Image-Like Graph Representations for Improved Molecular Property Prediction

Toni Sagayaraj
Brown University
antony_sagayaraj@brown.edu

Carsten Eickhoff
Brown University
carsten@brown.edu

Abstract

Research into deep learning models for molecular property prediction has primarily focused on the development of better Graph Neural Network (GNN) architectures. Though new GNN variants continue to improve performance, their modifications share a common theme of alleviating problems intrinsic to their fundamental graph-to-graph nature. In this work, we examine these limitations and propose a new molecular representation that bypasses the need for GNNs entirely, dubbed CubeMol. Our fixed-dimensional stochastic representation, when paired with a transformer model, exceeds the performance of state-of-the-art GNN models and provides a path for scalability.

1 Introduction

The field of drug discovery has always been resource intensive due to the sheer number of candidates screened for any biological target. Even with modern methods of "rational" drug design and libraries of known bioactive compounds, the average development cost of a new drug hovers at around 2.8 billion dollars [1]. In order to find better drug candidates, *in-silico* molecular property prediction serves to vastly accelerate this process by improving the efficacy of screening. In recent years, deep learning approaches have completely overtaken traditional methods using molecular fingerprints, hand-engineered features, and machine-learning algorithms due to their ability to leverage large datasets for improved performance.

The application of deep neural networks to property prediction remains a nontrivial task due to the inherent graph structure of molecular data, which cannot be handled by traditional feedforward or convolutional architectures. For this reason, the vast majority of advancements in property prediction have been achieved with Graph Neural Networks (GNNs), network architectures that restructure information propagation along the connectivity of graphs. These molecular graphs are represented by adjacency matrices along with feature vectors for each node and edge.

The majority of GNNs fall into two categories: Message Passing Neural Networks (MPNNs) and Graph Convolutional Networks (GCNs). Though these GNN variants have been specialized to work with molecular graphs, their methods of information propagation and readout functions hamper their expressive power and performance. We find that the primary bottlenecks of GNN performance arise from their graph-to-graph architecture, since constraining networks to transform sets of node embeddings poses challenges to learning deep representations that capture global structure and information.

In this work, we propose CubeMol, a novel stochastic representation of molecular graphs that bypasses the restrictive nature of GNNs entirely. Our fixed-dimensional random walk representations not only allow for the application of pre-existing architectures like ConvNets and Transformers, but also implicitly encode multiple views of the molecule's structure. Additionally, the stochastic nature of the representation provides natural data augmentation to improve model generalization. Our model exceeds the performance of other state-of-the-art GNNs in multiple property prediction tasks.

2 Limitations of Current GNNs

2.1 Long Distance Information Propagation

Information exchange between two distant nodes has long been known to be difficult for GNNs to accomplish. In their recent work, Alon Yahav (2021) explain this phenomenon through architectural bottlenecks for message propagation, which leads to "over-squashing" [2]. This stems from the fact that the receptive field of each node grows with each step of message aggregation. Given that communication between distant nodes requires many propagation steps, messages from exponentially more nodes are compressed into the same fixed-size vector. This limits the effective transmission range to the local neighborhood of any node, similar to the bottleneck for long-range interactions in RNN seq2seq models. Because global information is often vital for molecular properties, this aspect of GNNs limits their expressivity and performance for such tasks.

2.2 Readout Functions And Expressivity

Without the readout function, GNNs are graph-to-graph networks. Most property prediction models operate on sets of node embeddings, iteratively transforming them based on their connectivity with other nodes as well as the information contained in their edges. As molecular properties are often graph-based classification or regression tasks, generating a single fixed-dimensional representation from node embeddings is necessary. As a consequence, the diversity of information that can be captured from node embeddings is fundamentally limited. In addition to the standard average pooling function used, many pooling variants have been developed to mitigate this issue. Though set2set pooling mitigates this information loss via an LSTM, its sequential nature prevents the simultaneous consideration of all node embeddings [3]. Given the known difficulty with communicating global information, further limiting the ability of nodes to assemble their local information could make a GNN more akin to boosting many shallow models rather than a single cohesive deep model. Given the bottlenecks associated with any pooling methods, a method of avoiding a many-to-one compression at all should further increase the expressive power of GNNs.

2.3 Higher-Order and Multi-Scale Structures

A network's ability to capture molecular substructures and multi-scale features must govern its performance on global properties that rely on their interactions. For example, the proximity of different functional groups or their interactions with aromatic and conjugated systems greatly alters overall chemical properties. However, the other limitations discussed and the simple message aggregation schemes of most GNNs limit the ability to represent these features. If higher-order structures can only be captured by multiple message passing rounds, over-squashing directly limits the ability of GNNs to identify them. Furthermore, the fact that all molecular information is encoded in node representations leaves no trivial method of storing multi-scale features. If standard GNNs utilized these features for property prediction, they would need to be computed and incorporated into all node representations simultaneously. Otherwise, handcrafted architectural changes like graph coarsening are needed to account for hierarchical structures [4].

3 Related Work

Leveraging Multi-Scale Information LanczosNet aims to better capture multi-scale information using a low-rank approximation of the graph Laplacian using the Lanczos algorithm. From this representation, matrix powers can be easily calculated to gather multi-scale information. These features are then transformed by learnable spectral filters in order to increase the overall model capacity

4 Methods

4.1 CubeMol Representation

Much like Cubist paintings seek to represent 3D forms by combining multiple perspectives, CubeMol combines many parallel random walks to produce a dense fixed-dimensional representation of a graph. Rather than a representation that explicitly indicates the connectivity of a graph, the random walks of CubeMol serve to provide redundant cues of both local and global structure. These walks branch out from one another and are arranged in a grid, forming an image.

After choosing the grid size, a random starting node is chosen for the top left corner. Then, branching random walks fill out the tiles in the pattern specified in Figure 1. Though there are many arbitrary choices for how the walks can fill the grid, we designed the pattern to capture multiple levels of locality. Because there are many branch points and locations
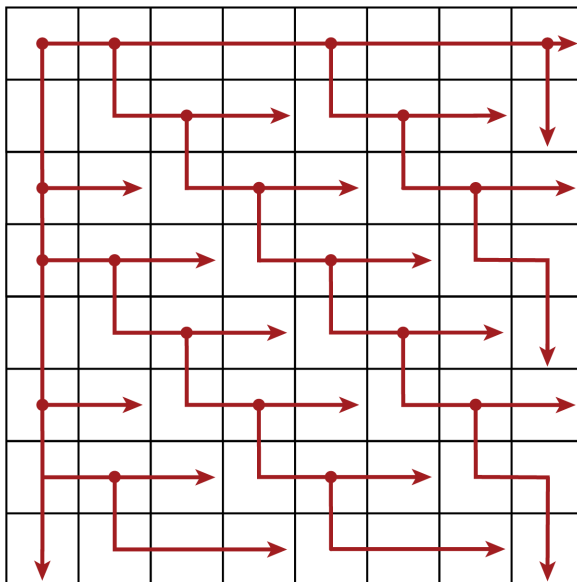
Figure 1: Figure 1: An 8x8 grid demonstrating the branching random-walk pattern of CubeMol. The red nodes indicate where the random walks branch, and the arrows indicate the direction.

where walks that have diverged remain close to each other, any given node in a tile is likely to have neighbors that are varied in distance on the original graph.

For each "pixel" representing a node in the graph, the channel dimensions contain both the node and edge features with the previous node in the walk. In molecular graphs, these correspond to the atom and bond features typically used by GNNs.

Given that the grid size is large enough for the random walks to traverse the entire graph, CubeMol should contain more than enough information to reconstruct the original graph. We avoid edge cases that could limit this information by preventing backtracking and branches that lead to the same neighbor if possible. The stochastic nature of this representation naturally provides data augmentation, as many distinct CubeMol tensors can be generated from a single graph.

4.2 Vision Transformers for Graphs

Recently, transformer models have proliferated the space of image classification thanks to the Vision Transformer (ViT) [5]. Dosovitskiy et al. found that pure transformers applied to sequences of image patches were able to achieve comparable results to state-of-the-art CNNs while requiring fewer computational resources. We chose to use a modified ViT model rather than a CNN because its self-attention is better suited to gather and compile information from the CubeMol representation.

Whereas the vanilla ViT segments the image into non-overlapping patches, we also use two types of additional patches shown in Figure 2. Edge patches cover information overlapping between two patches on the edges of the grid, and inner patches cover the intersections between four patches. These serve to facilitate learning the connection between these patches, since nodes on the edges between two patches have information that pertains to one another. We chose to add patches in this limited manner because of the $O(n^2d)$ computational complexity of attention on a sequence with n elements.

In addition to these patches, we maintained the standard of adding a learnable token to the sequence, whose final output would be used for the property prediction task. Before passing into the Transformer Encoder, all tokens were concatenated with a 1D learnable positional encoding.

Dataset	QM8	ESOL	CEP
Units	MAE in eV ($\times 10^{-3}$)	RMSE in log Mol/L	Percent
GGRNet [9]	13.94 ± 0.30	–	–
MPNN [10]	11.30 ± 0.31	0.47 ± 0.03	1.37 ± 0.09
PAGTN [11]	10.20 ± 0.30	0.55 ± 0.06	–
LanczosNet [4]	9.58 ± 0.14	–	–
Path MPNN [10]	8.70 ± 0.06	0.41 ± 0.02	1.23 ± 0.08
CubeMol+ 32x32	**8.41 ± 0**	**–**	**–**

Table 1 : Mean and std error on QM8, ESOL, and CEP for baselines and CubeMol models.

5 Experiments

5.1 Data

All datasets were split into a 80/10/10 ratio for training, validation, and testing. To compare our model against baselines, we chose

- QM8 [6]: Dataset of 21786 organic molecules with 8 or fewer heavy atoms (C, O, N, F). Prediction of 12 electronic spectra values calculated by DFT.
- ESOL [7]: Dataset of 1144 molecules for prediction of aqueous solubility
- CEP [8]: Dataset of 20000 organic molecules from The Harvard Clean Energy Project. Prediction of photovoltaic efficiency

5.2 Model Architecture & Training

For the CubeMol representation, we tested both 16x16 as well as 32x32 grids on QM8 to check the impact of grid size on performance. For the other two smaller datasets, we also tried a 64x64 size. For data augmentation, 50 sets of CubeMol representations were generated for each training dataset and grid size. On each epoch, a set is randomly chosen without replacement such that the model loops through all sets every 50 epochs.

For our transformer, we decided to use 6 attention layers, 16 attention heads, patch size of 4, 0.1 dropout on the transformer layers, and a final MLP layer with 1024 units. We tested the effectiveness of the additional patches, denoted by "+". All networks were trained for 1000 epochs, with a batch size of 64 and learning rate of 1×10^{-3} using the Adam optimizer. The validation set was evaluated at every epoch and the model checkpoint with the lowest validation loss was chosen for the final results on the test set.

6 Discussion

The results of our experiments are shown in Table 1. On QM8, our model was able to achieve SOTA by a significant margin. This final performance, however, is likely not close to the true power of the CubeMol transformer. We did not perform any hyperparameter tuning, and model continued to improve even close to the end of the 1000 training epochs. This leads us to believe that larger, better-tuned models with appropriate learning rate schedules can lead to significant improvements. In fact, this behavior was somewhat implied in Dosovitskiy et al. [5] with their experiments. These mainly focused on pre-training the ViT on large amounts of data before fine-tuning on smaller image-recognition datasets. When directly trained on these smaller datasets like CIFAR-100, however, they reached nowhere near the same performance.

The performances of the baselines represented side-by-side reflect the limitations of GNNs stated prior. Of all networks, GGRNet [9] performed the worst, ranking even below the MPNN [12] implemented by Flam-Shepherd et al. [10]. Though GGRNet implements a more complex MPNN, its readout function simply averages the node vectors and passes the result through an MLP for the result. The simpler MPNN uses the set2set readout instead, which is what likely lead to the better performance.

The Path Augmented Graph Transformer (PAGTN) [11] is arguably the best direct comparison to our CubeMol model. The PAGTN uses a transformer to operate on graphs like an MPNN, with the addition of path features and attention to all nodes simultaneously. The CubeMol Transformer can be viewed as reversing this construction, by explicitly laying out many paths across the molecule and then using a transformer to incorporate the information.

The Path MPNN [10] is arguably the best direct comparison to our CubeMol model. Though it does not use a transformer, the use of graph attention to aggregate messages across high order paths makes it analogous to the CubeMol transformer. The CubeMol Transformer can be viewed as reversing this construction, by explicitly laying out many paths across the molecule and then using a transformer to incorporate the information. Without the limitation of needing to store multi-scale information within the nodes of the graphs, the CubeMol likely surpasses the performance of the Path MPNN due to a larger model capacity and expressive power.

Though the Path Augmented Graph Transformer (PAGTN) [11] uses a transformer to operate on graphs like an MPNN, its performance is once again limited by the readout function. Even with global attention to all nodes simultaneously, it appears that the sum operator used to aggregate hidden representations limits its power compared to the Path MPNN, which uses the set2set model.

LanzcosNet [4] appears to be able to successfully utilize multi-scale information from the graph Laplacian and avoid limited model capacity, outperforming both the MPNN and PAGTN. However, the inaccuracies from the low-rank approximation provided by the Lanczos algorithm likely limits what the network can learn from the representation.

7 Conclusion and Future Works

In this paper, we identified limitations to GNNs and proposed the CubeMol representation to avoid the use of GNNs entirely. The construction of CubeMol as fixed-dimensional representation composed of random walks directly tackles the issues faced by GNNs, which pertain to capturing global information and multi-scale structures. When combined with a patch-based transformer, the performance of CubeMol surpasses state-of-the-art GNNs on QM8.

The concepts introduced in this work can be further expanded to greatly improve the performance of deep learning models on graph data. Since the CubeMol representation is a tensor like any image, the entire suite of powerful image-recognition models can now be applied to molecular property prediction. These include all of the scaling benefits, improvements to the efficiency of self-attention, and architectures for deeper transformer models.

In hindsight, our particular CubeMol grid representation can be deconstructed to be much simpler and robust. Since the grid is eventually broken down into a sequence of patches for the transformer, these can be replaced by a sequence of random walks centered around each nodes. An algorithmic manner of effectively covering subgraphs with redundant copies around each node would both reduce the number of patches needed per graph and allow the transformer to dynamically scale to larger graphs.

References

[1] Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 2020.

[2] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2021.

[3] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2016.

[4] Renjie Liao, Zhizhen Zhao, Raquel Urtasun, and Richard S. Zemel. Lanczosnet: Multi-scale deep graph convolutional networks. *arXiv preprint arXiv:1901.01484*, 2019.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012. PMID: 23088335.

[7] John S. Delaney. Esol: estimating aqueous solubility directly from molecular structure. *Journal of Chemical Information and Computer Sciences*, 44(3):1000–1005, 2004. PMID: 15154768.

[8] Johannes Hachmann, Roberto Olivares-Amaya, Sule Atahan-Evrenk, Carlos Amador-Bedolla, Roel S. Sánchez-Carrera, Aryeh Gold-Parker, Leslie Vogt, Anna M. Brockway, and Alán Aspuru-Guzik. The harvard clean energy project: Large-scale computational screening and design of organic photovoltaics on the world community grid. *The Journal of Physical Chemistry Letters*, 2(17):2241–2251, 2011.

[9] Hiroyuki Shindo and Yuji Matsumoto. Gated graph recursive neural networks for molecular property prediction. *arXiv preprint arXiv:1909.00259*, 2019.

[10] Daniel Flam-Shepherd, Tony Wu, Pascal Friederich, and Alan Aspuru-Guzik. Neural message passing on high order paths. *arXiv preprint arXiv:2002.10413*, 2020.

[11] Benson Chen, Regina Barzilay, and Tommi Jaakkola. Path-augmented graph transformer network. *arXiv preprint arXiv:1905.12712*, 2019.

[12] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017.