# REPRODUCIBLE MACHINE LEARNING: A REVIEW OF REPRESENTATION LEARNING APPROACHES FOR DRUG DISCOVERY

**Boury Mbodj**
boury.mbodj@mail.mcgill.ca

**Supervised by: William L. Hamilton**
wlh@cs.mcgill.ca

January 2020

**ABSTRACT**

Many advances in machine learning have transformed numerous industries, including search, transportation, speech recognition, and healthcare. More recently, a diverse set of representation learning approaches for molecular data are being proposed to accelerate the process of virtual screening in the drug discovery pipelines. Until present, quantum simulation methods such as Density Functional Theory, which are computationally expensive, are being used to calculate molecular quantum properties. In 2017, Gilmer et al. published "Neural Message Passing for Quantum Chemistry" [1] and presented a framework that groups multiple graph neural networks as well as a novel edge network to handle edge states in molecular graphs. The combination of the latter to an ensemble yielded state of the art results in the application of quantum property predictions. In this project, we examined whether the authors' [1] claim holds, by reproducing a subset of the baselines methods mentioned in the paper.

## 1 Introduction

Reproducibility is a cornerstone of scientific methods, and it is critical for machine learning research. If the results of existing studies cannot be replicated due to inconsistent experimental and publication process, it casts doubt on the validity of the initial findings. As the theme of our project, we attempt to reproduce the findings mentioned in "Neural Message Passing for Quantum Chemistry" [1].

Drug design is a complex, lengthy and expensive process where quantum chemistry calculations are frequently required. Quantum Mechanical (QM) methods such as Density Functional Theory (DFT) [2] have emerged as numerical approximations to Schrödinger's [3] equation method to calculate molecular properties. However, the computational complexity of such a highly accurate QM method is at least $O(N_e^3)$ where $N_e$ is the number of electrons [1], and is thus restricted to relatively small systems.

In recent years, to address the shortcomings of QM simulation methods and accelerate the drug discovery process, machine learning models have been proposed and claimed as both practical and innovative approaches. Much effort has been devoted to developing featurization techniques. Molecules are complex entities, and different methods from chemical descriptor vectors to 2D graph representations and 3D electrostatic grid representations have been developed. In 2017, Gilmer et al. released their paper in which they reformulate a subset of existing machine learning models into a single common framework called Message Passing Neural Networks (MPNNs), which they apply to a chemical property prediction task. This framework includes several graph neural networks such as Interaction Networks [4], Laplacian Based Methods [5] [6], Molecular Fingerprints [7] and Deep Tensor Neural Networks without being exhaustive [8]. We perform an investigation of a subset of the baseline models in MPNNs [1], as well as propose new baselines for both hand-engineered and graph representation featurization methods using QM9 [9] [10] as a benchmark.

This report is structured as follows: In Section 2, we review related work. In Sections 3 and 4, we outline the featurization methods we used, the data set, as well as the setup. In Section 5, we attempt to reproduce the findings of the paper and evaluate the results. Finally, in Sections 6 and 7, we discuss our proposed baselines and the outcomes of our models.

## 2 Related work

A diverse set of work on molecular descriptors, which is certainly out of the scope of this report, has been presented to advance the drug discovery process. In 2015, Duveneaud et al. introduced Molecular FingerPrints [7], which is an architecture that generalizes a method for learning vector representations of small molecules, or fingerprints using deep convolutional neural networks. The neural graph fingerprints allow graph inputs of arbitrary size and shape and are based on the same logic as circular fingerprints.
Inspired by the many-body Hamiltonian applied to the interactions of atoms, Schütt et al. proposed Deep Tensor Neural Networks (DTNN) [8], a custom deep network architecture for molecular data. DTNN has shown to reach chemical accuracy on a small set of molecular dynamics trajectories as well as QM9 [9] [10].

## 3 Featurizations

In this project, we will focus on 2D graph molecular descriptors, which usually include information about the molecule covalent and aromatic bonds, and 3D handcrafted descriptors which also incorporates spatial relationships.

### 3.1 Coulomb Matrix

The Coulomb Matrix [11] is a low-level molecular descriptor that is invariant to translations and rotations. Specifically, for each molecule, the descriptor is constructed by using Cartesian coordinates $Z_I$, and nuclear charges $R_I$, which is the same information that enters the Hamiltonian for an electronic structure calculation. For any Molecule $C_{IJ}$, the Coulomb Matrix [11] is defined as:

$$C_{IJ} = \begin{cases} 0.5 Z_I^{2.4} & I = J \\ \frac{Z_I Z_J}{R_J - R_I} & I \neq J \end{cases} \tag{1}$$

The diagonal elements correspond to the potential of the free atom, and the off-diagonal elements represent the inter-atomic Coulomb repulsion between nuclear charges in the system [11].

### 3.2 Bag of Bonds

The Bag of Bonds (BoB) [12] is a hand-engineered descriptor that is similar in concept and inspired by the natural language processing Bag of Words [13]. It has been presented as one of the early models that achieved remarkable accuracy on predictions throughout the chemical compound space. More specifically, BoB [12] is a vector composed of bags of particular bond types where each entry in every bag is computed in a similar fashion as in the Coulomb Matrix [11]. This descriptor is invariant under molecular rotations and translations, as well as row and column permutations.

### 3.3 Message Passing Neural Networks

Message Passing Neural Networks (MPNNs) [1] outlines a general framework of mainly spatial-based GNNs for supervised learning. On an undirected graph $G = (V, E)$ with node features $x_v$, edge features $e_{vw}$ and a neighborhood defined as $N(v)$, a message passing process is composed of two phases, a messaging phase (2) and (3) and a readout phase(4). The general messaging phase is defined by the following formula from Gilmer et al. [1]:

$$m_v^{t+1} = \sum_{w \epsilon N(v)} M_t(h_v^t, h_w^t, e_{vw}) \tag{2}$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{3}$$

Where the message $m_v^{t+1}$ is the transition function that propagates information and $h_v^{t+1}$ denotes the hidden states that are updated for $T$ iterations. The readout phase (4), which is computed with a readout function $R$, generates a representation of the entire graph based on Node hidden representations.

$$\hat{y} = R(\{h_v^t | v \epsilon G\}). \tag{4}$$

In this report, we will refer to MPNN [1] as the novel variation (enn-s2s), which has achieved impressive results with the continuous edge network message function, a gated recurrent unit (GRU) update function and a Set2Set [14] readout function.

### 3.4 Gated Graph Recurrent Neural Network

Inspired from previous work on Graph Neural Networks (GNNs) (Scarselli et al., 2009) [15], this paper presents a feed-forward neural network architecture for processing graphs as inputs that output sequences. The main difference from the GNNs [15] methodology is the use of Gated Recurrent Units [16] to unroll the recurrence for a fixed number of steps T, and the use back-propagation through time (BPTT) in order to compute gradients. The Gated graph neural network [17] is used as the main baseline in MPNNs [1] and is thus included in the MPNNs family of models. The message and update functions are respectively defined in (5) and (6) [18] as:

$$m_v^{t+1} = W_t \sum_{w \epsilon N(v)} h_w^t \tag{5}$$

$$h_v^{t+1} = GRU(h_v^t, m_v^{t+1}) \tag{6}$$

## 4 Dataset and Setup

### 4.1 Dataset

As in the reviewed paper, we used the Quantum Machine 9 (QM9) dataset [9] [10], a standard benchmark for molecular machine learning that corresponds to the GDB-9 subset of all neutral molecules, with up to nine atoms, not counting hydrogen. QM9 is comprised of 134k molecules with their corresponding equilibrium geometries, frontier orbital eigenvalues, dipole moments, harmonic frequencies, polarizabilities, and thermochemical energetics [9]. All molecules are modeled using Density 8 Functional Theory (B3LYP/6-31G(2df,p) based DFT). Gilmer et al. take an additional step and group the properties of the molecules in four categories: atomization energies/tightness of bonds, fundamental vibration frequency, states of electrons, and measures of spatial distributions.

### 4.2 Setup

In order to reproduce the experiments, we used PyTorch Geometric [19], a library for deep learning on irregularly structured input data such as graphs, point clouds, and manifolds, built upon PyTorch [20]. All graph neural network models were reproduced using the PyTorch Geometric [19] library and its example implementations as a starting point. For the hand-engineered molecular descriptors, we used the Molecular Machine Learning Toolkit [21] implementations of Bag of Bonds and variations of Coulomb Matrices.
We ran the experiments using Google Colab GPU and two Intel(R) Xeon(R) 2.30GHz CPUs, and trained the models over different configurations.

### 4.3 Input Representation

The input representation at the node level consisted of different atomic properties such as the atomic nuclear charge, the hybridization state, and other properties listed in Table 1 below. For the reproducibility task, the nodes represent the atoms and the edges represent the bonds for the MPNN [1].

| Feature | Description |
|---|---|
| Atom type | H, C, N, O, F one-hot or null |
| Atomic number | Integer electronic charge |
| Acceptor | If the atom accepts electrons |
| Donor | If the atom donates electrons |
| Aromatic | If the atom is part of an aromatic system |
| Hybridization | SP, SP2, or SP3 (one-hot or null) |
| Number of Hydrogens | Integer |

**Table 1: Atom featurization**

### 4.4 Model and Validation

We used kernel ridge regression with a Laplacian kernel for all hand-engineered descriptors, and a neural network on top of the graph representation learning featurizers to complete the regression tasks. Random splitting of molecular

data with the i.i.d assumption is not the best indicator of the performance of a model. However, due to resource limitations, we proceed with this common technique using a subset of the dataset. Thirteen thousand molecules were used to complete the experiments, eleven thousand for training, one thousand for testing, and one thousand for validating and reporting the error (MAE) with a held-out cross-validation strategy. For all of the tasks, we first standardized the target values using Scikit-learn Standard Scaler [22] so that all targets have a mean of zero and unit variance. In order to provide a better understanding of our results, we recorded the mean and standard deviation of all the targets before the normalization on Table 2.

Twelve target properties were used instead of thirteen as we were not able to identify the target named "Omega" in the QM9 dataset. The listed properties are the norm of dipole moment mu ($Debye$), the norm of isotropic polarizability alpha ($Bohr^3$), the highest occupied molecular orbital energy HOMO ($Hartree$), the lowest unoccupied molecular orbital energy LUMO ($Hartree$), the gap ($Hartree$) between HOMO and LUMO, the electronic spatial extent R2 ($Bohr^2$), the zero-point vibrational energy ZPVE ($Hartree$), the atomization energy at 0 Kelvin U0 ($Hartree$), atomization energy at room temperature U ($Hartree$), the enthalpy of atomization at room temperature H ($Hartree$), the atomization of free energy at room temperature G ($Hartree$) and the heat capacity at room temperature Cv ($cal/mol/K$).

| Target | Mean | Standard Deviation |
|---|---|---|
| **mu** | 2.62 | 1.51 |
| **alpha** | 65.36 | 8.97 |
| **HOMO** | -0.24 | 0.03 |
| **LUMO** | 0.01 | 0.06 |
| **gap** | 0.25 | 0.05 |
| **R2** | 955.46 | 246.86 |
| **ZPVE** | 0.13 | 0.03 |
| **U0** | -360.48 | 43.49 |
| **U** | -360.47 | 43.48 |
| **H** | -360.47 | 43.48 |
| **G** | -360.57 | 43.49 |
| **CV** | 28.41 | 4.54 |

**Table 2: Dataset mean and standard deviation (13000 molecules)**

### 4.5 Training

All models were trained with the same parameters as in the reviewed paper to provide a fair assessment. Indeed, each graph neural network model was trained using a mean squared error (MSE) loss function, stochastic gradient descent (SGD) with the Adam optimizer [23], and an initial learning rate of 0.001 that can decrease down to 0.00001 with the use of a Pytorch plateau scheduler. A batch size of 128 was used instead of 20 to speed up the training, with a limit of 300 epochs.

## 5 Experiments and Evaluations

#### 5.0.1 Evaluation of MPNNs MAE

The purpose of our first experiment is to verify the main claim of the selected paper [1], which is the attainment of higher accuracy results compared to a selected number of baselines. The Gated Graph Recurrent Neural Network (GG-NN) [17] reported in Table 3 uses the matrix multiplication message function with no featurization on the edges. However, GG-NN does take into account discrete edge types.

| Target | BOB | CM | GG-NN | MPNN |
|---|---|---|---|---|
| **mu** | 0.6724 | 0.8078 | 0.7639 | **0.1523** |
| **alpha** | 0.7782 | 1.4809 | 0.9228 | **0.3847** |
| **HOMO** | 0.0074 | 0.017 | 0.0081 | **0.0034** |
| **LUMO** | 0.0106 | 0.0171 | 0.0111 | **0.0037** |
| **gap** | 0.0122 | 0.0196 | 0.0118 | **0.0065** |
| **R2** | 22.2674 | 37.7441 | 76.7278 | **2.5781** |
| **ZPVE** | 0.0007 | 0.0010 | 0.0007 | **0.0004** |
| **U0** | 0.7360 | 3.4899 | **0.3955** | 0.5545 |
| **U** | 0.7360 | 3.4895 | **0.2621** | 0.5218 |
| **H** | 0.7360 | 3.4898 | **0.3899** | 0.3991 |
| **G** | 0.7360 | 3.4901 | **0.4603** | 0.4632 |
| **CV** | 0.4156 | 0.6600 | 0.5121 | **0.1516** |

**Table 3: Comparison of Previous Approaches with Graph Baseline Models (GG-NN and MPNN)**

As we mentioned in Section 3, spatial relationships such as edge distances were not incorporated in our graph models due to the high cost of the re-implementation effort. The handcrafted molecular descriptors thus have a significant advantage over the graph models.
However, in Table 3, we observe a pattern similar to the results obtained in MPNNs [1]. In general, the molecular graph descriptors surpass the hand-engineered ones, with the MPPN achieving the highest accuracy. The GG-NN [17] has the highest R2 MAE, and the Bag of Bonds (BOB) also generally performs better than the Coulomb Matrix as in [1]. We note that BoB and CM, which are trained on top of a kernel ridge regression model, have more stable outcomes than the GG-NN and MPNN, which can be caused by the non-convexity of deep neural networks.

In contrast to the findings in MPNNs [1], we observe that GG-NN [17] performs relatively better than all other models on targets that are related to atomization energies (U, U0, H, and G). The GG-NN relative MAE difference compared to hand-engineered descriptors is also lower in our experiments. Multiple factors, such as the exclusion of distance features in our graph models, can explain the latter observation. Important aspects to take into consideration are the difference in the number of data points, and the complexity of molecules used in our experiments (11000 molecules) compared to the reviewed paper (135000 molecules), which are considerably higher. Indeed, graph neural network models prediction ability tends to be dependent on the quantity of data used and the richness of the features.

Another point to explain the smaller difference in MAE between the MPNNs and the hand-engineered baseline models compared to [1] is the possible dissimilarity in the choice of the kernel for the ridge regressor. For both BoB [12] and CM [11] featurizations, there were no specifications about the ridge regression kernel that was used, which could either be a Gaussian or Laplacian. We used the Laplacian kernel in our experiments as they are often favored for their ability to utilize information in non-local chemical compound space.

#### 5.0.2 Evaluation of MPNNs training strategies

We then investigate the setting of joint learning on all 12 targets and report the average MAE on Table 4.

| Model | MPNN | GG-NN |
|---|---|---|
| **Joint training** | 0.1576 | 0.4907 |
| **Individual training** | 0.4344 | 6.7055 |

**Table 4: Average MAE from training both MPNN and GG-NN jointly and individually**

In contrast to the findings of the original paper, the MAE obtained when jointly training the model with all targets is lower than that of the individual training. A thorough analysis will be needed to explain the difference in findings.

#### 5.0.3 Evaluation of MPNNs MAE based on dataset training size

Finally, we experimented with different sample sizes and analyzed the change in performance in Table 5. Unlike the previous experiments, we set the number epochs to 100 instead of 300 due to the more significant amount of data points used.

| Dataset Size | N=11k | N=35k | N=58k |
|---|---|---|---|
| **MPNN** | 0.1742 | 0.1307 | 0.09698 |
| **GG-NN** | 0.5059 | 0.4769 | 0.4653 |

**Table 5: Results from training both MPNN and GG-NN on different sized training sets (N denotes the number of training samples)**

As expected, the MAE ratio decreases as we augment the training datasets for both MPNN and GG-NN.

## 6 Proposed Baselines

One of the ideal features for the construction of a Quantitative structure–activity relationship (QSAR) descriptor is the ability to generate different values for structurally different molecules, even if the structural differences are small. Theoretically, a graph neural network that is capable of passing the Weisfeiler-Lehman (WL) graph isomorphism test would have a higher probability of meeting this requirement. The WL 1-dimensional form, "naïve vertex refinement," is analogous to neighbor aggregation [24] in GNNs such as GraphSAGE. Indeed it has been proven that with the right parameter initialization, GNNs can have the same expressiveness as the 1-WL algorithm [25].

There have been recent works that have been proven to be more powerful in representational capacities such as the Graph Isomorphism Network (GIN) [24] and k-GNNs [25]. However, we only consider strong baselines that were proposed either around the same time as MPPNs [1] or before, in order to make a fairer comparison. Along with GraphSAGE, we propose a variant of the Coulomb Matrix as another strong hand-engineered molecular descriptor and explain in more detail the reasoning behind our approach below.

#### 6.0.1 Coulomb Matrix Eigenspectrum

In the MPNNs paper, the Bag of Bonds is the molecular descriptor that has the second-lowest MAE among handcrafted featurizers. In order to get a better assessment of the performance of the hand-engineered features, we propose another competitive variant of the Coulomb Matrix named Coulomb Matrix Eigenspectrum.
One of the main issues of the Coulomb Matrix is that it is not invariant to permutations and re-indexing of the atoms [11]. This issue can be tackled by using the Coulomb sorted eigenspectrum, where the Coulomb Matrix is replaced by a feature vector of the eigenvalues, sorted in descending order. This variant of the Coulomb Matrix is invariant to permutation of atoms indices and has a lower dimension and thus a smaller computational time complexity.

#### 6.0.2 GraphSAGE

The design of MPNN's that can generalize effectively to larger graphs than those appearing in the training set is one of the future directions discussed in the investigated paper. To this end, we present GraphSAGE [26] as another strong baseline as part of the MPNNs framework. It is an inductive representation learning algorithm that has yielded impressive results on large-scale graphs. This model learns functions that generate the embeddings for a node by sampling and aggregating features and topological information from the Node's neighborhood.

The message functions in GraphSAGE correspond to the aggregator functions in "Inductive Representation Learning on Large Graphs" [26]. Among the three aggregator functions that were presented in the paper (mean, pooling, and LSTM aggregators) [26], we choose with the mean aggregator as the message function (7) and the Set2Set [14] readout function as in the other graph models mentioned above.

$$h_v^{t+1} = \sigma\left(W_t \sum_{w \epsilon N(v)} \frac{h^t}{|N(v)|} + B_t h_w^t\right) \tag{7}$$

#### 6.0.3 Modification of the featurization methodology

As this is a graph level task, we defy the natural convention of using the edges as the bonds. Instead, we add a naive implementation of the bonds at the node level for both the Gated Graph Recurrent Neural Network and our GraphSAGE variant, which we denote as GG-NN+Bond and GSage+Bond respectively.

## 7 Experiments and Evaluations

We conduct the experiments in this section using the same parameters that we mentioned in our reproduction section above.

### 7.0.1 Evaluation of Proposed Baselines

| Target | BOB | CM | CM-E | GG-NN | GG-NN+Bond | MPNN | GSAGE+Bond |
|---|---|---|---|---|---|---|---|
| **mu** | 0.6724 | 0.8078 | 0.8632 | 0.7639 | **0.0407** | 0.1523 | 0.1404 |
| **alpha** | 0.7782 | 1.4809 | 1.1377 | 0.9228 | **0.2690** | 0.3847 | 0.6388 |
| **HOMO** | 0.0074 | 0.017 | 0.0111 | 0.0081 | **0.0014** | 0.0034 | 0.0041 |
| **LUMO** | 0.0106 | 0.0171 | 0.0166 | 0.0111 | **0.0021** | 0.0037 | 0.0049 |
| **gap** | 0.0122 | 0.0196 | 0.0202 | 0.0118 | **0.0021** | 0.0065 | 0.0068 |
| **R2** | 22.2674 | 37.7441 | 48.7628 | 76.7278 | 2.7471 | **2.5781** | 5.0963 |
| **ZPVE** | 0.0007 | 0.0010 | 0.0024 | 0.0007 | **0.0004** | **0.0004** | 0.0036 |
| **U0** | 0.7360 | 3.4899 | 0.6403 | **0.3955** | 1.4430 | 0.5545 | 2.5040 |
| **U** | 0.7360 | 3.4895 | 0.6403 | **0.2621** | 0.8049 | 0.5218 | 2.5040 |
| **H** | 0.7360 | 3.4898 | 0.6403 | **0.3899** | 1.085 | 0.3991 | 1.9943 |
| **G** | 0.7360 | 3.4901 | 0.6403 | **0.4603** | 1.3811 | 0.4632 | 2.0114 |
| **CV** | 0.4156 | 0.6600 | 0.6066 | 0.5121 | 0.1602 | **0.1516** | 0.2853 |

**Table 6: Comparison of Previous Approaches with Graph Baseline Model (GG-NN), MPNN and ours (GG-NN+Bond and GSAGE+Bond)**

Our results show that the Coulomb Matrix Eigenspectrum (CM-E) generally performs better than the original Coulomb Matrix. We also note that it achieves higher performance on the target properties related to the atomization energies/tightness of bonds (U0, U, H, G) than the Bag of Bonds.

Although a lower accuracy than GG-NN on the U0, U, H, and G was reported for GG-NN + bonds, we also observe a significant performance increase on several target properties results, which defeat all other baselines. The simple addition of the bonds as part of the Node features thus seems to have a positive impact on the GG-NN prediction ability for most properties.

The MPNN and our proposed graph model attain close MAE on mu, HOMO, LUMO, gap, and CV. Even though the MPNN [1] outperforms our proposed graphSAGE variant [26] on this specific application, our results still demonstrate that GSAGE+Bond is a robust baseline with promising results. We note that due to resource limitations, the assessment above, with the use of small sample size is not necessarily a good indicator of GraphSAGE's potential especially for large graphs.

Two main advantages of using GraphSAGE over the MPNN are for (1) its ability to generate embeddings for nodes that were not present during training and (2) its capacity to use neighborhood sub-sampling for an effective batch-training algorithm. The per-batch space and time complexity for GraphSAGE is [26]:

$$\prod_{i=1}^{K} S_i$$

where $K$ is the number of layers and $S$ is the fixed neighborhood set. This subsampling strategy is difficult to implement with graph neural networks with edge features, and the investigated paper recorded unsuccessful attempts at combining a similar methodology called "towers" with the edge network message function.

## 8 Discussion and Conclusion

In this report, we investigated the reproducibility of the published machine learning paper "Neural Message Passing for Quantum Chemistry" [1]. A series of experiments were conducted in order to verify the findings of the authors. The main difficulties that we encountered were the unavailability of the original paper's code, the inability to identify the last target property labeled as "Omega" and the lack of clarity of some of the methodologies that were used, such as the choice of the kernel for the ridge regression model.

We present a set of results, highlighting as well the difficulty in reproducing works in machine learning with resource limitations. Although we did not incorporate spatial relationships in the initial features of the graph featurizers that we experimented with, they still outperformed the hand-engineered molecular descriptors. The main limitation of both the Coulomb Matrix (CM) and the Bag of Bonds (BoB) model is the constant dimension, as the length of the feature vectors depends on the number of atoms in the largest molecule of interest.

The improvement of the performance of the GG-NN and GSAGE with the naive addition of the bond as part of the node features, demonstrates the power of graph representation models when we integrate rich data at the initialization phase.

Our proposed GSAGE+Bond baseline achieves results close to the MPNN for most target properties except for those associated with the atomization energies. It also allows us to address better the shortcomings that were highlighted in MPNNs. Graph neural networks with edge features are more expensive than those without as the intermediate edge-based activations need to be stored. The node sampling strategy introduced in GraphSAGE [26] is more efficient to improve computing and memory usage for large graphs. The inductive property of GraphSAGE could potentially help overcome several key issues concerning the limited amount of labeled data when building machine learning models on molecules.

Although we get a lower MAE with GG-NN +Bond, we maintain GraphSAGE as our proposed baseline for large graphs, as using GG-NN can be problematic for large graphs. Indeed, GG-NN needs to run the recurrent function multiple times over all nodes, requiring the intermediate states of all nodes to be stored in memory [27] and thus demands a higher computational space capacity.
Reproducing MPNNs [1] allowed us to get a better assessment of the amount of effort needed to implement the reviewed sophisticated models, which can be quite expensive. 2D descriptors such as SMILES (Simplified molecular-input line-entry system) based methods, are in practice less cumbersome and could also be used as effective alternatives. Overall, the reproduced graph representation learning featurizers achieved higher performance than the hand-engineered ones, with the MPNN (enn-s2s) attaining the lowest MAE. We thus validate the reproducibility of the paper "Neural Message Passing for Quantum Chemistry".

## 9 Acknowledgements

The author would like to thank William L. Hamilton for supervising her for this project and the helpful discussions on this work. She would also like to thank Matthias Fey for support on the PyTorch Geometric library.

## References

[1] Patrick F. Riley Oriol Vinyals George E. Dahl Justin Gilmer, Samuel S. Schoenholz. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[2] Klaus Capelle. A bird's-eye view of density-functional theory. *arXiv preprint arXiv:0211443*, 2006.

[3] Xavier Oriols Albert Benseny, David Tena. On the classical schrodinger equation. *arXiv preprint arXiv:1607.00168*, 2016.

[4] Matthew Lai Danilo Rezende Koray Kavukcuoglu Peter W. Battaglia, Razvan Pascanu. Interaction networks for learning about objects, relations and physics. *arXiv preprint arXiv:1612.00222*, 2016.

[5] Arthur Szlam Yann LeCun Joan Bruna, Wojciech Zaremba. Spectral networks and deep locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2014.

[6] Michaël Defferrard Xavier Bresson Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:11606.09375*, 2016.

[7] Jorge Aguilera-Iparraguirre Rafael Gomez-Bombarelli Timothy Hirzel Al an Aspuru-Guzik Ryan P. Adams David Duvenaud, Dougal Maclaurin. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292*, 2015.

[8] Stefan Chmiela Klaus R. Muller Alexandre Tkatchenko Kristof T. Schutt, Farhad Arbabzadah. Quantum-chemical insights from deep tensor neural networks. *arXiv preprint arXiv:1609.08259*, 2016.

[9] Lorenz C. Blum Jean-Louis Reymond Lars Ruddigkeit, Ruud van Deursen. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *J. Chem. Inf. Model. 2012, 52, 11, 2864-2875*, 2012.

[10] Matthias Rupp O. Anatole von Lilienfeld Raghunathan Ramakrishnan, Pavlo O. Dral. Quantum chemistry structures and properties of 134 kilo molecules. *Sci Data 1, 140022 (2014) doi:10.1038/sdata.2014.22*, 2014.

[11] Klaus-Robert Muller O. Anatole von Lilienfeld Matthias Rupp, Alexandre Tkatchenko. Fast and accurate modeling of molecular atomization energies with machine learning. *arXiv preprint arXiv:1109.2618v1*, 2011.

[12] O. Anatole von Lilienfeld David J. Yaron Christopher R. Collins, Geoffrey J. Gordon. Constant size molecular descriptors for use with machine learning. *arXiv preprint arXiv:1701.06649*, 2017.

[13] Jialu Liu. Image retrieval based on bag-of-words model. *arXiv preprint arXiv:1304.5168*, 2013.

[14] Manjunath Kudlur Oriol Vinyals, Samy Bengio. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391v4*, 2015.

[15] Ah Chung Tsoi Markus Hagenbuchner Gabriele Monfardini Franco Scarselli, Marco Gori. The graph neural network model. *IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 20, N*, 2009.

[16] KyungHyun Cho Yoshua Bengio Junyoung Chung, Caglar Gulcehre. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[17] Marc Brockschmidt Daniel Tarlow Yujia Li, Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493v4*, 2017.

[18] Jian Tang William L. Hamilton. Graph representation learning tutorial. *AAAI*, 2019.

[19] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Pytorch, team pytorch. 2017.

[21] M. S. Butrico M. D. Fuge D. C. Elton, Z. Boukouvalas and P. W. Chung. Applying machine learning techniques to predict the properties of energetic materials. *Scientific Reports 8,9059*, 2018.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Jimmy Lei Ba Diederik P. Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980v9*, 2014.

[24] Jure Leskovec Stefanie Jegelka Keyulu Xu, Weihua Hu. How powerful are graph neural networks. *arXiv preprint arXiv:11810.00826*, 2019.

[25] Matthias Fey William L. Hamilton Jan Eric Lenssen Gaurav Rattan Martin Grohe Christopher Morris, Martin Ritzer. Weisfeiler and leman go neural: Higher-order graph neural networks. *arXiv preprint arXiv:1810.02244*, 2018.

[26] Jure Leskovec William L. Hamilton, Rex Ying. Inductive representation learning on large graphs. *arXiv preprint arXiv:706.02216*, 2017.

[27] Fengwen Chen Guodong Long Zonghan Wu, Shirui Pan. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596v4*, 2019.