Multiresolution Graph Transformers and Wavelet Positional Encoding for Learning Hierarchical Structures

Nhat Khang Ngo [*1] Truong Son Hy [*2] Risi Kondor [3]

Abstract

Contemporary graph learning algorithms are not well-defined for large molecules since they do not consider the hierarchical interactions among the atoms, which are essential to determine the molecular properties of macromolecules. In this work, we propose Multiresolution Graph Transformers (MGT), the first graph transformer architecture that can learn to represent large molecules at multiple scales. MGT can learn to produce representations for the atoms and group them into meaningful functional groups or repeating units. We also introduce Wavelet Positional Encoding (WavePE), a new positional encoding method that can guarantee localization in both spectral and spatial domains. Our proposed model achieves competitive results on two macromolecule datasets consisting of polymers and peptides, and one drug-like molecule dataset. Importantly, our model outperforms other state-of-the-art methods and achieves chemical accuracy in estimating molecular properties (e.g., GAP, HOMO and LUMO) calculated by Density Functional Theory (DFT) in the polymers dataset. Furthermore, the visualizations, including clustering results on macromolecules and low-dimensional spaces of their representations, demonstrate the capability of our methodology in learning to represent long-range and hierarchical structures. Our PyTorch implementation is publicly available at https://github.com/HySonLab/Multires-Graph-Transformer.

1. Introduction

Macromolecules are long-range and hierarchical structures as they consist of many substructures. While small molecules in existing datasets (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014; Sterling & Irwin, 2015) comprise less than 50 atoms connected by simple rings and bonds, this number in a macromolecule can be dozens or even hundreds. Substructures such as repeating units and functional groups are intrinsic parts of macromolecules; they present unique chemical reactions regardless of other compositions in the same molecules (Jerry, 1992). Therefore, studying the multiresolution characteristic of large molecules is imperative to gain comprehensive knowledge about real-life materials like polymers or proteins (Schmid, 2022). In recent years, several works (Anand et al., 2022; Gaul & Cuesta-Lopez, 2022; Depta et al., 2022) have been proposed to apply machine learning algorithms to learn macromolecules at multiple scales. These approaches, however, rely on thorough feature selection and extraction, which are inefficient when learning from large databases of multicomponent materials (Anand et al., 2022).

Message passing is a prevailing paradigm for designing neural networks that operate on graph-structured data. Previous studies (Gilmer et al., 2017; Kipf & Welling, 2016; Veličković et al., 2018; Corso et al., 2020; Xu et al., 2019b) have proposed different strategies to perform message passing on graphs and achieved remarkable results across various domains. However, message-passing-dominated graph neural networks (GNNs) possess several limitations, such as limited expressiveness capability (Morris et al., 2019; Xu et al., 2019b), over-smoothing (Chen et al., 2020; Li et al., 2018; Oono & Suzuki, 2020), over-squashing (Alon & Yahav, 2021) issues. These two shortcomings hinder GNNs from making good predictions on long-range and hierarchically structured data. Furthermore, the molecular properties of large molecules are formed not only by interactions among atoms within neighborhoods but also by distant atoms. Therefore, local information is not sufficient to model macromolecules.

Transformers are classes of deep learning models that leverage self-attention mechanisms to handle long-range dependencies in various data domains, such as natural language processing (Vaswani et al., 2017; Devlin et al., 2019) or computer vision (Dosovitskiy et al., 2021; Liu et al., 2021). In graph domains, Transformer-like architectures (Kreuzer

[*]Equal contribution [1]FPT Software AI Center, Hanoi, Vietnam [2]Halıcıoğlu Data Science Institute, University of California San Diego, La Jolla, USA [3]Department of Computer Science, University of Chicago. Correspondence to: Truong Son Hy <tshy@ucsd.edu>.

The 2023 ICML Workshop on Computational Biology. Baltimore, Maryland, USA, 2023. Copyright 2023 by the author(s).

et al., 2021; Dwivedi & Bresson, 2020a; Rampášek et al., 2022) have proved their effectiveness in learning node representations as they can overcome the over-smoothing and over-squashing issues by directly measuring the pairwise relationships between the nodes. Node positional representations can be derived based on spectral (Dwivedi & Bresson, 2020b; Dwivedi et al., 2022a) or spatial (You et al., 2019; Li et al., 2020) domains. Most existing spectral-based methods decompose the graph Laplacian into sets of eigenvectors and eigenvalues. However, these eigenvectors have sign ambiguity and are unstable due to eigenvalue multiplicities. On the other hand, spatial-based approaches compute the shortest distances among the nodes; however, these encoding methods do not consider the structural similarity between nodes and their neighborhoods (Chen et al., 2022).

Our contributions in this are two-fold:

- We design Multiresolution Graph Transformer (MGT) and Wavelet Positional Encoding (WavePE) to operate on macromolecules at multiple scales.
- We show the effectiveness of our methodology by reporting its superior performance on three molecular property prediction benchmarks. These datasets contain macromolecules, i.e. peptides and polymers, that are highly hierarchical and consist of up to hundreds of atoms. Our model achieves important chemical accuracy in DFT approximation for the polymers dataset.

2. Wavelet Positional Encoding

2.1. Spectral Graph Wavelets

Let $\mathcal{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of an undirected graph $G = (\mathcal{V}, \mathcal{E})$. The normalized graph Laplacian is defined as $\mathcal{L} = \mathcal{I}_n - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$, where $\mathcal{I}_n$ is the identity matrix and $\mathcal{D}$ is the diagonal matrix of node degrees. $\mathcal{L}$ can be decomposed into a complete set of orthonormal eigenvectors $U = (u_1, u_2, ..., u_n)$ associated with real and non-negative eigenvalues $\{\lambda\}_1^n$. While graph Fourier transform uses U as a set of bases to project the graph signal from the vertex domain to the spectral domain, graph wavelet transform constructs a set of spectral graph wavelets as bases for this projection via:

$$\psi_s = U \Sigma_s U^T$$

where $\Sigma_s = \text{diag}(g(s\lambda_1), g(s\lambda_2), ..., g(s\lambda_n))$ is a scaling matrix of eigenvalues, $\psi_s = (\psi_{s1}, \psi_{s2}, ..., \psi_{sn})$ and each wavelet ψ_{si} indicates how a signal diffuses away from node i at scale s; we choose $g(s\lambda) = e^{-s\lambda}$ as a heat kernel (Donnat et al., 2018). Since a node's neighborhoods can be adjusted by varying the scaling parameter s (Xu et al., 2019a), using multiple sets of wavelets at different scales can provide comprehensive information on the graph's structure. It means that larger values of s_i correspond to larger neighborhoods surrounding a center node. Figure 2 illustrates how wavelets can be used to determine neighborhoods at different scales on a molecular graph. In this work, we leverage this property of graph wavelets to generate node positional representations that can capture the structural information of a center node on the graph at different resolutions. We employ k diffusion matrices $\{\psi_{s_i}\}_{i=1}^k$ in which each ψ_{s_i} has a size of $n \times n$, resulting in a tensor of graph wavelets $\mathcal{P} \in \mathbb{R}^{n \times n \times k}$. Additionally, WavePE is a generalized version of RWPE (Dwivedi et al., 2022a) as the random walk process can be regarded as a type of discrete diffusion. In the following section, we demonstrate the use of tensor contractions to generate a tensor of node positional representations $\mathbf{P} \in \mathbb{R}^{n \times k}$ from $\mathcal{P}$. In general, Fig.1a demonstrates our wavelet positional encoding method.

3. Multiresolution Graph Transformers

In this section, we present Multiresolution Graph Transformers (MGT), a neural network architecture for learning hierarchical structures. MGT uses Transformers to yield the representations of macromolecules at different resolutions. While previous work either neglects the hierarchical characteristics of large molecules or fails to model global interactions between distant atoms, our proposed approach can satisfy these two properties via multiresolution analysis.

Figs. 1b and 1c show an overview of our framework. MGT consists of three main components: an atom-level encoder, a module to extract substructures, and a substructure-level encoder. We use a graph transformer to generate the atomic embeddings. Then, substructures present in molecules are extracted by a learning-to-cluster algorithm. The molecular graph is coarsened into a set of substructures, and we use a pure Transformer encoder to learn their relations.

3.1. Atom-Level Encoder

To utilize the proposed wavelet positional encoding demonstrated in Section 2, we leverage the design of the graph transformer proposed in (Rampášek et al., 2022), which is a general, powerful, and scalable Graph Transformer (GraphGPS) for graph representation learning. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of a graph with n nodes and e edges; $\mathbf{X}^l$ and $\mathbf{E}^l$ are node and edge features at layer l-th, respectively. In addition, $\mathbf{X}^0 \in \mathbb{R}^{n \times d}$ and $\mathbf{E}^0 \in \mathbb{R}^{e \times d}$ are initial atom and bond features embedded in d-dimensional spaces created by two embedding layers. The wavelet positional vectors $\mathbf{p} \in \mathbb{R}^{n \times k}$ are fed to an encoder (e.g., a feed-forward neural network or a linear transformation), yielding a tensor of positional features $\mathbf{P} \in \mathbb{R}^{n \times d_p}$. We let $\mathbf{X}^0 := \text{concat}(\mathbf{X}^0, \mathbf{P})$ to produce new node features $\mathbf{X}^0 \in \mathbb{R}^{n \times (d + d_p)}$. From here, we define $d := d + d_p$,

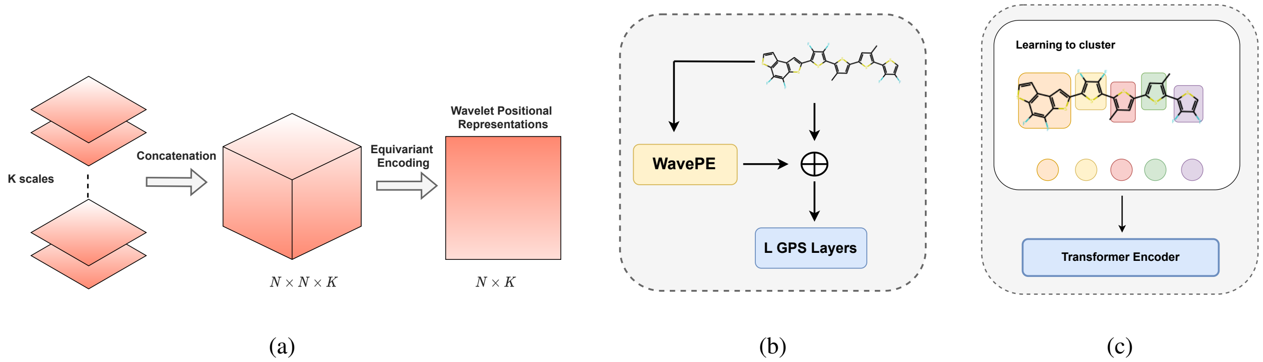
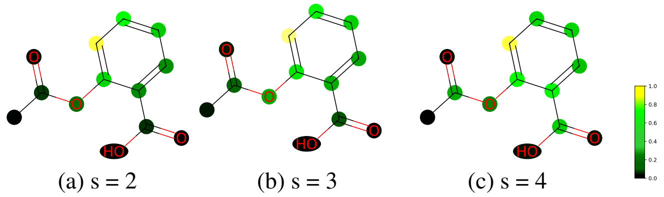


Figure 1. Overview of Wavelet Positional Encoding (WavePE) and Multiresolution Graph Transformer (MGT). a) k diffusion matrices of size $N \times N$ are stacked together to produce a wavelets tensor with size $N \times N \times K$ which are contracted by equivariant encoding methods to yield a tensor of positional representation $N \times k$. b) Atomic representations are derived by passing the molecular graph augmented with positional features through L GPS layers. c) A macromolecule is decomposed into several substructures in which the features are aggregated from the atom-level outputs, resulting in a set of substructures that are moved to a Transformer encoder.



(a) s = 2 (b) s = 3 (c) s = 4

Figure 2. Visualization of some of the wavelets with scaling parameters on the Aspirin $C_9H_8O_4$ molecular graph with 13 nodes (i.e. heavy atoms). The center node is colored yellow. The colors varying from bright to dark illustrate the diffusion rate from the center node to the others, i.e. nodes that are closer to the center node have brighter colors. Low-scale wavelets are highly localized, whereas the high-scale wavelets can spread out more nodes on the molecular graphs

and for convenience, the output dimensions of all layers are equal to d.

Each layer of GraphGPS uses a message-passing neural network (MPNN^l) to exchange information (i.e., messages) within the neighborhood and a self-attention layer (SA^l) described in Eq. (18) to compute global interactions among distant nodes:

$$\mathbf{X}_L^{l+1}, \mathbf{E}^{l+1} = \text{MPNN}^l(\mathbf{X}^l, \mathbf{E}^l, \mathbf{A}) \quad (1)$$

$$\mathbf{X}_G^{l+1} = \text{SA}^l(\mathbf{X}^l) \quad (2)$$

$$\mathbf{X}^{l+1} = \text{FFN}^l(\mathbf{X}_L^{l+1} + \mathbf{X}_G^{l+1}) \quad (3)$$

where $\mathbf{X}_L^{l+1}$ and $\mathbf{X}_G^{l+1}$ are node local and global representations; they are unified into $\mathbf{X}^{l+1}$ via Eq. 3. Popular techniques such as Dropout (Srivastava et al., 2014) and normalization (Ioffe & Szegedy, 2015; Ba et al., 2016) are omitted for the sake of clarity. By feeding the molecular graph through L layers, we attain two tensors $\mathbf{X}_a := \mathbf{X}^L$ and $\mathbf{E}_a := \mathbf{E}^L$ indicating the node and edge embeddings, respectively.

3.2. Learning to Cluster

In this work, we use a message-passing neural network augmented with differentiable pooling layers (Ying et al., 2018; Hy & Kondor, 2023) to cluster the atoms into substructures automatically:

$$\mathbf{Z} = \text{MPNN}_\text{e}(\mathbf{X}_a, \mathbf{E}_a, \mathbf{A}) \quad (4)$$

$$\mathbf{S} = \text{Softmax}(\text{MPNN}_\text{c}(\mathbf{X}_a, \mathbf{E}_a, \mathbf{A})) \quad (5)$$

where MPNN_e and MPNN_c are two-layer message-passing networks that learn to generate node embeddings ($\mathbf{Z} \in \mathbb{R}^{n \times d}$) and a clustering matrix ($\mathbf{S} \in \mathbb{R}^{n \times C}$), respectively; C denotes the number of substructures in molecules. A tensor of features $\mathbf{X}_s \in \mathbb{R}^{C \times d}$ for the substructures is computed:

$$\mathbf{X}_s = \mathbf{S}^T\mathbf{Z} \quad (6)$$

This learning-to-cluster module is placed after the atom-level encoder. Intuitively, atom nodes updated with both local and global information should be classified into accurate substructures.

3.3. Substructure-level Encoder

Given a set of substructures $\mathcal{V}_s$ with a tensor of features $\mathbf{X}_s \in \mathbb{R}^{C \times d}$, we forward $\mathbf{X}_s$ to L conventional Transformer encoder layers (Vaswani et al., 2017) to capture their pairwise semantics:

$$\mathbf{H}_1^{l+1} = \text{Norm}(\text{SA}^l(\mathbf{H}^l) + \mathbf{H}^l) \quad (7)$$

$$\mathbf{H}^{l+1} = \text{Norm}(\text{FFN}(\mathbf{H}_1^{l+1}) + \mathbf{H}_1^{l+1}) \quad (8)$$

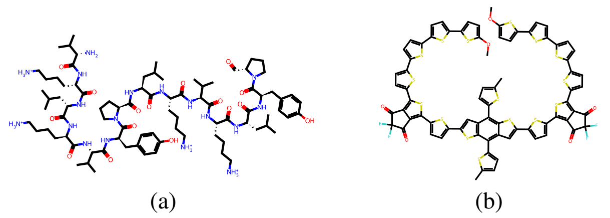


Figure 3. Examples of two macromolecules. a) An example of a peptide that consists of many functional groups. b) An example of a polymer that consists of many repeating units

where SA refers to (multi-head) self-attention described in Eq. (18), and $\mathbf{H}^0$ is equal to $\mathbf{X}_s$. Additionally, we add a long-range skip connection to alleviate gradient vanishing as:

$$\mathbf{H}_s = \text{FFN}(\text{concat}(\mathbf{H}^0, \mathbf{H}^L)) \tag{9}$$

$\mathbf{H}_s \in \mathbb{R}^{C \times d}$ is the output indicating the representations for the substructures. Finally, we aggregate all C vectors $h_s \in \mathbf{H}_s$ to result in a unique representation $z \in \mathbb{R}^d$ for the molecules (refer to Section C.2), before feeding it to a feed-forward network to compute the final output $y \in \mathbb{R}^c$ for property prediction:

$$z = \zeta(\{h_s\}_{s=1}^{C}) \tag{10}$$

$$\hat{y} = \text{FFN}(z) \tag{11}$$

4. Experiments

We empirically validate our proposed approach in two types of macromolecules including peptides and polymers. Figure 3 illustrates two examples of macromolecules in the datasets. Our PyTorch implementation is publicly available at `https://github.com/HySonLab/Multires-Graph-Transformer`.

4.1. Polymer Property Prediction

Experimental Setup We use a polymer dataset proposed in (St. John et al., 2019). Each polymer is associated with three types of density functional theory (DFT) (Hohenberg & Kohn, 1964) properties including the first excitation energy of the monomer calculated with time-dependent DFT (GAP), the energy of the highest occupied molecular orbital for the monomer (HOMO), and the lowest unoccupied molecular orbital of the monomer (LUMO). The dataset is split into train/validation/test subsets with a ratio of 8:1:1, respectively. For training, we normalize all learning targets with a mean of 0 and a standard deviation of 1.

Results As shown in Table 1, our MGT models achieve the lowest MAE scores across three properties. In addition, WavePE can attain comparable results with LapPE and RWPE for this task. We observe that the vanilla Transformer has the poorest performance. This demonstrates that computing global information without the awareness of locality is not sufficient for macromolecular modeling. As described in Section 3, MGT is an extended version of GPS. In particular, a learning-to-cluster module and a substructure-level Transformer encoder are extensions to GPS. The better performance of MGT, as a result, indicates that our methodology in modeling hierarchical structures is appropriate and reasonable.

There are two important benchmark error levels: (1) “DFT error”, the estimated average error of the DFT approximation to nature; and (2) “chemical accuracy”, the target error that has been established by the chemistry community. Estimates of DFT error and chemical accuracy are provided by (Faber et al., 2017). Our model is the only one to achieve the chemical accuracy for all the three molecular properties.

4.2. Peptides Property Prediction

Experimental Setup We run experiments on two real-world datasets including (1) Peptides-struct and (2) Peptides-func (Dwivedi et al., 2022b). The two datasets are multi-label graph classification problems and share the same peptide molecular graphs, but with different tasks. While the former consists of 10 classes based on peptides function, the latter is used to predict 11 aggregated 3D properties of peptides at the graph level. For a fair comparison, we follow the experimental and evaluation setting of (Dwivedi et al., 2022b) with the same train/test split ratio. We use mean absolute error (MAE) and average precision (AP) to evaluate the method’s performance for Peptides-struct and Peptides-func, respectively.

Results Table 2 shows that our proposed MGT + WavePE achieves the best performances in two peptide prediction tasks. In addition to WavePE, MGT + RWPE also attains the second-best performances. The superiority of WavePE to RWPE can be explained as mentioned in Section 2 that WavePE is a generalized version of RWPE. In particular, our proposed MGT outperforms all the baselines in the Petides-func task by a large margin and decreases the MAE score to less than 0.25 in the Petides-struct task.

5. Conclusion

In this paper, we introduce a novel architecture, Multiresolution Graph Transformer (MGT), and a new atomic positional encoding named WavePE that is based on multiresolution analysis and wavelet theory. We have shown competitive experimental results on two macromolecule datasets of polymers and peptides. We believe our model and implementation will certainly advance the field of DFT approximation for large-scale molecular structures.

References

Alon, U. and Yahav, E. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=i80OPhOCVH2.

Anand, G., Ghosh, S., Zhang, L., Anupam, A., Freeman, C. L., Ortner, C., Eisenbach, M., and Kermode, J. R. Exploiting machine learning in multiscale modelling of materials. *Journal of The Institution of Engineers (India): Series D*, Nov 2022. ISSN 2250-2130. doi: 10.1007/s40033-022-00424-z. URL https://doi.org/10.1007/s40033-022-00424-z.

Ba, L. J., Kiros, J. R., and Hinton, G. E. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., and Sun, X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3438–3445, Apr. 2020. doi: 10.1609/aaai.v34i04.5747. URL https://ojs.aaai.org/index.php/AAAI/article/view/5747.

Chen, D., O'Bray, L., and Borgwardt, K. Structure-aware transformer for graph representation learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, 2022.

Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Veličković, P. Principal neighbourhood aggregation for graph nets. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13260–13271. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/99cad265a1768cc2dd013f0e740300ae-Paper.pdf.

Depta, P. N., Dosta, M., Wenzel, W., Kozlowska, M., and Heinrich, S. Hierarchical coarse-grained strategy for macromolecular self-assembly: Application to hepatitis b virus-like particles. *International Journal of Molecular Sciences*, 23(23), 2022. ISSN 1422-0067. doi: 10.3390/ijms232314699. URL https://www.mdpi.com/1422-0067/23/23/14699.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Donnat, C., Zitnik, M., Hallac, D., and Leskovec, J. Learning structural node embeddings via diffusion wavelets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 1320–1329, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220025. URL https://doi.org/10.1145/3219819.3220025.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Dwivedi, V. P. and Bresson, X. A generalization of transformer networks to graphs. *CoRR*, abs/2012.09699, 2020a. URL https://arxiv.org/abs/2012.09699.

Dwivedi, V. P. and Bresson, X. A generalization of transformer networks to graphs. *CoRR*, abs/2012.09699, 2020b. URL https://arxiv.org/abs/2012.09699.

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=wTTjnvGphYj.

Dwivedi, V. P., Rampasek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. Long range graph benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022b. URL https://openreview.net/forum?id=in7XC5RcjEn.

Faber, F. A., Hutchison, L., Huang, B., Gilmer, J., Schoenholz, S. S., Dahl, G. E., Vinyals, O., Kearnes, S., Riley, P. F., and von Lilienfeld, O. A. Machine learning prediction errors better than dft accuracy. *arXiv preprint arXiv:1702.05532*, 2017.

Gaul, C. and Cuesta-Lopez, S. Machine learning for screening large organic molecules. *arXiv preprint arXiv:2211.15415*, 2022.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

Hohenberg, P. and Kohn, W. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964. doi: 10.1103/PhysRev.136.B864. URL https://link.aps.org/doi/10.1103/PhysRev.136.B864.

Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., and Leskovec, J. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

Hy, T. S. and Kondor, R. Multiresolution equivariant graph variational autoencoder. *Machine Learning: Science and Technology*, 4(1):015031, mar 2023. doi: 10.1088/2632-2153/acc0d8. URL https://dx.doi.org/10.1088/2632-2153/acc0d8.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

Jerry, M. Advanced organic chemistry: reactions, mechanisms and structure, 1992.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Kreuzer, D., Beaini, D., Hamilton, W., Létourneau, V., and Tossou, P. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Li, P., Wang, Y., Wang, H., and Leskovec, J. Distance encoding: Design provably more powerful neural networks for graph representation learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33014602. URL https://doi.org/10.1609/aaai.v33i01.33014602.

Oono, K. and Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1ldO2EFPr.

Ramakrishnan, R., Dral, P., Rupp, M., and von Lilienfeld, A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 08 2014. doi: 10.1038/sdata.2014.22.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a General, Powerful, Scalable Graph Transformer. *arXiv:2205.12454*, 2022.

Ruddigkeit, L., Deursen, R., Blum, L., and Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52, 10 2012. doi: 10.1021/ci300415d.

Schmid, F. Understanding and modeling polymers: The challenge of multiple scales. *ACS Polymers Au*, 0(0):null, 2022. doi: 10.1021/acspolymersau.2c00049. URL https://doi.org/10.1021/acspolymersau.2c00049.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

St. John, P. C., Phillips, C., Kemper, T. W., Wilson, A. N., Guan, Y., Crowley, M. F., Nimlos, M. R., and Larsen, R. E. Message-passing neural networks for high-throughput polymer screening. *The Journal of chemical physics*, 150 (23):234111, 2019.

Sterling, T. and Irwin, J. Zinc 15 - ligand discovery for everyone. *Journal of chemical information and modeling*, 55, 10 2015. doi: 10.1021/acs.jcim.5b00559.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Xu, B., Shen, H., Cao, Q., Qiu, Y., and Cheng, X. Graph wavelet neural network. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=H1ewdiR5tQ.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL https://openreview.net/forum?id=ryGs6iA5Km.

Ying, R., You, J., Morris, C., Ren, X., Hamilton, W. L., and Leskovec, J. Hierarchical graph representation learning with differentiable pooling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 4805–4815, Red Hook, NY, USA, 2018. Curran Associates Inc.

You, J., Ying, R., and Leskovec, J. Position-aware graph neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7134–7143. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/you19b.html.

A. Additional Tables

Table 1. Experimental results on the polymer property prediction task. All the methods are trained in four different random seeds and evaluated by MAE ↓. Our methods are able to attain better performance across three DFT properties of polymers while having less number of parameters. All the properties are measured in eV.

Model	No. Params	Property		
		GAP	HOMO	LUMO
DFT error		1.2	2.0	2.6
Chemical accuracy		0.043	0.043	0.043
GCN	527k	0.1094 ± 0.0020	0.0648 ± 0.0005	0.0864 ± 0.0014
GCN + Virtual Node	557k	0.0589 ± 0.0004	0.0458 ± 0.0007	0.0482 ± 0.0010
GINE	527k	0.1018 ± 0.0026	0.0749 ± 0.0042	0.0764 ± 0.0028
GINE + Virtual Node	557k	0.0870 ± 0.0040	0.0565 ± 0.0050	0.0524 ± 0.0010
GPS	600k	0.0467 ± 0.0010	0.0322 ± 0.0020	0.0385 ± 0.0006
Transformer + LapPE	700k	0.2949 ± 0.0481	0.1200 ± 0.0206	0.1547 ± 0.0127
MGT + LapPE (ours)	499k	**0.0378** ± **0.0004**	**0.0270** ± **0.0010**	0.030 ± 0.0006
MGT + RWPE (ours)	499k	0.0384 ± 0.0015	0.0274 ± 0.0005	**0.0290** ± **0.0007**
MGT + WavePE (ours)	499k	0.0387 ± 0.0011	0.0283 ± 0.0004	0.0290 ± 0.0010

Table 2. Results on peptides property prediction

Model	No.Params	Peptides-struct	Peptides-func
		MAE ↓	AP ↑
GCN	508k	0.3496 ± 0.0013	0.5930 ± 0.0023
GINE	476k	0.3547 ± 0.0045	0.5498 ± 0.0079
GatedGCN	509k	0.3420 ± 0.0013	0.5864 ± 0.0077
GatedGCN + RWPE	506k	0.3357 ± 0.0006	0.6069 ± 0.0035
Transformer + LapPE	488k	0.2529 ± 0.0016	0.6326 ± 0.0126
SAN + LapPE	493k	0.2683 ± 0.0043	0.6384 ± 0.0121
SAN + RWPE	500k	0.2545 ± 0.0012	0.6562 ± 0.0075
MGT + LapPE (ours)	499k	0.2488 ± 0.0014	0.6728 ± 0.0152
MGT + RWPE (ours)	499k	0.2496 ± 0.0009	0.6709 ± 0.0083
MGT + WavePE (ours)	499k	**0.2453** ± **0.0025**	**0.6817** ± **0.0064**

B. Implementation Details

B.1. Multiresolution Graph Transformer

In this section, we elaborate on the architecture and hyperparameters used to train and evaluate our MGT to achieve the above numerical results. Table 3 show details of the hyperparameters used for MGT in all the experiments. In particular, we use the atom and bond encoder modules provided by OGB (Hu et al., 2021) to attribute the molecular graph. We use two GPS layers to compute the atom-level representations and two Transformer layers for calculating the substructure-level representations. For learning to cluster, we use a 2-layer message-passing network to compute $\mathbf{Z}$ and $\mathbf{S}$ mentioned in Eq. (4)

(5) as follows:

$$\mathbf{Z}_a^1, \mathbf{E}_a^1 = \text{GatedGCN}^1(\mathbf{X}_a, \mathbf{E}_a, \mathbf{A}) \tag{12}$$

$$\mathbf{Z}_a^1 = \text{Batchnorm}(\text{ReLU}(\mathbf{Z}_a^1)) \tag{13}$$

$$\mathbf{Z}_a^2, \mathbf{E}_a^2 = \text{GatedGCN}^2(\mathbf{Z}_a^1, \mathbf{E}_a^1, \mathbf{A}) \tag{14}$$

$$\mathbf{Z}_a^2 = \text{Batchnorm}(\text{ReLU}(\mathbf{Z}_a^2)) \tag{15}$$

$$\mathbf{Z} = \text{concat}(\mathbf{Z}_a^1, \mathbf{Z}_a^2) \tag{16}$$

$$\mathbf{Z} = \text{FFN}(\mathbf{Z}) \tag{17}$$

S is computed similarly with an auxiliary Softmax operation on the output to produce a probabilistic clustering matrix.

Table 3. The hyperparameters for MGT

Hyperparameters	Values
No. Epoch	200
Emb Dim	84
Batch size	128
Learning rate	0.001
Dropout	0.25
Attention Dropout	0.5
Diffusion Step (K)	[1, 2, 3, 4, 5]
No. Head	4
Activation	ReLU
Normalization	Batchnorm
No. Cluster	10
λ_1	0.001
λ_2	0.001

B.2. Baselines used in Polymer Property Prediction

Table 4 shows the implementation of the baselines we used in the polymer experiments. All the models are designed to have approximately 500 to 700k learnable parameters. For fair comparisons, all the models are trained in 50 epochs with a learning rate of 0.001 and batch size of 128.

Table 4. The detailed settings of baselines for polymer property prediction

Model	No. Layer	Embed Dim	No. Params
GCN	5	300	527k
GCN + Virtual Node	5	156	557k
GINE	5	156	527k
GINE + Virtual Node	5	120	557k
GPS	3	120	600k
Transformer + LapPE	6	120	700k

C. Background

C.1. Notation

A molecule can be represented as an undirected graph in which nodes are the atoms and edges are the valency bonds between them. In paticular, we refer to a molecular graph as $G = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X}, \mathbf{P}, \mathcal{V}_s)$, where G is an undirected graph having $\mathcal{V}$

($|\mathcal{V}| = n$) and $\mathcal{E}$ as sets of nodes and edges respectively; also, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the graph's adjacency matrix. When a graph is attributed, we augment G with a set of node feature vectors $\mathcal{X} = \{x_1, ..., x_n\}, x_i \in \mathbb{R}^d$ and a set of node positional vectors $\mathcal{P} = \{p_1, ..., p_n\}, p_i \in \mathbb{R}^p$. These two types of attributes are stored in $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{P} \in \mathbb{R}^{n \times p}$ correspondingly. In addition to the atom-level representation of G, $\mathcal{V}_s = \{v_{s_1}, ..., v_{s_k}\}$ denotes the substructure set in which $v_{s_i} \subset \mathcal{V}$, i.e. v_{s_i} is a subset of atoms of the molecule.

C.2. Hierachical Learning on Molecules

Molecular property prediction is regarded as a graph-level learning task. We need to aggregate node embeddings into graph-level vectors which are then fed to a classifier to make predictions on graphs. Specifically, a function $f : \mathcal{V} \rightarrow \mathcal{Z}$ that maps the atom $u \in \mathcal{V}$ to a d_o-dimensional vector $z_u \in \mathcal{Z} \subset \mathbb{R}^{d_o}$ should learn to produce atom-level representations. Most existing graph neural networks compute the vector $z = \zeta(\{f(u)|u \in \mathcal{V}\})$ that indicates a representation for the entire molecular graph, where ζ can be sum, mean, max, or more sophisticated operators. For hierarchical learning, substructure-level representations can be derived in addition to atom-level representations by aggregating node representations in the same substructures as $z_s = \zeta(\{f(u)|u \in v_s \wedge v_s \in \mathcal{V}_s\})$. Instead of atom vectors, we aggregate the substructure vectors to represent the entire graph, i.e. $z = \zeta(\{z_s|z_s \in \mathcal{V}_s\})$. Finally, a classifier g given z as inputs is trained to predict the molecular properties.

C.3. Transformers on Graphs

While GNNs learn node embeddings by leveraging the graph structure via local message-passing mechanisms, Transformers disregard localities and directly infer the relations between pairs of nodes using only node attributes. In other words, the node connectivity is not utilized in pure transformer-like architectures (Vaswani et al., 2017), reducing the graph conditions to a set learning problem. Given a tensor of node features $\mathbf{X} \in \mathbb{R}^{n \times d}$, Transformers compute three matrices including query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) via three linear transformations $\mathbf{Q} = \mathbf{X}\mathbf{W}_q^T$, $\mathbf{K} = \mathbf{X}\mathbf{W}_k^T$, and $\mathbf{V} = \mathbf{X}\mathbf{W}_v^T$. A self-attention tensor ($\mathbf{H}$) can be computed as follows:

$$\mathbf{H} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_o}})\mathbf{V} \tag{18}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$ are learnable parameters in $\mathbb{R}^{d_o \times d}$, resulting in $\mathbf{H} \in \mathbb{R}^{n \times d_o}$. Furthermore, each $\mathbf{H}$ in Eq. 18 denotes an attention head. To improve effectiveness, multiple $\{\mathbf{H}\}_{i=1}^h$ are computed, which is known as multi-head attention. All of the attention heads are concatenated to form a final tensor: $\mathbf{H}_o = \text{concat}(\mathbf{H}_1, ..., \mathbf{H}_h)$, where h is the number of attention heads. Finally, the output $\mathbf{X}'$, i.e. new node representations, can be computed by feeding $\mathbf{H}_o$ into a feed-forward neural network (FFN), i.e. $\mathbf{X}' = \text{FFN}(\mathbf{H}_o)$. It is easy to see that Transformers operating on inputs without positional encoding are permutation invariant.

Positional Encoding As pure Transformer encoders only model sets of nodes without being cognizant of the graph structures, positional or structural information between nodes and their neighborhoods should be incorporated into node features. In particular, node positional representations can be added or concatenated with node features, resulting in comprehensive inputs for Transformer-like architectures operating on graph-structured data.