Universal Representation of Chemical Entities for Machine Learning

Harold Benoit
Signal Processing Laboratory LTS2
EPFL
harold.benoit@alumni.epfl.ch

Abstract

Molecular and material machine learning have seen tremendous progress in the past few years, especially thanks to the usage of GNNs. Data representation of chemical entities is a key component to the success of ML in these fields. Nonetheless, almost of the work done is focused on model architecture or a data representation tightly coupled with some specific architecture design (SE-3 transformers, ...).

Thus, to the best of our knowledge, we have developed the first universal and model-agnostic graph representation of chemical entities. In this work, we show how this representation allows us to get good performance on a wide variety of molecular and crystal datasets, discuss the different components of the data representation by how they affect downstream performance, and the possibility of transfer learning.

This new data representation paves the way for a possible crossover of knowledge between molecular and matieral science, whether through transfer learning or joint effort in modelling/data collection spanning multiple chemical fields.

1 Introduction

The main goal of this work is to develop a universal and model-agnostic graph representation of chemical entities for machine learning. This would possibly allow for crossover of knowledge between molecular and material science, whether literally using transfer learning or by having joint effort in modeling or data collection spanning multiple chemical fields.

One additional goal is also to try to enable the representation to go beyond the current "static" view of chemical entities widely used everywhere. To be more precise, for example, when representing a molecule as a graph, the usual procedure is to define the edges using interatomic bonds or some cut-off distance. This process often assumes that the molecule is in its minimum-energy state e.g. "static".

This assumption is in conflict with the fact that molecules are dynamic objects. One example is the usage of shape-altering catalysts that bring reactant molecules together by "bending them", thus accelerating the rate of reaction. The current representations in machine learning do not allow enough flexibility to represent this "bending" process.

2 Background and related work

This section serves as an overview of the work done on chemical entities (molecules and materials) in machine learning. More specifically, we will mainly talk about the usage of graphs, whether through specific model architecture or representation learning.

We will conclude that, to the best of our knowledge, no universal and model-agnostic representation of chemical entities exists.

Preprint. Under review.

2.1 GNNs

In the latest advances in molecular[23] and material machine learning[13], the usage of graph-based learning techniques has been ubiquitous. Indeed, graphs naturally describe objects with rich structural and spatial information.

2.2 Molecular Graph Networks

Molecules are essentially atoms and bonds interconnecting atoms, which naturally lead themselves to graph representations. Compared to to the SMILES textual representation, molecular graphs provide a more exhaustive representation.

As a result, graph-based machine learning models have evolved much faster than sequence-based models for molecules. Additionally, more and more general graph learning papers[2, 8, 12] employ molecular graph datasets to examine the performance of their algorithms as well.

Furthermore, the usage of 3D information of atoms has started to gain traction in the research. This has resulted in several ways of encoding 3D information in the graph representation or the model architecture.

One line of research for 3D molecular graphs has been equivariant graph neural networks (EGNNs), including SE(3)-transformers[7], PaiNN[14], NequIP [18], Noisy Nodes[22]. The raw input of these methods usually contains the absolute information, such as coordinates in the Cartesian coordinate system. One downside is that the networks components of EGNNs need to be carefully designed to satisfy equivariance under continous 3D roto-translations.

Another category of methods take purely relative 3D information as input, such as distances between atoms, angles between bonds, angles between planes, etc. Hence, the network becomes naturally invariant to 3D roto-translations. The development of these methods is still ongoing. The usage of distance between atoms seems general to all methods, but the research diverges on the additional geometries to use whether it be angles or torsions or both.

Examples are DimeNet [20], GemNet[19], and Directional MPNN[21] which propose directional message embeddings. Although the input is still 2D molecular graphs, they consider not only the distances between atoms but also the spatial directions, which are calculated by atoms' 2D coordinates. They use directional information by transforming messages based on the angle between atoms.

However, how different geometries (distance, angle or torsion) contribute to the information aggregation process still lacks rigorous justification. There is no established standard spatial information learning method for now.

SphereNet[25] proposes spherical message passing, as an attempt to unify all geometries in a single framework and obtain complete identification of 3D graph structures, by using the spherical coordinate system.

2.3 Molecular Representation Learning (MRL)

Another line of work in molecular machine learning focuses less on the right network architecture and more on molecular representation learning.

Wang et al.[16] use chemical reactions to assist the learning. Using a contrastive loss, they preserve the equivalence of molecules with respect to chemical reactions in the embedding space by forcing the sum of the reactant embeddings and the sum of product embeddings to be equal.

A more recent attempt, Uni-Mol[29], proposes a framework, incorporating 3D informations by using 3D molecules conformation, to pre-train SE(3)-transformers (mentioned at 2.2) on huge unlabelled datasets. It achieves excellent performance in many molecular downstream tasks.

2.4 Polymers

Polymers are at the intersection between molecules and crystals. Indeed, in contrast to organic molecules, polymers are often not well-defined single structure but *an ensemble of similar molecules*. But they do not follow an exact lattice pattern like crystals. Polymers may be more easily described as stochastic objects.

Aldeghi et al.[17] introduce a graph-based representation for polymers and a weighted directed neural message passing scheme to appropriately take into account the stochasticity by weighing the edges by their probabilities.

2.5 Crystal Graph Networks

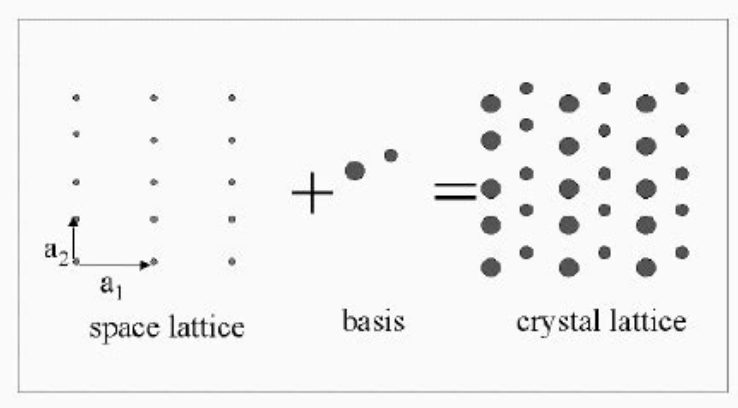

Figure 1: Lattice and basis are the two components forming a crystal

A crystal is a solid material in which the atoms, molecules, or ions are arranged in an orderly repeating pattern extending in all three spatial dimensions. The repeating pattern is called the crystal lattice, and the basic unit of the lattice is called the unit cell.

The spatial arrangement between atoms naturally leads to a graph representation, although one needs to be careful about how to encode the repeating pattern of the lattice in a graph. In our representation, we tackle this problem by taking into account the periodic boundary conditions when computing the distances (section 3.1.5).

In materials science, GNNs have also been shown[13] to provide substantial performance improvements for crystal property predictions compared to descriptor-based machine learning models.

There has been rapid progress in the development of GNN architectures for predicting material properties such as SchNet[3], Crystal Graph Convolutional Neural Networks (CGCNN)[4], improved Crystal Graph Convolutional Neural Networks (iCGCNN)[9], OrbNet[10], etc.

This family of models represents crystalline material as a graph with one node for each constituent atom and edges corresponding to bonds. A common theme is the use of elemental properties as node features and distances and/or bond valences as edge features.

One of the latest, ALIGNN[13], makes an explicit effort to incorporate bond angles, the justification being that many important material properties (especially electronic properties such as band gaps) are highly sensitive to structural features such as bond angles and local geometric distortions

MatErials Graph Network (MEGNet)[5] is an attempt to unify molecules and crystals. A MEGNet model is made up of multiple MEGNet blocks. A MEGNet block is a graph-to-graph mapping, taking in as input nodes, edges and global attributes. The main contribution is that their MEGNet model works both with molecules and crystals, although the two have different attributes (nodes, edges and global), with the molecule representation being richer.

3 Universal Representation

As we have seen, a lot of work has been done on both molecular and crystalline machine learning.

Nonetheless, almost all of the work are either model-specific or have a special data representation tightly coupled with a specific model architecture.

Thus, to the best of our knowledge, no universal and model-agnostic graph representation of chemical entities for machine learning exists.

A good representation should capture the relevant chemical properties of the object or at least its underlying structure from which the properties are derived (i.e. the presence of a covalent bond between two atoms is derived from the distance between the two atoms, thus it is theoretically sufficient to have the atoms positions without making the covalent bond explicit.)

One key point to remember is that the constraints imposed by universality and model-agnosticity will inevitably impose constraints on the representation. Thus, state-of-the-art results are not to be expected on e.g. molecular property prediction, as we cannot add molecular-specific or crystal-specific features without breaking universality.

3.1 Representation description

The graph is composed of nodes, with node attributes. Each atom of the chemical entity is one node. The edges, with edge attributes, are defined in 3.1.4.

3.1.1 Nodes

Given the constraint of universality, atoms can only be described in a general context, mostly facts from the periodic table.

3.1.2 Numerical node attributes

Given an atom or element a , its numerical attributes or features are:

- The atomic number of the element.
- The atomic radius of the element in Ångstroms. It is not defined for radioactive elements and noble gases.
- The atomic mass of the element in amu.
- Average ionic radius of the element in Ångstroms. The average is taken over all oxidation states of the element for which data is present.
- Average cationic radius of the element in Ångstroms. The average is taken over all oxidation states of the element for which data is present.
- Average anionic radius of the element in Ångstroms. The average is taken over all negative oxidation states of the element for which data is present.
- Maximum oxidation state for element.
- Minimum oxidation state for element.
- Periodic table row (also known as periods) of the element. There are 7 possible periods.
- Periodic table group (also known as column) of the element.There are 18 possible groups.

One might remark that atomic number, oxidation state, and periodic table features are *integer* features and should, in theory, be expanded into one-hot vectors. Nonetheless, in practice[2], it has been observed that *integer* features with high class cardinality (such as atomic number) are better kept in integer form for downstream performance.

3.1.3 Boolean node features

Knowledge about the element classification in general groups can be useful. The boolean feature for a given element is defined as:

- True if element is noble gas.
- True if element is a post-transition or poor metal.
- True if element is a rare earth metal (i.e. lanthanoid or actinoid)
- True if element is a metal.
- True if element is a metalloid.
- True if element is an alkali metal.

- True if element is an alkaline earth metal (group II).
- True if element is a halogen.
- True if element is a chalcogen.
- True if element is a lanthanoid.
- True if element is a actinoid.
- True if this element can be quadrupolar.

Whether Boolean features are useful or not will be investigated in the benchmarks section.

3.1.4 Edges

To achieve our goal of going beyond "static" views of chemical entities, we cannot rely on bonds or cutoff distances to define our edges. Indeed, it would be ideal that the same chemical entity in different conformations would map approximately to the same representations (at the exception of the inter-atom distances).

The solution found to this problem is to assume that any atom can interact with every other atom (as they may move further or closer from each other), and thus they are all connected. This results in a complete graph.

3.1.5 Edge attributes

For an edge e between atom a_1 and a_2, there are two edge attributes:

- The result of testing if a_1 and a_2 are bonded, resulting into three possible outcomes: [YES, NO, NO DATA]. The result is then converted into a one-hot vector of length 3.
- The euclidean distance between a_1 and a_2. For crystals, the periodic boundary conditions are taken into account i.e. the distance between a_1 and the closest mirror image of a_2.

3.1.6 Invariance to 3D roto-translations

Any good description must satisfy the necessary rotational, translational, and permutational invariances.

We achieve the rotational and translational invariances by only using relative geometries such as distances between atoms instead of absolute geometries such as cartesian coordinates.

Permutational invariance is directly achieved by using a graph representation.

3.2 Representation implementation

The code for the data representation relies on two excellent libraries that we must cite: *RDKit*[28] and *pymatgen*[1].

3.2.1 Input format

To make sure that the data representation pipeline is applicable on most datasets, one should support the most readily-available format.

For molecules, the SMILES textual format is supported. All other molecule format such as .mol, supported by *RDKit*, can also be very easily integrated.

For crystals, the **pymatgen Structure** format is supported.

3.2.2 3D coordinate generation

To be able to support the SMILES format, a 3D coordinate generation pipeline, using RDKit, was created.

The pipeline takes a SMILES string, converts into a molecule, hydrogens are made explicit, and finally, the third version of the ETKDG[11] method is used to generate the conformers. The process is seeded to ensure reproducibilty.

The generation process can be unstable and fail for some molecules (if they are too big, ...) and thus, they need to be skipped.

3.2.3 Explicit hydrogens

For our data representation, in the case of molecules, the hydrogens are implicit by default. Whether it is useful or not to make them explicit will be studied in the next section.

4 Benchmarks

After defining our representation, it is now time to check whether it translates to acceptable performance on a multitude of downstream tasks and datasets. It was chosen that a single model would be used on all tasks and datasets.

Indeed, the aim of this work is not to produce SOTA results but to produce a good universal and model-agnostic representation. Spending time fine tuning several models would introduce a "modelling bias", artificially inflating the capacity of the data representation to be useful in many contexts (i.e. datasets **and** models).

4.1 Model

The model is heavily inspired from the GeneralGNN[1] model from the Spektral[27] library, considered to be a good starting point for any general graph-related task.

It is made up of:

- 3 message passing layers.
- Aggregation function: global add pool
- 2 fully-connected layers with dimension 256.

A message passing layer is defined as:

- GeneralConv[2] with all the default parameters. The number of input and output channels is 256.
- BatchNorm
- PReLU

4.2 Training

The results reported in the below tables follow the given training routine:

- Number of epochs:100
- Optimizer: Adam
- Learning rate: Starting at $\gamma = 10^{-3}$ it exponentially decays by $\epsilon = 0.999$ every 50 steps of training.
- Batch size: 32
- No hyper-parameter tuning
- The reported loss/AUC is the best validation loss/AUC over all epochs (i.e. a validation epoch is ran at the end of every training epoch).
- The datasets are split 80% training, 20% validation. Random splitting is applied for crystal datasets and scaffold splitting is applied for molecular datasets.
- Every experiment is run with 3 different splitting seeds (100, 200, 300) and we report mean and standard deviation.

[1] https://graphneural.network/models/#generalgnn

[2] https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch_geometric.nn.conv.GeneralConv

The training was done on an internal cluster and here are its characteristics:

- 2 x Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz (24 cores each)
- 256GB of RAM
- 4TB of SSD storage
- 3 x GeForce Ti1080 GPU with 11GB of RAM each
- Ubuntu 20.04LTS

4.3 Results

We will denote our representation by UCR, which stands for "Universal Chemical Representation".

4.3.1 Molecular datasets

4 datasets are used in this section: FreeSolv, Lipophilicity, BACE, and BBBP. They are downloaded using DeepChem[26] S3 storage. Each dataset $D = (X, y)$ comes in the form of X being a list of SMILES textual description and y a vector of numerical or boolean target. The data representation uses explicit hydrogens in this case. The results can be found in Table 1.

Task	**Physical chemistry**		**Bioactivity**	**Pharmacokinetics**
Dataset	**FreeSolv**	**Lipophilicity**	**BACE**	**BBP**
Samples	642	4200	1513	2050
Metrics	**RMSE**		**AUC**	
AttentiveFP	1.371(0.446)	0.783(0.036)	0.850(0.017)	0.872(0.024)
MolMapNet	1.398(0.312)	0.731(0.012)	0.868(0.094)	0.911(0.013)
GLAM	1.319(0.346)	**0.596(0.025)**	**0.888(0.033)**	**0.932(0.015)**
UCR	**1.204(0.042)**	1.270(0.020)	0.703(0.0036)	0.662533(0.1466)

Table 1: Performance comparison on datasets of molecular properties. All datasets are scaffold split 80-20. All methods are run with three different split seeds and then we take the average score and the standard deviation (in parentheses). The highlighted text with bold-black style means the best. Results taken from the GLAM paper[24].

As can be seen, the performance varies quite a lot between datasets. Indeed, the model seems to perform very well on **FreeSolv**, moderately well on **BACE**, consistently poorly on **Lipophilicity** and with high variance on **BBBP**.

4.3.2 Crystal datasets

All datasets are from the benchmark test suite for supervised machine learning on inorganic materials, Matbench[6]. Each dataset $D = (X, y)$ comes in the form of X being a list of **pymatgen.Structure** and y a vector of numerical or boolean target. The data representation uses numerical **and** boolean features, in this case. The results can be found in Table 2.

As a good first step, UCR consistently beats Dummy. Nonetheless, UCR seems to perform poorly on **phonons** (most likely due to underfitting, as the dataset is quite small, only 1265 samples) and **log_gvrh**. Finally, it can be noted that UCR performs as well as other methods in **dieletric** (CGCNN v2019[4]) and in **perovskites** (CrabNet[15]).

4.4 Explicit hydrogens or not?

We would like to investigate the impact of explicit hydrogens on performance on molecular tasks. The results of this comparison can be seen in Table 3. Based on these results, it seems hard to draw a clear confusion as to whether or not it is better to make hydrogens explicit, although there does seem to be an advantage for implicit hydrogen (BBBP performance improvement is drastic).

Overall, in practice, it seems that on smaller datasets, making hydrogen explicit makes the representation slightly too complex, thus hindering the fitting of the model. But, when training on larger

Matbench				
Dataset	**dieletric**	**log_gvrh**	**phonons**	**perovskites**
Samples	4764	10987	1265	18928
Metrics	**MAE**			
ALIGNN	0.3449(0.0871)	**0.0715(0.0006)**	29.5385(2.1148)	**0.0288(0.0009)**
MODNet	**0.2711(0.0714)**	0.0731(0.0007)	34.2751(2.0781)	0.0908(0.0028)
SchNEt	0.3277(0.0829)	0.0796(0.0022)	38.9636(1.9760)	0.0342(0.0005)
CGCNN v2019	0.5988(0.0833)	0.0895(0.0016)	57.7635(12.3109)	0.0452(0.0007)
CrabNet	0.3234(0.0714)	0.1014(0.0017)	55.1114(5.7317)	0.4065(0.0069)
MegNet	0.3391(0.0745)	0.0871(0.0013)	**28.7606(2.5767)**	0.0352(0.0016)
AMMExpress v2020	0.3150(0.0672)	0.0874(0.0020)	56.1706(6.7981)	0.2005(0.0085)
UCR	0.5604(0.0561)	0.2486(0.0260)	157.12(16.258)	0.4215(0.0127)
Dummy	0.8088(0.0718)	0.2931(0.0031)	323.9822(17.7269)	0.5660(0.0048)

Table 2: Performance comparison on tasks from MatBench[6]. Results are taken from the official MatBench[6] site. For the results of UCR, all datasets are random split 80-20 and the training is ran with three different split seeds and then we take the average score and the standard deviation (in parentheses). The highlighted text with bold-black style means the best. Dummy is the classifier that ouputs the mean of the dataset.

Task	**Physical chemistry**		**Bioactivity**		**Pharmacokinetics**
Dataset	**FreeSolv**	**Lipophilicity**	**BACE**		**BBBP**
Metrics	**MAE**			**AUC**	
No Hydrogen	0.804(0.038)	**0.895(0.026)**	**0.865(0.085)**	0.697(0.0005)	**0.789(0.004)**
Hydrogen	**0.794(0.036)**	0.913(0.013)	0.900(0.067)	**0.703(0.036)**	0.662(0.146)

Table 3: Performance comparison between explicit and implicit hydrogen on datasets of molecular properties. All datasets are scaffold split 80-20. All methods are run with three different split seeds and then we take the average score and the standard deviation (in parentheses). The highlighted text with bold-black style means the best.

datasets (e.g. QM9[3]), it was observed that explicit hydrogen could prove massively beneficial to approach SOTA on some tasks.

4.5 Boolean features or not?

Matbench				
Dataset	**dieletric**	**log_gvrh**	**phonons**	**perovskites**
Metrics	**MAE**			
No Boolean	0.6199(0.060)	**0.2333(0.0136)**	164.32(16.287)	0.4279(0.0164)
Boolean	**0.5604(0.0561)**	0.2486(0.0260)	**157.12(16.258)**	**0.4215(0.0127)**

Table 4: Performance comparison between adding or not boolean features on tasks from MatBench[6]. For the results of UCR, all datasets are random split 80-20. All methods are run with three different split seeds and then we take the average score and the standard deviation (in parentheses). The highlighted text with bold-black style means the best.

As the boolean features almost only encompass features relevant to materials, we tested the performance improvement solely on the crystal datasets. As can be seen in Table 4, adding boolean features does seem to be helpful, as it improves performance in almost all tasks.

4.6 Effect of randomness of 3D generation on results

Given that the 3D coordinates generation process (discussed in Section 3.2.2) requires sampling a diverse set of conformers of a molecule, it is a possibility that randomness affects the quality of the data representation. Thus, we would like to know how much variation can be incurred by the generation process.

[3]The QM9 performance comparison is not reported because of a lack of time and resources.

To test this, we will do 3 training runs, each with a different seed for 3D coordinates generation, but keeping the same split seed. We can then compute the coefficient of variation[4] (COV) of the loss. Given that the COV is a relative measure (dimensionless), it takes into account the scale of the data and will let us compare between datasets.

To give further context to the observed variation, we also report the COV of the training that varies the split seed, while keeping the same generation seed. The results are reported in Table 5.

Task	**Physical chemistry**		**Bioactivity**		**Pharmacokinetics**
Dataset	**FreeSolv**	**Lipophilicity**	**BACE**		**BBBBP**
Metrics	**Coefficient of variation**				
Split seed	4.54%	1.43%	7.52%	0.52%	22.14%
Generation seed	3.44%	2.07%	5.79%	0.32%	0.41%

Table 5: Comparison of coefficient of variation of the loss between varying the split seed (while keeping the same generation seed) and varying the generation seed (while keeping the same split seed). All datasets are scaffold split 80-20. All methods are run with three different split/generation seeds.

As we can see, randomness of the generation process does affect the performance of the model, and brings about as much as variation as varying the split seed (except for **BBBP**). Thus, it seems like a good advice to tune the generation seed if performance is important.

5 Discussion

5.1 Transfer Learning

Given the universality of our representation, a natural question arises of whether "knowledge" can be transferred from a molecular-trained model to a crystal-trained model and vice-versa.

To investigate this question, we used two datasets. First, the QM9[5] dataset and the HOMO-LUMO gap target. Secondly, the MatBench[6] "matbench_mp_gap" and the band gap target.

The HOMO-LUMO gap in chemistry and the band gap in solid-state physics are similar in that they both refer to the energy gap between the highest occupied energy level and the lowest unoccupied energy level in a material. Nonetheless, they are also different in the sense that they are usually used to describe different aspects of the materials.

Given this similarity, one can hypothesise that a transfer of knowledge is possible between models.

To test our hypothesis, the test we applied was to train a model on QM9 for 4 epochs, and then took the same model as a starting state to train on MatBench for 4 epochs. We then compared the performance with a model trained on MatBench for 8 epochs (to approximately match the training budget).

No clear improvement in performance by using the pre-trained model was found. Whether this is the case because the pre-trained model didn't converge yet or because there isn't any knowledge to be transferred is hard to conclude.

5.2 Extension to polymers

Extending the representation to polymers should prove to be fairly easy, following the lead of Aldeghi et al.[17], one could add an edge feature indicating the probability of an edge, to take into account the stochastic nature of polymers.

5.3 Extension to inorganic bulk materials

Extending the representation to inorganic bulk materials only requires to implement the ability to compute distances between atoms. This is left for future work.

[4] https://en.wikipedia.org/wiki/Coefficient_of_variation

[5] http://quantum-machine.org/datasets/

5.4 Limitations

This representation is obviously not without its flaws. Here are its main flaws:

- Memory heavy and high dimension representation because of the usage of a complete graph, which scales quadratically in the number of atoms.
- No global attribute (e.g. temperature). As there is no consensus yet on how to use global graph information in GNNs, the restriction of model-agnositicity forbids us from adding it to the representation. Nonetheless, the representation can be very easily extended with global information when the model is specified. For interested users, ready-to-use universal global features are already available in the codebase.
- Supporting SMILES textual description and the requirement of 3D information of our data representation implies that we must have molecule conformer generation pipeline. This pipeline suffers two issues: randomness (discussed in 4.6) and failure (some large molecules are not supported). Nonetheless, over the 4 molecular datasets represented, only 14 molecules have been skipped.

5.5 Future work

Given a universal representation, some possibilities to leverage this advantage are transfer learning of molecular pre-trained models to crystallographic datasets and vice-versa , or generative modelling.

6 Conclusion

In this work, we have developed a universal and model-agnostic graph representation of chemical entities. We have shown how this representation allows us to get good performance on a wide variety of molecular and crystal datasets. We have discussed the different components of the data representation and how they possibly affect downstream performance. We also have discussed the possibility of doing transfer learning, and how extending the representation to several other chemical entities should require little work. Finally, we discussed the limitations of our approach and possible future work.

Acknowledgements

Many thanks to Daniel Probst, my supervisor at LTS2, for his help and insights during the whole project.

Code availability

The code for this project can be found at https://github.com/HaroldBenoit/chem-ml-repr.

References

[1] Shyue Ping Ong et al. “Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis”. In: *Computational Materials Science* 68 (Feb. 1, 2013), pp. 314–319. ISSN: 0927-0256. DOI: 10.1016/j.commatsci.2012.10.028. URL: https://www.sciencedirect.com/science/article/pii/S0927025612006295 (visited on 01/08/2023).

[2] Justin Gilmer et al. “Neural message passing for Quantum chemistry”. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. Sydney, NSW, Australia: JMLR.org, Aug. 6, 2017, pp. 1263–1272. (Visited on 01/05/2023).

[3] Kristof T. Schütt et al. *SchNet: A continuous-filter convolutional neural network for modeling quantum interactions*. Dec. 19, 2017. DOI: 10.48550/arXiv.1706.08566. arXiv: 1706.08566[physics,stat]. URL: http://arxiv.org/abs/1706.08566 (visited on 01/05/2023).

[4] Tian Xie and Jeffrey C. Grossman. "Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties". In: *Physical Review Letters* 120.14 (Apr. 6, 2018), p. 145301. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.120.145301. arXiv: 1710.10324[cond-mat]. URL: http://arxiv.org/abs/1710.10324 (visited on 01/05/2023).

[5] Chi Chen et al. "Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals". In: *Chemistry of Materials* 31.9 (May 14, 2019). Publisher: American Chemical Society, pp. 3564–3572. ISSN: 0897-4756. DOI: 10.1021/acs.chemmater.9b01294. URL: https://doi.org/10.1021/acs.chemmater.9b01294 (visited on 01/05/2023).

[6] Alexander Dunn et al. "Benchmarking materials property prediction methods: the Matbench test set and Automatminer reference algorithm". In: *npj Computational Materials* 6.1 (Sept. 15, 2020). Number: 1 Publisher: Nature Publishing Group, pp. 1–10. ISSN: 2057-3960. DOI: 10.1038/s41524-020-00406-3. URL: https://www.nature.com/articles/s41524-020-00406-3 (visited on 01/15/2023).

[7] Fabian B. Fuchs et al. *SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks*. Nov. 24, 2020. arXiv: 2006.10503[cs,stat]. URL: http://arxiv.org/abs/2006.10503 (visited on 01/05/2023).

[8] Weihua Hu et al. *Strategies for Pre-training Graph Neural Networks*. Feb. 18, 2020. DOI: 10.48550/arXiv.1905.12265. arXiv: 1905.12265[cs,stat]. URL: http://arxiv.org/abs/1905.12265 (visited on 01/05/2023).

[9] Cheol Woo Park and Chris Wolverton. "Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery". In: *Physical Review Materials* 4.6 (June 1, 2020). Publisher: American Physical Society, p. 063801. DOI: 10.1103/PhysRevMaterials.4.063801. URL: https://link.aps.org/doi/10.1103/PhysRevMaterials.4.063801 (visited on 01/05/2023).

[10] Zhuoran Qiao et al. "OrbNet: Deep Learning for Quantum Chemistry Using Symmetry-Adapted Atomic-Orbital Features". In: *The Journal of Chemical Physics* 153.12 (Sept. 28, 2020), p. 124111. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/5.0021955. arXiv: 2007.08026[physics]. URL: http://arxiv.org/abs/2007.08026 (visited on 01/05/2023).

[11] Shuzhe Wang et al. "Improving Conformer Generation for Small Rings and Macrocycles Based on Distance Geometry and Experimental Torsional-Angle Preferences". In: *Journal of Chemical Information and Modeling* 60.4 (Apr. 27, 2020). Publisher: American Chemical Society, pp. 2044–2058. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.0c00025. URL: https://doi.org/10.1021/acs.jcim.0c00025 (visited on 01/08/2023).

[12] Yuning You et al. "Graph Contrastive Learning with Augmentations". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 5812–5823. URL: https://proceedings.neurips.cc/paper/2020/hash/3fe230348e9a12c13120749e3f9fa4cd-Abstract.html (visited on 01/05/2023).

[13] Kamal Choudhary and Brian DeCost. "Atomistic Line Graph Neural Network for improved materials property predictions". In: *npj Computational Materials* 7.1 (Nov. 15, 2021). Number: 1 Publisher: Nature Publishing Group, pp. 1–8. ISSN: 2057-3960. DOI: 10.1038/s41524-021-00650-1. URL: https://www.nature.com/articles/s41524-021-00650-1 (visited on 01/05/2023).

[14] Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. "Equivariant message passing for the prediction of tensorial properties and molecular spectra". In: (June 7, 2021). arXiv: 2102.03150[physics]. URL: http://arxiv.org/abs/2102.03150 (visited on 01/05/2023).

[15] Anthony Yu-Tung Wang et al. "Compositionally restricted attention-based network for materials property predictions". In: *npj Computational Materials* 7.1 (May 28, 2021). Number: 1 Publisher: Nature Publishing Group, pp. 1–10. ISSN: 2057-3960. DOI: 10.1038/s41524-021-00545-1. URL: https://www.nature.com/articles/s41524-021-00545-1 (visited on 01/17/2023).

[16] Hongwei Wang et al. *Chemical-Reaction-Aware Molecule Representation Learning*. Sept. 22, 2021. DOI: 10.48550/arXiv.2109.09888. arXiv: 2109.09888[physics,q-bio]. URL: http://arxiv.org/abs/2109.09888 (visited on 01/07/2023).

[17] Matteo Aldeghi and Connor W. Coley. "A graph representation of molecular ensembles for polymer property prediction". In: *Chemical Science* 13.35 (2022), pp. 10486–10498. ISSN: 2041-6520, 2041-6539. DOI: 10.1039/D2SC02839E. arXiv: 2205.08619[cond-mat]. URL: http://arxiv.org/abs/2205.08619 (visited on 01/07/2023).

[18] Simon Batzner et al. "E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials". In: *Nature Communications* 13.1 (May 4, 2022), p. 2453. ISSN: 2041-1723. DOI: 10.1038/s41467-022-29939-5. arXiv: 2101.03164[cond-mat,physics:physics]. URL: http://arxiv.org/abs/2101.03164 (visited on 01/05/2023).

[19] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. *GemNet: Universal Directional Graph Neural Networks for Molecules*. Apr. 5, 2022. arXiv: 2106.08903[physics,stat]. URL: http://arxiv.org/abs/2106.08903 (visited on 01/05/2023).

[20] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. *Directional Message Passing for Molecular Graphs*. Apr. 5, 2022. arXiv: 2003.03123[physics,stat]. URL: http://arxiv.org/abs/2003.03123 (visited on 01/05/2023).

[21] Johannes Gasteiger, Chandan Yeshwanth, and Stephan Günnemann. *Directional Message Passing on Molecular Graphs via Synthetic Coordinates*. Apr. 5, 2022. arXiv: 2111.04718[physics,q-bio]. URL: http://arxiv.org/abs/2111.04718 (visited on 01/05/2023).

[22] Jonathan Godwin et al. *Simple GNN Regularisation for 3D Molecular Property Prediction & Beyond*. Mar. 15, 2022. arXiv: 2106.07971[cs]. URL: http://arxiv.org/abs/2106.07971 (visited on 01/05/2023).

[23] Zhichun Guo et al. *Graph-based Molecular Representation Learning*. July 8, 2022. arXiv: 2207.04869[cs,q-bio]. URL: http://arxiv.org/abs/2207.04869 (visited on 01/05/2023).

[24] Yuquan Li et al. "An adaptive graph learning method for automated molecular interactions and properties predictions". In: *Nature Machine Intelligence* 4.7 (July 2022). Number: 7 Publisher: Nature Publishing Group, pp. 645–651. ISSN: 2522-5839. DOI: 10.1038/s42256-022-00501-8. URL: https://www.nature.com/articles/s42256-022-00501-8 (visited on 01/15/2023).

[25] Yi Liu et al. *Spherical Message Passing for 3D Graph Networks*. Nov. 24, 2022. arXiv: 2102.05013[cs]. URL: http://arxiv.org/abs/2102.05013 (visited on 01/05/2023).

[26] *Deep Learning for the Life Sciences [Book]*. ISBN: 9781492039839. URL: https://www.oreilly.com/library/view/deep-learning-for/9781492039822/ (visited on 01/17/2023).

[27] Daniele Grattarola and Cesare Alippi. "Graph Neural Networks in TensorFlow and Keras with Spektral". In: ().

[28] *RDKit*. URL: https://rdkit.org/ (visited on 01/08/2023).

[29] Gengmo Zhou et al. "Uni-Mol: A Universal 3D Molecular Representation Learning Framework". In: (), p. 23.