqueness: The representation provides a unique mapping for each atomic structure, corresponding distinctly to the specific property value it represents.

5. Continuity: Minor variations in atomic structure are reflected in corresponding changes within the molecular representation.

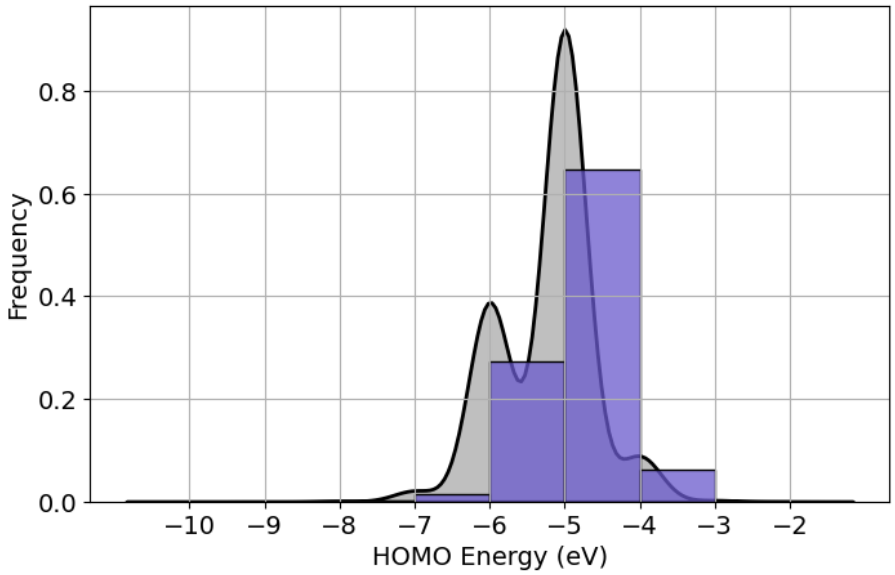


Figure 2.3: Distribution of the HOMO energy level of molecules in QM9.

6. Compactness: An efficient representation remains condensed and minimises the number of features to reduce computational load while still providing sufficient detail for accurate modelling.

7. Computational Efficiency: Although we have access to more computational resources than ever in history, calculating the descriptor should still be substantially less resource-intensive than direct computation of the physical properties it represents.

2.3 Many-body tensor representation (MBTR)

The selected representation approach for the purpose of this thesis is many-body tensor representation (MBTR), which is widely accepted in the realm of materials informatics. This advanced technique is designed to provide an informative description of the structural motifs, capturing two-body and higher-order interactions inherent within the molecular systems. MBTR employs tensors (i.e., multi-dimensional numerical arrays) to represent various atomic and molecular features. Significantly,

MBTR is effective in describing both finite systems, e.g., small organic molecules and biomolecules, and periodic systems, e.g., extended solid-state materials [9].

We utilise Dscribe library to implement MBTR and apply it to the QM9 dataset. Dscribe is an open-source software package that provides feature engineering for machine learning models in atomistic materials simulations. It includes implementations for representation methods such as Coulomb matrix, Smooth Overlap of Atomic Positions (SOAP), and MBTR, facilitating property predictions like formation energy for solids and ionic charges for atoms in organic molecules [8].

MBTR represents the molecular complex structure by breaking it into multiple-body terms (such as species, interatomic distances, bond angles, dihedral angles). Each body includes fixed-length vectors and uses a geometric function (g_k). In the scope of this thesis, we only focus on values of $k = 1, 2, 3$. One-body terms of MBTR ($k = 1$) encode the type of elements constituting the molecule. Two-body terms ($k = 2$) encode the pairwise distances between the composing atoms using Euclidean distances or inverse distances. It is worth highlighting that ($k = 2$) encode distances between all atom pairs, which is independent of the molecular bonds. Lastly, three-body terms ($k = 3$) encode the angular distributions for any triplets of atoms within the molecule. A simple illustration of how MBTR encodes the 3D structure of a water molecule is visualized in Figure 2.4 [7].

In the implementation of MBTR, while using the Dscribe library, the following geometry functions can be configured for atoms l, m and n in a molecule [8]:

- $g_1(Z_l) : Z_l$ (atomic number of l)
- $g_2(R_l, R_m) : |R_l - R_m|$ (Euclidean distance) or $\frac{1}{|R_l - R_m|}$ (inverse Euclidean distance)
- $g_3(R_l, R_m, R_n) : \angle(R_l - R_m, R_n - R_m)$ (angle) or $\cos(\angle(R_l - R_m, R_n - R_m))$

Initiating g_2 with inverse distance emphasises the influence of atoms based on

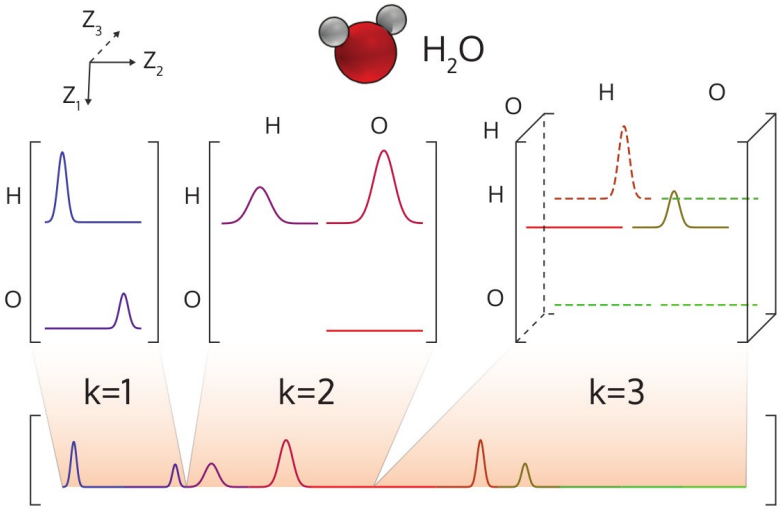


Figure 2.4: Figure by paper [8], illustrates the output of MBTR for a water molecule. The figure presents the distributions of MBTR for k values of 1, 2, and 3, encompassing various combinations of chemical elements.

their proximity. Atoms in close spatial proximity exert a more pronounced effect on the final representation, while those farther apart contribute negligibly. This approach ensures that the contributions of closer atoms are accurately reflected in the model, thereby enhancing the precision and reliability of HOMO energy level predictions.

The scalar values of g_k are broadened by applying a Gaussian function, as illustrated in Equations (1), (2), and (3). Employing Gaussian in this context is a widely used statistical technique referred to as Kernel Density Estimation (KDE). This non-parametric approach is utilised to estimate the probability density function of a random variable from a finite data sample [15]. KDE provides a smooth and continuous approximation of the underlying data distribution.

$$D_1^l(x) = \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{(x-g_1(Z_l))^2}{2\sigma_1^2}} \tag{1}$$

$$D_2^{l,m}(x) = \frac{1}{\sigma_2\sqrt{2\pi}} e^{-\frac{(x-g_2(R_l,R_m))^2}{2\sigma_2^2}} \tag{2}$$

$$D_3^{l,m,n}(x) = \frac{1}{\sigma_3\sqrt{2\pi}} e^{-\frac{(x-g_3(R_l,R_m,R_n))^2}{2\sigma_3^2}} \tag{3}$$

The variable x spans the entire range of possible values for g_k. Moreover, the parameter σ_k denotes the standard deviation of the Gaussian kernel and serves as a crucial user-defined hyperparameter in the MBTR. This parameter impacts the distribution of the k-body interactions by controlling the width of the Gaussian curves. As depicted in Figure 2.5, the selection of σ_k significantly influences the resulting representation. Specifically, larger σ_k values yield broader curves with less pronounced peaks, whereas smaller σ_k values result in narrower curves with sharper spikes. Proper tuning of σ_k is essential for optimising the MBTR's ability to effectively capture the molecule's relevant structural information for the ML models.

Furthermore, the D_k distributions for each set of k atoms in a molecule are summed, optionally using a weighting function to adjust the contributions of different atoms, as shown in the following equations:

$$\mathrm{MBTR}_1^{Z_1}(x) = \sum_{l=1}^{|Z_1|} w_1^l D_1^l(x) \tag{4}$$

$$\mathrm{MBTR}_2^{Z_1,Z_2}(x) = \sum_{l=1}^{|Z_1|}\sum_{m=1}^{|Z_2|} w_2^{l,m} D_2^{l,m}(x) \tag{5}$$

$$\mathrm{MBTR}_3^{Z_1,Z_2,Z_3}(x) = \sum_{l=1}^{|Z_1|}\sum_{m=1}^{|Z_2|}\sum_{n=1}^{|Z_3|} w_3^{l,m,n} D_3^{l,m,n}(x) \tag{6}$$

Where l, m, and n denote arbitrary atoms in a molecule, and w_k represents the chosen weighting function. Figure 2.6 illustrates how MBTR_2 encodes the D_2 distribution of all possible atom pairs in a CO_2 molecule when a unity weighting function is applied. In this figure, (a) depicts the distribution of inverse distances between all the atoms in the molecule, with the x-axis measured in 1/Å. However,

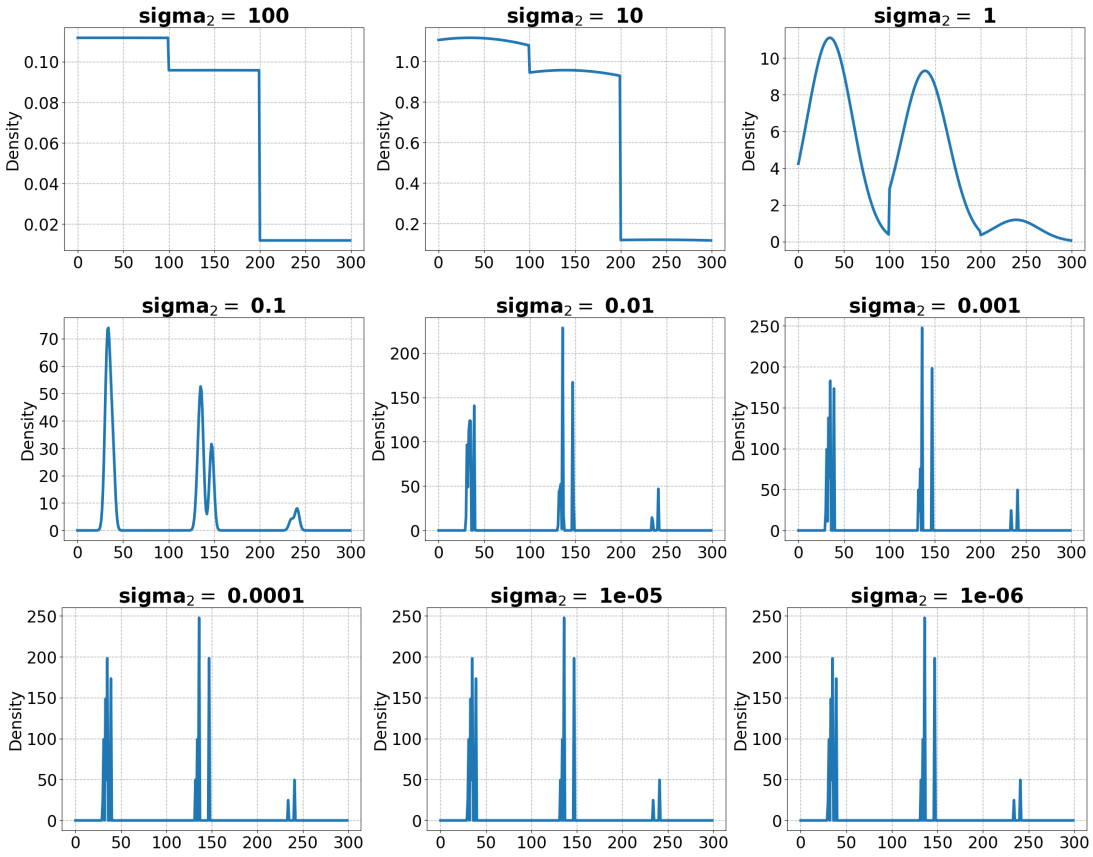
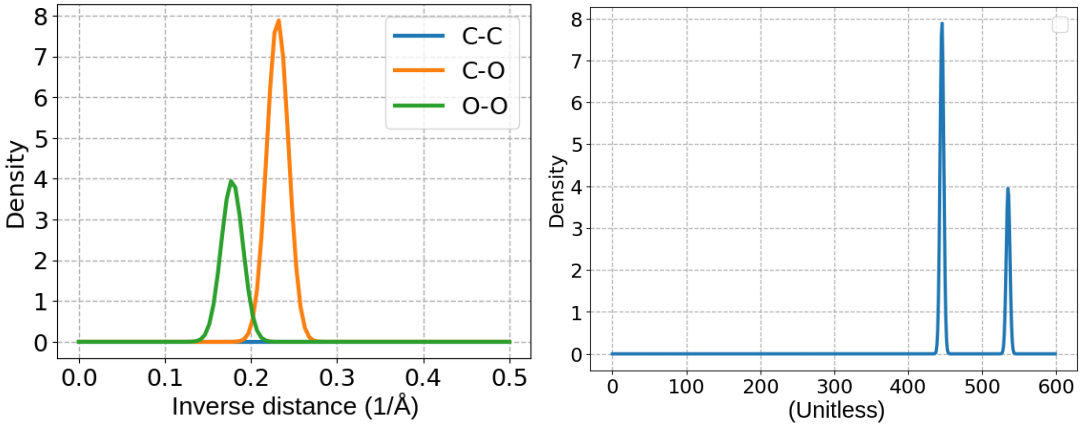


Figure 2.5: MBTR representation of a propane molecule for different values of the hyperparameter σ. Each plot demonstrates how varying σ affects the shape of the representation curves.

in (b), the concatenation of all distributions into the final MBTR_2 representation results in an x-axis that is unitless.

In this thesis, only MBTR_2 was employed, as the findings of Stuke et al. [7] indicate that including $k = 1, 3$ in the molecular representation has no significant impact on the prediction of HOMO energy when using a kernel ridge regression model. Furthermore, incorporating $k = 1, 3$ increases the size of the representation, which significantly raises the computational resources required for testing and learning from this representation by the ML model. Therefore, our model was trained exclusively on the transformed interatomic distances, as corroborated by preceding



(a) (b)

Figure 2.6: (a): The Gaussian transformation of inverse inter-atomic distances in the molecule CO_2, with the x-axis in 1/Å. Note how the pair of carbon and carbon exhibits a flat line at 0, as there is only one carbon atom in CO_2. (b): Visualization of $MBTR_2$ for the molecule CO_2, showing the influence of each atom pair on $MBTR_2$. The x-axis is unitless.

papers [7] [16]. Additionally, we utilised inverse distances to measure interatomic distances and applied a unit weighting function for simplicity and ease of interpretation. However, σ_k was treated as a hyperparameter and optimised to enhance the accuracy of the computational model.

When implementing MBTR for computational modelling, the continuous distribution representing the molecule is discretised into a grid. Sampling N_x points from the $MBTR_2$ distribution yields a representation of size $N_e^2 \times N_x$, where N_e denotes the number of molecular constituent elements [9]. The value of N_x is defined by the user during the initialisation of MBTR and determines the size of the resulting feature vector. In this thesis, N_x is optimised as a hyperparameter due to its potential impact on the model's predictive accuracy.

2.4 Data preprocessing

To develop a robust and reliable model, the molecular MBTR representation, serving as the model's input, is further preprocessed through data binning along the y-axis. This additional preprocessing step effectively reduces input noise and decreases the computational cost of model development. To evaluate the efficiency and effectiveness of this binning approach, two techniques for y-axis binning were examined:

1. Logarithmic scaling
2. Equal Width Binning

2.4.1 Logarithmic scaling

In feature engineering, logarithmic scaling or logarithmic transformation is commonly employed for heavily skewed data or long-tailed distributions. This technique compresses larger values in the dataset by reflecting their magnitude, while expanding the range of smaller values. When analysing the distribution of generated $MBTR_2$ representation values for the QM9 dataset, we observe that the representations consist of numerous small values, indicating atom pairs at large distances. If we merely truncate these values instead of scaling them, the majority would reduce to zero, leading to an unstable and uninformative representative distribution for a molecule.

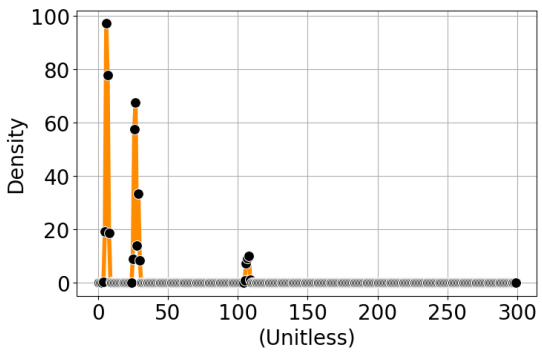
Therefore, the molecular representation values generated by MBTR are first multiplied by 10 and transformed by a logarithm to highlight even the smallest differences between values. This transformation reduces the impact of larger values, resulting in a more symmetrical and normal distribution. Additionally, by subtracting the minimum transformed value, which is always negative, from all the transformed values, we ensure that the transformed representation remains positive.

This approach not only preserves the integrity of the data but also enhances the model's capacity to learn from subtle differences. Let X denote the set of values representing a molecule by MBTR. For each $x \in X$, the transformation $T(x)$ is defined as follows:

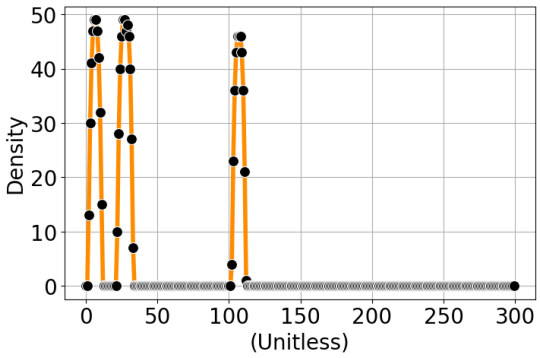
$$T(x) = \begin{cases} -\min\left(\lfloor \log_2(10X) \rfloor\right) + 1 & \text{if } x = 0 \\ 0 & \text{if } \frac{1}{10} \leq x < \frac{2}{10} \\ \lfloor \log_2(10x) \rfloor - \min\left(\lfloor \log_2(10X) \rfloor\right) + 1 & \text{if } 0 < x < \frac{1}{10} \text{ or } x > \frac{2}{10} \end{cases} \tag{2.1}$$

The purpose of this mapping function is to filter out very small values within a specific range ($\frac{1}{10} \leq x < \frac{2}{10}$). Given that the MBTR representation forms a continuous distribution, omitting some small values does not result in the loss of valuable information. However, the case where $x = 0$ should be treated as a special circumstance. This case, which indicates the absence of an atom pair in a molecule, should be represented with a unique value in the final transformation.

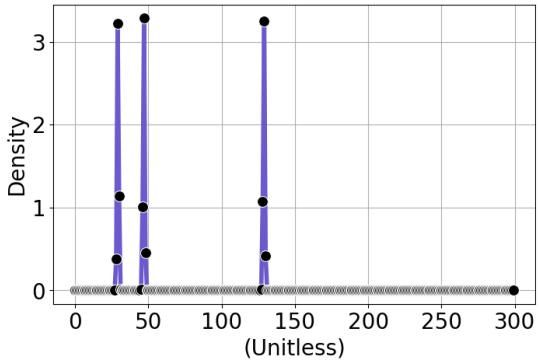
Finally, the remaining values are normalised and shifted to remain positive. In this case, the MBTR values indicate either very small interatomic distances, or very large ones, both of which have a significant influence on the HOMO energy value. This transformation maps the MBTR values to a set of discrete numbers with lower variance, resulting in a more stable representation. Figure 2.7 illustrates this binning approach for two molecules found in the QM9 dataset: Butane, a highly flammable gas with various applications, and Hydrogen Cyanide, a highly toxic compound used in manufacturing.



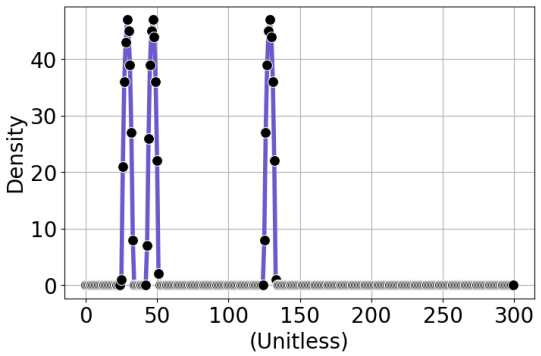
(a) MBTR representation of Butane



(b) MBTR representation of Butane after log transformation



(c) MBTR representation of Hydrogen Cyanide



(d) MBTR representation of Hydrogen Cyanide after log transformation

Figure 2.7: Transformation of the MBTR representation using logarithmic scaling. Figures (a) and (b) display the MBTR representation of butane (C_4H_{10}) before and after transformation. Similarly, Figures (c) and (d) present the MBTR output for hydrogen cyanide (HCN) before and after transformation. The logarithmic scaling preserves the original distribution while highlighting smaller spikes and reducing variance.

2.4.2 Equal-width binning

In this linear method of binning, the minimum and maximum of each molecular MBTR representation are determined. The subtraction of the minimum from the maximum yields the range of the MBTR representation values. Subsequently, this range can be divided into a specified number of non-overlapping bins. This technique, also called “data bucketing”, is used to categorize data into a limited number of bins during preprocessing phase for creating histograms. Essentially, an appropriately chosen bin size or width is crucial for binning the data into various intervals.

This allows for the counting of data points within each interval, thereby determining the frequency of occurrences in each bin.

A binned dataset is only able to represent the original dataset as long as it is determined with a rational number of bins. Therefore, number of bins could be effective in revealing the characteristics of data. Using excessively large bin numbers (more narrow bins) leads to capturing the random normal variability of the data. This phenomenon called "under smoothing" captures the noise in the dataset rather than emphasizing on the underlying distribution. In contrast, "over smoothing" is able to lower the noise and highlight the general pattern by using smaller bin numbers (wider bins). Therefore, selecting the right number of bins serves to diminish the noise within the MBTR output values, effectively grouping proximal values [17].

By translating the representation values into a finite set of categories, the model can distinguish between molecules more effectively. An excessively broad or overly limited range of model's input values would likely obfuscate the discernment of distinct inputs, limiting the model's learning predictability. There are many methods and standards for selecting the optimal number of bins; however, in this thesis, we decided to explore the impact of varying bin numbers by conducting multiple experiments. We compared the models' prediction accuracies to determine an optimal binned representation of the QM9 dataset that contains sufficient information for the model to learn.

2.5 Many-body atom distances (MBAD) representation

As previously discussed in Section 2.3, the MBTR offers a Gaussian kernel-based representation. However, Transformer architectures are traditionally designed to

process discrete inputs, such as tokens or words in natural language processing tasks. Consequently, when integrating MBTR with Transformers, it is necessary to discretise the MBTR output into non-overlapping bins as a data preprocessing step.

The MBTR method is distinguished by its continuous representation of fixed size, a feature particularly advantageous for classical models such as Kernel Ridge Regression (KRR). In contrast, Transformer models and Recurrent Neural Networks (RNNs) are designed to process sequential data of variable lengths without fixed size constraints. RNNs achieve this by maintaining hidden states across sequences, while Transformers utilise self-attention mechanisms to dynamically focus on different parts of the input sequence, enabling efficient handling and analysis of diverse molecular lengths.

Given the mentioned inherent Transformers' attributes, it is advantageous to employ a discrete representation of molecules with no constraint regarding the outputs' length. This ensures the compatibility of the representation method with the Transformer's input processing mechanism, allowing the model to effectively analyse and interpret the molecular data. The contradiction of the MBTR's and Transformer's innate unique characteristics inspired us to propose a modification to the MBTR structure. We adopted the MBTR to suit the Transformers architecture the best. Our aim was to make direct comparisons between conventional models that rely on original MBTR and Transformers using tailored MBTR for a prediction task. The set of experiments in this thesis ensures a fair comparison between such.

Considering the inherent properties of Transformers, it is advantageous to employ a discrete representation of molecules without constraints on output length. This ensures compatibility with the Transformer's input processing mechanism, allowing the model to effectively analyse and interpret molecular data. The contrasting characteristics of MBTR and Transformer architectures prompted us to propose a modification to the MBTR structure. We adapted MBTR to better suit the

Transformer architecture. Our objective was to facilitate direct comparisons between conventional models which utilise the original MBTR and Transformers using the tailored MBTR for predictive modeling. The experiments conducted in this thesis ensure a fair comparison between these two approaches.

Our proposed adaptation, Many Body Atom Distances (MBAD), exclusively utilises K_2, in line with the choice made in Section 2.3. The principal difference in this new approach lies in the omission of the Gaussian function, which is intended to smooth the interatomic distances. This alteration permits the direct extraction of Euclidean distances from MBTR and focuses on the comparison of all interatomic distances within a given molecule. Given the relatively small size of the organic molecules in the QM9 dataset, the use of a cutoff distance, which limits the range of interatomic distances considered, could result in the loss of valuable structural information. By including all interatomic distances without a cutoff, we ensure a comprehensive representation of the molecular structure, capturing subtle variations that are crucial for accurate modeling and analysis. This approach maximises the extraction of pertinent information, thereby enhancing the performance and reliability of the predictive models.

MBAD, in comparison to MBTR, results in representations of variable lengths for each molecule in the dataset. Nevertheless, it remains necessary to categorize these interatomic distances into segments. This categorization is essential as it transforms the continuous real-value distance data into discrete integer intervals, making it suitable for Transformer processing. In Sections 2.4.1 and 2.4.2 of our previous experiments, we tried different binning methods. Given the additional complexities and interpretative constraints imposed by logarithmic binning, we decided to bin the MBAD representation using only the equal length binning method. This process involves segregating all interatomic distances extracted by MBAD from the longest to the shortest into bins of equal lengths, thereby facilitating the analysis of

discretised inputs suitable for Transformer model processing.

2.6 Transformer neural networks

Transformers represent a groundbreaking class of neural networks that have significantly advanced artificial intelligence. Initially conceived for tasks in natural language processing, their primary function is to transform input sequences into corresponding output sequences. A quintessential application of Transformers is in machine translation, where they can convert a series of sentences from one language into another.

Notably, Transformers have exhibited substantial improvements in performance metrics over some older models, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. These earlier models are particularly designed for processing sequential data that exhibits inherent interdependencies, such as those found in linguistic data where the meaning of each word or phrase can depend significantly on its context within a sentence or a broader text corpus. They analyse elements one at a time. This sequential processing involves maintaining a "hidden state"—essentially a form of short-term memory—that retains and updates information about the input processed up to each step.

Due to their recurrent nature, RNNs and LSTMs take one input at a time, updating the hidden state based on the current input and the previous state. This design inherently ties each step's output to its priorly seen inputs, leading to challenges in learning long-range dependencies within the data. These challenges are exemplified by issues such as vanishing gradients, where the gradient signal becomes too small to make meaningful updates, and exploding gradients, where overly large gradient values can lead to unstable training processes. Such issues often result in the network failing to retain earlier learned information effectively. In contrast, Transformers revolutionize this approach by utilizing an attention mechanism and

by bypassing the need for a memory state that evolves over time, which eliminates the input-output dependencies. Consequently, this new architecture supports more robust learning, especially in scenarios involving long input sequences.

Furthermore, the attention mechanism central to Transformers provides a significant computational advantage by enabling parallel processing of inputs. Unlike RNNs, which must process data sequentially, Transformers can handle multiple data points simultaneously, drastically enhancing computational efficiency. This capability allows Transformers not only to process larger datasets more effectively but also to scale up with increased computational resources, leading to superior performance across numerous benchmarks.

The foundational architecture of the Transformer was first introduced in the seminal paper "Attention is All You Need" [18]. This architecture, as illustrated in Figure 2.8, comprises two main components: an encoder and a decoder. This dual-component structure has laid the groundwork for subsequent state-of-the-art models such as Bidirectional Encoder Representations from Transformers (BERT) [19] and Generative Pre-Trained Transformers (GPT) [20]. BERT, initially developed by Google, employs only the encoder portion of the original Transformer architecture. In contrast, variations such as the GPT, developed by OpenAI, utilise solely the decoder component. Each of these components—encoder and decoder—is further subdivided into multiple layers, the specifics of which are elaborated upon in the subsequent subsection.

2.6.1 Tokenization

Tokenization in NLP is a necessary data preparation method in which larger texts are segmented into smaller units such as phrases, words, or even characters. This method contributes to simplification and standardising of human language to make it more manageable for language models. Such models typically perform more ef-

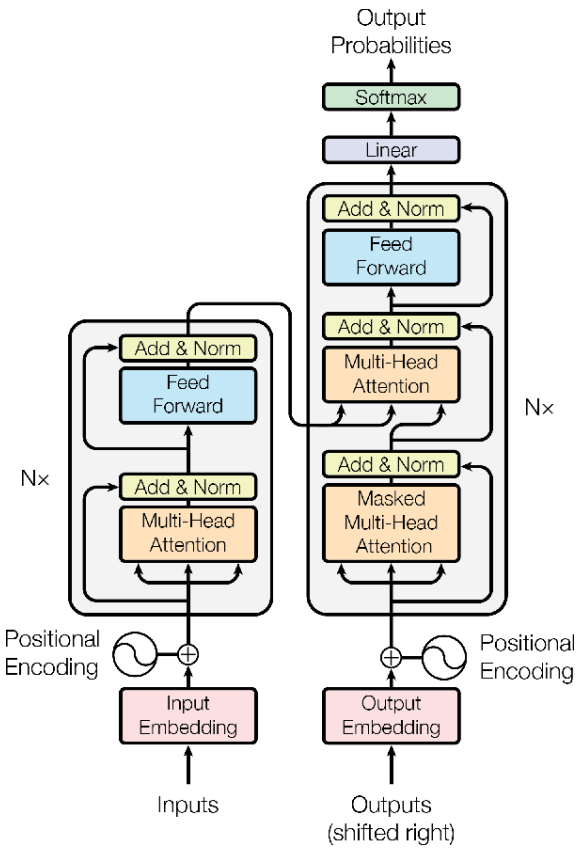


Figure 2.8: The dual-component structure of Transformers, consisting of an encoder and a decoder. The figure is reprinted from "Attention is All You Need" by Vaswani et al. [18].

ficiently with smaller units because they can analyse and interpret the information they contain more effectively. The size of tokens and the tokenization method vary depending on the knowledge we strategically intend to extract.

2.6.2 Padding

Padding is another critical technique in the field of NLP. Padding tokens refer to the process of adding a specified “padding token” to the beginning, end, or both ends of a token sequence to achieve uniformity in length among multiple sequences. The padding token does not have any particular meaning and only ensures to equalise the sequences’ lengths. In larger datasets, sequences within each batch can be padded to

the length of the longest sequence in that batch. This method of dynamic padding, considering the variability in sequence lengths, can improve memory efficiency while speeding up the learning process. As a result of dynamic padding, the language model receives a fixed-size and properly structured input that is ready for further processing.

2.6.3 Embeddings

Embeddings are one of the fundamental concepts in machine learning, especially NLP. Embeddings represent complex data structures such as images, audio, and text. Such objects are translated by embeddings into a lower dimensional space while preserving their relevant properties. Translated embeddings are able to maintain a relationship to the original data, presenting its traits and remarks. In NLP, a high-dimensional numerical embedding represents each token. Tokens are transformed into vectors such that tokens with similar meanings are located close to each other in the embedding space. Moreover, these embeddings contain information such as the token's semantic meaning, syntactic information, and contextual meaning.

Embedding size or dimension plays an important role in determining both the model's capacity and performance. Larger embeddings can capture more nuanced and detailed information about each token. They can encode a greater variety of syntactic and semantic features leading to a deeper understanding of language. In contrast, smaller embeddings are less capable of capturing data's complex features but they require less memory. Therefore, embedding dimension in a model is a deterministic hyperparameter which needs to be fine-tuned based on the data complexity and the model design. In the seminal paper introducing the Transformer architecture, the authors specified an embedding dimension of 512 [18]. However, subsequent advancements in model capabilities, particularly in versions of OpenAI's GPT-3, have seen embedding dimensions expand significantly, reaching up to 12,288

in the most advanced configurations [21].

2.6.4 Encoder

The encoder is a crucial component of the Transformer architecture, primarily tasked with creating a continuous representation of the input sequence. As stated in [the original paper], the encoder includes six layers, each identical in structure. The main feature of each layer is a self-attention sub-layer that focuses on different parts of the input sequence based on their relevance to each other. This mechanism is contemplated in the following subsection. Next in each layer, there is a sub-layer of the position-wise feed-forward network (FFN), consisting of two linear transformations. These transformations are applied to the positions of input elements independently, yet each position undergoes the same transformation, using the same weights and biases.

Figure 2.8 demonstrates that each two encoder's sub-layers—both self-attention and FFN—is wrapped by a residual connection followed by layer normalisation. Thus, the output from each sub-layer is calculated as $\text{LayerNorm}(x + \text{Sublayer}(x))$, where $\text{Sublayer}(x)$ represents the specific operation carried out by the sub-layer. The residual connection and the normalisation plays an important role in stabilizing the learning process and stimulating the model's training by combining the input and output of each layer, thereby limiting problems such as vanishing gradients.

2.6.5 Decoder

The decoder consists of six stacked layers, commencing by receiving the previous decoder's output. Similar to the encoder, the decoder contains a layer of self-attention at the very bottom. This sub-layer of self-attention allows Transformers to dynamically attend to different parts of the output. However, there is a significant difference between the encoder's self-attention sub-layer and the decoder's self-attention sub-

layer. The self-attention in the decoder uses masking to prevent a position from being influenced by subsequent positions. This ensures that the prediction of a position in the output sequence depends only on the knowledge available to the transformer up to that position.

As illustrated in Figure 2.8 (right side), the transformer decoder also includes an encoder-decoder attention (cross-attention) sub-layer. This sub-layer facilitates the alignment of the encoder's output with the output from the previous self-attention sub-layer in the decoder. Consequently, the outputs of the encoder and the decoder are combined, allowing the most relevant parts of the input sequence to influence the prediction of each position in the output sequence. Finally, each sub-layer in the transformer is wrapped with a residual connection followed by layer normalisation, mirroring the structure employed in the encoder.

2.6.6 Multi-headed self-attention mechanism

The self-attention mechanism is essential to the Transformers architecture as it allows these models a nuanced contextual understanding of the input sequence. This mechanism concentrates on a sequence's component dynamically, adjusting its focus based on the surrounding context. It is analogous to how humans are able to selectively focus on a particular part of a text. The attention function's input consists of query and key vectors, each with a dimension of d_k, and value vectors of dimension d_v. These vectors are produced through linear transformations of the input sequence. Practically this function operates simultaneously on a set of vectors compacted into matrices. The attention function is mathematically represented as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.2}$$

Where Q, K, and V are matrices of a set of queries, keys, and values respectively. Additionally, the Transformer architecture employs multi-head self-attention,

which consists of h layer of individual self-attention functions. Each layer performs the attention function over a different projection of the input value, yielding d_v-dimensional outputs. Finally, as illustrated in Figure 2.9, the output from all layers are concatenated and linearly projected, resulting in the final output values. This innovative approach allows the Transformer to simultaneously address information from distinct inputs representations at different positions.

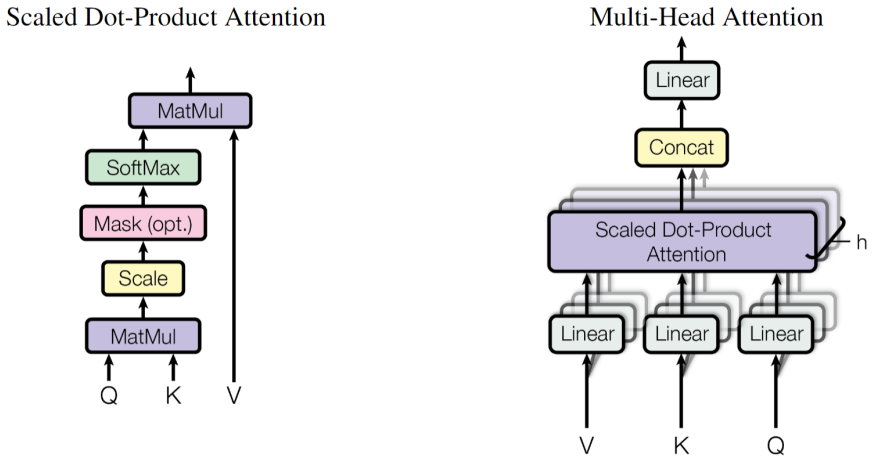


Figure 2.9: Illustrating the attention mechanism's core (on the left) and the multi-head attention mechanism (on the right). The figure is reprinted from "Attention is All You Need" by Vaswani et al. [18].

2.7 Positional encoding

Transformers are specially structured to process sequential data following the prior similar models such as RNNs. Such models are equipped and adapted carefully to consider the order of the data points when generating the outputs. Although in the sequential data, the inherent order of the input elements is pivotal to their interpretation, the attention mechanism is naturally permutation invariant. In Transformers architecture, the order of entries are introduced by a smart solution called positional encoding. The main purpose of positional encoding is to inject positional informa-

tion into these models.

To elucidate further, positional embeddings are vectors of the same size as the input token embeddings. Therefore, they can be summed with the token embeddings at the initial steps of the decoder or encoder stack. Adding the positional encoding to the input embeddings ensures that the model learns not only the significance of each token but also its position in the sequence. There are multiple options for encoding functions which must be chosen based on the input data and the project. The one mentioned below was introduced by the original paper [18]:

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.3}$$

$$\mathrm{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.4}$$

Where pos is the position of the token in the sequence, i is the dimension index, and d_{model} is the dimensionality of the token embeddings. The use of sinusoidal functions helps the model easily learn to attend by relative positions since for any fixed offset k, $\mathrm{PE}(pos_k)$ can be represented as a linear function of $\mathrm{PE}(pos)$.

2.8 MBAD and Transformers integration

MBAD was designed to output the molecular interatomic distances and the elements constituting each molecule. This design of MBAD geared us towards adapting the NLP techniques and Transformers architecture to suit our research goal the most. Inspired by the tokenization technique, we map each pair of composing elements using the formula shown in Equation 2.5, where Z_1 and Z_2 are the atomic numbers of the first and second elements, respectively. Coefficient 118, which is the number of chemical elements in the periodic table, ensures an injective mapping. Considering how MBAD is an invariant representation, same as MBTR, it preserves an order when calculating the distances of neighbouring atoms in a molecule. Therefore, the

mapping of each molecule certainly remains similar, even under molecular rotations.

$$T(Z_1, Z_2) = (118 \cdot Z_1) + Z_2 \tag{2.5}$$

In order to benefit from the Transformer technology in the field of material science and property prediction, a few adjustments were made to the model. We incorporated the numerically encoded atoms in a molecule (Figure 2.10) as the model's input. Meanwhile, the inter-atomic distances extracted by the MBTR representation integrate into the model as positional embeddings. The inter-atomic distances are transformed to an embedding by a randomly initialised neural network. Furthermore, the resulting mapped embeddings is incorporated into the model architecture as positional embeddings, where it is added to the input embeddings.

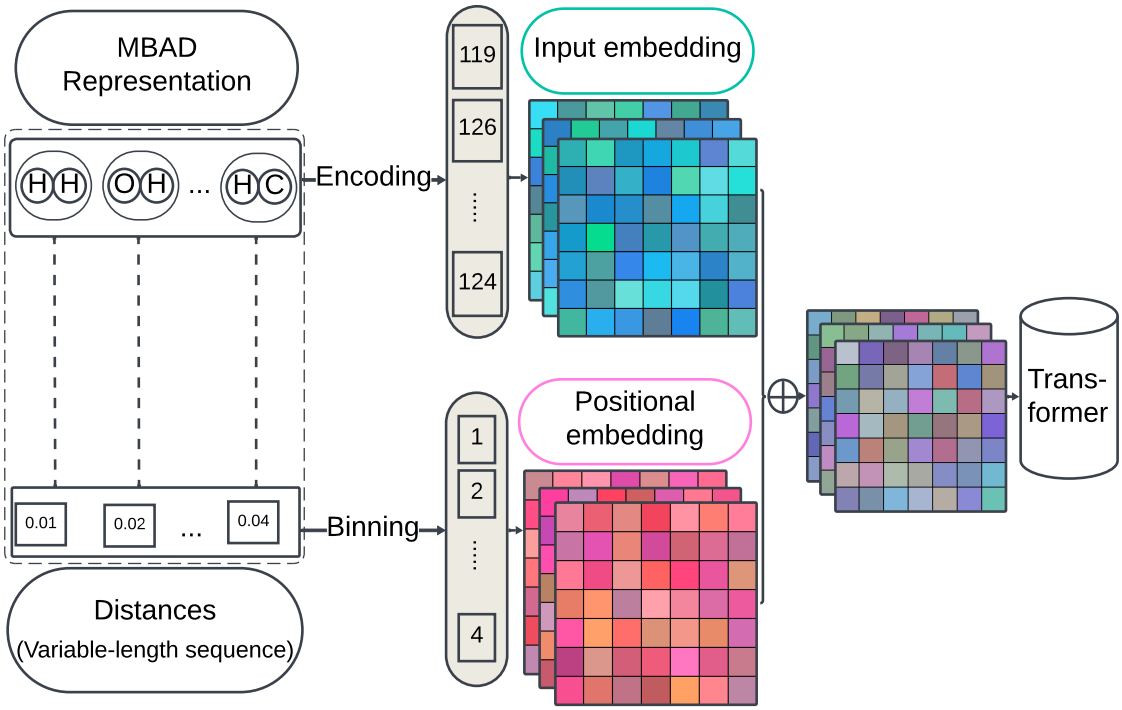


Figure 2.10: Transformers architecture is adapted to be most suited for analyzing the interatomic distances

2.9 Evaluation metrics

There are metrics for evaluating the model's performance. A suitable metric is essential to the model's learning. Moreover, a common metric allows us to challenge and assess how accurately a model predicts the target value compared to other available techniques and models. Numerous metrics are popular for evaluation, each providing unique insights.

2.9.1 Mean squared error

Mean Square Error (MSE) is a loss function which is used to penalise the model corresponding to the margin between the predicted and the true value. MSE highlights the difference between targeted and predicted values more pronouncedly, by squaring their subtraction. It also normalises the error against the number of the observations in the dataset. The mathematical formula to calculate the MSE is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{2.6}$$

Where the n is the number of data points, $\hat{y}_i$ is the predicted value by the model and y_i is the true target.

2.9.2 Mean absolute error

Mean Absolute Error (MAE) is another loss function. It measures the average magnitude absolute difference between the target and the predicted values. It is defined mathematically by the following formula:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{2.7}$$

Where the n denotes the number of observations, y_i the true value and $\hat{y}_i$ the

predicted value. Unlike MSE, MAE is less sensitive to outliers because it employs a linear, rather than quadratic, penalty for errors.

2.9.3 R-squared score

The R-squared score (R^2-score) measures the proportion of variance in the dependent variable that is predictable from the independent variables. In other words, the R^2-score evaluates how well the model is able to predict the target based on the observations. It ranges from 0 to 1 and is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.8}$$

where $\bar{y}$ represents the mean of the true values, y_i the true values, $\hat{y}_i$ the predicted values by the model, and n the number of observations. In this thesis, we report R^2-scores of trained models on the test set, where a higher score indicating a better fit of the model to the data.

2.9.4 Applying evaluation metrics

In practice, MSE, MAE, and R^2-score are used to evaluate the Transformer model's performance on the validation and test sets. By comparing these evaluation metrics, we gain insights into how well the model generalises to unseen data and how accurately it is able to predict. Furthermore, our set of evaluation methods is aligned with reported metrics of other research, effectively serving as a standard in the relevant field. This standardisation facilitates benchmarking and ensures consistency in assessing model performance across different studies and applications.

3 Model Evaluation Results

In order to discuss the research goals of this thesis, we analyse and compare the results of different models. Each model is trained on the training set and the reported performance metrics are evaluated over the test set.

3.1 Analysis of MBTR

MBTR representation is a common tool for representing molecular structures in various computational analyses. It describes molecules' many-body interactions as fixed-size continuous representations, providing machine learning models with novel structured features. We exclusively encoded the molecular interatomic distances for the Transformer neural network to predict the HOMO energy. Figure 3.1 illustrates the Transformer model's learning curve with a decaying learning rate, plotted by measuring the model's loss at each epoch. It also compares the model's loss on the training set against the evaluation one. The loss function is plotted on a logarithmic scale to visualise the nuanced changes in the model's learning more evidently.

As demonstrated in Figure 3.1, after inputting the processed MBTR representation to the Transformer neural network, the model's weights are initialised randomly. Therefore, the model begins with a high loss and error rate, which decrease as the model starts learning from the represented molecules until around epoch 70. Since this epoch, minimal improvement can be observed in the evaluation loss. Meanwhile, the training loss falls dramatically after epoch 70, as the model starts memorising

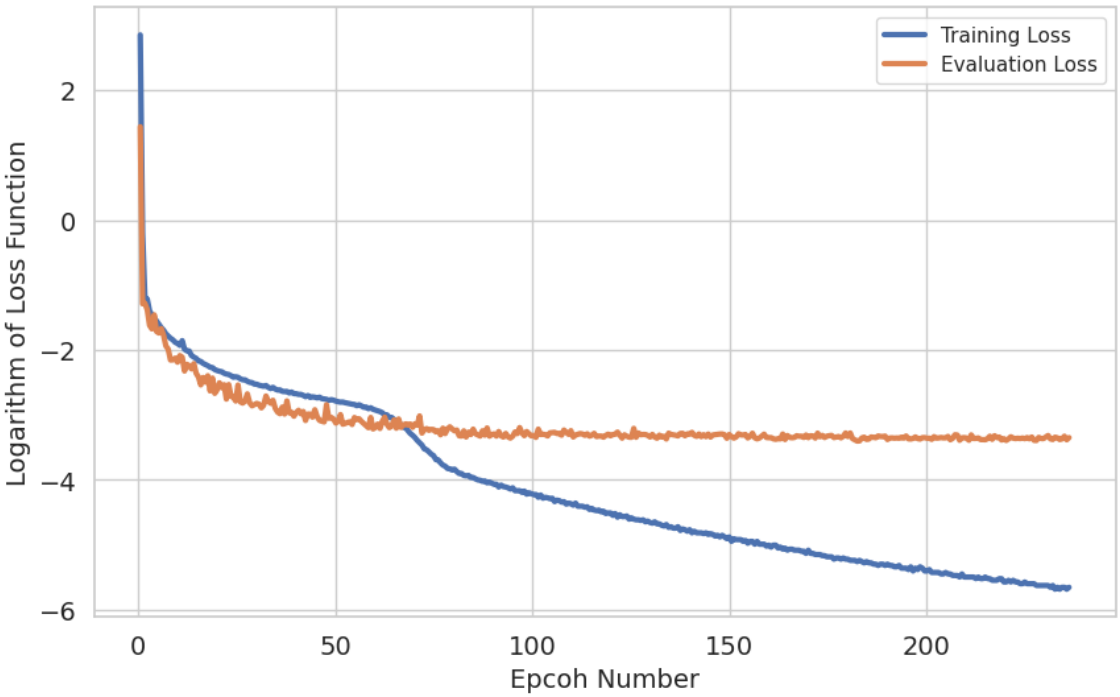


Figure 3.1: Training and evaluation of the Transformer neural network on MBTR representation of QM9 dataset. Around epoch 70, the model begins to overfit over the training set of data which barely improves the prediction accuracy on the evaluation set.

the entries and consequently overfitting on the training dataset.

3.1.1 Learning rate

In Figure 3.2 , we examine the effect of different learning rates and the convergence speed during training. Utilising the proper learning rates impacts the learning process evidently. As demonstrated in this figure, with larger learning rates (e.g., 0.001), the model misses the loss function's local minima, hence the learning curve is relatively flat. Conversely, opting for a smaller learning rates (e.g., $1e-6$) leads to slow learning and increases the probability of getting stuck in a local optimum. This observation led us to conclude that we could benefit from decay learning rate to adjust dynamically during the training process.

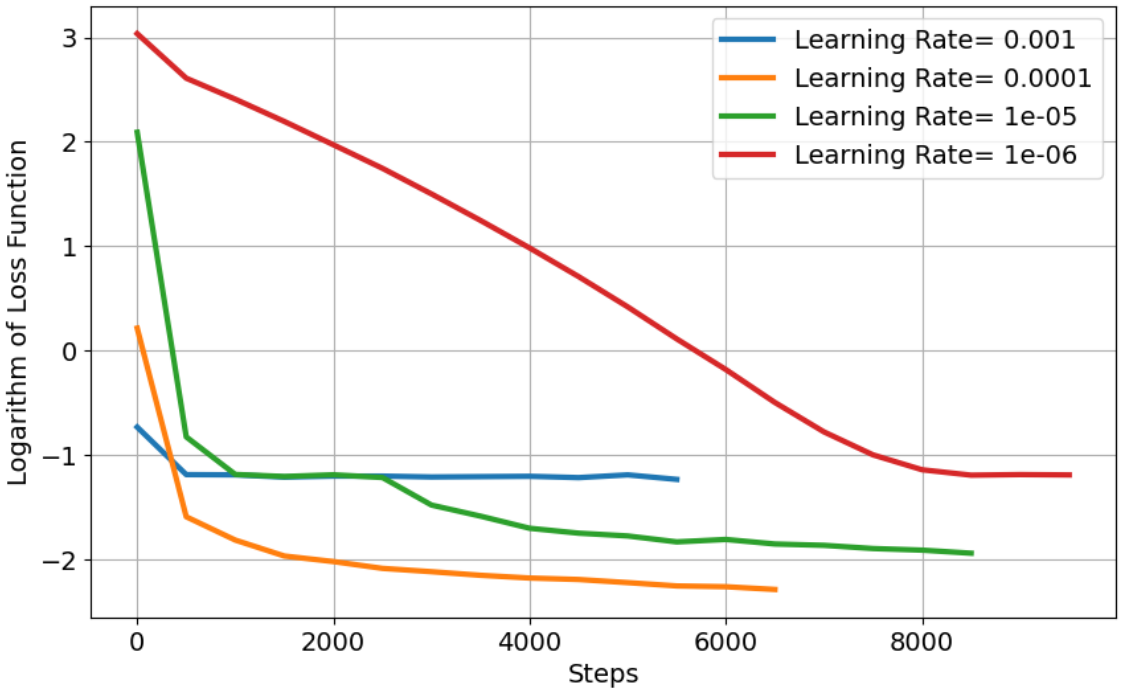


Figure 3.2: Logarithm of loss function of the Transformer model while training on the dataset represented by MBTR with different learning rates (ranging from 1e-3 to 1e-6) and a constant batch size of 64.

3.1.2 Sigma

Although in machine learning models there are a number of hyperparameters that require to be fine-tuned, there are also some introduced by the MBTR. There are 19 hyperparameters affecting MBTR representation when initialisation [22]. One of these hyperparameters is sigma, as we stated in Section 2.3. In MBTR, sigma (σ) refers to a coefficient in the Gaussian function that determines the smoothness or narrowness of the features when representing a molecule.

Reported in Table 3.1, we compare the MAE across different Transformer models after being represented by MBTRs of different sigma settings (i.e., ranging from 100 to $1e-6$). The lowest MAE, indicating the best model performance, is observed at $\sigma = 0.1$ with an MAE of 0.1829 and an R^2-score of 0.7884. Conversely, the worst performance corresponds to $\sigma = 1e-6$ and $\sigma = 100$, yielding an MAE of 0.3551 and

0.3719 respectively. In conclusion, this comparison demonstrates the critical role of sigma in determining the quality of the information captured by the representation method. If sigma is too large or too small, the model is unable to effectively learn the structural features of the molecules.

Table 3.1: Transformer model performance in predicting HOMO energy across MBTRs with different sigma initialisations. Optimal result is characterised by a low MAE and a high R^2-score.

Sigma	**MAE**	R^2**-score**
100	0.3719	0.0445
10	0.3551	0.1661
1	0.2592	0.5565
0.1	**0.1829**	**0.7884**
0.01	0.2055	0.7431
0.001	0.2003	0.7274
$1e-4$	0.2083	0.7069
$1e-5$	0.2282	0.6671
$1e-6$	0.3551	0.1661

3.1.3 Comparative discretisation methods

During the data preprocessing, stated in section 2.4, we proposed two methods to discretise the MBTR representation outputs, preparatory to entering the Transformer model. One of the proposed binning methods is to map the continuous values generated by MBTR to a set of equal length intervals. As a result, the interval indices, transformed discrete integers, are structured and preprocessed for input into the Transformer neural network.

Although the equal width binning technique is straightforward and increases the interpretability, it introduces a new hyperparameter which requires optimisation. In this method, the selection of the right number of bins obviously determines the

effectiveness of our data preparation. Therefore, we examined the impact of different binning numbers on the model performance. The results are listed in Table 3.2.

Table 3.2: Comparison of MAE in HOMO energy predictions by Transformer models trained with segmented MBTR representations into N equal-length bins (lower MAE is the indication of better performance).

Number of bins N	**MAE**
10	0.3471
500	0.1383
1000	**0.1273**
2000	**0.1275**
10000	0.1742

As highlighted in Table 3.2, $N = 1000$ and $N = 2000$ are the optimal numbers of bins when discretising the molecular representation by MBTR. If we employ a smaller number of larger-sized buckets (e.g., 10 bins), the transformation causes a significant loss of information, which leads to a larger error margin in the predictions. Moreover, using an excessively large number of bins (e.g., 10000 bins) when implementing the technique can counteract its intended purpose of denoising and data processing.

An alternative approach for binning the MBTR output is the application of logarithmic scaling, as introduced in Equation 2.1. This method is designed to limit noise while preserving both zero values and spikes in the representations. This method maps the represented values to a finite set of integers without the need for any hyperparameters, thereby eliminating the necessity for further optimisation. The results of the two discretisation methods are compared in Table 3.3, where key machine learning model settings, e.g., batch size, learning rate, were kept constant to ensure a fair comparison. Based on the results, we conclude that both binning methods are equally effective, each demonstrating unique strengths.

Table 3.3: Comparison of accuracy in HOMO energy predictions by Transformer models trained with different methods of discretising the MBTR representation (lower MAE and higher R^2-score indicate better performance).

Binning method	**MAE**	R^2**-score**
Logarithmic scaling	0.1245	0.88
Equal Width	0.1274	0.87

3.1.4 MBTR size

As discussed in Section 2.3, when encoding interatomic distances in a molecule using MBTR, the size of the representation depends on both the number of unique elements in the molecule and the number of sampling points used to discretise the x-axis. The MBTR transforms continuous distributions of distances into fixed-size vectors by sampling from the distribution. The number of sampling points is a hyperparameter that requires fine-tuning when configuring the MBTR settings, as it affects the resolution of the representation and the computational cost. In Table 3.4, we evaluate the Transformer model's performance using MBTR representations with varying numbers of sampling points.

Table 3.4 presents the MAE of Transformer models trained on MBTR representations of varying sizes. The number of sampling points in these representations ranges from 20 to 80, resulting in overall representation vectors of sizes between 300 and 1200. Given that the Transformer models were trained using NVIDIA GPUs, it is essential to consider the trade-off between input sequence length and batch size. An increase in the representation size requires a corresponding decrease in the batch size due to memory limitations. We utilised two NVIDIA A100 GPUs with a total of 80GB of GPU memory, which dictated the maximum feasible batch size, as indicated in Table 3.4.

Table 3.4 proves that sampling the distance between each combination of atom pairs 20, 30, and 40 times results in optimal model performance. That is, represen-

Table 3.4: Performance of Transformer models trained on MBTR representations with varying sizes. A lower MAE indicates better predictive performance of the models. As the representation size increases, the batch size is reduced due to GPU memory limitations.

Sampling number	**MBTR vector size**	**Batch size**	**MAE**
20	300	**128**	**0.1273**
30	450	**128**	**0.1299**
40	600	**80**	**0.1264**
50	750	50	0.1349
60	900	40	0.1451
70	1050	40	0.1452
80	1200	40	0.1316

tation lists of lengths 300, 450, and 600, respectively, provide sufficient molecular structure information for effective model learning. Moreover, smaller representation lists allow for the use of larger batch sizes, which, in turn, improves the training speed of the model.

3.2 Analysis of MBAD and model adaptation

Following the implementation of the original MBTR, a continuous fixed-size representation tool, we determined that adjustments to the representation method were necessary, given the inherent structure of Transformers. Transformer input typically consists of tokenized sequences of variable length. As such, it is essential to adapt our representation tool to align with the input processing constraints of Transformers. Therefore, in Section 2.5, we introduced MBAD, a variable-length representation tool to encode the molecular interatomic distance.

Considering the Transformer architecture as introduced in Section 2.6, we incorporated key components, e.g., positional embedding, to address the specific chal-

lenges of our project. As was stated in Section 2.8, chemical elements in the MBAD representation output are encoded in pairs and subsequently fed into the model. In this procedure, the interatomic distances captured by MBAD are discretised into bins and incorporated as positional encodings. Subsequently, the positional encodings and the input are mapped into embeddings and aggregated through summation. This tailored model effectively integrates all molecular structural information to predict the HOMO energy robustly. Figure 3.3 presents the performance of such model over the training and evaluation set by tracking the logarithm of the loss function after each epoch.

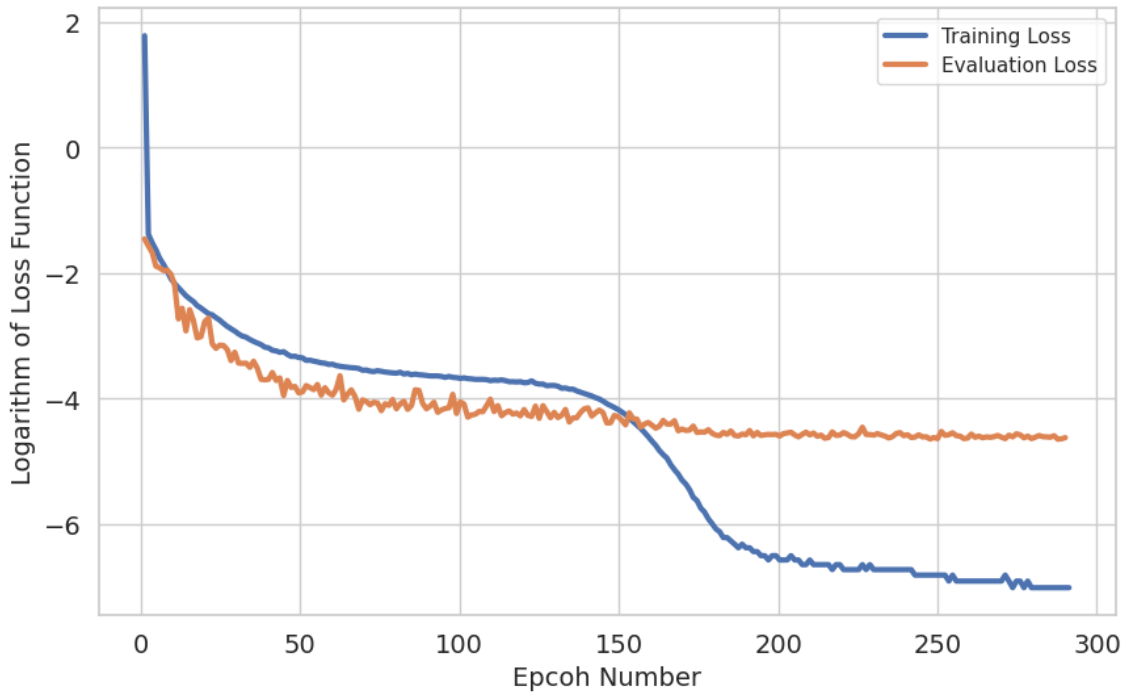


Figure 3.3: Learning curve of Transformer-based model trained over MBAD representation of the molecular data, depicting the logarithm of the training and evaluation loss at each epoch.

3.2.1 Discretising MBAD

MBAD representation output, specifically the encoded distances between every atom pair in a molecule, consists of sequential continuous values. Consequently, it requires discretisation similar to the process applied to the MBTR output. Although two different discretisation methods were introduced in Section 2.4, both demonstrated comparable efficacy when examining their effect on using MBTR representation with the machine learning model (reported in Table 3.3). Thus, the equal-width binning method was selected for discretising the MBAD representation output to further preprocess the data. This approach is straightforward and interpretable, maintaining relevant feature information without compromising the model's generalisation ability, assuming that the number of buckets is properly optimised. Table 3.5 presents an analysis of how varying the number of bins affects the Transformer's predictive performance when applied to the MBAD representation.

Table 3.5: MAE and R^2-score of model predicting the HOMO energy molecular dataset represented by MBAD and discretised into N bins.

Number of bins (N)	**MAE**	**R^2-score**
5	0.15	0.84
10	0.11	0.91
50	0.086	0.94
80	**0.0847**	**0.95**
100	**0.0847**	**0.95**
500	0.086	0.94
1000	0.096	0.92
2000	0.099	0.92
5000	0.12	0.88
10000	0.16	0.79

Table 3.5 reports the impact of segmenting the data into varying numbers of

intervals (N) on the model's performance in predicting the target variable after 100 epochs. Specifically, when N is too small, the model is barely able to capture nuanced relationships within the data. Conversely, an excessively large number of intervals (e.g., $N = 10000$) introduces a substantial amount of noise, diluting meaningful patterns and thereby degrading the model's predictive accuracy. Therefore, Table 3.5 proves that an optimal balance is achieved when the MBAD representation is divided into approximately 80 to 100 bins.

3.2.2 Hyperparameter optimisation

In this section, we investigate the complex relation of the hyperparameters and the model's accuracy in performing the prediction task. The following variables were studied:

- Embedding size tested over the range of [24,144,456,768,1200,1800], reflecting on the size of the model.
- Batch size with a testing range of $[24, 36, 64, 128, 512, 1024]$ which initiates the number of samples the model processes at a step.
- Dropout rate examined at values [0,0.2,0.4,0.6] to prevent overfitting.
- Number of hidden layers explored over [2,4,8,12,16,24], indicating the depth of the model.

A grid search was conducted to explore all combinations of these manipulating variables. Each model of different initiative setting was trained on 4 NVIDIA Ampere A100 GPUs, with 40 GB of memory, and a limited compute window of 30 hours. The Adafactor algorithm was employed to dynamically adjust the learning rate based on the scale of parameters and their gradients, eliminating the need for a fixed initial

learning rate. This adaptive learning rate optimisation technique offers the significant advantage of reduced memory consumption as well [23]. In case the model was able to complete 100 epochs within the mentioned time frame, it was subsequently evaluated on a test set for comparison. Models that failed to converge within the allotted time or exceeded the memory constraint were considered too complex or inefficient for the available computational resources. Among the 255 successfully completed models, 50 models with the lowest prediction MAE on unseen portion of data were selected for further analysis. The distribution of each hyperparameter in this selected set of models was closely examined to identify patterns and trends, as illustrated in Figure 3.4.

Figure 3.4(a) illustrates the 4 largest embedding dimensions (i.e., 456, 768, 1200 and 1800) are associated with high-performing models, indicating a positive correlation between increased embedding size and enhanced model efficacy. While larger embedding sizes such as 1200 and 1800, enable the model to capture more intricate patterns, they also raise the risk of overfitting, resulting in diminished performance on test sets. Moreover, larger models require substantial memory resources, thus elevating the risk of memory overflow errors, especially when implied with complex models. Figure 3.4(b) demonstrates that smaller batch sizes, such as 24, 36, and 64, are predominantly linked with the most effective models. Utilizing smaller batch size has a regularisation effect on the learning process and enhances the model's simplicity. Although training with smaller batch sizes prolongs the process, it introduces beneficial noise into the learning dynamics, enabling the final model to generalise more effectively on unseen data [24]. Another hyperparameter examined in the 50 best-performing models is the batch size, as presented in Figure 3.4(c). A model of 4 to 8 hidden layers includes sufficient parameters to learn nuisance patterns in data. However, adding more layers can overly complicate the model, leading it to memorise data rather than learn from it, which often results in overfitting and

poor performance on unseen data. Figure 3.4(d) reveals that the implementation of dropout rates within the tested spectrum remains relatively uniform across the highest-performing models, although a slight increase is observed in the number of models with dropout rate of 0. Considering that models with fewer hidden layers accounted for the largest share among the high-performing models, it is reasonable to conclude that such models may not necessitate the use of a dropout rate.

Figure 3.5 presents a parallel coordinate plot that compares several promising hyperparameter configurations in accordance to their model's performance. In this plot, each line represents a different model configuration, with axes corresponding to specific hyperparameters, while colour indicates the performance metric (i.e., MAE on the test set). To maintain clarity, a threshold of 0.09 was set to exclude models with higher error rates. It is evident from the plot that increasing hyperparameters—embedding size and the number of hidden layers—leads to improved model performance. However, models with both a largest embedding size and a highest number of hidden layers encountered memory limitations during training. The figure also illustrates a trade-off between these two hyperparameters: a smaller embedding size can be offset by a deeper model, and vice versa. Thus, achieving a balance between the two hyperparameters helps optimise the error rate of the predicted target.

Figure 3.5 also highlights a correlation between the number of hidden layers and the dropout rate among models with lower MAE (represented by shades of red). An excessive increase in the number of hidden layers leads to overfitting, which can be mitigated by a higher dropout rate. The dropout rate and batch size play a crucial role in regulating the model's performance. By increasing the dropout rate or reducing the batch size, we can prevent the model from becoming overly focused on small details in the dataset, thereby reducing the risk of overfitting at the expense of losing generalisation capabilities. However, excessively high dropout

rates combined with a small batch size leads to an underfitted model.

As a result of this thorough optimisation experiment, we identified the most effective hyperparameters for fine-tuning. The best-performing model (embedding size = 768, batch size = 24, dropout rate = 0.2, and 8 hidden layers) is highlighted by a black dashed line in Figure 3.5. This configuration significantly improved the model's prediction accuracy and generalisation on the test set.

3.3 Comparing models with MBTR and MBAD

Ultimately, after thoroughly analysing and optimising both approaches, we compared the performance of MBTR with the original Transformer architecture and MBAD with the adapted Transformer architecture. Both models were optimised and evaluated on the same test set, and the results are presented in Table 3.6.

Table 3.6: Comparison of model performance with standard MBTR and adapted approach applied with MBAD representation, highlighting the improvement in accuracy and generalisation achieved by the adapted solution.

Model	**MAE**	R^2**-score**
Transformer and MBTR	0.1239	0.89
Adapted Transformer and MBAD	**0.0676**	**0.96**

The adapted Transformer model with the MBAD representation achieved a significantly lower MAE of 0.0676 compared to 0.1239 from the Transformer with MBTR, indicating a substantial improvement in prediction accuracy. These results suggest that the modifications introduced in the adapted Transformer architecture, combined with the MBAD representation, contribute to enhanced model performance. The adapted architecture appears to be better suited for capturing the complex interactions modeled by MBAD, leading to improved accuracy and generalisation capabilities.

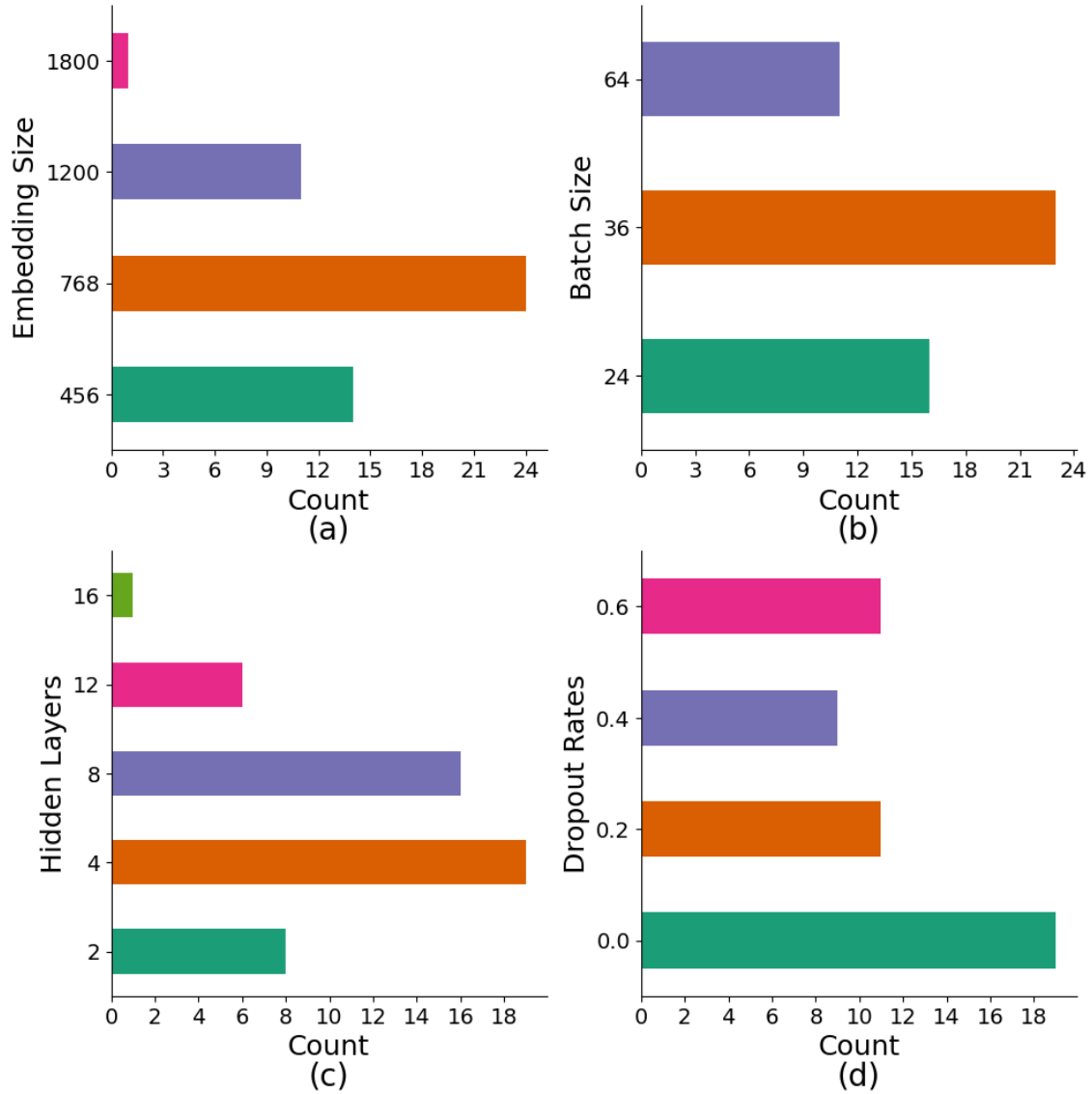


Figure 3.4: Distribution of hyperparameter values among the 50 models with the lowest MAE (best performance) on the test set after 100 epochs of training. The most commonly observed values for the experimented variables are embedding size = 768, batch size = 36, number of hidden layers = 4 and dropout rate=0. (a) Larger embedding sizes (>= 456) are the only values to be seen among the top-performing models. (b) Only the three smallest batch sizes were present among the leading models. (c) The middle range of hidden layers, specifically 4 and 8 layers, predominantly produced the lowest prediction errors. (d) Dropout rates were uniformly distributed among the best models, with a slight advantage observed for a dropout rate of 0.

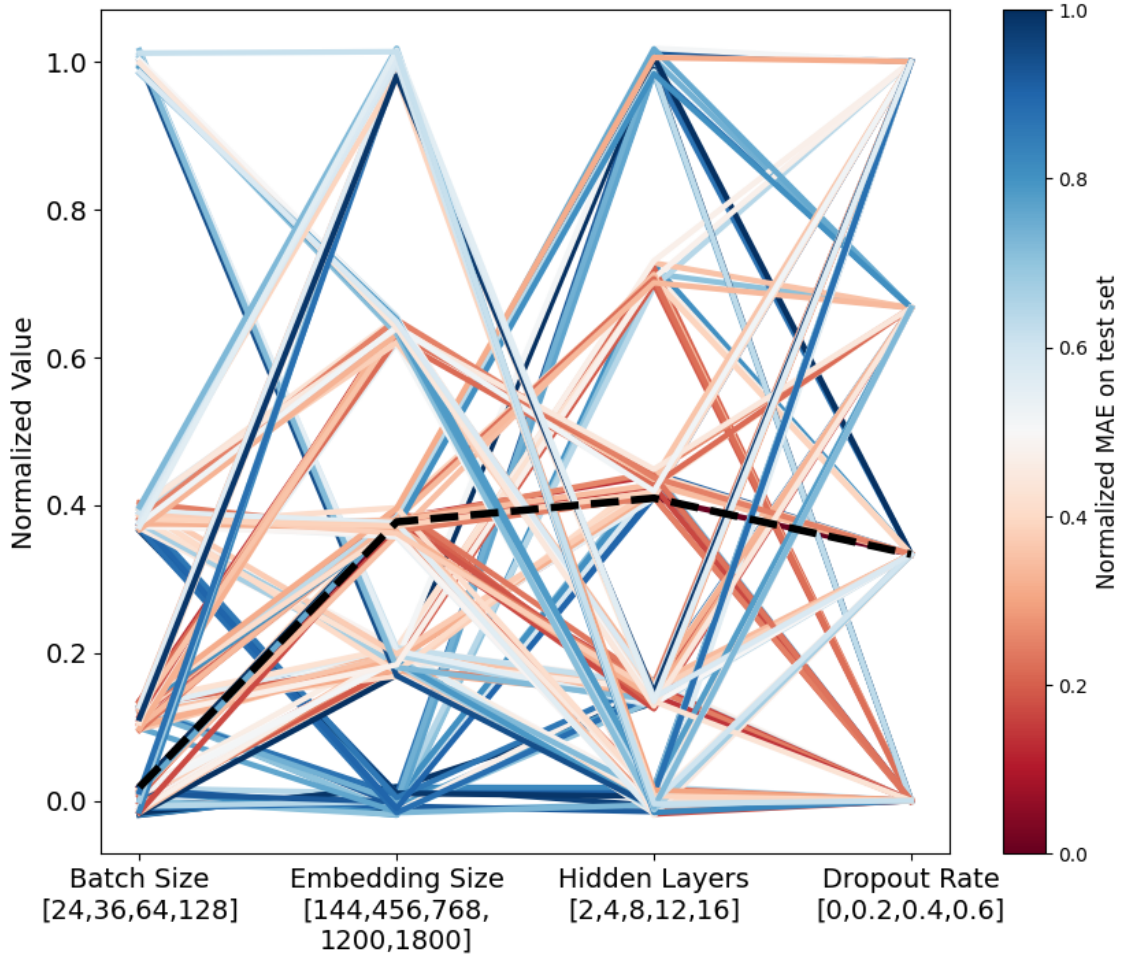


Figure 3.5: Parallel coordinate plot comparing the hyperparameters of the Transformer-based model, filtered by a predictive performance threshold of MAE < 0.09. The dashed line represents the configuration that yields to best-performing model, achieving an MAE of 0.062 after 100 epochs.

3.4 Comparing to the state of the art

In order to further examine the effectiveness of our adaptation of Transformer architecture with a novel representation tailored for such a model, in Table 3.7, we compared its performance with several state-of-the-art (SOTA) models that have been widely applied for predicting HOMO energy in materials science. This comparison allows us to highlight advantages and potential limitations of our approach to existing techniques.

The first model compared is Kernel Ridge Regression (KRR), a regularised learning algorithm that captures nonlinear relationships by applying a kernel function to implicitly map data into a higher-dimensional space [25]. Several studies [26], [7], [27] reported KRR performance for predicting modeling of HOMO energy on the QM9 dataset. Although there are some differences in their validation methods (e.g., cross-validation techniques and representation methods), in Table 3.7, we presented the best-reported KRR which is applied over the MBTR representation of QM9 dataset. Wavelet Scattering Regression is a technique that uses wavelet transforms to capture multiscale, translation-invariant features from data [28]. Lastly, with the lowest MAE in Table 3.7, is the Cormorant model, a neural network that allows for rotationally covariant prediction of properties of complex physical systems [29].

There are also graph-based models in Table 3.7, such as Deep Tensor Neural Network (DTNN), a deep neural network approach that studies quantum-chemical many-body systems by projecting spatial and chemical features onto trainable embeddings [30]. Expanding upon DTNN, SchNet was developed to study atomistic systems. SchNet is another deep learning framework for quantum-chemical property predictions by implementing continuous-filter convolutions [31].

Table 3.7: Performance comparison of the adapted Transformer model with MBAD representation and state-of-the-art (SOTA) models on HOMO energy prediction. A dash ("-") indicates that the corresponding metric was not evaluated in the original source.

Model	**MAE**	R^2**-score**
Transformer (MBAD)	**0.06766 ± 0.00258**	**0.96169 ± 0.00273**
Kernel Ridge Regression (MBTR)	0.086±0.001	0.950±0.002
Deep Tensor Neural Network	0.16	0.9
Cormorant	0.034 ± 0.002	-
SchNet	0.041	-
WaveScatt	0.085	-

Pinpointed in Table 3.7, our adapted Transformer model using MBAD surpassed more conventional models, such as KRR and DTNN by an MAE of 0.0676. However, both SchNet and Cormorant demonstrated superior performance in capturing complex molecular interactions. This is likely due to the design of these models, which are specifically tailored to study molecular structures, contributing to their enhanced accuracy.

4 Conclusion and Future Work

4.1 Conclusion

In this thesis, we investigated the capabilities of Transformer-based models in predicting material quantum chemical properties, specifically focusing on the HOMO energy level. Our motivation stemmed from the impressive results achieved by Transformers across various domains and the potential benefits of applying novel models in the realm of material science, a field that has not yet been extensively explored.

Our analysis was divided into two distinct stages. Initially, the molecular dataset (i.e., QM9) was represented using MBTR, a well-known method that encodes the structural features of molecules into a fixed-length continuous vector for computational modelling. During the first stage, we retained the original MBTR and Transformer architecture with minimal modifications, such as discretising the continuous representation, to test the baseline performance. This approach yielded an MAE of 0.123, demonstrating that even with minimal changes, the Transformer model can reasonably predict HOMO energy levels compared to regression models such as KRR.

We also examined hyperparameters and different approaches to data discretisation to maximise the efficiency of using MBTR and the Transformer model in predicting molecular orbital energy. MBTR's initial hyperparameters, such as sigma

(σ) and representation size, influence the descriptiveness and precision of the representation method in interpreting molecular structures. Therefore, they have a significant impact on the prediction accuracy of the Transformer model. Meanwhile, when molecules are discretised using different techniques, as long as these techniques preserve the main characteristics of the original representation, they lead to similar prediction accuracy by the Transformer model.

As the Transformer neural network is capable of analysing sequences of variable length, we developed a more suitable representation approach during the second stage of our research. The Many-Body Atom Distance (MBAD) is a discretised representation tailored to fit the Transformer model while descriptively presenting molecular structural information. Molecules are introduced to the Transformer model as combinations of atom pairs. Using this novel representation method, the Transformer model receives encoded atomic numbers as input and interatomic distances as positional encodings. By leveraging a purposefully designed representation and modifications to the Transformer architecture, we achieved a significantly lower MAE of 0.071 compared to our first approach.

Hyperparameters were fine-tuned diligently throughout both stages of the study. Unlike MBTR, MBAD is a simpler representation with no hyperparameters, eliminating the need for optimisation. In contrast, optimising MBTR, which has a total of 15 hyperparameters, can be overly complex. However, the Transformer model itself includes numerous hyperparameters that significantly influence the model's learning process. When applying MBAD with the Transformer-based model, an extensive grid search revealed that the model's hyperparameters are heavily correlated. While fine-tuning the hyperparameters proved to be highly resource-intensive, it was crucial for achieving more reliable predictions of the HOMO energy level.

Since MBAD, unlike MBTR, does not constrain molecules to fixed-length representations, it is faster and more memory-efficient. In datasets containing molecules

of various sizes, MBTR produces a highly sparse representation. This sparsity is not only inefficient but also increases the risk of overfitting. On the other hand, the Transformer neural network, due to its large number of parameters, is a resource-intensive model. This complexity, depending on the allocated resources, can lead to memory errors when analysing large molecules, even with the MBAD representation.

In conclusion, our findings manifest the potential of leveraging the Transformer neural network for the unconventional task of predicting quantum chemical properties of organic molecules. Our adaptation of the original Transformer architecture, along with our simple variable-length representation, enhances the Transformer model's understanding of the molecular structure. Through the model's learning process, the Transformer attends to each pair of atoms in relation to other pairs in the molecule and gradually captures an understanding of molecular interatomic distances with regard to the targeted property.

4.2 Recommendation for future work

While the current study has demonstrated the efficacy of Transformer-based models for predicting material properties using the QM9 dataset, several promising avenues for future research remain unexplored. Expanding the scope of this research to include larger and more challenging datasets represents a key next step. The QM9 dataset, though popular and well-suited for comparative studies, contains relatively simple organic molecules. Applying the Transformer model to more complex datasets with a greater diversity of molecular structures would provide a more comprehensive evaluation of its generalisation capabilities. This would also allow the model to encounter molecular representations that are more challenging, testing its ability to learn from larger and more varied input spaces.

Another critical aspect of future work lies in leveraging the power of Transformer models for transfer learning. One of the primary strengths of the Transformer ar-

chitecture in fields such as NLP has been its ability to benefit from pretraining on vast datasets, which allows the model to transfer learned knowledge to specific tasks with limited data. In the context of materials science, pretraining a Transformer model on a large and chemically diverse dataset could provide it with a broad understanding of molecular structures and interactions, enabling it to make more accurate predictions on specialized tasks like quantum chemical energy prediction.

Additionally, the Transformer-based model proposed in this work could be extended to predict other material properties beyond those derived from electronic structures. Investigating its performance on properties that are influenced by different physical or chemical factors, such as thermal conductivity, mechanical strength, or solubility, could provide valuable insights into the model's versatility. Studying such properties would test the model's ability to generalize beyond electronic structure predictions and adapt to diverse property types further validating its potential as a robust tool for materials science research.

In conclusion, this work highlights the promising capabilities of Transformer-based models in materials property prediction and lays the foundation for further exploration. By expanding to larger datasets, utilising transfer learning, and addressing diverse property predictions, future studies can unlock the full potential of this approach, advancing the role of machine learning in materials informatics.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale", *ArXiv*, vol. abs/2010.11929, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:225039882.

[2] K.-D. Luong and A. Singh, "Application of transformers in cheminformatics", *Journal of Chemical Information and Modeling*, 2024.

[3] S. Islam, H. Elmekki, A. Elsebai, *et al.*, "A comprehensive survey on applications of transformers for deep learning tasks", *Expert Systems with Applications*, vol. 241, p. 122 666, 2024, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.122666. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423031688.

[4] K. Rajan, "Materials informatics", *Materials Today*, vol. 8, no. 10, pp. 38–45, 2005, ISSN: 1369-7021. DOI: https://doi.org/10.1016/S1369-7021(05)71123-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369702105711238.

[5] G. Huang, Y. Guo, Y. Chen, and Z. Nie, "Application of machine learning in material synthesis and property prediction", *Materials*, vol. 16, no. 17, 2023, ISSN: 1996-1944. DOI: 10.3390/ma16175977. [Online]. Available: https://www.mdpi.com/1996-1944/16/17/5977.

[6] S. F. Dinesh Behera Amie Philpot. "Understanding homo and lumo". (), [Online]. Available: https://www.ossila.com/pages/homo-lumo#vb-cb-comparison (visited on 10/30/2024).

[7] A. Stuke, M. Todorović, M. Rupp, *et al.*, "Chemical diversity in molecular orbital energy predictions with kernel ridge regression", English, *Journal of Chemical Physics*, vol. 150, no. 20, pp. 1–13, May 2019, | openaire: EC/H2020/676580/EU//NoMaD, ISSN: 0021-9606. DOI: 10.1063/1.5086105.

[8] L. Himanen, M. O. Jäger, E. V. Morooka, *et al.*, "Dscribe: Library of descriptors for machine learning in materials science", *Computer Physics Communications*, vol. 247, p. 106 949, 2020, ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2019.106949. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465519303042.

[9] H. Huo and M. Rupp, "Unified representation of molecules and crystals for machine learning", *Machine Learning: Science and Technology*, vol. 3, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:109774536.

[10] G. Lei, R. Docherty, and S. J. Cooper, "Materials science in the era of large language models: A perspective††electronic supplementary information (esi) available. see doi: Https://doi.org/10.1039/d4dd00074a", *Digital Discovery*, vol. 3, no. 7, pp. 1257–1272, 2024, ISSN: 2635-098X. DOI: https://doi.org/10.1039/d4dd00074a. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2635098X24001190.

[11] S. Yu, N. Ran, and J. Liu, "Large-language models: The game-changers for materials science research", *Artificial Intelligence Chemistry*, vol. 2, no. 2, p. 100 076, 2024, ISSN: 2949-7477. DOI: https://doi.org/10.1016/j.aichem.2024.100076. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2949747724000344.

[12] R. Ramakrishnan, P. O. Dral, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules", *Scientific Data*, vol. 1, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:15367821.

[13] H. Kim, J. Y. Park, and S. Choi, "Energy refinement and analysis of structures in the qm9 database via a highly accurate quantum chemical method", *Scientific Data*, vol. 6, no. 1, p. 109, Jul. 2019, ISSN: 2052-4463. DOI: 10.1038/s41597-019-0121-7. [Online]. Available: https://doi.org/10.1038/s41597-019-0121-7.

[14] S. Lu, Z. Gao, D. He, L. Zhang, and G. Ke, "Data-driven quantum chemical property prediction leveraging 3d conformations with uni-mol+", *Nature Communications*, vol. 15, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271906551.

[15] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. CRC Press, 1986, vol. 26.

[16] G. Zhou, Z. Gao, Q. Ding, *et al.*, "Uni-mol: A universal 3d molecular representation learning framework", *ChemRxiv*, 2022. DOI: 10.26434/chemrxiv-2022-jjm0j-v2.

[17] L. St-Pierre, Y. A. Sari, and M. Kumral, "Creation of histograms for data in various mineral resource and engineering problems: A review of existing methods and a proposed new method to define bin number", *Natural Resources Research*, vol. 26, pp. 201–212, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:63363387.

[18] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need", NIPS'17, pp. 6000–6010, 2017.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", in *North American Chapter of the Association for Computational Linguistics*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:52967399.

[20] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training", 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49313245.

[21] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners", in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[22] A. Stuke, P. Rinke, and M. Todorović, "Efficient hyperparameter tuning for kernel ridge regression with bayesian optimization", *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035 022, Jun. 2021. DOI: 10.1088/2632-2153/abee59. [Online]. Available: https://dx.doi.org/10.1088/2632-2153/abee59.

[23] N. M. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost", *ArXiv*, vol. abs/1804.04235, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4786918.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, Book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org.

[25] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning* (Springer Series in Statistics). New York, NY, USA: Springer New York Inc., 2001.

[26] Z. Wu, B. Ramsundar, E. N. Feinberg, *et al.*, "Moleculenet: A benchmark for molecular machine learning", *Chemical Science*, vol. 9, pp. 513–530, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:217680306.

[27] M. Glavatskikh, J. Leguy, G. Hunault, T. Cauchy, and B. D. Mota, "Dataset's chemical diversity limits the generalizability of machine learning predictions", *Journal of Cheminformatics*, vol. 11, no. 1, 2019, ISSN: 1758-2946. DOI: https://doi.org/10.1186/s13321-019-0391-2.

[28] M. Hirn, S. Mallat, and N. Poilvert, "Wavelet Scattering Regression of Quantum Chemical Energies", *arXiv e-prints*, arXiv:1605.04654, arXiv:1605.04654, May 2016. DOI: 10.48550/arXiv.1605.04654. arXiv: 1605.04654 [math.CA].

[29] B. Anderson, T.-S. Hy, and R. Kondor, "Cormorant: Covariant molecular neural networks", in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[30] K. T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks", *Nature Communications*, vol. 8, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:18666195.

[31] K. Schütt, P.-J. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions", in *Neural Information Processing Systems*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:4106658.