ALMA MATER STUDIORUM

UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE

AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

AI for Industry

SEMI-SUPERVISED LEARNING IN GRAPH NEURAL NETWORKS FOR STRUCTURAL AND PROPERTY PREDICTION APPLIED TO ADVANCED FUNCTIONAL MATERIALS DESIGN

SUPERVISOR

Michele Lombardi

CANDIDATE

Tommaso Cortecchia

CO-SUPERVISOR

Francesco Mercuri

Academic year 2022-2023

Session 2nd

Abstract

Machine learning is becoming an integrating part of computational materials science, being used to predict materials properties, accelerate simulations, design new structures, and predict synthesis routes of new materials. But its efficacy is undermined by problems of data scarcity and portability challenges.

This work explores the potential of graph neural networks in developing a unified predictor for material properties. The goal is to create a versatile molecular model using atomic number and relative distances as exclusive features. The model aims to handle diverse molecular classes, scales, and theory levels, enhancing precision in predicting material properties, even with limited data.

To achieve this, inspired by recent advances in Natural Language Processing, we propose a Masked Molecular Modeling task, training the model in a semi-supervised manner without explicit labels. This task allows the model to predict the atomic type of masked atoms in a molecular structure, giving the opportunity to aggregate diverse data sources and mitigating data scarcity issues. We also assess the capacity of the model to perform property prediction, even with masked elements, and compare it with state-of-the-art approaches.

By incorporating a graph attention mechanism, we not only enhance the model's performance but also gain valuable insights into its internal representation and processing. This contributes to meaningful explanations and a deeper understanding of the model's workings.

Contents

Chapter 1

Introduction

In recent times, researchers witnessed big steps forward in using machine learning for chemistry and materials science. The addition of data-driven methodologies in the field is changing the game, accelerating discovery, simulation, and design of new materials. Graph Neural Networks (GNNs) are drawing increasing interests, since they can directly work on a graph or structural representation of molecules and materials, enabling them to access all relevant information needed to characterize materials.

In particular, the application of GNNs has shown great promise for predicting molecular properties. However, the current limitations mainly stem from the scarcity of available data, which is both expensive and time-consuming to generate. Furthermore, there are portability issues with learning models due to variations in specific tasks, molecular classes, and computational methods applied to create different datasets.

Current state-of-the-art (SOTA) approaches often rely on complex and domain-specific models, limiting their applicability across different material classes, scales, and theoretical frameworks. This work explores the potential of employing GNNs to overcome these limitations and develop a unified predictor that leverages atomic number and relative distances of elements as the only features needed. By encoding atomic number and spatial distances as node and edge features, respectively, our model aims to reconcile different material classes, scales, theoretical methods and relative parameters under a single framework.

Drawing inspiration from recent breakthroughs in Natural Language Processing, our objective is to create a comprehensive and extensive molecular model that captures a vast amount of knowledge regarding atomistic relationships and properties. This model will serve as an equivalent representation for molecules, similar to the way modern language models incorporate understanding and generation of language.

Our envisioned model aims to possess the capability to handle diverse molecular classes, such as drugs and crystals, spanning various scales ranging from small molecules to nano-structured systems. Furthermore, it should be compatible with different computational methods, accommodating variations in complexity. By achieving this versatility, the model can be specialized to perform specific tasks, such as predicting specific material properties, with precision and accuracy. This specialization can be achieved even when confronted with limited data, which is the bottleneck of machine learning for materials science today.

To accomplish the development of such a model, we explore the potential of training it in a semi-supervised manner, eliminating the need for explicit labels or targets and relying solely on the molecular structure. In this regard, we propose the adoption of a Masked Molecular Modelling task, inspired by the concept of Masked Language Modeling. In this task, the model is presented with a molecule structure where certain nodes (atoms) have been masked, and its objective is to predict the atomic type of the masked atoms. This approach enables the aggregation of diverse data from various sources, circumventing the need to rely on specific available targets and their computation, thereby mitigating the issue of data scarcity. By leveraging this novel task, the model could exploit a broader range of data, enhancing its learning capabilities and extending its applicability to different material design scenarios.

The work is organized as follows: in Chapter 2, we introduce the fundamentals of computational materials science, GNNs and the variants used in this work. We then conduct a comprehensive review of the state of the art in the application of GNNs to materials science. Chapter 3 presents the three datasets and the architecture of the model used in this work, along with an explanation of the training objectives. Lastly, in Chapter 4, we illustrate quantitative results by comparing the model's performance to SOTA models. We also provide a detailed examination of the attention maps produced by the model, which offer valuable insights from an interpretability perspective and somewhat align with classical molecular models. We conclude with final considerations about the model's limitations and potential future work.

Chapter 2

Background

2.1 Computational Materials Science

Computational materials science (CMS) is a subfield of materials science that uses modeling, simulation, theory, and informatics to understand materials and their properties. The first theoretical calculations in chemistry were those of Walter Heitler and Fritz London in 1927, using valence bond theory. Computational chemistry has its roots in the early attempts made by theoretical physicists, beginning in 1928, to solve the Schrödinger equation using hand-cranked calculating machines. The first known use of computational methods in materials science was in the 1950s, when quantum mechanics was used to study the atomic structure of metals. In the 1960s, computational methods were employed to study the electronic structure of materials, leading to the development of density functional theory. The 1970s saw the first first-principles electronic structure calculations, which allowed for the prediction of materials properties without relying on experimental data. In the 1980s, the development of more powerful computers enabled the study of larger and more complex materials systems. The 1990s introduced the concept of ”ab initio” materials modeling, which involved the use of first-principles calculations to predict materials properties and behavior. In the 2000s, machine learning and data-driven approaches began to be used in materials science [55, 2, 28].

Some of the most significant developments in computational materials science include density-functional theory (DFT) [40, 49] and multiscale simulation models. Both approaches have been recognized with the Nobel Prize in chemistry, but their impact extends beyond chemistry and affects all disciplines of fundamental natural science. Today, major themes in the field include uncertainty quantification and propagation throughout simulations for decision making, data infrastructure for sharing simulation inputs and results, high-throughput materials design and discovery, and new approaches given significant increases in computing power.

The foundation of Computational Chemistry lies in quantum mechanics, classical mechanics, statistical mechanics, and thermodynamics. These theoretical frameworks are used to describe the behavior of atoms, molecules, and materials at different scales. Most popular methods include:

- Density Functional Theory (DFT) [49]: as mentioned earlier, DFT is a widely used computational method to study electronic structure, properties, and stability of materials from first principles.

- Molecular Dynamics (MD): MD simulations model the motion of atoms and molecules in a material over time, allowing the study of material dynamics and thermodynamics.

- Monte Carlo (MC) methods: MC simulations use statistical sampling to explore the configuration space of materials and calculate thermodynamic properties.

- Quantum Monte Carlo (QMC) [12]: QMC methods provide more accurate electronic structure calculations than DFT but are computationally expensive and limited to small systems.

- Machine Learning (ML): ML techniques have been integrated to accelerate calculations, predict material properties, and aid in materials discovery.

Today, many challenges lie behind the utilization of computational methods:

- Multiscale and multi-physics modeling: addressing material problems with important features at multiple length scales requires the development of efficient and accurate multiscale modeling techniques. Addressing problems of multi-physics nature, such as thermo-mechanical and electromagnetic phenomena, also presents significant challenges [9]
- Computational resources: as materials science problems become more complex, the demand for computational resources increases, necessitating the development of more efficient algorithms and the use of advanced computing technologies, such as parallel computing, cloud computing, and GPU computing [38].
- Uncertainty Quantification: quantifying and propagating uncertainties throughout simulations is essential for making informed decisions based on computational materials science research.
- Data Veracity: ensuring the accuracy and reliability of data used in data-driven materials science is a critical challenge, as it directly impacts the quality of predictions and insights [39].
- Data Infrastructure: developing robust data infrastructure for sharing simulation inputs and results is crucial for facilitating collaboration and accelerating materials discovery.

2.2 Emergence of Machine Learning

Data science and machine learning have seamlessly integrated into the fabric of natural sciences, emerging as the fourth pillar alongside experiment, theory, and simulation [51]. Throughout the materials development cycle, machine learning methods are increasingly pervasive, influencing various stages such as discovering initial candidate materials through property prediction [74], database screening [46], and even inverse materials design [60]. This influence extends to the detailed analysis of materials within machine learning-accelerated simulations [29], the prediction of synthesis conditions [53], and automated analysis of experimental data [47], as well as experimental planning [37].

The machine learning models employed in the domains of chemistry and materials science exhibit a diverse range of methods. These encompass classical machine learning models like decision tree ensembles to state-of-the-art deep learning methods such as convolutional neural networks [50] and sequence models [72, 4], initially developed for challenges in computer vision and natural language processing.

The growing interest from the scientific community towards machine learning solutions can be attributed to their particular speed and efficiency compared to traditional experimental and/or computational methods. Despite being widely used, DFT is still too slow to be applied to large systems (scaling as $O(N_e^3)$ where N_e is the number of electrons). For example, to run the DFT calculation on a single 9 heavy atom molecule in QM9 takes around an hour on a single core of a Xeon E5-2660 (2.2 GHz) using a version of Gaussian G09 (ES64L-G09RevD.01) [42]. What typically can take several hours using DFT simulations, can be achieved in a mere fraction of time with machine learning models. And that is really what is attaining the attention of the scientists. Machine learning could enable rapid screening and selection of materials, alleviating the computational burden these practices entail, and assist in the design process of such materials.

But all this comes with a cost. Modern AI systems are data-hungry and require a lot of resources and optimization to work effectively. Currently available datasets cover only a minimum part of the entire chemical space, which

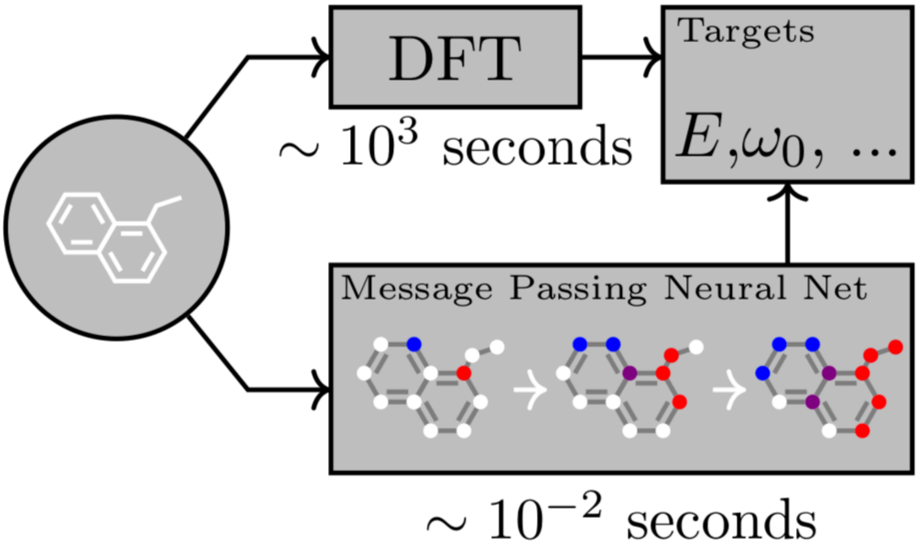


Figure 2.1: Neural Networks are fairly faster than traditional simulation methods. Taken from [33]

is estimated to contain around 10^{80} potentially meaningful chemical combinations. Options like transfer learning and physics-informed neural networks [64] are still being explored .

There are still challenges and limitations in the application of machine learning and deep learning in materials science. Researchers are continuously working to improve the accuracy, efficiency, and interpretability of these models to better understand and predict material properties and behavior [17, 1, 69].

2.3 Graph Neural Networks

GNNs, or Graph Neural Networks, are a class of machine learning models specifically designed to operate on graph-structured data. They have gained significant attention and popularity in recent years due to their effectiveness in various domains, including social networks, recommender systems, bioinformatics, and computer vision.

Traditional neural networks are designed to process grid-like data, such as images or sequences, where the input has a fixed shape and connections between elements are typically uniform. However, many real-world problems involve data with irregular and interconnected structures, such as social networks, citation networks, or molecule structures. GNNs are designed to handle such data by leveraging the underlying graph structure.

At a high level, GNNs aim to learn node representations by aggregating information from neighboring nodes in the graph. This aggregation process typically involves a series of message passing steps, where nodes exchange information with their neighbors and update their own representations accordingly. By iteratively performing these steps, GNNs are able to capture both local and global information about the graph, enabling effective learning and prediction tasks.

The core components of a GNN include:

- *Node representations*: Each node in the graph is associated with a feature vector, which represents its characteristics or attributes. These features can be provided as input or learned as part of the GNN training process.

- *Message passing*: GNNs propagate information through the edges of the graph. At each step, a node aggregates and updates its representation based on the representations of its neighboring nodes. This process allows nodes to exchange information and capture relational dependencies.

- *Graph pooling*: GNNs can aggregate information from the entire graph to produce a graph-level representation. This can be achieved through pooling operations that summarize node-level features into a single graph-level representation.

- *Readout/Output*: After the message passing and pooling steps, the learned node and graph representations can be used for various downstream tasks, such as node classification, link prediction, graph classification, or recommendation.

GNNs have been successful in various applications. For example, in social network analysis, GNNs can learn representations of users and predict their interests or behaviors. In drug discovery, GNNs can predict molecular properties and assist in the design of new drugs [85]. In recommender systems, GNNs can model user-item interactions and provide personalized recommendations [26].

It's worth noting that GNNs have different architectures and variations, including Graph Convolutional Networks (GCNs) [48], GraphSAGE [35], Graph Attention Networks (GAT) [76], and Graph Transformers [23], among others. Each variation introduces specific modifications to the basic GNN framework to address different challenges or improve performance in specific domains.

2.3.1 Message Passing Paradigm

A graph is given by a tuple $\mathcal{G} = (V, E, X, W)$, where V is the set of vertices, E is the set of edges, X are node attributes and W are the edge attributes.

The message passing algorithm is the fundamental component of GNNs that allows nodes to exchange information with their neighboring nodes inside the graph. The algorithm enables GNNs to capture and propagate information throughout the graph, incorporating both local and global dependencies. A message block of a GNN typically consists of the following steps:

1. *Initialization*: Each node in the graph is initialized with a feature vector, provided as input or learned as an embedding.

$$h_v^0 = x_v, \quad \forall v \in V$$

2. *Message Computation and Aggregation*: In this step, each node aggregates information from its neighboring nodes:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

   The specific aggregation function $\sum$ can vary, some popular choices include mean aggregation, sum aggregation, or weighted sum based on the edge connections.

3. *Update function*: the update step combines the current node representation with the transformed messages, allowing the node to integrate information from its neighbors:

$$h_v^{t+1} = U_t(h_v^t, m_v^t)$$

The message passing algorithm is typically performed iteratively for multiple steps. In each iteration, the nodes exchange messages, aggregate them, and update their representations based on the aggregated information. The iterative nature of the algorithm allows nodes to gather information from increasingly distant parts of the graph.

The readout phase computes a feature vector for the whole graph using some readout function R according to

$$\hat{y} = R(\{h_v^T | v \in \mathcal{G}\})$$

The message functions M_t, vertex update functions U_t and readout function R are all learned differentiable functions. R operates on the set of node states and must be invariant to permutations of the node states in order for the message passing to be invariant to graph isomorphism.

2.3.2 Tasks

Three primary types of tasks associated with GNNs are:

Node-level Tasks: node-level tasks involve making predictions or classifications for individual nodes within a graph. The goal is to learn representations of nodes that capture their local and global context. Examples of node-level tasks include:

1. *Node Classification*: Assigning labels or categories to nodes based on their attributes and connectivity.

2. *Node Regression*: Predicting continuous values for nodes, such as predicting property values of molecules in a chemical graph.

3. *Node Clustering*: Grouping nodes into clusters based on their structural similarities.

Edge-level Tasks: edge-level tasks involve predicting properties or relationships between pairs of nodes (edges) within a graph. These tasks often focus on capturing pairwise interactions and dependencies. Examples of edge-level tasks include:

1. *Link Prediction*: Predicting missing or future connections between nodes in a graph, commonly used in social networks or recommendation systems.

2. *Relation Extraction*: Identifying relationships between entities in a knowledge graph or natural language processing context.

3. *Edge Classification*: Assigning labels to edges to capture specific interactions or associations between nodes.

Graph-level Tasks: graph-level tasks focus on making predictions or classifications for entire graphs. The objective is to capture the overall structure and interactions within the graph. Examples of graph-level tasks include:

1. *Graph Classification*: Assigning a label or category to the entire graph, such as classifying molecular graphs as different chemical compounds.
2. *Graph Regression*: Predicting a continuous value for the entire graph, like predicting the properties of a material represented by its crystal structure graph.
3. *Graph Generation*: Generating new graphs that share specific characteristics or properties with the input data.

2.3.3 Expressiveness of GNNs

Traditional feed-forward networks (multi-layer perceptrons) are known to be universal approximators: they can approximate any smooth function to any desired accuracy [41]. However, GNNs pose unique challenges and opportunities due to the intricate nature of graph-structured data.

GNNs operate by propagating and aggregating information across graph nodes, enabling them to capture complex relationships and structural patterns within graphs. While they have demonstrated remarkable performance on various datasets, they also exhibit a phenomenon where they excel on some datasets but underperform on others [87, 78]. This has prompted researchers to delve deeper into understanding the underlying power and limitations of GNNs.

To shed light on the expressive power of GNNs, researchers have drawn connections to the Weisfeiler-Lehman (WL) test [83], a classical problem in graph theory that addresses graph isomorphism. The WL test, initially believed to be a polynomial-time solution for graph isomorphism, leverages iterative graph recoloring to distinguish between different types of graph structures. However, it was later found to be insufficient for certain cases.

In any case, graph neural networks have been demonstrated to be as expressive as the 1-WL test [57, 68]. Graph Isomorphism Networks (GIN) [87] were introduced to explore the relationship between GNNs and the WL test. These networks were designed to be as powerful as possible in terms of distinguishing between graph structures. Interestingly, the expressiveness of GIN is closely related to the WL algorithm. This connection not only offers a new perspective on graph neural networks but also serves as a bridge between classical graph theory and modern machine learning techniques.

Moreover, extensions of the WL test, such as the k-WL test [43], and Invariant Graph Networks [54], have further enriched our understanding of GNN expressiveness. These higher-order graph architectures demonstrate varying degrees of power in distinguishing graph structures, leading to the emergence of the Weisfeiler-Lehman hierarchy. Such developments provide a theoretical foundation for assessing and comparing different graph neural network architectures.

While the theoretical insights have expanded our understanding of GNN capabilities, the practical implications are still being explored. Recent benchmarks have shown that the performance of provably powerful graph neural network algorithms might not necessarily surpass that of older techniques in practice [91, 24]. This highlights the intricate interplay between expressivity, generalization, and the underlying notion of graph similarity in specific applications.

Weisfeiler-Lehman isomorphism test The Weisfeiler-Lehman Isomorphism Test works by producing for each graph a canonical form. If the canonical forms of two graphs are not equivalent, then the graphs are definitively not isomorphic. However, as mentioned above, it is possible for two non-isomorphic graphs to share a canonical form, so this test alone cannot provide conclusive evidence that two graphs are isomorphic.

- At iteration i we assign to each node a tuple $L_{i,n}$ containing the node's old compressed label and a multiset of the node's neighbors' compressed labels (a multiset is a set where elements may appear multiple times).

- At each iteration we will additionally assign a new "compressed" label $C_{i,n}$ to each node n's set of labels. Any two nodes with the same $L_{i,n}$ will get the same compressed label.

1. To begin, we initialize $C_{0,n} = 1$ for all nodes n. At iteration i of the algorithm (beginning with $i = 1$), for each node n, we set $L_{i,n}$ to be a tuple containing the node's old label $C_{i-1,n}$ and the multiset of compressed node labels $C_{i-1,m}$ from all nodes m neighboring n from the previous iteration $(i - 1)$.

2. We then complete iteration i by setting $C_{i,n}$ to be a new "compressed" label, such as a hash of $L_{i,n}$. Any two nodes with the same labels $L_{i,n}$ must get the same compressed label $C_{i,n}$.

3. Partition the nodes in the graph by their compressed label. Repeat $2 + 3$ for up to N (the number of nodes) iterations, or until there is no change in the partition of nodes by compressed label from one iteration to the next.

When using this method to determine graph isomorphism, it may be applied in parallel to the two graphs. The algorithm may be terminated early after an iteration if the sizes of partitions of nodes partitioned by compressed labels diverge between the two graphs; if this is the case, the graphs are not isomorphic.

2.3.4 GAT

GAT (Graph Attention) Networks were introduced by Velickovic et al. [76] in 2018.

A single attentional layer is composed as follows: the input is a set of node features h $= \{h_1, \ldots, h_n\}, h_i \in \mathbb{R}^F$, where N is the number of nodes, and F

Algorithm 1 1-WL (color refinement)

```
Input: $\mathcal{G} = (V, E, X_v)$
$C_{0,v} \leftarrow 1$ for all $v \in V$
repeat
    for all $v \in V$ do
        $L_{i,v} \leftarrow (C_{i-1,v}, \{\{C_{i-1,w} : w \in \mathcal{N}(v)\}\})$ for all $v \in V$
        $C_{i,v} \leftarrow hash(L_{i,v})$
until $(C_{i,v})_{v \in V} = (C_{i-1,v})_{v \in V}$
return $\{\{C_{i,v} : v \in V\}\}$
```

is the dimensionality of the features in each node. First, each node feature is mapped to a higher-level feature through a learnable linear transformation, parameterized by weight matrix $\mathbf{W} \in \mathbb{R}^{F' \times F}$. Then a shared attentional mechanism $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$ is applied to compute *attention coefficients*

$$e_{ij} = a(h_i, h_j), \tag{2.1}$$

indicating the importance of node j's features to node i. The most general formulation of attention allows node i to attend to every node j of the input. In GAT, the graph topology is injected in the computation by allowing node i to attend to every node $j \in N_i$, where N_i is the neighborhood of i. For each node i attention weights are normalized by applying a softmax function:

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum\limits_{k \in N_i} \exp(e_{ik})} \tag{2.2}$$

The attention mechanism a consists in a single-layer feed-forward network, followed by a LeakyReLU nonlinearity:

$$a(h_i, h_j) = \text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}h_i \| \mathbf{W}h_j]\right), \tag{2.3}$$

with a $\in \mathbb{R}^{2F'}$, and ^{T} and $||$ are the transposition and concatenation operators respectively. The attentional mechanism described in the paper refers to the original implementation of Bahdanau et al. [5], but the framework is agnostic to the specific choice of the attentional setup.

Fully expanded, attention coefficient result in:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}h_i \| \mathbf{W}h_j])\right)}{\sum\limits_{k \in N_i} \exp\left(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}h_i \| \mathbf{W}h_k])\right)} \tag{2.4}$$

The output for each node is obtained as a linear combination of the neighbors features weighted by the attention coefficients, eventually followed by a nonlinearity σ :

$$h_i' = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \mathbf{W} h_j\right) \tag{2.5}$$

The mechanism can be extended to be a *multi-head attention* [75], with K independent attention computations which outputs are then concatenated to obtain the final output feature, represented by:

$$h_i' = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k h_j\right) \tag{2.6}$$

Specially, when performing multi-head attention on the final (prediction) layer of the network, concatenation is no longer sensible - instead, averaging is employed, the nonlinearity (usually a softmax or logistic sigmoid for classification problems) is applied in the end:

$$h_i' = \sigma\left(\sum_{k=1}^{K} \sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k h_j\right) \tag{2.7}$$

Figure 2.2 shows the aggregation process of a GAT layer.

A subsequent study by Brody et al. [10] highlights a constraint in the expressive capabilities of GAT networks. They formally introduce the concepts of *static* and *dynamic* attention, asserting that GAT networks exclusively offer *static* attention, which is inherently less expressive compared to the more encompassing *dynamic* attention. In the end, they demonstrate their claim in a controlled problem where the original GAT cannot even fit the training

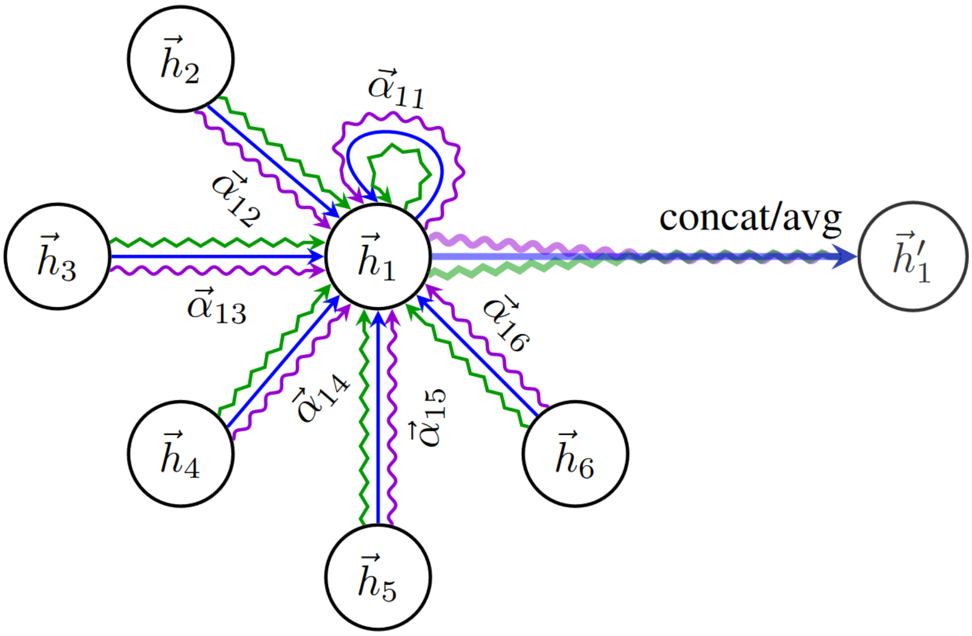


Figure 2.2: An illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain h'_1. Extracted from [76].

data. The proposed fix is a simple modification of the order of operations in Eq.(2.3):

$$\text{GAT}: \qquad a(h_i, h_j) = \text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}h_i \| \mathbf{W}h_j]\right) \tag{2.8}$$

$$\text{GATv2}: \qquad a(h_i, h_j) = \mathbf{a}^T \text{LeakyReLU}\left(\mathbf{W}[h_i \| h_j]\right) \tag{2.9}$$

2.3.5 EGAT

The work of Wang et. al. [79] introduces the so-called edge-integrated attention mechanism (EGAT), integrating edge data in the message-passing operation, which is not covered in the original implementation of GAT. Moreover, edge features are updated with the adjacent node features to generate higher level representations of edges too.

The node update happens similarly to the original GAT implementation, but integrates edge features in the computation. First, node and edge features are mapped to higher-level features through linear transformations with weight matrix $\mathbf{W_h} \in \mathbb{R}^{F'_H \times F_H}$ and $\mathbf{W_h} \in \mathbb{R}^{F'_E \times F_E}$ respectively, then an edge-integrated attention mechanism generates attention weights α_{ij}

$$w_{ij} = a(\mathbf{W_h} h_i, \mathbf{W_h} h_j, \mathbf{W_e} e_{ij}) \tag{2.10}$$

which are then normalized through a softmax function:

$$\alpha_{ij} = \text{softmax}_j(w_{ij}) = \frac{\exp(w_{ij})}{\sum\limits_{k \in \mathcal{N}_i} \exp(w_{ik})} \tag{2.11}$$

where $\mathcal{N}_i$ represents the neighborhood of node i.

The attention mechanism a is chosen as a single layer feed-forward network followed by a LeakyReLU activation, similarly to GAT:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W_h} h_i \| \mathbf{W_h} h_j \| \mathbf{W_e} e_{ij}])\right)}{\sum\limits_{k \in N_i} \exp\left(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W_h} h_i \| \mathbf{W_h} h_k \| \mathbf{W_e} e_{ik}])\right)} \tag{2.12}$$

The resulting weights are then applied to perform a weighted sum on neighboring node features, followed by a non-linearity σ:

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} [\mathbf{W_h} h_j \| \mathbf{W_e} e_{ij}]\right) \tag{2.13}$$

Edge features are updated using the *node-transit strategy*, which uses nodes as transit ports of edge features. Firstly, the nodes aggregate the adjacent edge features with the edge-integrated attention mechanism:

$$\beta_{ij} = \text{softmax}_j(\text{LeakyReLU}(\mathbf{b^T}[\mathbf{W_h} h_i \| \mathbf{W_h} h_j \| \mathbf{W_e} e_{ij}])) \tag{2.14}$$

$$e'_i = \sum_{j \in \mathcal{N}_i} (\beta_{ij} \mathbf{W_e} e_{ij}) \tag{2.15}$$

The aggregated edge features and node features are used to generate the higher-level edge features through a multi-layer perceptron (MLP):

$$e'_{ij} = \text{MLP}(h_i, h_j, e'_i, e'_j, e_i j) \tag{2.16}$$

2.4 Graph Neural Networks for Materials Science

State-of-the-art architectures Early developments in neural networks for molecular graphs date back to the 90s and 2000s, without explicitly referring to the term graph neural network [56, 7]. Message passing neural networks (MPNNs), with edge features capturing bond information, have been applied to molecular [33] and crystal graphs [84]. D-MPNN introduces directed edge embeddings and message passing between edges [89]. Masked self-attention layers, inspired by natural language processing models [75], have been suggested for graph attention networks [76] and explicitly used for molecules in Attentive Fingerprint models [86] and for Crystal Graph Attention networks [70].

Beyond graph models focusing on chemical graphs, a significant category is dedicated to models explicitly designed for learning quantum properties. These models typically take atomic numbers and positions as input, training on data derived from approximate solutions of the steady-state Schrödinger equation. The QM9 dataset, a widely recognized benchmark dataset [67, 66], encompasses 12 quantum properties of small molecules containing up to nine atoms, excluding hydrogen. SchNet [71], one of the earliest graph networks achieving chemical accuracy on QM9, utilizes convolutional filters for interatomic distances and incorporates skip connections between node updates. An enhancement to SchNet involves updating positional features along graph edges, as demonstrated by Jørgensen et al. [45].

MEGNet [15] explores the application of GNNs to crystals, incorporating geometric information and leveraging global properties such as temperature, crucial for solid-state crystalline systems. DimeNet [31, 32] addresses the dependence of molecular potential energy on bond angles, utilizing edge embedding with message passing steps from atomic triplets and bond pairs to incorporate angular features. This formalism has been adopted by other recent GNNs [16, 90], and extended to include dihedral angles [30, 14].

For directed edge updates with explicit angle plus node information, as seen in DimeNet, message passing essentially operates on higher-order paths [27] or k-pairs of atoms [57]. However, this becomes impractical for fully connected larger graphs due to the exponential increase in multi-node interactions. Models like MXMNet [90] address this by using multiplex graphs, selectively considering specific edges when calculating bond angles in higher-order pathways [16].

Graph representation Graph networks often utilize the chemical graph directly as input, catering to both molecules [33] and inorganic compounds [70, 84]. This approach offers advantages over compositional or fixed-sized vector representations, providing flexibility and scalability. Consequently, GNNs find applications in tasks like drug design or material screening [86], where knowledge of functional groups, scaffolds, or the full chemical structure and its topology is crucial. In molecular applications, the chemical graph is commonly extracted from SMILES codes and augmented with features obtained from cheminformatics software.

For chemistry-related tasks, the connectivity of atoms in molecules often contains sufficient information to predict molecular properties without relying on exact geometry. However, geometry or stereochemical information can be considered e.g, through additional edge features representing the distance between atoms [15]. In contrast, materials applications face challenges as atom connectivity is not well defined in most cases, necessitating the extraction of graphs from crystal structures based on distance heuristics. The chemical graph alone may not suffice to accurately predict quantum-mechanical or electronic-structure properties in materials tasks that depend strongly on the exact molecular geometry.

For tasks involving geometric dependencies, such as predicting potential energy surfaces of molecules and materials [14], it becomes evident that geometric information is essential. The representation of positional and geometric information to learn quantum properties has been explored in previous works, leading to a variety of descriptors. Many of those descriptors expand geometric information into symmetry or basis functions, and are typically employed in conventional machine learning models such as neural networks and Gaussian processes.

Geometric information, such as distances, bonds and dihedral angles, has also been widely used for node or edge representation in graph neural networks. Angles or distances are similarly expanded into Gaussian-like, radial and spherical Fourier-Bessel functions. For molecules, attributes like chirality, aromaticity, hybridization, presence of rings, are valid information to be included to the structural representation. For solid crystals and periodic structures, the periodicity and space group symmetries are additional symmetries to be added to the representation for GNNs.

Chapter 3

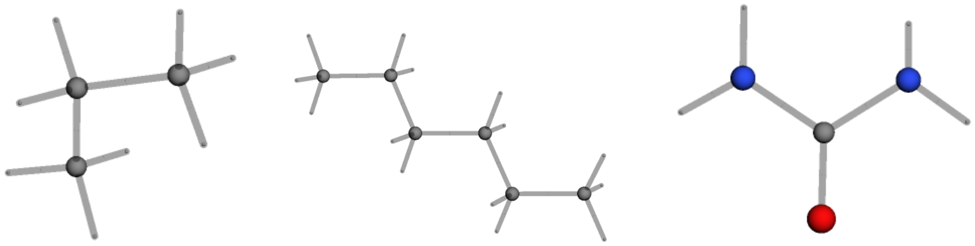
Methodology

3.1 Datasets

QM9 The QM9 dataset, short for ”Quantum Chemistry of Molecular Structures 9” [67, 66], is a widely used dataset in the field of computational chemistry and machine learning. It was introduced to serve the comparative analysis of existing methods, the development of new methods, such as hybrid quantum mechanics/machine learning, and the systematic identification of structure-property relationships. It contains data related to the structural properties of around 134 000 small organic molecules. Each molecule is represented by its atomic structure, including the types of atoms, their coordinates, and the chemical bonds between them. For each molecule, the dataset provides a range of quantum chemical properties, calculated at the B3LYP/6-31G(2df,p) level of quantum chemistry, including but not limited to:

- HOMO (Highest Occupied Molecular Orbital)
- LUMO (Lowest Unoccupied Molecular Orbital)
- Band gap
- Internal energies
- Thermodynamic properties

The QM9 dataset is valuable for tasks related to drug discovery, property prediction, and understanding the electronic and structural properties of organic molecules.

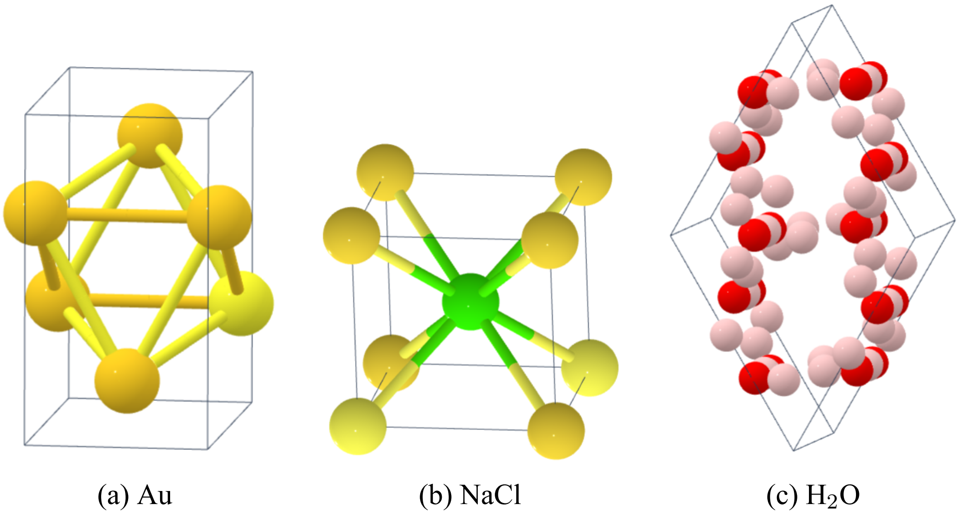


(a) Propane (b) Hexane (c) Urea

Figure 3.1: Molecules sampled from QM9

Materials Project The Materials Project [44] (MP) is a widely recognized and influential initiative in the field of materials science and computational materials research. It's a collaborative effort to provide open-access materials data and computational tools to researchers and the public. Its primary mission is to accelerate materials discovery and innovation by offering comprehensive data on a wide range of materials. MP maintains a vast and continuously expanding database of materials, including inorganic compounds and crystalline structures.

Since Materials Project is constantly being updated, the work of Chen et al. [15] has produced a subset of MP - nominally, MP2018.6.1 - serving as benchmark for later studies, including this one. The crystal data set comprises the DFT-computed energies and band gaps of 69 640 crystals from the Materials Project obtained via the Python Materials Genomics (pymatgen) [62] interface to the Materials Application Programming Interface (API) [61] on June 1, 2018. Available targets in this dataset are formation energy per atom and band gap.



(a) Au (b) NaCl (c) H_2O

Figure 3.2: Unit cells of crystals sampled from MP, along with their crystal lattice.

CSIRO Graphene Oxide Dataset The CSIRO Graphene Oxide dataset (GO) [6] is a collection of electronically neutral graphene oxide nanoflake and periodic graphene oxide sheet final configurations for use in data-driven studies. The dataset includes 20 396 nanoflake final configurations, accompanied by a list of 830 features extracted from the simulations. The dataset was generated using DFTB simulations to train machine learning models. A study published in 2019 identified 25 archetypal 'pure' graphene oxide structures and three prototypes that are the truly representative averages in 224-dimensional space [58].

For the purpose of this work, a subset of 7 000 samples has been validated and extracted from the original dataset. Out of the 830 features accompanying graphene sheets, we are interested in 5 of them, to be used as targets for property prediction:

- Total energy
- Fermi energy
- Ionization potential
- Electronegativity

- Electron affinity

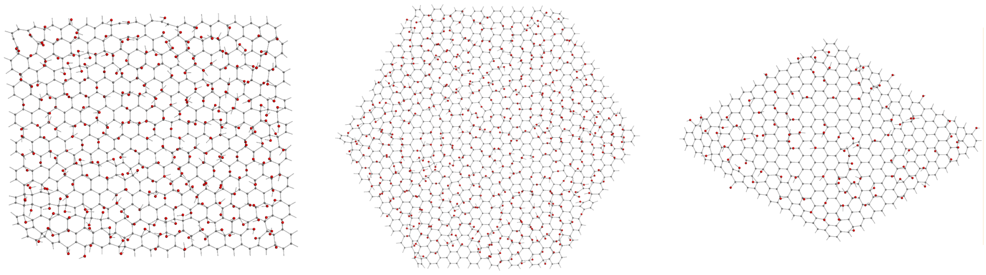


Figure 3.3: Graphene oxide sheets sampled from GO

QM9 and MP are commonly used benchmarks for molecular and crystalline systems, respectively. Over the years, many works have focused on one or the other, often both are taken in consideration. On the other hand, less effort has been dedicated to nanostructured systems in general. Working on these systems has many complications: nanostructured materials encompass a wide range of structures, including nanoparticles, nanowires, nanotubes, and more; each of these structures can have different sizes, shapes, and surface properties; at the nanoscale, size effects become prominent, and the properties of materials can differ significantly from their bulk counterparts. Understanding and modeling these size-dependent effects can be computationally intensive and may require specialized techniques. There is a lack of widely accepted benchmark datasets for nanostructured materials. Creating such datasets can be challenging due to the diversity of structures and properties, not mentioning the challenges related to obtaining synthetic data through simulations, which are usually extremely resource-intensive and time-consuming. The CSIRO Graphene Oxide dataset has been selected as a benchmark to assess the performance of the model for a particular instance of this wide class of materials (i.e.,graphenes).

3.2 Structure Representation

This study aims to develop a straightforward representation method that can be used for any type of material. Various common representation formats exist for different material classes:

- Small molecules: SMILES (Simplified Molecular Input Line Entry System) [81, 82, 80], SMARTS (SMILES Extended) [19, 20], InChI (International Chemical Identifier) [73], MDL Molfile [18], Chemical Markup Language (CML) [13, 59], XYZ Coordinate Format [88],...
- Crystals: (CIF) Crystallographic Information File [34, 11], PDB (Protein Data Bank) [3, 8], COD (Crystallography Open Database)[22], XYZ,...
- Nanographenes: CIF, PDB, MOL, XYZ,...

Each representation differ in the description of the structure: SMILES, InChI and SMARTS are a compact way to describe composition and topology (bonds and substructures) of a molecule, but don't contain spatial information; other representations (e.g., XYZ) contain only information about atomic positions; other richer representation, like MOL, PDB, CIF, are dictated by necessities of standardization and data exchange, but are not available for every structure; not all representations are compatible with any material class. Inside this representation zoo, what really do small molecules, crystals and nanomaterials all have in common? They all can be represented in term of their composition (atomic types) and structure (atomic positions). This is the starting point of the work. Each sample is converted to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where

- $\mathcal{V}$ is the set of N_v atomic numbers of the elements composing the structure
- $\mathcal{E}$ is the set of N_e bond distances

Bonds are selected according to a cutoff distance of 4 Å. Using this cutoff, no structure is forming isolated atoms and can be used for training. Bond distances are further expanded on a basis of Gaussian:

$$e_{ij} = \exp\left(-\frac{(r_{ij} - r_0)^2}{\sigma^2}\right)$$

where r_{ij} is the distance between atoms i and j,r_0 takes values at 100 locations linearly placed between 0 and 5, and the width $\sigma = 0.5$

3.3 Model Architecture

The proposed architecture is designed as a unified model capable of making predictions simultaneously at the node, edge, and graph levels.

The process begins with an input graph, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of atomic numbers and $\mathcal{E}$ is the set of bond distances, expanded using a basis of Gaussians (see previous section). An embedding layer is employed to map the atomic numbers to high-dimensional features, providing a richer representation of the atoms in the graph. The node and edge features undergo updates using the Edge-Featured Graph Attention (EGAT) mechanism described in Section (2.3.5). Each convolutional layer t has n_t independent attention heads; the output features are obtained as the concatenation of the output of each head. After T repetitions, the computation takes different paths:

- Graph prediction: the high-level atomic features are fed to a Set2Set layer computing a pooling function [77]. Unlike conventional LSTMs, Set2Set layers are irrespective of the order of the input sequence, which makes them well-suited as an aggregation operator. The same is applied to bond attributes. The resulting outputs are then concatenated and fed to a gated MLP [52] to obtain the final result.

- Node and Edge prediction: instead of following the graph prediction

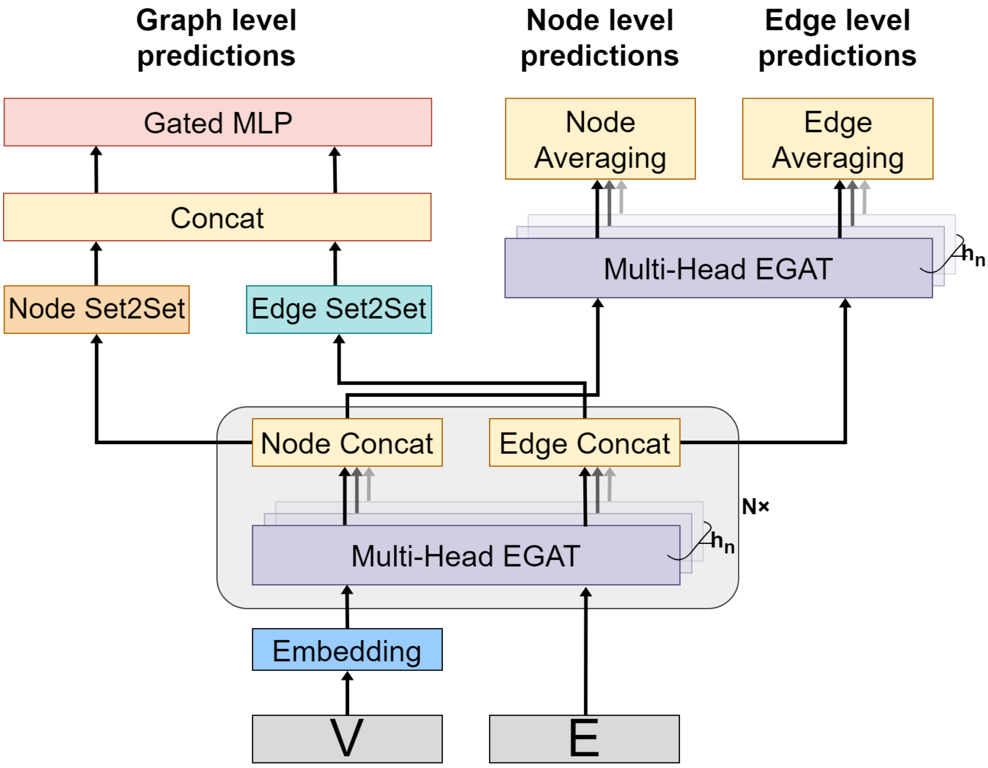


Figure 3.4: Schematic Representation of the network architecture. Multi-head EGAT produces h vectors which are then concatenated before feeding them to the next convolutional layer. For graph-level prediction, the set of node attributes and the set of edge attributes are input to Set2Set layers and outputs are concatenated; finally, the concatenated output is passed through a gated MLP producing the final prediction. For node and edge level predictions, the EGAT hidden dimension is set to 1 and the output comes from averaging across the heads.

path, atomic and bond features are directed to a final EGATConv layer equipped with n_{OUT} attention heads. The predictions for both node and graph levels are obtained by averaging the results across all the heads.

A schematic representation of the proposed architecture is available in Figure 3.4.

3.4 Tasks

The model has been trained on node classification and graph property prediction on a variety of targets for each dataset.

Masked Molecular Modelling Taking inspiration from Large Language Models, a similar type of task applied to molecular structures is introduced. The main idea is to use node embeddings to reconstruct the node types from the representation. Given the set of atomic elements contained in the input structure, some of them are randomly extracted and their atomic number is masked. The task of the network is to predict the atomic number of the masked nodes, optimizing the expectation of samples. This is similar to how first popular language models (e.g., BERT [21]) were pre-trained through a masked language model objective. The task is formulated as a multi-class node-classification problem. We minimize the cross-entropy between the actual atomic numbers f_i and the predicted ones, computed on the masked nodes only.

$$\mathcal{L}_n = -\mathbb{E}_{v_i \sim \mathcal{V}_M} \left[\sum_{m=1}^{K_n} f_{im} \log(\sigma(g_{\theta_n,i}(\mathcal{G}))) \right], \tag{3.1}$$

where $\mathcal{V}_M$ is the set of masked nodes, K_n is the number of atom types, $g_{\theta_n,i}$ computes the prediction for the masked node v_i.

Property Prediction The final goal of the model is to offer predictions on specific properties associated with a given structure. The available datasets often come with a series of properties associated with the structure (e.g., formation energy, Fermi energy, band gap, ...). For this reason, a graph-level prediction task is combined with the node prediction task, optimizing the mean-squared-error between the predicted and the target value of the property:

$$\mathcal{L}_p = \|y - g_{\theta_p}(\mathcal{G})\|^2 \tag{3.2}$$

The total loss to be minimized is thus given by

$$\mathcal{L}_t = \mathcal{L}_n + \mathcal{L}_p \tag{3.3}$$

Chapter 4

Results

The same architecture has been trained and tested on three different tasks:

1. *Property prediction only*: the model receives an integral graph structure as the input, and outputs the graph-level property.

2. *Masked node prediction only*: the model receives a graph that contains 20% of masked nodes, and returns in output the labeled nodes.

3. *Masked node prediction + property prediction*: the model receives a graph that contains 20% of masked nodes, and outputs the graph-level property together with labeled nodes.

For each property available in the datasets described in Section 3.1 a different model has been trained. Each model has been trained and tested on a 80:10:10 split, for 500 epochs on an Nvidia A40 GPU, using Adam optimizer with initial learning rate set to $0,001$ along with a cosine decay schedule, early stopping if no improvement is made for 150 subsequent epochs. Since the focus of this work is about making structural predictions through masked molecular modeling, from now on we will refer to the model as MaMoMo (Masked Molecular Model).

The node model is used to assess the ability to understand atomic interactions by asking to reconstruct the original structure given some missing atoms.

The property model serves to assess the capacity to provide accurate prediction on targets of interest. The node+property model has the objective of evaluating whether training the model jointly on both node-level and graph-level tasks has any adverse impact on its performance.

It is worth noting that there is a minor flaw in the current representation of MP structures. MP crystals have sizes ranging from a single element to many dozens of atoms. This is due to them being represented with their unit cell i.e, the smallest repeating unit having the full symmetry of the crystal structure, and the periodic boundary conditions (PBCs). The structures with the smallest number of atoms have been enlarged by replicating the unit cell on every direction. Although this is fine for masked molecular modelling, this is conceptually wrong for property prediction, since target values are referred to a theoretically infinite structure, and it should be necessary to use PBCs or to consider unit cells only, for a better consistency. This issue will need to be addressed in future work.

4.1 Masked Node Prediction

The model reaches very good results when trained exclusively for masked node prediction. Overall accuracy and F1 score are above 99% for GO and QM9 datasets, which have a little number of classes compared to MP, where the accuracy and F1 are 95% and 92% respectively. Combining structural prediction with property prediction causes the model to lose precision, though the effect is more evident when the prediction is related to extensive properties of the structures (e.g., total energy, enthalpy, Gibbs free energy,...). In general, when the model struggles more to fit a property, also masked node prediction is affected, but this is understandable given the combined loss.

Tables 4.1,4.2 and 4.3 resume the accuracy and F1 scores for each of the trained models. More details are available in Appendix A and B.

model	Accuracy	F1
node-only	99.998%	99.997%
E_{tot}*	86.102%	77.498%
ζ_0	99.998%	99.997%
χ	99.996%	99.995%
EA	99.998%	99.997%
IP	99.998%	99.997%

Table 4.1: Comparison of prediction metrics between different models on CSIRO Graphene Oxide Dataset. E_{tot}: total energy; ζ_0: Fermi energy; χ: electronegativity; EA: eletron affinity; IP: ionization potential. *Extensive property.

4.2 Property Prediction

For the GO dataset, the only available benchmark is coming from GrapheNet, a convolutional neural network under development at CNR-ISMN of Bologna. For this reason, we have trained a MEGNet [15] model to be used as a further comparison element. The code and the training loop are taken from the original implementation but we have increased the number of parameters in order to achieve comparable results. For QM9 and MP, there is already a rich literature, and we added two models – SchNet [71] for QM9 and CGNN [84] for MP – as further reference, together with the already cited MEGNet.

Tables 4.4,4.5 and 4.6 illustrate the comparison of the mean absolute errors (MAEs) between the performance of different models applied to each of the datasets. Results exhibit a comparable order of magnitude and show great promise. We even improved on some targets for the GO dataset. Nevertheless, the model requires further refinement to attain the chemical accuracy targets outlined by Faber et al. [25] across various properties.

Performance degradation was expected in property prediction with masked nodes. We recall that the model receives a degraded graph as input, and has to both reconstruct the structure and predict the target property. While this is quite evident in QM9, where molecules are small and removing a fifth of the atoms has a more substantial impact, it is less so in MP. Notably, in graphenes, there’s even an improvement in performance for certain targets. Two likely

model	Accuracy	F1
node-only	99.986%	99.936%
ϵ_{HOMO}	99.982%	99.930%
ϵ_{LUMO}	99.972%	99.829%
$\Delta\epsilon$	99.982%	99.905%
$ZPVE$*	99.978%	99.873%
μ	99.834%	98.794%
α	99.195%	94.823%
$\langle R^2\rangle$*	98.619%	89.842%
U_0*	97.860%	86.845%
U*	96.306%	80.315%
H*	97.653%	86.837%
G*	97.281%	85.947%
C_v	99.789%	97.892%

Table 4.2: Comparison of prediction metrics between different models on QM9 Dataset. ϵ_{HOMO}: highest occupied molecular orbital; ϵ_{LUMO}: lowest unoccupied molecular orbital; $\Delta\epsilon$: energy gap; $ZPVE$: zero-point vibrational energy; μ: dipole moment; α: isotropic polarizability; $\langle R^2\rangle$: electronic spatial extent; U_0: internal energy at 0 K; U: internal energy at 298 K; H: enthalpy at 298 K; G: Gibbs free energy at 298 K; C_v: heat capacity at 298 K. *Extensive property.

model	Accuracy	F1
node-only	95.751%	92.983%
E_f	94.681%	91.385%
$\Delta\epsilon$	92.221%	87.048%

Table 4.3: Comparison of prediction metrics between different models on Materials Project Dataset. E_f: formation energy per atom; $\Delta\epsilon$: band gap.

explanations emerge: 1) CSIRO's graphene oxides exhibit regular structures with only three different atomic types, simplifying the reconstruction process, and 2) instructing the model to learn structural information has effectively enhanced its understanding of the system.

Extensive properties are those whose values scale proportionally with the size of the underlying structure. It is clear that our model faces challenges when handling this kind of properties. The authors of MEGNet integrate a global state feature, evolving alongside node and edge features, to the architecture. It's plausible that a significant portion of information regarding the overall graph property, whether intensive or extensive, resides within these global state features. This work has taken inspiration from MEGNet in the adoption of Set2Set layers as the final node and edge aggregation operator, but has dropped the global state. A subsequent work from the authors of MEGNet, M3GNet [14], also drops the global state feature, but introduces differentiated aggregation layers depending on whether the target property is an intensive property or an extensive one.

This leads us to consider the possibility that there may be limitations in the capacity of Set2Set layers, which have been adopted as the final aggregation operators for nodes and edges in our work. Problems can lie in using them as the sole aggregation mechanism, in the initialization of their weights, or also in the handling of extensive values without any kind of normalization (e.g., fitting the model on per atom properties instead of total ones), and should be better investigated in a future work.

It should be noted that, for some properties of QM9 – specifically, electronic spatial extent ($\langle R^2 \rangle$), internal energy at 0K (U_0), internal energy at 298.15K (U), enthalpy(H) – results are not comparable, since we have later discovered that previous works have used a modified version of QM9 dataset, unfortunately not available anymore, where these properties are on completely different distribution and scale. Table 4.5 underscores the properties where discrepancies exist. While errors may seem huge in absolute terms, in relative

property	units	GrapheNet	MEGNet	MaMoMo (graph)	MaMoMo (node+graph)
E_{tot}*	eV	3.920	104.243	1.047e+4	9.9e+3
ζ_0	eV	0.047	0.047	0.048	0.040
χ	eV	0.067	0.070	0.065	0.060
EA	eV	0.084	0.080	0.078	0.075
IP	eV	0.064	0.079	0.071	0.070

Table 4.4: Comparison of MAEs between different models on CSIRO Graphene Oxide Dataset. *Extensive property.

	units	SchNet	MEGNet	MaMoMo (graph)	MaMoMo (node+graph)
ϵ_{HOMO}	eV	0.041	0.038	0.069	0.193
ϵ_{LUMO}	eV	0.044	0.031	0.063	0.214
$\Delta\epsilon$	eV	0.066	0.061	0.101	0.260
$ZPVE$*	meV	1.43	1.40	3.894	88.23
μ	D	0.050	0.040	0.242	0.342
α	bohr^{-3}	0.081	0.083	0.235	0.723
$\underline{\langle R^2 \rangle *}$	bohr^{-2}	0.302	0.265	8.724	22.944
$\underline{U_0 *}$	eV	0.012	0.009	17.480	142.9
$\underline{U *}$	eV	0.013	0.010	12.873	139.5
$\underline{H *}$	eV	0.012	0.010	10.915	145.5
$\underline{G *}$	eV	0.012	0.010	12.136	154.7
C_v	cal (mol K)$^{-1}$	0.029	0.030	0.064	0.158

Table 4.5: Comparison of MAEs between different models on QM9 Dataset. *Extensive property. Underlined targets are not comparable because of a mismatch in the dataset used by the works.

term they are still very good. For a more comprehensive evaluation of the results, please refer to Appendix A and B.

4.3 Attention Maps

One of the remarkable advantages of incorporating graph attention mechanisms into our model is its inherent ability to extract and visualize attention maps. These maps provide a transparent view of how the model processes information within a graph. By visualizing which connections the model pays the most attention to, we gain information into its decision-making process,

property	units	CGNN	MEGNet	MaMoMo (graph)	MaMoMo (node+graph)
E_f	eV atom^{-1}	0.039	0.028	0.064	0.098
$\Delta\epsilon$	eV	0.388	0.330	0.407	0.408

Table 4.6: Comparison of MAEs between different models on Materials Project Dataset.

getting valuable insights into the learned representations.

Stunningly, the attention maps that the model has learned are resembling very closely what is the typical representation commonly used among the scientific community. Having set a quite high distance threshold (4 Å) for connecting atoms when representing a structure, the resulting graphs are very densely connected and thus very different from the classical view – most bond distances are below 2 Å. This makes even more impressive that the model was able to learn representations that are meaningful to us even when given such complex starting structures.

The attention maps also highlight the differences lying among the three structural classes that are present in the dataset. When dealing with molecules, such as those found in the QM9 dataset, much of the essential information can often be inferred from stoichiometry (the ratios of elements) and topology (the arrangement of atoms). In this context, the primary focus of the model is to extract meaningful chemical bonds between atoms. These bonds serve as critical features, and the model leverages them to make accurate predictions about molecular properties. Bonds in molecules are well-defined and provide a clear framework for understanding the molecule's structure and behavior.

On the contrary, the realm of crystals presents a different challenge. Crystalline structures are inherently complex, characterized by the arrangement of atoms or ions in a repeating and three-dimensional lattice. In crystals, interactions between atoms occur in a highly intricate and coordinated manner. In such a system, the traditional concept of individual atomic bonds, as observed in molecules, is not particularly meaningful. Instead, the behavior of a crystal emerges from the collective interactions of its constituent atoms within the lattice. These interactions involve long-range forces and are influenced by factors like crystal symmetry, periodicity, and defects.

Nanographenes possess characteristics common to both the molecular compositions found in QM9 and the crystalline structures encountered in MP. Firstly, they mostly fall within the realm of organic materials, primarily composed of carbon and hydrogen atoms, and feature robust, directional bonds akin to those observed in molecular systems. Secondly, as nanostructured materials they exhibit a form of short-range periodicity, for example the hexagonal carbon lattice in graphene oxide, reminiscent of the crystalline periodicity observed in crystals. However, unless we are dealing with pristine structures, like pure graphene sheets, periodicity is frequently disrupted by the presence of defects, dislocations, grain boundaries, adsorbates, and various forms of disorder. GO structures exhibit functional groups, primarily consisting of oxygen atoms (O) and hydroxyl groups (OH), bonded to the carbon atoms on their surface. Imperfections profoundly impact the properties of nanostructured materials. For instance, in the case of graphene sheets, properties can vary significantly based on factors such as shape, dimension, the density of defects, the influence of adsorbates, and more. Therefore, comprehensive analysis and modeling of nanostructures require a holistic consideration of these diverse characteristics, rendering them particularly challenging to work with.

It's worth noting that nanostructured materials are often examined using the same simulation methods employed for crystalline materials. However, a fundamental distinction arises in the treatment of boundaries. In crystalline simulations, the lattice structure dictates the simulation cell's definition, whereas in nanostructured simulations large simulation cells, aimed at minimizing boundary effects and ensuring a more realistic representation of these intricate systems, are defined.

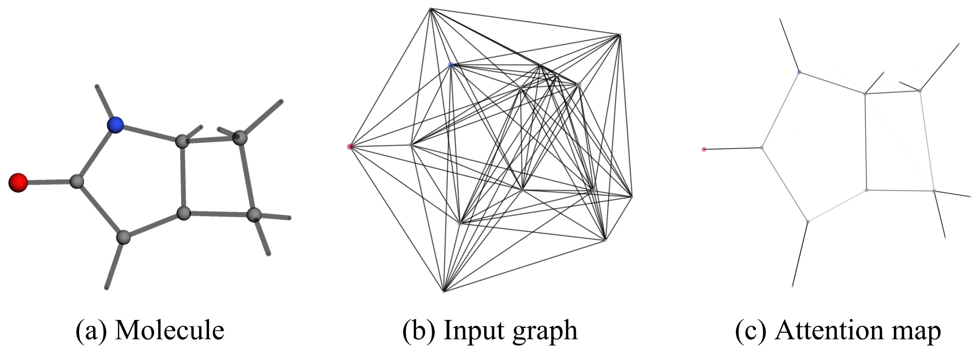

(a) Molecule (b) Input graph (c) Attention map

Figure 4.1: A molecule randomly extracted from QM9 test set. 4.1a: visualization of the molecule using a molecular graphics toolkit, 4.1b: graph structure input to the network, 4.1c: one attention map extracted from the network.

All these differences are clear when visualizing the attention maps. For QM9 and GO, certain layers perfectly reproduce the topology of the structure on many samples; Figure 4.1 and 4.3 show the comparison between the input structure visualized with py3dmol (`https://pypi.org/project/py3Dmol/`), a molecular graphics software, and one attention map extracted from a convolutional layer of the network. The resemblance is striking, also considering what the network receives as input. Recalling what we mentioned above, we should not expect crystals to show an ordered (bond-wise) structure, and so it is. Most attention maps resemble what is presented in figure 4.2. Other attention maps focus on different kinds of interaction, for example hydrogen bonds, or boundaries in graphene sheets; during masked molecular modelling, attention is deviated by the presence of masked nodes, suggesting that the model is focusing on nearby elements to infer the masked atom type. More images are available in Appendix C

4.4 Conclusions

Before drawing the conclusions, we recall the reasons that have motivated us to start this work:

- The world of computational chemistry is increasingly oriented towards

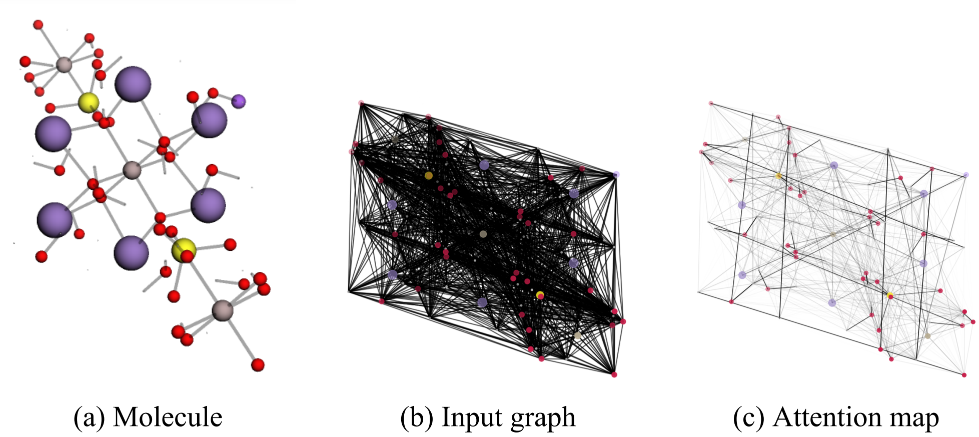


Figure 4.2: A molecule randomly extracted from MP test set. 4.2a: visualization of the molecule using a molecular graphics toolkit, 4.2b: graph structure input to the network, 4.2c: one attention map extracted from the network.

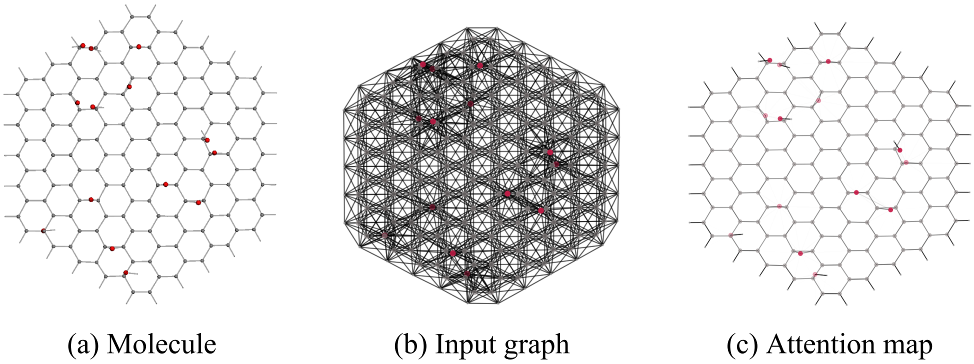


Figure 4.3: A molecule randomly extracted from GO test set. 4.3a: visualization of the molecule using a molecular graphics toolkit, 4.3b: graph structure input to the network, 4.3c: one attention map extracted from the network.

data science and machine learning, which are emerging as the fourth pillar of natural sciences.

- Data-driven CMS has the power to speed-up the research, but relies on the availability of datasets, which production is expensive and time-consuming.
- The chemical space is largely still unexplored. Moreover, there are many branches of CMS addressing very different classes, at different scales and theoretical levels.
- Datasets are rarely comparable, and reproducibility is a problem even when provided with the parametrization. There is an open research field towards the creation of data integration frameworks.
- Structural representation for ML models must be simple and flexible enough to handle all the aforementioned differences.

The one presented here is an initial exploration of the possibility of establishing a simple, unified framework for the prediction of properties on materials. Its inception was driven by the aspiration to create a generic model, trained to perform structural prediction on a multitude of structures, spanning from simple molecules to crystals and large nanostructured systems. The goal of this model is to learn intrinsic quantomechanical properties of the matter, and then be easily tuned to specific tasks, classes and theory levels, requiring little data. This adaptability is crucial given the scarcity and costliness of data production, a significant bottleneck in this field.

The work showed here presents just a preliminary assessment of the method and the architecture. Despite minimal hyperparameter optimization, it has obtained promising results, demonstrating its ability to handle various properties and classes. However, vulnerabilities in certain tasks have surfaced, and further improvement are necessary to reach chemical accuracy.

Last but not least, we have showed how this approach is enriched by an interpretable internal representation manifested in the attention maps. This not only enhances our understanding of the models but also paves the way for advancements in future research.

4.5 Future Perspectives

The model presented in Section 3.3 has been designed to also perform edge-level predictions, in addition to node- and graph-level predictions. Masked molecular modeling, as presented until now, is a ”simple” task, considering that the model still has visibility of the distances between masked atoms and their neighborhood. An advanced alternative, as proposed by Hao et al. [36], involves sampling a fraction of edges from the structure, then masking both the edges and the connected nodes, tasking the model with their prediction. This requires the model to capture more information about interatomic relations and the arrangement of atoms within the structure.

After having validated the capacity of the model to perform property prediction, the natural direction of future works would be to collect and aggregate as many data as possible from every possible source, in order to cover as much as possible the chemical space, and train the model on structural prediction only – which, at that point, would be the only meaningful option to integrate different datasets, given their incompatibilities. Such model should represent the equivalent of what Large Language Models represent for language: a foundational model, capturing general knowledge, flexible enough to be tuned to specific tasks without much effort [63, 65]. Of course, such envisioned model implies a massive work in term of data integration and requires the collaboration of many branches of the scientific community. Other options to be explored could be student-teacher approaches, active learning and transfer learning.

Appendix A

Supplementary Tables

Target	Unit	MAE	MSE	R2
E_{tot}	eV	1.047e+4	1.766e+8	0.578
ζ_0	eV	0.048	0.006	0.910
χ	eV	0.065	0.016	0.794
EA	eV	0.078	0.030	0.738
IP	eV	0.071	0.023	0.757

Table A.1: Property-only model: detailed metrics for GO targets.

Target	Unit	MAE	MSE	R2
ϵ_{HOMO}	eV	0.069	0.010	0.971
ϵ_{LUMO}	eV	0.063	0.008	0.995
$\Delta\epsilon$	eV	0.101	0.021	0.988
$ZPVE$	meV	3.894	1099.0	0.999
μ	D	0.242	0.136	0.941
α	bohr$^-3$	0.235	0.203	0.997
$\langle R^2 \rangle$	bohr$^-2$	8.724	630.353	0.992
U_0	eV	17.480	3425.9	0.997
U	eV	12.873	1706.9	0.999
H	eV	10.915	1202.9	0.999
G	eV	12.136	1294.3	0.999
C_v	cal(mol K)$^-1$	0.064	0.028	0.998

Table A.2: Property-only model: detailed metrics for QM9 targets.

Target	Unit	MAE	MSE	R2
E_f	eV atom1	0.064	0.016	0.986
$\Delta\epsilon$	eV	0.407	0.495	0.803

Table A.3: Property-only model: detailed metrics for MP targets.

Target	Unit	MAE	MSE	R2
E_{tot}	eV	9.9e+3($\pm$2.6e+2)	1.6e+8($\pm$9.2e+6)	0.608($\pm$2.1e–2)
ζ_0	eV	0.040($\pm$5.6e–4)	0.005($\pm$1.8e–4)	0.936($\pm$2.5e–3)
χ	eV	0.060($\pm$7.1e–4)	0.015($\pm$3.3e–4)	0.803($\pm$4.3e–3)
EA	eV	0.075($\pm$8.8e–4)	0.030($\pm$7.2e–4)	0.738($\pm$6.1e–3)
IP	eV	0.070($\pm$1.2e–3)	0.025($\pm$6.3e–4)	0.738($\pm$6.6e–3)

Table A.4: Node+property model: detailed metrics for GO targets. Mean and standard deviation values are derived from 30 iterations on the dataset, with distinct random masking of the nodes.

Target	Unit	MAE	MSE	R2
ϵ_{HOMO}	eV	0.193($\pm$1e–3)	0.077($\pm$2e–2)	$0.78(6 \pm 5$e–2)
ϵ_{LUMO}	eV	0.214($\pm$9e–4)	0.084($\pm$1e–3)	0.947($\pm$6e–4)
$\Delta\epsilon$	eV	0.260($\pm$1e–3)	0.123($\pm$1e–3)	0.927($\pm$6e–4)
$ZPVE$	meV	88.237($\pm$7e–1)	1.6e+4($\pm$3e+2)	0.979($\pm$4e–4)
μ	D	0.342($\pm$2e–3)	0.241($\pm$4e–3)	0.895($\pm$1e–3)
α	bohr^{-}3	0.723($\pm$5e–3)	1.226($\pm$3e–2)	0.981($\pm$5e–4)
$\langle R^2 \rangle$	bohr^{-}2	22.944($\pm$1e–1)	1.3e+3($\pm$5e+1)	0.983($\pm$6e–4)
U_0	eV	142.9($\pm$1e+0)	4.6e+4($\pm$1e+3)	0.961($\pm$1e–3)
U	eV	139.5($\pm$9e–1)	4.9e+4($\pm$4e+3)	0.958($\pm$3e–3)
H	eV	145.5($\pm$1e+0)	4.8e+4($\pm$2e+3)	0.959($\pm$1e–3)
G	eV	154.7($\pm$1e+0)	5.7e+4($\pm$3e+3)	0.951($\pm$2e–3)
C_v	cal(mol K)$^{-}$1	0.158($\pm$1e–3)	0.057($\pm$3e–3)	0.997($\pm$2e–4)

Table A.5: Node+property model: detailed metrics for QM9 targets. Mean and standard deviation values are derived from 30 iterations on the dataset, with distinct random masking of the nodes.

Target	Unit	MAE	MSE	R2
E_f	eV atom1	0.098($\pm$6.9e–4)	0.025($\pm$3.3e–4)	0.978($\pm$2.9e–4)
$\Delta\epsilon$	eV	0.408($\pm$2.4e–3)	0.528($\pm$7.6e–3)	0.790($\pm$3.0e–3)

Table A.6: Node+property model: detailed metrics for MP targets. Mean and standard deviation values are derived from 30 iterations on the dataset, with distinct random masking of the nodes.

Target	Accuracy	F1 (macro)	F1 (weighted)
node-only	$99.998(\pm0.001)\%$	$99.997(\pm0.002)\%$	$99.998(\pm0.001)\%$
E_{tot}	$86.102(\pm0.106)\%$	$77.498(\pm0.165)\%$	$85.683(\pm0.112)\%$
ζ_0	$99.998(\pm0.002)\%$	$99.997(\pm0.002)\%$	$99.998(\pm0.002)\%$
χ	$99.996(\pm0.002)\%$	$99.995(\pm0.002)\%$	$99.996(\pm0.002)\%$
EA	$99.998(\pm0.002)\%$	$99.997(\pm0.002)\%$	$99.998(\pm0.002)\%$
IP	$99.998(\pm0.002)\%$	$99.997(\pm0.003)\%$	$99.998(\pm0.002)\%$

Table A.7: Detailed metrics for masked node prediction on GO. Mean and standard deviation values are derived from 30 iterations on the dataset, with distinct random masking of the nodes.

Target	Accuracy	F1 (macro)	F1 (weighted)
node-only	$99.986(\pm0.006)\%$	$99.936(\pm0.080)\%$	$99.986(\pm0.006)\%$
ϵ_{HOMO}	$99.982(\pm0.007)\%$	$99.930(\pm0.067)\%$	$99.982(\pm0.007)\%$
ϵ_{LUMO}	$99.972(\pm0.009)\%$	$99.829(\pm0.144)\%$	$99.972(\pm0.009)\%$
$\Delta\epsilon$	$99.982(\pm0.007)\%$	$99.905(\pm0.103)\%$	$99.982(\pm0.007)\%$
$ZPVE$	$99.978(\pm0.007)\%$	$99.873(\pm0.107)\%$	$99.978(\pm0.007)\%$
μ	$99.834(\pm0.021)\%$	$98.794(\pm0.361)\%$	$99.834(\pm0.021)\%$
α	$99.195(\pm0.041)\%$	$94.823(\pm0.724)\%$	$99.190(\pm0.042)\%$
$\langle R^2\rangle$	$98.619(\pm0.046)\%$	$89.842(\pm0.973)\%$	$98.609(\pm0.046)\%$
U_0	$97.860(\pm0.070)\%$	$86.845(\pm0.973)\%$	$97.821(\pm0.073)\%$
U	$96.306(\pm0.091)\%$	$80.315(\pm1.011)\%$	$96.165(\pm0.095)\%$
H	$97.653(\pm0.062)\%$	$86.837(\pm1.078)\%$	$97.606(\pm0.064)\%$
G	$97.281(\pm0.058)\%$	$85.947(\pm0.999)\%$	$97.206(\pm0.061)\%$
C_v	$99.789(\pm0.021)\%$	$97.892(\pm0.619)\%$	$99.789(\pm0.021)\%$

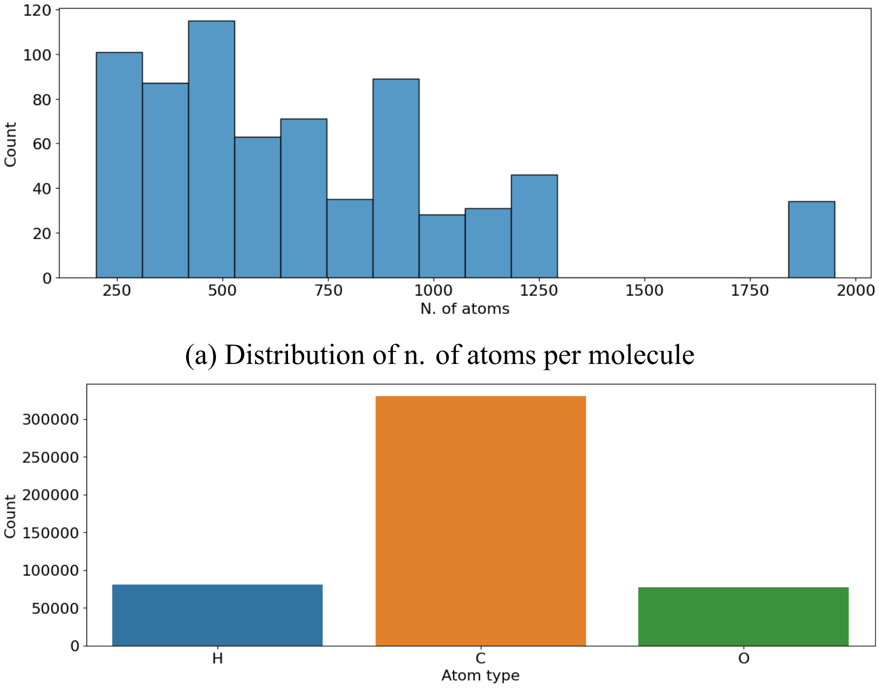
Table A.8: Detailed metrics for masked node prediction on QM9. Mean and standard deviation values are derived from 30 iterations on the dataset, with distinct random masking of the nodes.

Target	Accuracy	F1 (macro)	F1 (weighted)
node-only	$95.751(\pm0.091)\%$	$92.983(\pm0.265)\%$	$95.755(\pm0.091)\%$
E_f	$94.681(\pm0.062)\%$	$91.385(\pm0.211)\%$	$94.684(\pm0.062)\%$
$\Delta\epsilon$	$92.221(\pm0.135)\%$	$87.048(\pm0.296)\%$	$92.224(\pm0.134)\%$

Table A.9: Detailed metrics for masked node prediction on MP. Mean and standard deviation values are derived from 30 iterations on the dataset, with distinct random masking of the nodes.

Appendix B

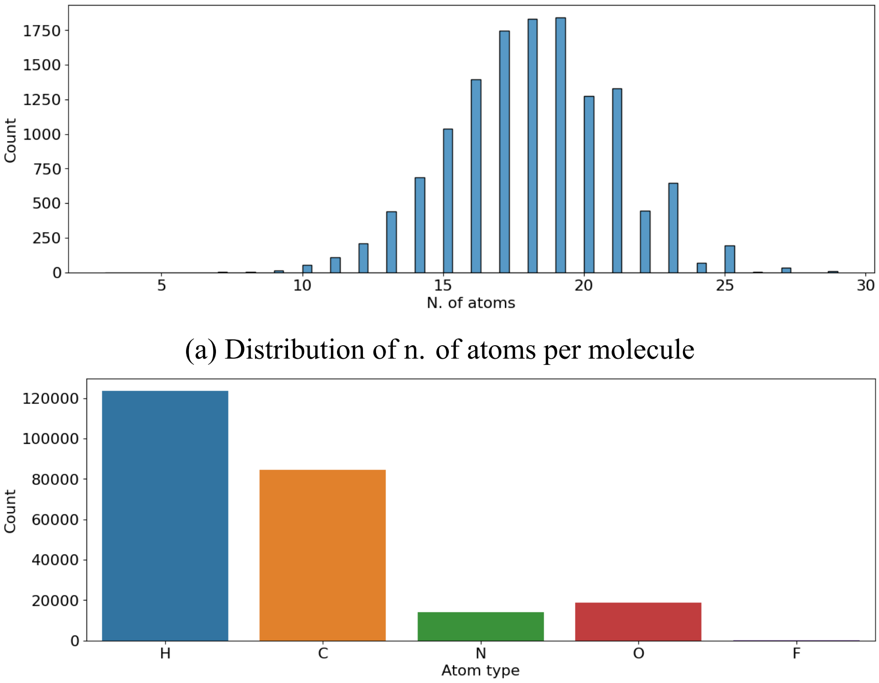
Supplementary Figures



(a) Distribution of n. of atoms per molecule

(b) Atom types distribution

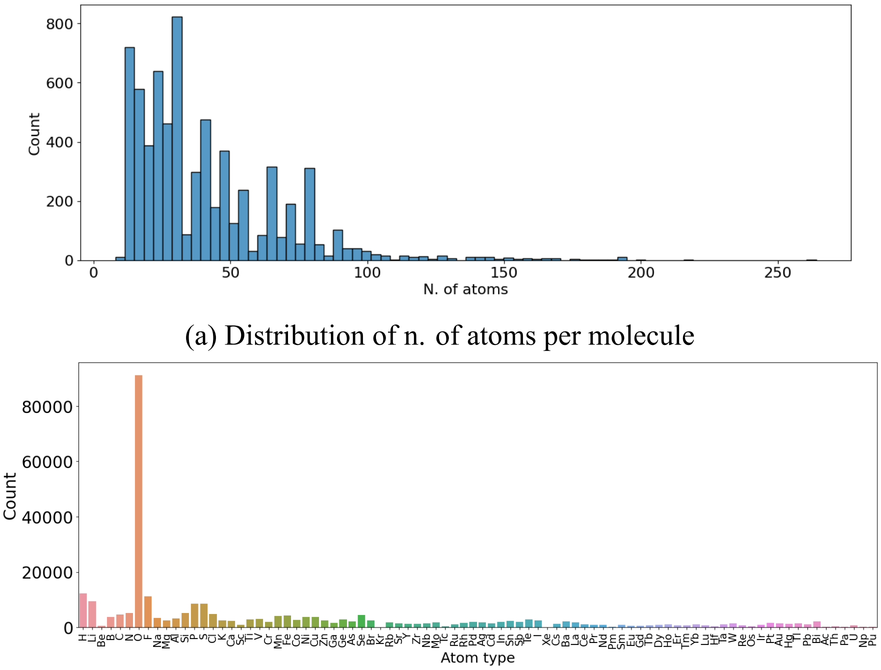
Figure B.1: Distribution of structure sizes (B.1a) and atom types (B.1b) over the entire GO dataset.



(a) Distribution of n. of atoms per molecule

(b) Atom types distribution

Figure B.2: Distribution of structure sizes (B.2a) and atom types (B.2b) over the entire QM9 dataset.



(a) Distribution of n. of atoms per molecule

(b) Atom types distribution

Figure B.3: Distribution of structure sizes (B.3a) and atom types (B.3b) over the entire MP dataset.

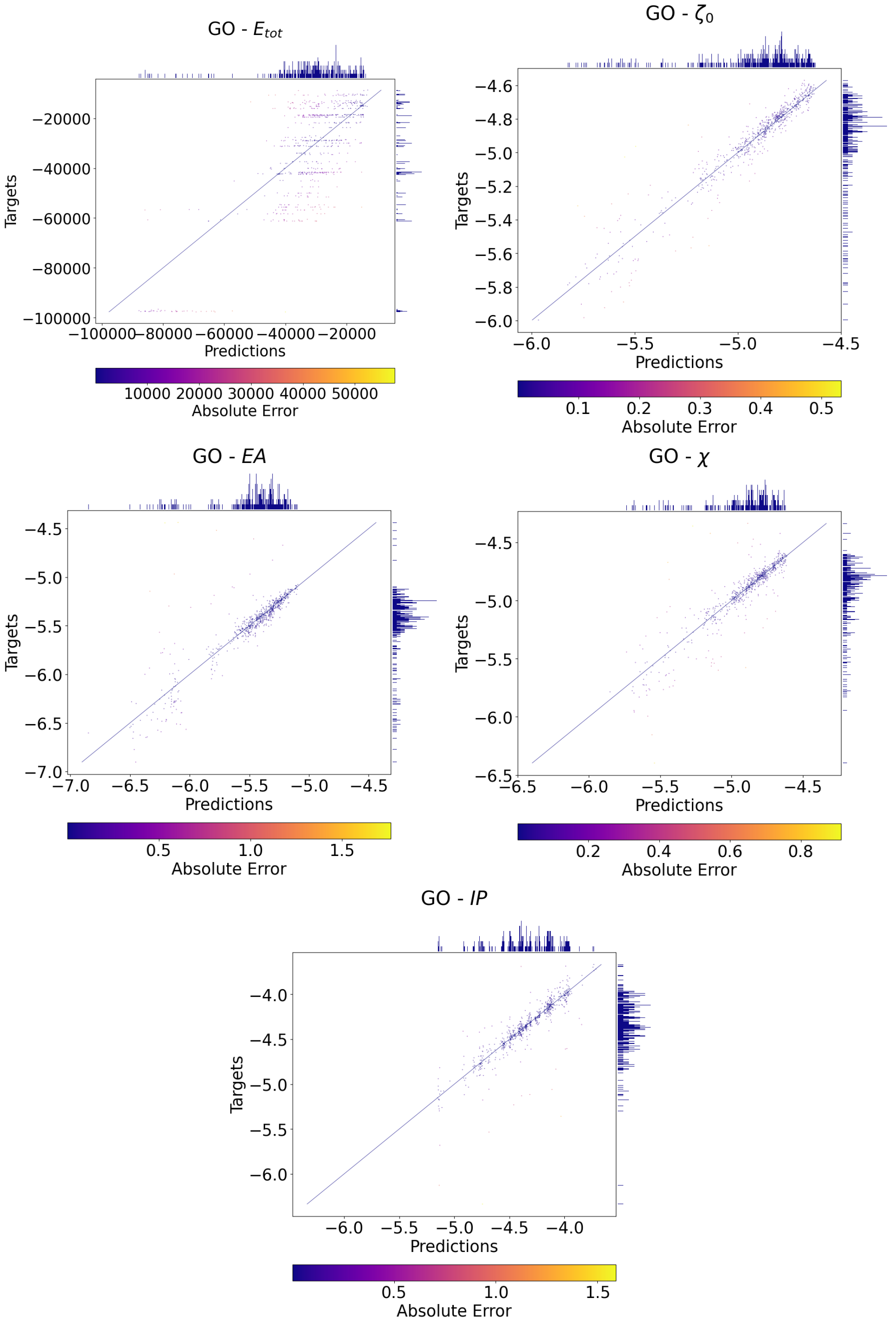


Figure B.4: Property-only model: parity plots for GO targets.

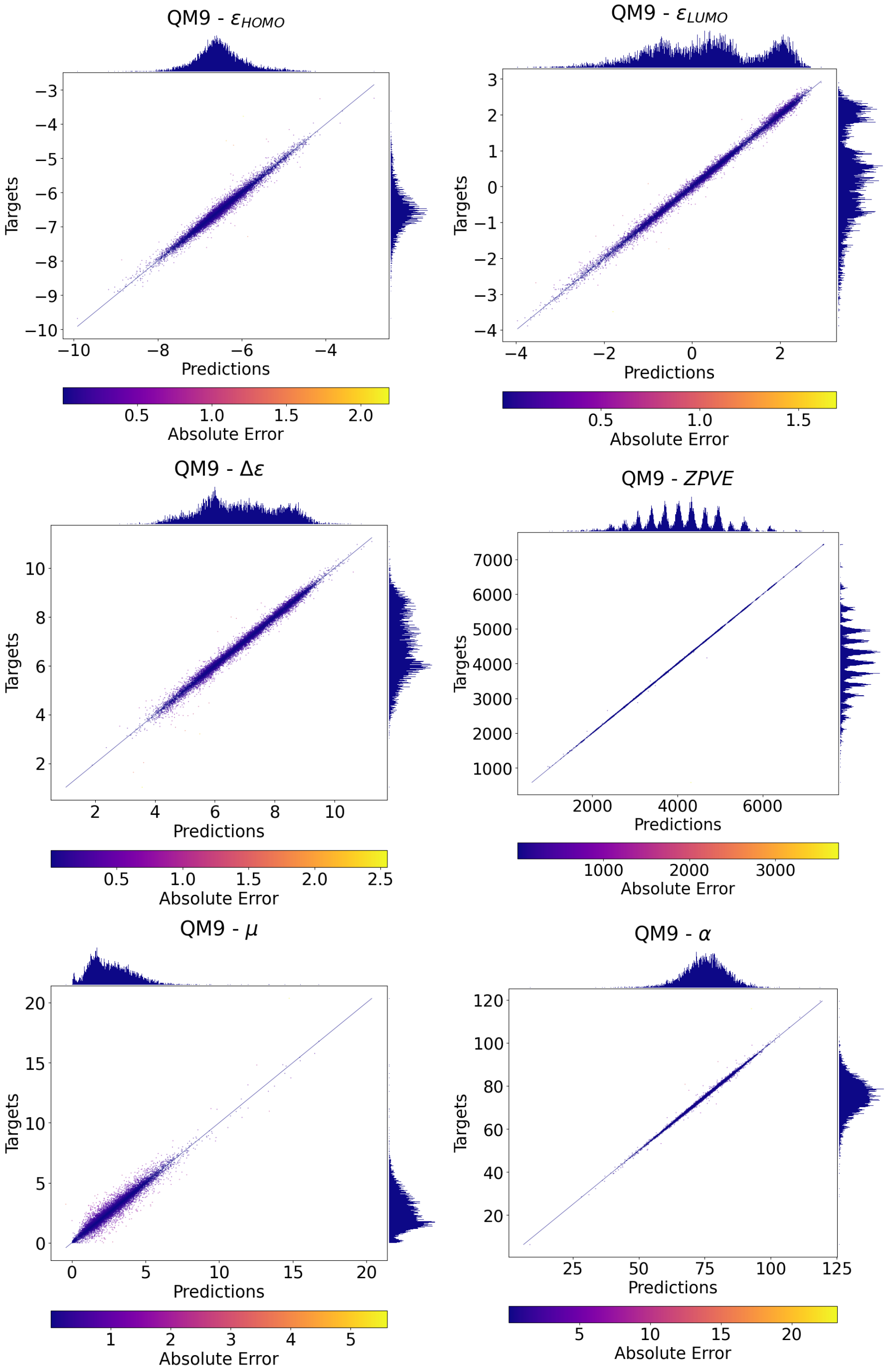


Figure B.5: Property-only model: parity plots for QM9 targets (1/2).

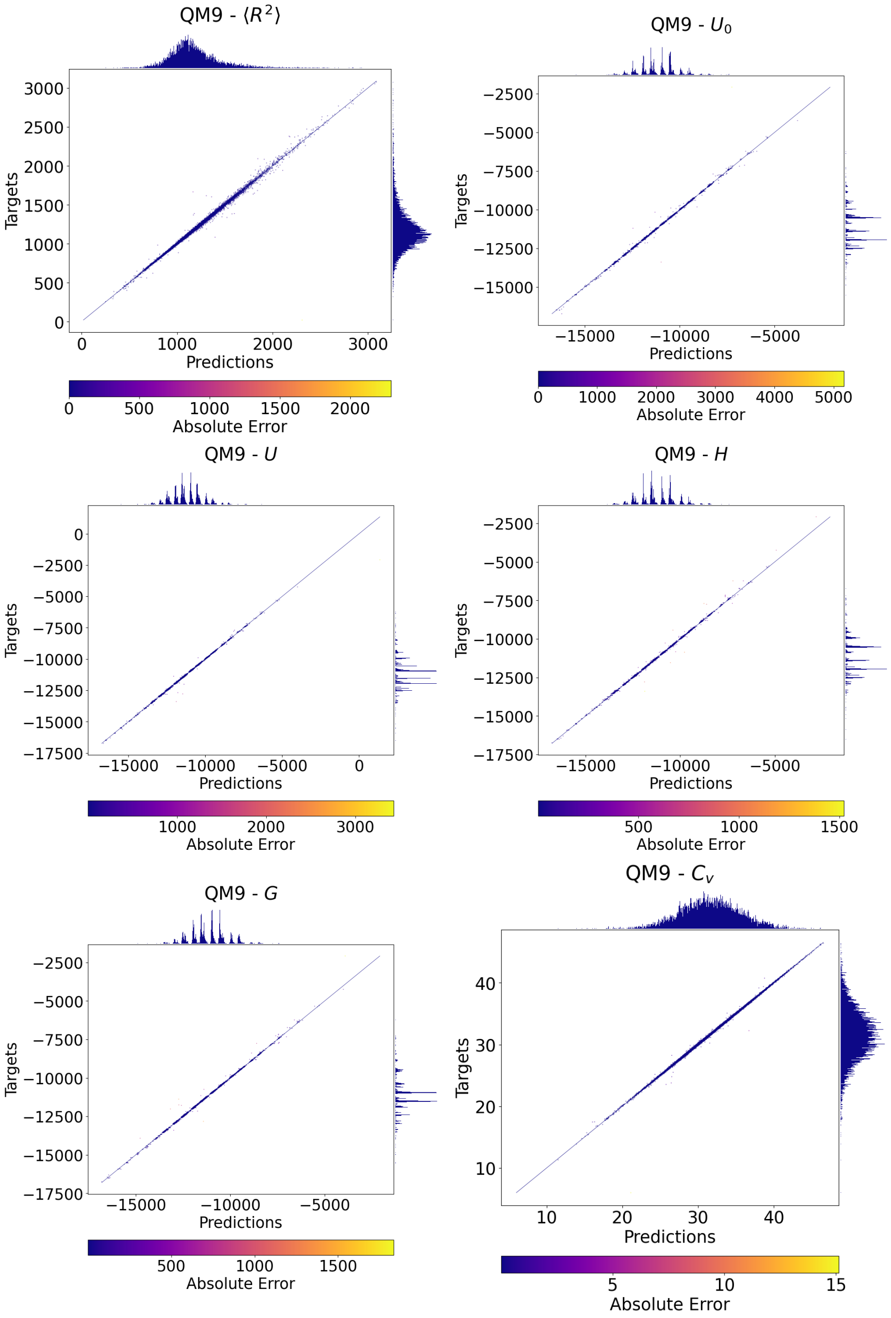


Figure B.6: Property-only model: parity plots for QM9 targets (2/2).

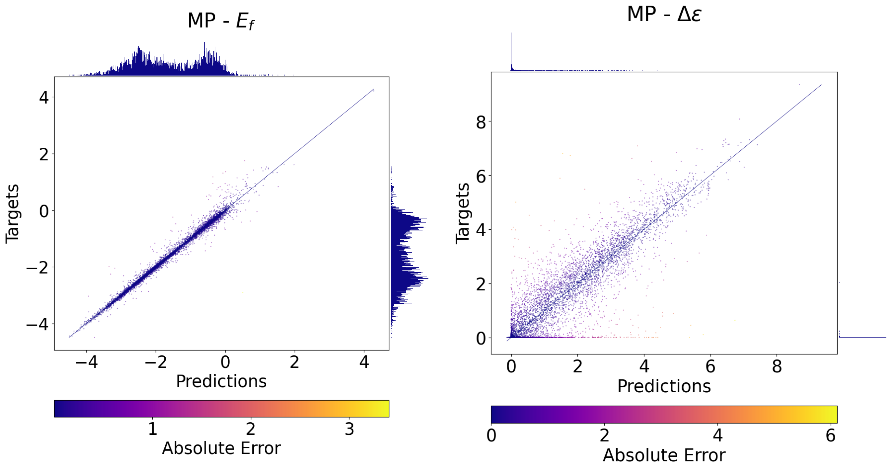


Figure B.7: Property-only model: parity plots for MP targets.

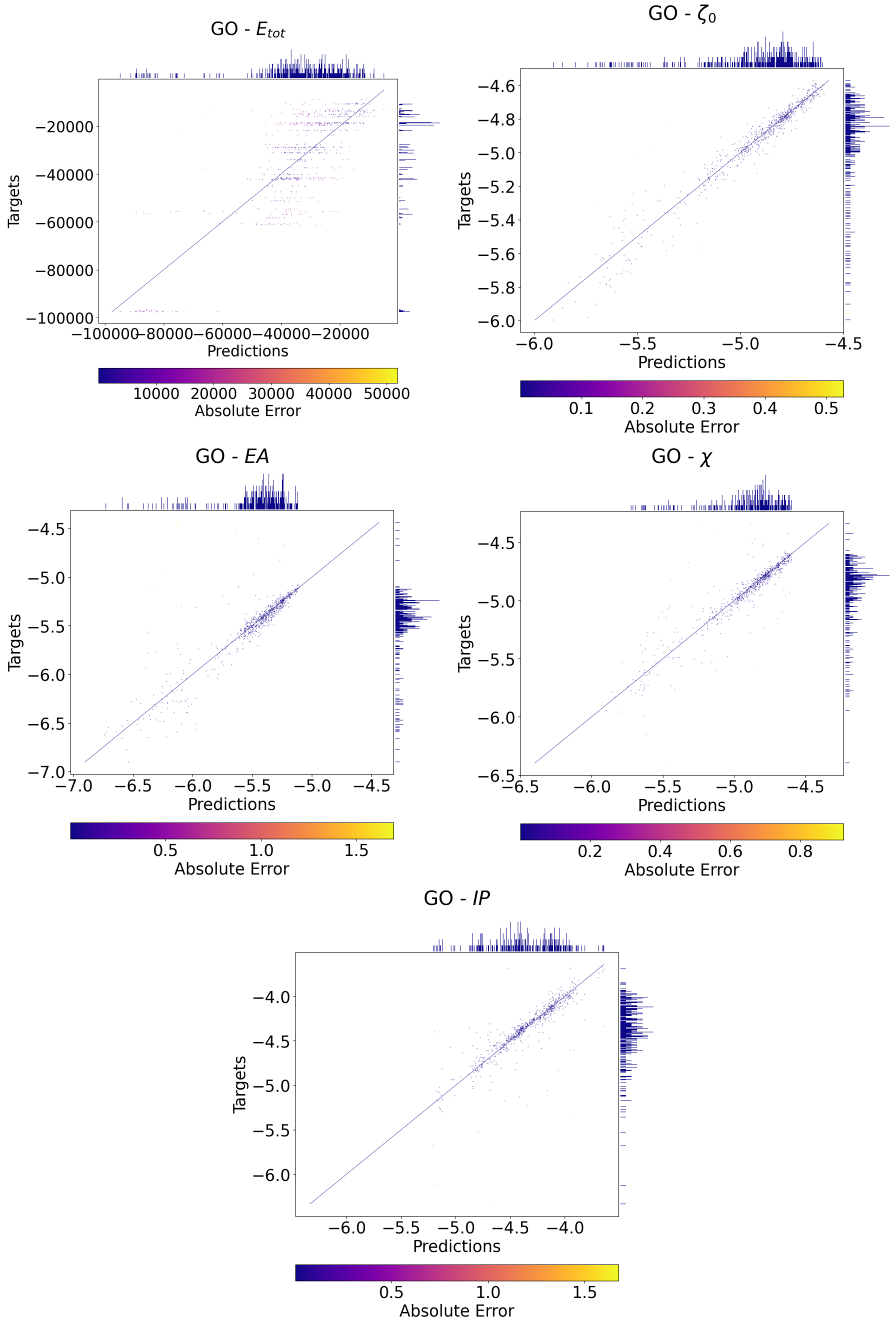


Figure B.8: Node+property model: examples of parity plots for GO targets.

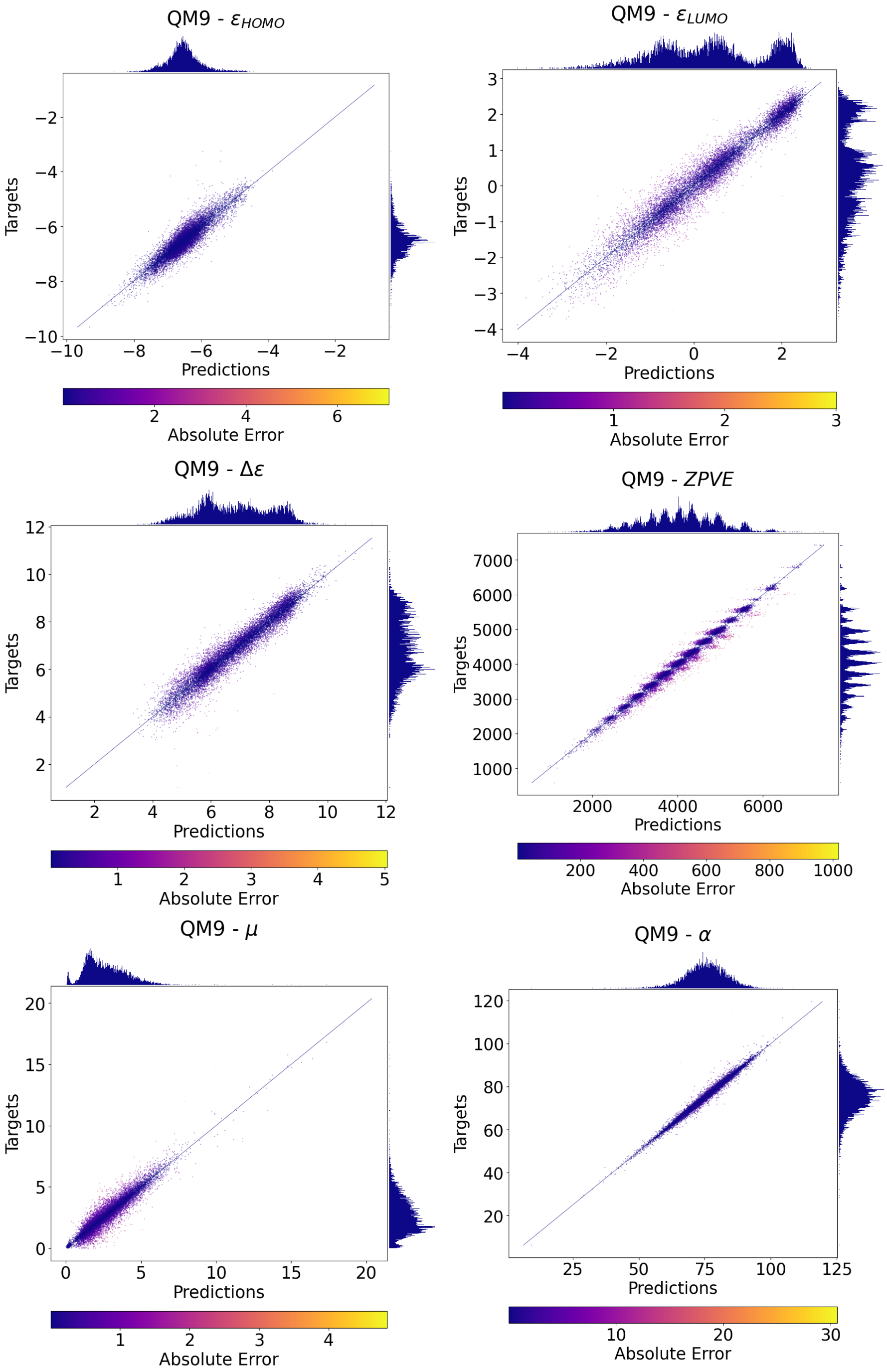


Figure B.9: Node+property model: examples of parity plots for QM9 targets (1/2).

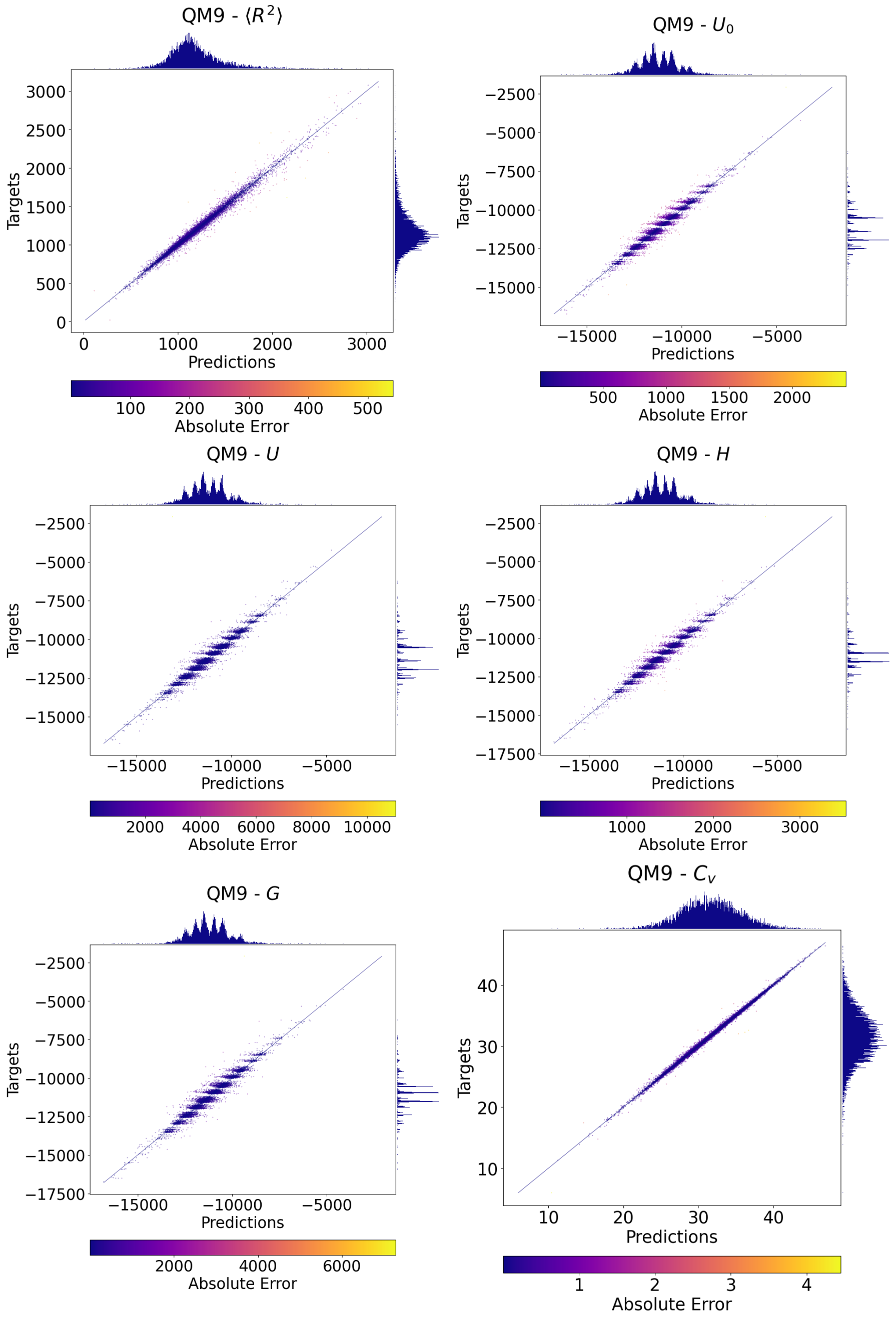


Figure B.10: Node+property model: examples of parity plots for QM9 targets (2/2).

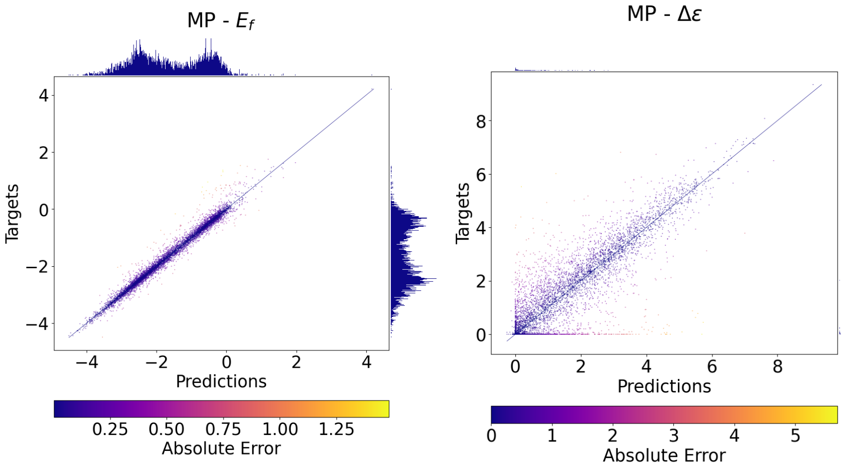


Figure B.11: Node+property model: examples of parity plots for MP targets.

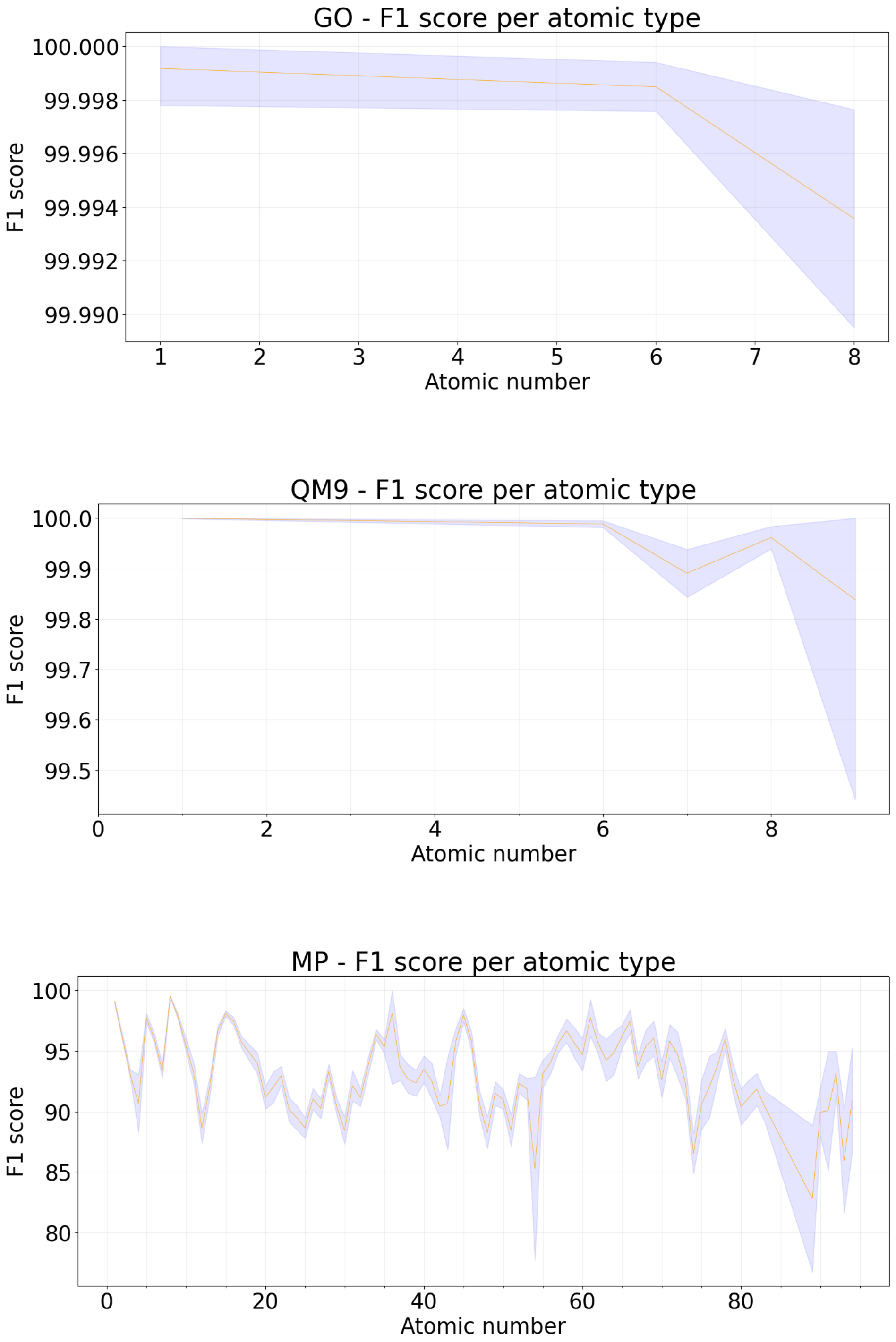


Figure B.12: Node+property model: example of F1 score per atom type for masked node prediction. Mean and standard deviation values are derived from 30 iterations on the dataset, with distinct random masking of the nodes.

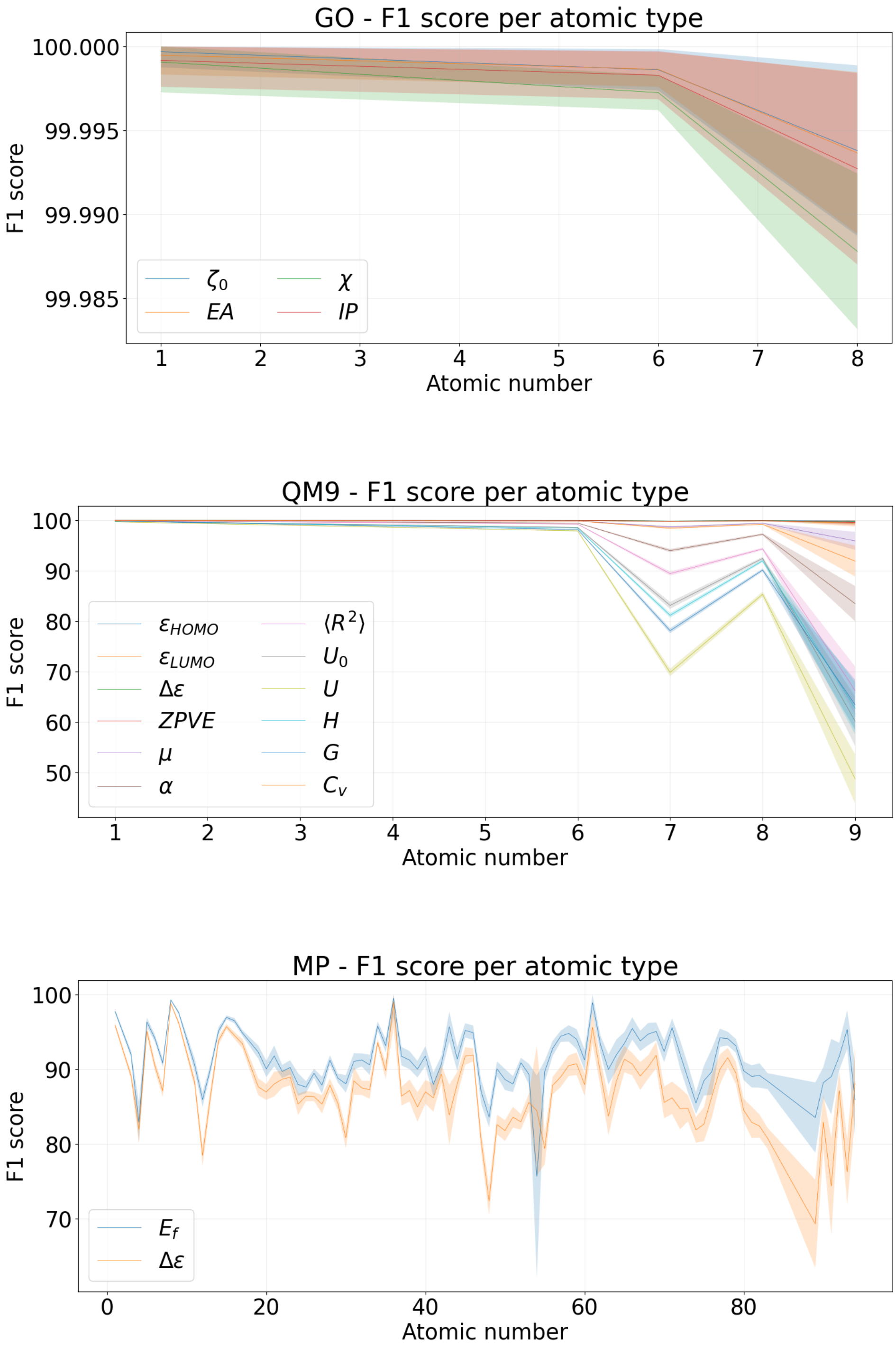
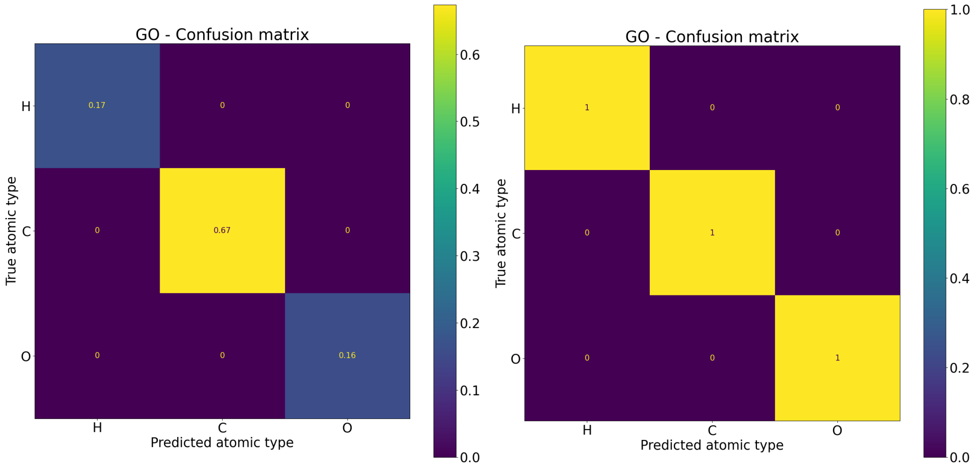


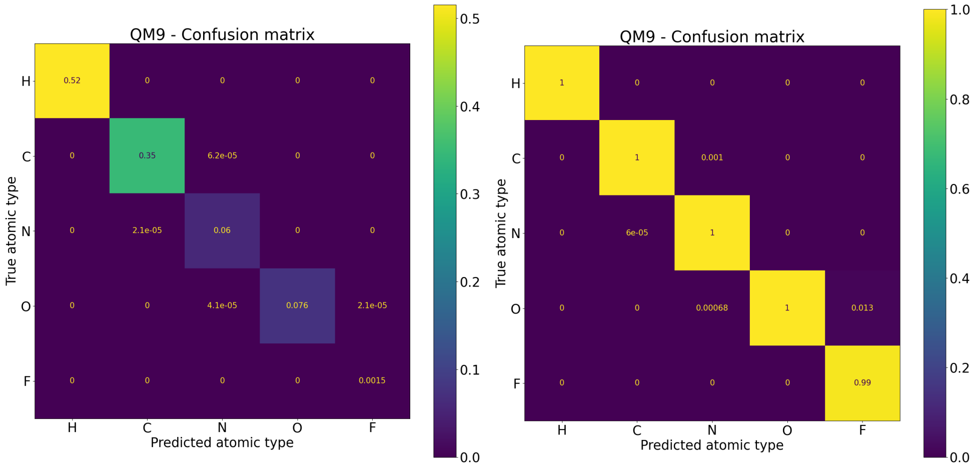
Figure B.13: Node+property model: example of F1 score per atom type for masked node prediction. Mean and standard deviation values are derived from 30 iterations on the dataset, with distinct random masking of the nodes.



(a) Normalized by total predictions

(b) Normalized by true predictions

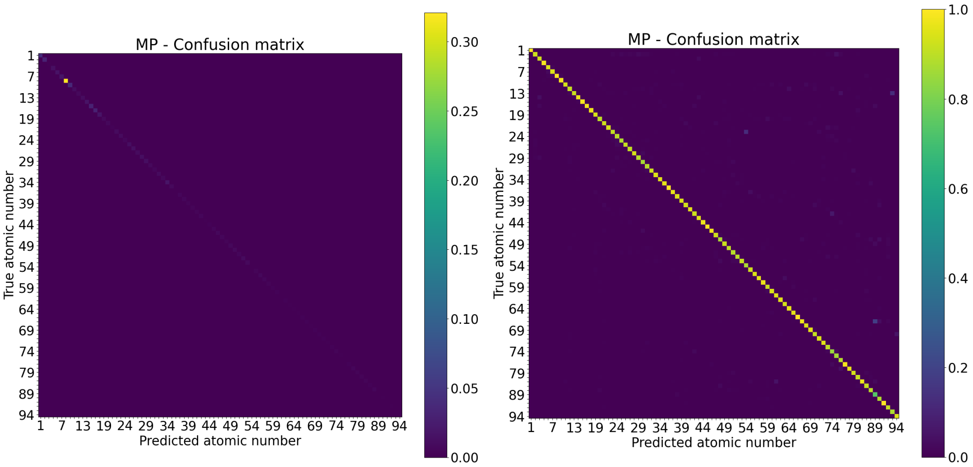
Figure B.14: Node-only model, masked node prediction, example of confusion matrix normalized by total (left) and by row (right), on GO dataset.



(a) Normalized by total predictions

(b) Normalized by true predictions

Figure B.15: Node-only model, masked node prediction, example of confusion matrix normalized by total (left) and by row (right), on QM9 dataset.



(a) Normalized by total predictions

(b) Normalized by true predictions

Figure B.16: Node-only model, masked node prediction, example of confusion matrix normalized by total (left) and by row (right), on MP dataset.

Appendix C

Attention Maps

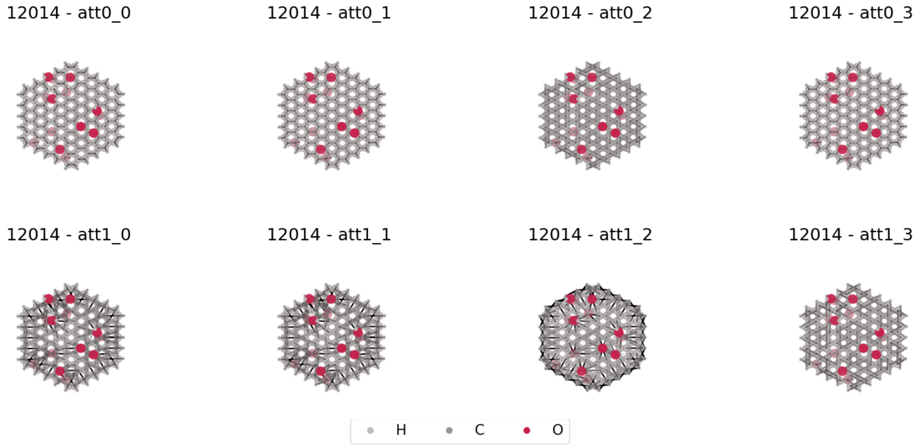


Figure C.1: Property-only model (E_{tot}): attention maps computed on a sample from GO. The last attentional layer is for node and edge prediction, so it is not showed here since it was not trained.

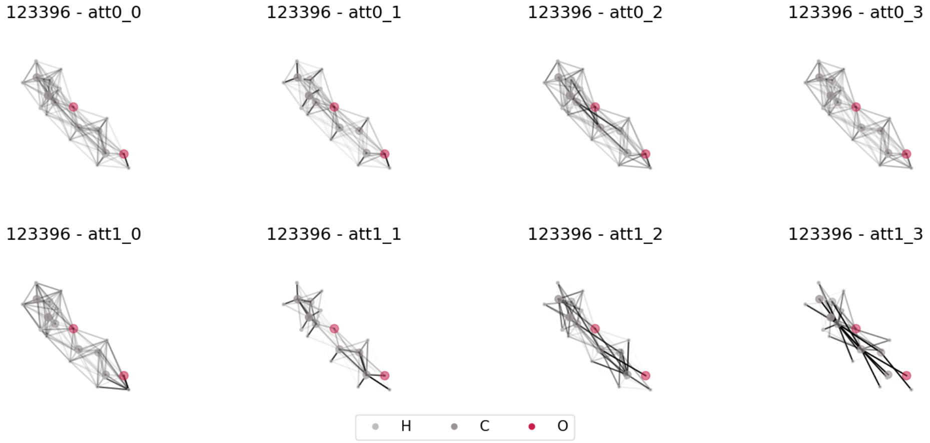


Figure C.2: Property-only model (U_0): attention maps computed on a sample from QM9. The last attentional layer is for node and edge prediction, so it is not showed here since it was not trained.

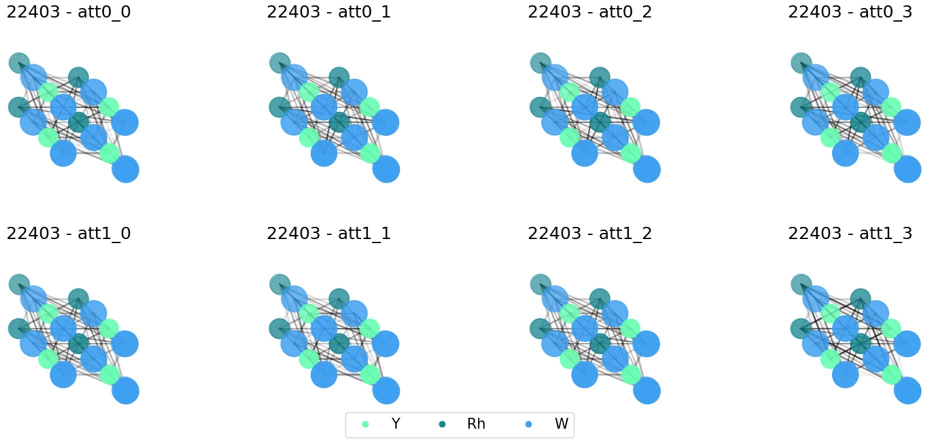


Figure C.3: Property-only model (E_f): attention maps computed on a sample from MP. The last attentional layer is for node and edge prediction, so it is not showed here since it was not trained.

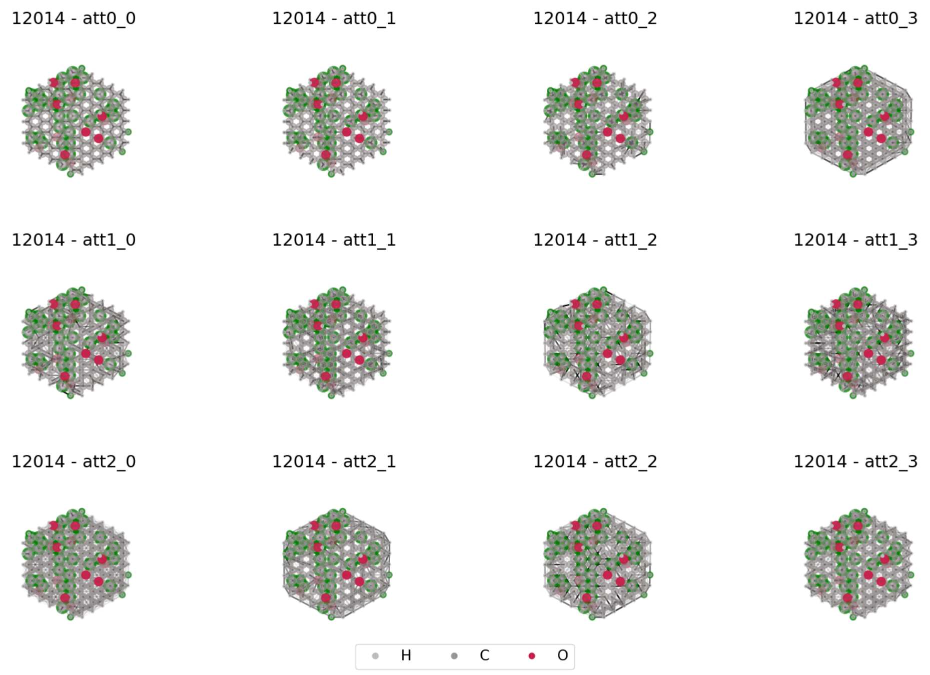


Figure C.4: Node-only model: attention maps computed on a sample from GO.

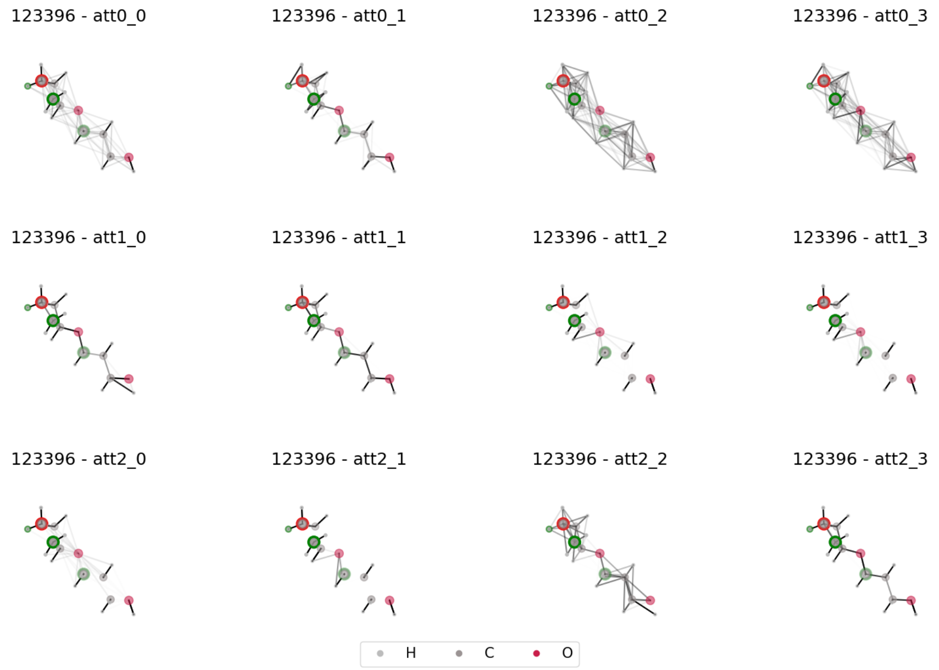


Figure C.5: Node-only model: attention maps computed on a sample from QM9.

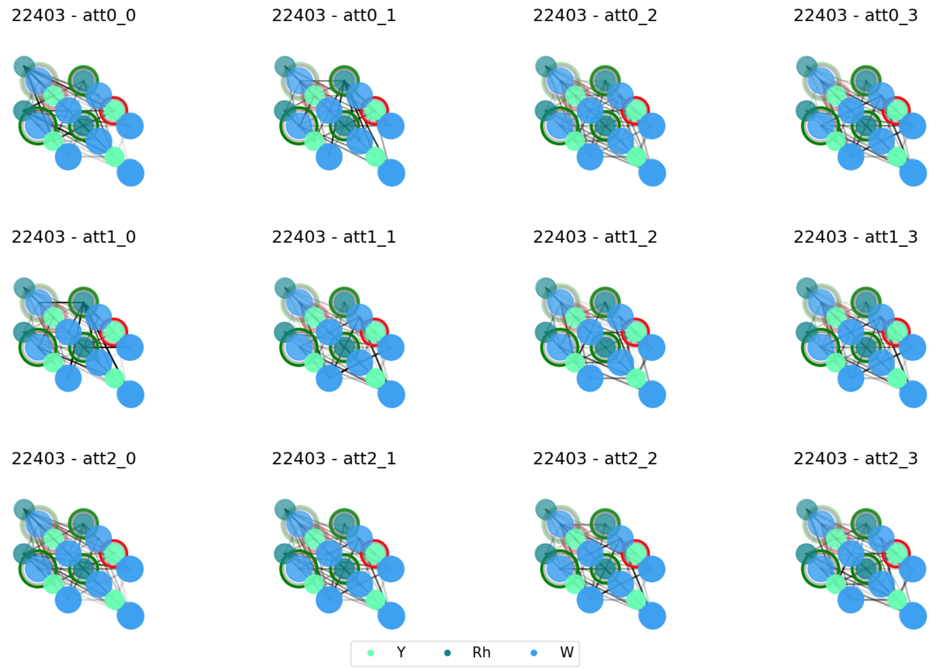


Figure C.6: Node-only model: attention maps computed on a sample from MP.

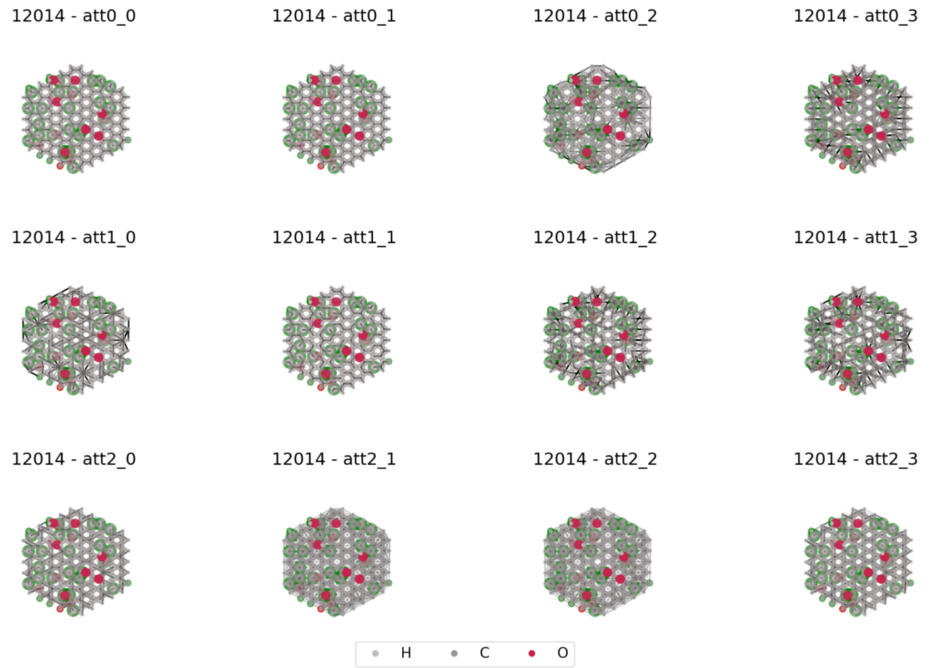


Figure C.7: Node+property model (E_{tot}): attention maps computed on a sample from GO.

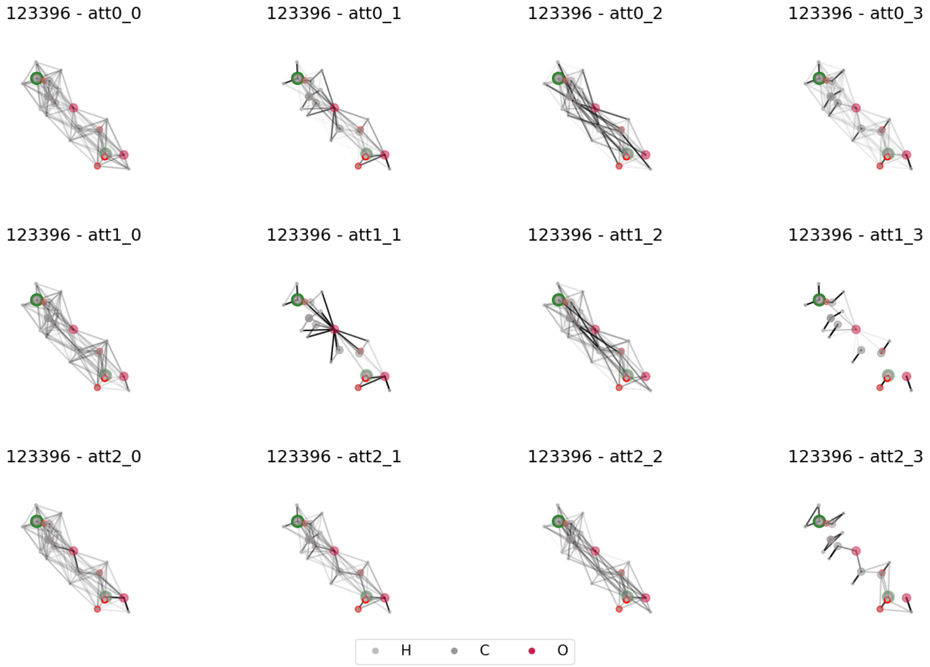


Figure C.8: Node+property model (U_0): attention maps computed on a sample from QM9.

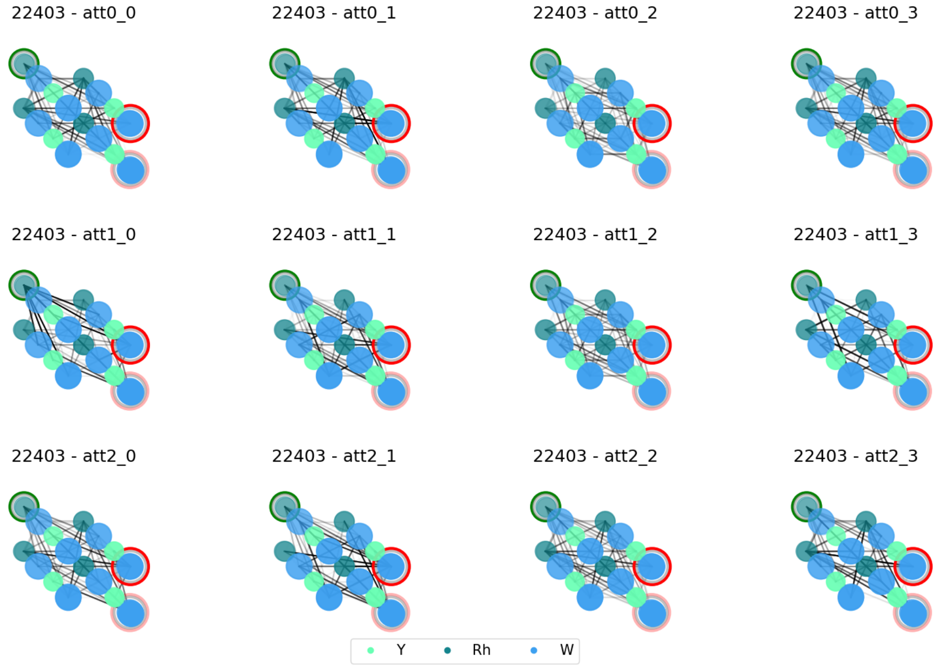


Figure C.9: Node+property model (E_f): attention maps computed on a sample from MP.

Bibliography

[1] A Strategic Approach to Machine Learning for Material Science: How to Tackle Real-World Challenges and Avoid Pitfalls | Chemistry of Materials. URL: https://pubs.acs.org/doi/10.1021/acs.chemmater.2c01333 (visited on 09/25/2023).

[2] M. P. Allen and D. J. Tildesley. Computer Simulation of Liquids - introduction. In M. P. Allen and D. J. Tildesley, editors, *Computer Simulation of Liquids*, pages 1–4. Oxford University Press, June 2017. ISBN: 978-0-19-880319-5. DOI: 10.1093/oso/9780198803195.003.0001. URL: https://doi.org/10.1093/oso/9780198803195.003.0001 (visited on 09/25/2023).

[3] Atomic Coordinate Entry Format Version 3.3. URL: https://www.wwpdb.org/documentation/file-format-content/format33/v3.3.html (visited on 09/25/2023).

[4] V. Bagal, R. Aggarwal, P. K. Vinod, and U. D. Priyakumar. MolGPT: Molecular Generation Using a Transformer-Decoder Model. en. *Journal of Chemical Information and Modeling*, 62(9):2064–2076, May 2022. ISSN: 1549-9596, 1549-960X. DOI: 10.1021/acs.jcim.1c00600. URL: https://pubs.acs.org/doi/10.1021/acs.jcim.1c00600 (visited on 07/05/2023).

[5] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, September 2014.

[6] A. Barnard, B. Motevalli Soumehsaraei, B. Sun, and L. Lai. Neutral Graphene Oxide Data Set. en. DOI: 10.25919/5e30b44a7c948. URL: https://researchdata.edu.au/neutral-graphene-oxide-data-set/1441627 (visited on 09/24/2023).

[7] J. Behler and M. Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters*, 98(14):146401, April 2007. DOI: 10.1103/PhysRevLett.98.146401. URL: https://link.aps.org/doi/10.1103/PhysRevLett.98.146401 (visited on 10/01/2023). Publisher: American Physical Society.

[8] H. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide Protein Data Bank. en. *Nature Structural & Molecular Biology*, 10(12):980–980, December 2003. ISSN: 1545-9985. DOI: 10.1038/nsb1203-980. URL: https://www.nature.com/articles/nsb1203-980 (visited on 09/25/2023). Number: 12 Publisher: Nature Publishing Group.

[9] R. Borst. Challenges in computational materials science: Multiple scales, multi-physics and evolving discontinuities. *Computational Materials Science*, 2040:2–70, July 2008. DOI: 10.1016/j.commatsci.2007.07.022.

[10] S. Brody, U. Alon, and E. Yahav. How Attentive are Graph Attention Networks? en, January 2022. URL: http://arxiv.org/abs/2105.14491 (visited on 07/05/2023). arXiv:2105.14491 [cs].

[11] I. D. Brown and B. McMahon. CIF: the computer language of crystallography. en. *Acta Crystallographica Section B: Structural Science*, 58(3):317–324, June 2002. ISSN: 0108-7681. DOI: 10.1107/S0108768102003464. URL: http://scripts.iucr.org/cgi-bin/paper?an0595 (visited on 09/25/2023). Number: ARRAY(0xa49f8d8) Publisher: International Union of Crystallography.

[12] D. Ceperley and B. Alder. Quantum Monte Carlo. English (US). *Science*, 231(4738):555–560, 1986. ISSN: 0036-8075. DOI: 10.1126/science.231.4738.555. Publisher: American Association for the Advancement of Science.

[13] Chemical Markup Language | CML. URL: https://www.xml-cml.org/ (visited on 09/25/2023).

[14] C. Chen and S. P. Ong. A universal graph deep learning interatomic potential for the periodic table. en. *Nature Computational Science*, 2(11):718–728, November 2022. ISSN: 2662-8457. DOI: 10.1038/s43588-022-00349-3. URL: https://www.nature.com/articles/s43588-022-00349-3 (visited on 07/05/2023).

[15] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. en. *Chemistry of Materials*, 31(9):3564–3572, May 2019. ISSN: 0897-4756, 1520-5002. DOI: 10.1021/acs.chemmater.9b01294. URL: https://pubs.acs.org/doi/10.1021/acs.chemmater.9b01294 (visited on 07/05/2023).

[16] K. Choudhary and B. DeCost. Atomistic Line Graph Neural Network for improved materials property predictions. en. *npj Computational Materials*, 7(1):1–8, November 2021. ISSN: 2057-3960. DOI: 10.1038/s41524-021-00650-1. URL: https://www.nature.com/articles/s41524-021-00650-1 (visited on 10/01/2023). Number: 1 Publisher: Nature Publishing Group.

[17] K. Choudhary, B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. L. Billinge, E. Holm, S. P. Ong, and C. Wolverton. Recent advances and applications of deep learning methods in materials science. en. *npj Computational Materials*, 8(1):1–26, April 2022. ISSN: 2057-3960. DOI: 10.1038/s41524-022-00734-6. URL: https://www.nature.com/articles/s41524-

022-00734-6 (visited on 09/25/2023). Number: 1 Publisher: Nature Publishing Group.

[18] A. Dalby, J. G. Nourse, W. D. Hounshell, A. K. I. Gushurst, D. L. Grier, B. A. Leland, and J. Laufer. Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited. *Journal of Chemical Information and Computer Sciences*, 32(3):244–255, May 1992. ISSN: 0095-2338. DOI: 10.1021/ci00007a012. URL: https://doi.org/10.1021/ci00007a012 (visited on 09/25/2023). Publisher: American Chemical Society.

[19] Daylight Theory: SMARTS - A Language for Describing Molecular Patterns. URL: https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html (visited on 09/25/2023).

[20] Daylight>SMARTS Tutorial. URL: https://www.daylight.com/dayhtml_tutorials/languages/smarts/index.html#INTRO (visited on 09/25/2023).

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics, June 2019. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423 (visited on 10/04/2023).

[22] R. T. Downs and M. Hall-Wallace. The American Mineralogist crystal structure database. en.

[23] V. P. Dwivedi and X. Bresson. A Generalization of Transformer Networks to Graphs. *ArXiv*, abs/2012.09699, 2020. URL: https://api.semanticscholar.org/CorpusID:229298019.

[24] V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson. Benchmarking Graph Neural Networks, December 2022. DOI: 10.48550/arXiv.2003.00982. URL: http://arxiv.org/abs/2003.00982 (visited on 10/01/2023). arXiv:2003.00982 [cs, stat].

[25] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, and O. A. Von Lilienfeld. Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. en. *Journal of Chemical Theory and Computation*, 13(11):5255–5264, November 2017. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/acs.jctc.7b00577. URL: https://pubs.acs.org/doi/10.1021/acs.jctc.7b00577 (visited on 07/05/2023).

[26] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin. Graph Neural Networks for Social Recommendation. en. In *The World Wide Web Conference*, pages 417–426, San Francisco CA USA. ACM, May 2019. ISBN: 978-1-4503-6674-8. DOI: 10.1145/3308558.3313488. URL: https://dl.acm.org/doi/10.1145/3308558.3313488 (visited on 10/02/2023).

[27] D. Flam-Shepherd, T. Wu, P. Friederich, and A. Aspuru-Guzik. Neural Message Passing on High Order Paths. *Machine Learning: Science and Technology*, 2, April 2021. DOI: 10.1088/2632-2153/abf5b8.

[28] D. Frenkel and B. Smit. Undesrtanding Molecular Simulation 2nd edition -introduction. In D. Frenkel and B. Smit, editors, *Understanding Molecular Simulation (Second Edition)*, pages 1–6. Academic Press, San Diego, January 2002. ISBN: 978-0-12-267351-1. DOI: 10.1016/B978-012267351-1/50003-1. URL: https://www.sciencedirect.com/science/article/pii/B9780122673511500031 (visited on 09/25/2023).

[29] P. Friederich, F. Häse, J. Proppe, and A. Aspuru-Guzik. Machine-learned potentials for next-generation matter simulations. eng. *Nature materials*, 20(6):750–761, June 2021. ISSN: 1476-4660. DOI: `10.1038/s41563-020-0777-6`. URL: `https://doi.org/10.1038/s41563-020-0777-6` (visited on 10/01/2023).

[30] J. Gasteiger, F. Becker, and S. Günnemann. GemNet: Universal Directional Graph Neural Networks for Molecules. en. In November 2021. URL: `https://openreview.net/forum?id=HS_sOaxS9K-` (visited on 10/01/2023).

[31] J. Gasteiger, S. Giri, J. Margraf, and S. Günnemann. *Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules*. November 2020.

[32] J. Gasteiger, J. Groß, and S. Günnemann. *Directional Message Passing for Molecular Graphs*. March 2020.

[33] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural Message Passing for Quantum Chemistry. en.

[34] S. R. Hall, F. H. Allen, and I. D. Brown. The crystallographic information file (CIF): a new standard archive file for crystallography. en. *Acta Crystallographica Section A: Foundations of Crystallography*, 47(6):655–685, November 1991. ISSN: 0108-7673. DOI: `10.1107/S010876739101067X`. URL: `//scripts.iucr.org/cgi-bin/paper?es0164` (visited on 09/25/2023). Number: 6 Publisher: International Union of Crystallography.

[35] W. Hamilton, Z. Ying, and J. Leskovec. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL: `https://proceedings.neurips.cc/paper_files/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html` (visited on 10/01/2023).

[36] Z. Hao, C. Lu, Z. Huang, H. Wang, Z. Hu, Q. Liu, E. Chen, and C. Lee. ASGN: An Active Semi-supervised Graph Neural Network for Molecular Property Prediction. en. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 731–752, Virtual Event CA USA. ACM, August 2020. ISBN: 978-1-4503-7998-4. DOI: 10.1145/3394486.3403117. URL: https://dl.acm.org/doi/10.1145/3394486.3403117 (visited on 07/31/2023).

[37] F. Häse, L. M. Roch, and A. Aspuru-Guzik. Next-Generation Experimentation with Self-Driving Laboratories. *Trends in Chemistry*, 1(3):282–291, June 2019. ISSN: 2589-5974. DOI: 10.1016/j.trechm.2019.02.007. URL: https://www.sciencedirect.com/science/article/pii/S258959741930019X (visited on 10/01/2023).

[38] T. Heine. Grand Challenges in Computational Materials Science: From Description to Prediction at all Scales. *Frontiers in Materials*, 1, 2014. ISSN: 2296-8016. URL: https://www.frontiersin.org/articles/10.3389/fmats.2014.00007 (visited on 09/25/2023).

[39] L. Himanen, A. Geurts, A. S. Foster, and P. Rinke. Data-Driven Materials Science: Status, Challenges, and Perspectives. en. *Advanced Science*, 6(21):1900808, 2019. ISSN: 2198-3844. DOI: 10.1002/advs.201900808. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.201900808 (visited on 09/25/2023). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/advs.201900808.

[40] P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. en. *Physical Review*, 136(3B):B864–B871, November 1964. ISSN: 0031-899X. DOI: 10.1103/PhysRev.136.B864. URL: https://link.aps.org/doi/10.1103/PhysRev.136.B864 (visited on 09/25/2023).

[41] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366,

January 1989. ISSN: 0893-6080. DOI: 10.1016/0893-6080(89)90020-8. URL: https://www.sciencedirect.com/science/article/pii/0893608089900208 (visited on 10/01/2023).

[42] B. Huang and O. A. von Lilienfeld. Quantum machine learning using atom-in-molecule-based fragments selected on-the-fly, August 2020. URL: http://arxiv.org/abs/1707.04146 (visited on 10/01/2023). arXiv:1707.04146 [physics].

[43] N. Huang and S. Villar. A Short Tutorial on The Weisfeiler-Lehman Test And Its Variants. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8533–8537, June 2021. DOI: 10.1109/ICASSP39728.2021.9413523. URL: http://arxiv.org/abs/2201.07083 (visited on 10/01/2023). arXiv:2201.07083 [cs, stat].

[44] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. en. *APL Materials*, 1(1):011002, July 2013. ISSN: 2166-532X. DOI: 10.1063/1.4812323. URL: https://pubs.aip.org/apm/article/1/1/011002/119685/Commentary-The-Materials-Project-A-materials (visited on 09/24/2023).

[45] P. B. Jørgensen, K. W. Jacobsen, and M. N. Schmidt. Neural Message Passing with Edge Updates for Predicting Properties of Molecules and Materials: 32nd Conference on Neural Information Processing Systems. In 2018.

[46] R. N. Jorissen and M. K. Gilson. Virtual screening of molecular databases using a support vector machine. eng. *Journal of Chemical Information and Modeling*, 45(3):549–561, 2005. ISSN: 1549-9596. DOI: 10.1021/ci049641u.

[47] S. V. Kalinin, C. Ophus, P. M. Voyles, R. Erni, D. Kepaptsoglou, V. Grillo, A. R. Lupini, M. P. Oxley, E. Schwenker, M. K. Y. Chan, J. Etheridge, X. Li, G. G. D. Han, M. Ziatdinov, N. Shibata, and S. J. Pennycook. Machine learning in scanning transmission electron microscopy. en. *Nature Reviews Methods Primers*, March 2022. ISSN: 2662-8449. URL: https://doi.org/10.1038/s43586-022-00095-w (visited on 10/01/2023). Publisher: York.

[48] T. N. Kipf and M. Welling. SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS. en, 2017.

[49] W. Kohn and L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Physical Review*, 140(4A):A1133–A1138, November 1965. DOI: 10.1103/PhysRev.140.A1133. URL: https://link.aps.org/doi/10.1103/PhysRev.140.A1133 (visited on 09/26/2023). Publisher: American Physical Society.

[50] Y. Lecun, K. Kavukcuoglu, and C. Farabet. *Convolutional Networks and Applications in Vision*. May 2010. DOI: 10.1109/ISCAS.2010.5537907. Journal Abbreviation: ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems Pages: 256 Publication Title: ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems.

[51] O. Lilienfeld. Introducing Machine Learning: Science and Technology. *Machine Learning: Science and Technology*, 1:010201, February 2020. DOI: 10.1088/2632-2153/ab6d5d.

[52] H. Liu, Z. Dai, D. R. So, and Q. V. Le. Pay Attention to MLPs. en, June 2021. URL: http://arxiv.org/abs/2105.08050 (visited on 07/05/2023). arXiv:2105.08050 [cs].

[53] Y. Luo, S. Bag, O. Zaremba, J. Andreo, S. Wuttke, M. Tsotsalas, and P. Friederich. *MOF Synthesis Prediction Enabled by Automatic Data Mining and Machine Learning*. August 2021. DOI: 10.26434/chemrxiv-2021-kgd0h.

[54] H. Maron, H. Ben-Hamu, N. Shamir, and Y. Lipman. Invariant and Equivariant Graph Networks, April 2019. DOI: 10.48550/arXiv.1812.09902. URL: http://arxiv.org/abs/1812.09902 (visited on 10/01/2023). arXiv:1812.09902 [cs, stat].

[55] R. M. Martin. Electronic Structure - DFT. In *Electronic Structure: Basic Theory and Practical Methods*, pages 119–134. Cambridge University Press, 1st edition, April 2004. ISBN: 978-0-521-53440-6 978-0-521-78285-2 978-0-511-80576-9. DOI: 10.1017/CBO9780511805769.008. URL: https://www.cambridge.org/core/product/identifier/9780511805769/type/book (visited on 09/25/2023).

[56] C. Merkwirth and T. Lengauer. Automatic Generation of Complementary Descriptors with Molecular Graph Networks. *Journal of chemical information and modeling*, 45:1159–68, September 2005. DOI: 10.1021/ci049613b.

[57] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. en. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4602–4609, July 2019. ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v33i01.33014602. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4384 (visited on 08/11/2023).

[58] B. Motevalli, A. J. Parker, B. Sun, and A. S. Barnard. The representative structure of graphene oxide nanoflakes from machine learning. en. *Nano Futures*, 3(4):045001, December 2019. ISSN: 2399-1984. DOI: 10.1088/2399-1984/ab58ac. URL: https://iopscience.

iop.org/article/10.1088/2399-1984/ab58ac (visited on 07/05/2023).

[59] P. Murray-Rust and H. S. Rzepa. Chemical Markup, XML, and the World Wide Web. 4. CML Schema. *Journal of Chemical Information and Computer Sciences*, 43(3):757–772, May 2003. ISSN: 0095-2338. DOI: 10.1021/ci0256541. URL: https://doi.org/10.1021/ci0256541 (visited on 09/25/2023). Publisher: American Chemical Society.

[60] J. Noh, G. H. Gu, S. Kim, and Y. Jung. Machine-enabled inverse design of inorganic solid materials: promises and challenges. en. *Chemical Science*, 11(19):4871–4881, May 2020. ISSN: 2041-6539. DOI: 10.1039/D0SC00594K. URL: https://pubs.rsc.org/en/content/articlelanding/2020/sc/d0sc00594k (visited on 10/01/2023). Publisher: The Royal Society of Chemistry.

[61] S. P. Ong, S. Cholia, A. Jain, M. Brafman, D. Gunter, G. Ceder, and K. A. Persson. The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles. *Computational Materials Science*, 97:209–215, February 2015. ISSN: 0927-0256. DOI: 10.1016/j.commatsci.2014.10.037. URL: https://www.sciencedirect.com/science/article/pii/S0927025614007113 (visited on 09/24/2023).

[62] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder. Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. en. *Computational Materials Science*, 68:314–319, February 2013. ISSN: 09270256. DOI: 10.1016/j.commatsci.2012.10.028. URL: https://linkinghub.elsevier.com/retrieve/pii/S0927025612006295 (visited on 09/24/2023).

[63] G. Paaß and S. Giesselbach. Pre-trained Language Models. en. In G. Paaß and S. Giesselbach, editors, *Foundation Models for Natural Language Processing: Pre-trained Language Models Integrating Media*, Artificial Intelligence: Foundations, Theory, and Algorithms, pages 19–78. Springer International Publishing, Cham, 2023. ISBN: 978-3-031-23190-2. DOI: 10.1007/978-3-031-23190-2_2. URL: https://doi.org/10.1007/978-3-031-23190-2_2 (visited on 10/02/2023).

[64] D. Raabe, J. R. Mianroodi, and J. Neugebauer. Accelerating the design of compositionally complex materials via physics-informed artificial intelligence. en. *Nature Computational Science*, 3(3):198–209, March 2023. ISSN: 2662-8457. DOI: 10.1038/s43588-023-00412-7. URL: https://www.nature.com/articles/s43588-023-00412-7 (visited on 07/05/2023).

[65] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners. en.

[66] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. en. *Scientific Data*, 1(1):140022, August 2014. ISSN: 2052-4463. DOI: 10.1038/sdata.2014.22. URL: https://www.nature.com/articles/sdata201422 (visited on 09/15/2023).

[67] L. Ruddigkeit, R. Van Deursen, L. C. Blum, and J.-L. Reymond. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. en. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, November 2012. ISSN: 1549-9596, 1549-960X. DOI: 10.1021/ci300415d. URL: https://pubs.acs.org/doi/10.1021/ci300415d (visited on 09/15/2023).

[68] R. Sato. A Survey on The Expressive Power of Graph Neural Networks, October 2020. URL: http://arxiv.org/abs/2003.04078 (visited on 10/02/2023). arXiv:2003.04078 [cs, stat].

[69] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques. Recent advances and applications of machine learning in solid-state materials science. en. *npj Computational Materials*, 5(1):1–36, August 2019. ISSN: 2057-3960. DOI: 10.1038/s41524-019-0221-0. URL: https://www.nature.com/articles/s41524-019-0221-0 (visited on 09/25/2023). Number: 1 Publisher: Nature Publishing Group.

[70] J. Schmidt, L. Pettersson, C. Verdozzi, S. Botti, and M. A. L. Marques. Crystal graph attention networks for the prediction of stable materials. *Science Advances*, 7(49):eabi7948, December 2021. DOI: 10.1126/sciadv.abi7948. URL: https://www.science.org/doi/10.1126/sciadv.abi7948 (visited on 09/30/2023). Publisher: American Association for the Advancement of Science.

[71] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. SchNet – A deep learning architecture for molecules and materials. en. *The Journal of Chemical Physics*, 148(24):241722, June 2018. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.5019779. URL: https://pubs.aip.org/aip/jcp/article/962591 (visited on 07/05/2023).

[72] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee. Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Central Science*, 5(9):1572–1583, September 2019. ISSN: 2374-7943. DOI: 10.1021/acscentsci.9b00576. URL: https://doi.org/10.1021/acscentsci.9b00576 (visited on 10/01/2023). Publisher: American Chemical Society.

[73] The InChI Trust and IUPAC – InChI Trust. en-GB. URL: https://www.inchi-trust.org/iupac/ (visited on 09/25/2023).

[74] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, and S. Zhao. Applications of

machine learning in drug discovery and development. en. *Nature Reviews Drug Discovery*, 18(6):463–477, June 2019. ISSN: 1474-1784. DOI: 10.1038/s41573-019-0024-5. URL: https://www.nature.com/articles/s41573-019-0024-5 (visited on 10/01/2023). Number: 6 Publisher: Nature Publishing Group.

[75] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[76] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. en, February 2018. URL: http://arxiv.org/abs/1710.10903 (visited on 07/05/2023). arXiv:1710.10903 [cs, stat].

[77] O. Vinyals, S. Bengio, and M. Kudlur. *Order Matters: Sequence to sequence for sets*. November 2016.

[78] Y. Wang and M. Zhang. Towards Better Evaluation of GNN Expressiveness with BREC Dataset. en, July 2023. URL: http://arxiv.org/abs/2304.07702 (visited on 08/11/2023). arXiv:2304.07702 [cs].

[79] Z. Wang, J. Chen, and H. Chen. EGAT: Edge-Featured Graph Attention Network. en. In I. Farkaš, P. Masulli, S. Otte, and S. Wermter, editors, *Artificial Neural Networks and Machine Learning – ICANN 2021*. Volume 12891, pages 253–264. Springer International Publishing, Cham, 2021. ISBN: 978-3-030-86361-6 978-3-030-86362-3. DOI: 10.1007/978-3-030-86362-3_21. URL: https://link.springer.com/10.1007/978-3-030-86362-3_21 (visited on 09/08/2023). Series Title: Lecture Notes in Computer Science.

[80] D. Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, February 1988. ISSN: 0095-2338. DOI: 10.1021/ci00057a005. URL: https://doi.org/10.1021/ci00057a005 (visited on 09/25/2023). Publisher: American Chemical Society.

[81] D. Weininger. SMILES. 3. DEPICT. Graphical depiction of chemical structures. *Journal of Chemical Information and Computer Sciences*, 30(3):237–243, August 1990. ISSN: 0095-2338. DOI: 10.1021/ci00067a005. URL: https://doi.org/10.1021/ci00067a005 (visited on 09/25/2023). Publisher: American Chemical Society.

[82] D. Weininger, A. Weininger, and J. L. Weininger. SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Sciences*, 29(2):97–101, May 1989. ISSN: 0095-2338. DOI: 10.1021/ci00062a008. URL: https://doi.org/10.1021/ci00062a008 (visited on 09/25/2023). Publisher: American Chemical Society.

[83] B. Weisfeiler and A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *nti, Series*, 2(9):12–16, 1968.

[84] T. Xie and J. C. Grossman. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. en. *Physical Review Letters*, 120(14):145301, April 2018. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.120.145301. URL: https://link.aps.org/doi/10.1103/PhysRevLett.120.145301 (visited on 07/05/2023).

[85] J. Xiong, Z. Xiong, K. Chen, H. Jiang, and M. Zheng. Graph neural networks for automated de novo drug design. *Drug Discovery Today*, 26(6):1382–1393, June 2021. ISSN: 1359-6446. DOI: 10.1016/j.

drudis.2021.02.011. URL: https://www.sciencedirect.com/science/article/pii/S1359644621000787 (visited on 10/02/2023).

[86] Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, and M. Zheng. Pushing the Boundaries of Molecular Representation for Drug Discovery with the Graph Attention Mechanism. eng. *Journal of Medicinal Chemistry*, 63(16):8749–8760, August 2020. ISSN: 1520-4804. DOI: 10.1021/acs.jmedchem.9b00959.

[87] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How Powerful are Graph Neural Networks? en, February 2019. URL: http://arxiv.org/abs/1810.00826 (visited on 08/11/2023). arXiv:1810.00826 [cs, stat].

[88] XYZ (format) - Open Babel. URL: https://openbabel.org/wiki/XYZ_%28format%29 (visited on 09/25/2023).

[89] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, A. Palmer, V. Settels, T. Jaakkola, K. Jensen, and R. Barzilay. Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, August 2019. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.9b00237. URL: https://doi.org/10.1021/acs.jcim.9b00237 (visited on 10/01/2023). Publisher: American Chemical Society.

[90] S. Zhang, Y. Liu, and L. Xie. Molecular Mechanics-Driven Graph Neural Network with Multiplex Graph for Molecular Structures. en, November 2020. URL: http://arxiv.org/abs/2011.07457 (visited on 07/05/2023). arXiv:2011.07457 [physics, q-bio].

[91] M. Zopf. *1-WL Expressiveness Is (Almost) All You Need*. July 2022. DOI: 10.1109/IJCNN55064.2022.9892655. Pages: 8.