

## 1 A Discussion on LRGB

2 One of the most widely used benchmark for assessing the long-range propagation capabilities of GNNs  
 3 is the Long Range Graph Benchmark (LRGB) [7]. The benchmark proposes five tasks: two molecular  
 4 property prediction tasks (Peptides-func and Peptides-struct), one molecular bond prediction task  
 5 (PCQM-Contact), and two computer vision tasks (PascalVOC-SP and COCO-SP). However, despite  
 6 initial rapid improvements, performance on LRGB has plateaued. Since its introduction in 2022, there  
 7 has been a noticeable deceleration in progress. Considering a set of 30 models on the Peptides-func  
 8 task, we observe a performance improvement by 6.5% in the first year, but only by 1.3% in the  
 9 second, and no significant gain in the third year [14, 19, 9, 16, 7, 32, 11, 20, 31, 2, 33, 8, 6, 21, 3, 12,  
 10 18, 30, 27, 22, 10, 4]. A similar trend exists for the other benchmark tasks as well.

11 Furthermore, a recent analysis on LRGB [1], as well as the benchmark’s sensitivity to hyperparameter  
 12 tuning [32], raises additional concerns about the long-range nature of its tasks. The analysis reveals  
 13 that only a subset of tasks genuinely require longer interactions, while the peptides tasks are effectively  
 14 local. This highlights the need for more focused benchmarks that explicitly and systematically test  
 15 long-range propagation capabilities of GNNs.

## 16 B Additional Experimental Analysis

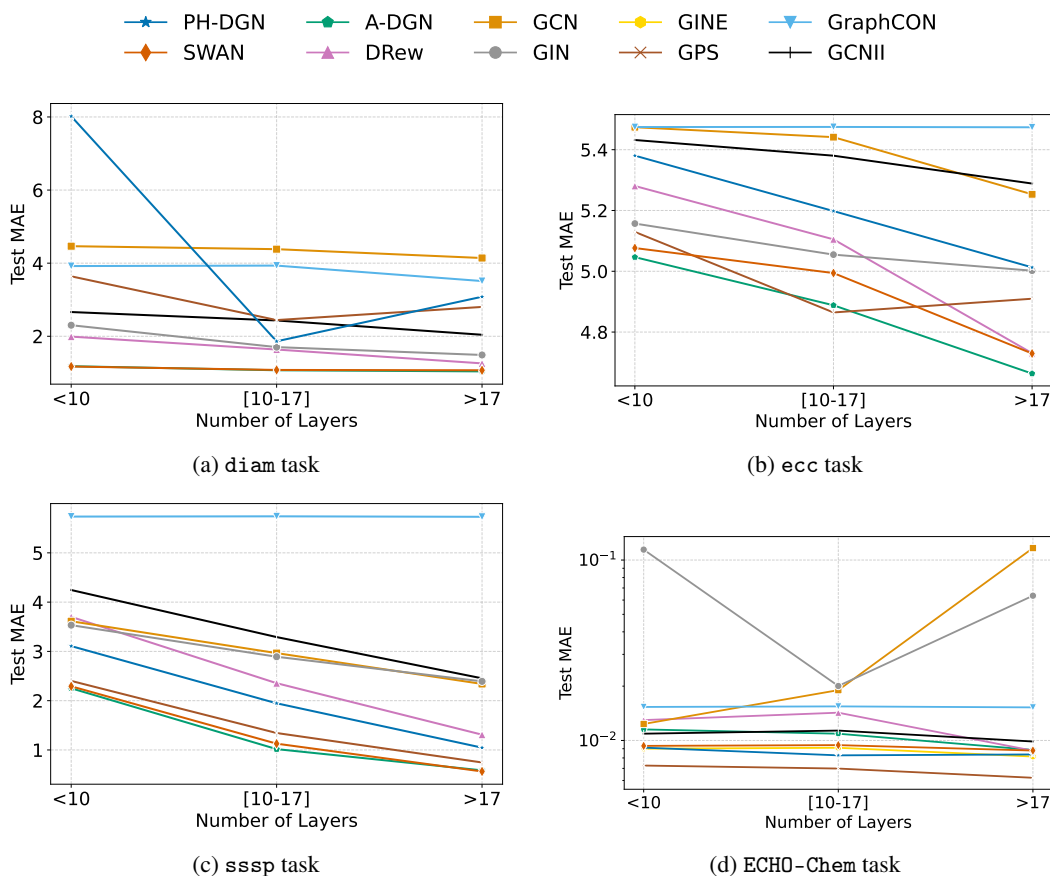


Figure 1: Test MAE at different numbers of GNN layers across tasks.

**Layer-wise Performance Analysis.** In Figure 1, we evaluate the impact of the radius of the explored neighborhood (i.e., the number of GNN layers) on test MAE across all tasks. We divide the results into three regimes: shallow ( $< 10$  layers), medium (10–17 layers), and deep ( $> 17$  layers). Therefore, in the shallow regime, GNNs perform short-range propagation; in the medium regime, they capture medium-range dependencies; and in the deep regime, they are able to model long-range interactions. A consistent pattern emerges across most tasks: deeper networks – especially those tailored for long-range propagation – tend to perform better, thus confirming the long-range nature of the proposed benchmarks. Specifically:

- On the `diam` task (Figure 1a), performance trends are model-dependent. Long-range models such as DRew and A-DGN remain stable or slightly improve, others like PH-DGN exhibit a large performance improvement moving from shallow to medium depth regime. This task, being graph-level and heavily reliant on global information by design, clearly benefits from increased depth and non-dissipative architectures which are able to perform many message-passing steps across multiple hops.
- For the `ecc` task (Figure 1b), we observe a consistent performance gain with increasing depth across nearly all models. Again, long-range architectures like A-DGN and SWAN, or the multi-hop GNN, DRew, show strong improvements in the deep regime, outperforming the others. This aligns with the intuition that eccentricity – being a node-level but globally-informed property – benefits from many message-passing layers to capture distant context, highlighting the strength of long-range architectures.
- In the case of `sssp` (Figure 1c) we again observe strong depth-related improvements, with the exception of GraphCON. Notably, SWAN, GPS, and DRew achieve large gains in the deep regime. Traditional models such as GCN and GIN or GraphCON exhibit plateau or degradation, revealing limited depth scalability.
- Finally, on the `ECH0-Chem` task (Figure 1d), the behavior differs. This task involves precise regression of atomic partial charges, where small errors matter. Most models show stable MAE across depths, except for GCN and GIN, which degrade significantly in the deep regime. Importantly, models with explicit long-range message-passing capabilities (A-DGN, SWAN, PH-DGN, GPS, and DRew) retain high accuracy even at  $> 17$  layers. This suggests their robustness in fine-grained, long-range molecular prediction tasks.

Overall, the observed patterns reveal a clear correlation between the number of message-passing layers and performance: models require many layers to perform well, confirming the long-range nature of these benchmarks. Remarkably, architectures explicitly designed to support many message-passing steps consistently outperform others, further confirming the long-range nature of our proposed benchmarks.

**Performance Across Graph Diameters.** Figure 2 reports test MAE across varying graph diameters for all tasks. This analysis highlights how different models handle increasingly long-range dependencies.

For the `diam` task (Figure 2a), most models show robust performance for small to moderate diameters, with a slight increase in MAE for very large diameters. Notably, GCN, GraphCON and GCNII architectures exhibit substantial degradation as diameter increases, suggesting poor scalability in capturing global structure on many message-passing steps. Again, non-dissipative architectures (i.e., A-DGN, PH-DGN, and SWAN), DRew and GPS remain consistently accurate across all graph diameters, demonstrating their capacity to generalize across different graph scales.

The `ecc` task (Figure 2b) reveals a characteristic U-shaped curve. Performance improves as diameter increases from small to moderate values, and deteriorates again for very large graphs. Here, all models follow a similar trend, although A-DGN and GPS tend to dominate in the optimal range.

In the `sssp` task (Figure 2c), increasing graph diameter consistently correlates with rising MAE. Model performance divides into three groups, with GraphCON exhibiting the worst performance both in terms of overall MAE and degradation with increasing diameter. GCN, GCNII, and GIN show similar values across the task and similar degradation trends. Finally, non-dissipative models, GPS Transformer and DRew, once again demonstrate remarkable and consistent performance even on difficult graphs. This trend reinforces the long-range nature of the task, where deeper or more expressive models are required to maintain strong performance.

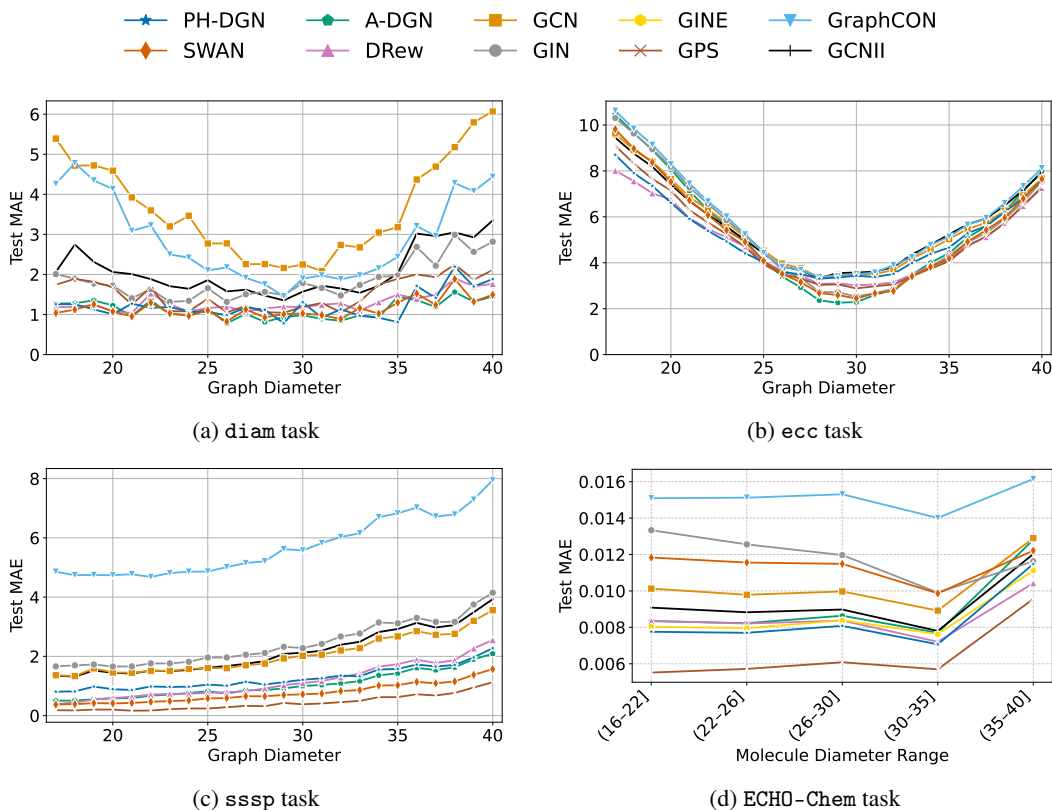


Figure 2: Test MAE at different graph diameters across synthetic and molecular tasks.

On the molecular ECHO-Chem task (Figure 2d), test MAE consistent across all ranges, but subtle trends emerge. Models like DRew and GPS show stability and even slight improvements for larger molecular graphs, while GCN and GIN degrade more noticeably, confirming their limited capacity to manage increasing molecular complexity. Interestingly, GINE performs substantially better than its counterpart GIN, suggesting that edge-level attributes play a crucial role in the ECHO-Chem regression task. Additionally, we note that all models exhibit a general performance drop when processing molecular graphs with a diameter greater than 35. We attribute this behavior to the original ChEMBL dataset’s distribution, which includes fewer graphs with diameters in the 35–40 range. This also impacts our ECHO-Chem dataset as illustrated in Figure 4. As a result, models have limited opportunity to learn effective representations for such large graphs, which likely contributes to the observed degradation.

Overall, this complementary diameter-wise analysis underlines the necessity for architectures capable of handling variable and large receptive fields. It also highlights that while shallow models may perform competitively on small graphs, their limitations become apparent in regimes requiring long-range reasoning.

## C Extended Results

In Table 1 we report additional result on synthetic benchmark. In particular the MAE, MSE and the training loss defined as  $\log_{10}(\text{MSE})$ .

Table 1: Test performance (mean  $\pm$  std) of different models across tasks. Lower is better. In bold the best model.

Model	Test Loss $\downarrow$	Test MSE $\downarrow$	Test MAE $\downarrow$
diam			
A-DGN	$-2.531 \pm 0.010$	$4.818 \pm 0.108$	$1.151 \pm 0.038$
DRew	<b><math>-2.635 \pm 0.020</math></b>	<b><math>3.756 \pm 0.170</math></b>	$1.243 \pm 0.047$
GCNII	$-2.227 \pm 0.026$	$9.696 \pm 0.568$	$2.005 \pm 0.093$
GCN	$-1.848 \pm 0.051$	$22.872 \pm 2.766$	$3.832 \pm 0.262$
GIN	$-2.356 \pm 0.066$	$7.238 \pm 1.153$	$1.630 \pm 0.161$
GPS	$-2.192 \pm 0.025$	$10.454 \pm 0.610$	$2.160 \pm 0.098$
GraphCON	$-1.995 \pm 0.037$	$16.427 \pm 1.419$	$2.969 \pm 0.189$
PH-DGN	$-2.416 \pm 0.181$	$6.699 \pm 2.728$	$1.627 \pm 0.398$
SWAN	$-2.517 \pm 0.023$	$4.950 \pm 0.265$	<b><math>1.121 \pm 0.070</math></b>
ecc			
A-DGN	$-1.649 \pm 0.006$	$35.967 \pm 0.492$	$4.981 \pm 0.037$
DRew	<b><math>-1.696 \pm 0.002</math></b>	<b><math>32.247 \pm 0.148</math></b>	<b><math>4.651 \pm 0.020</math></b>
GCNII	$-1.603 \pm 0.006$	$39.911 \pm 0.518$	$5.241 \pm 0.030$
GCN	$-1.606 \pm 0.005$	$39.706 \pm 0.460$	$5.233 \pm 0.034$
GIN	$-1.668 \pm 0.015$	$34.454 \pm 1.201$	$4.869 \pm 0.092$
GPS	$-1.682 \pm 0.003$	$33.346 \pm 0.226$	$4.758 \pm 0.021$
GraphCON	$-1.566 \pm 0.001$	$43.505 \pm 0.017$	$5.474 \pm 0.001$
PH-DGN	$-1.630 \pm 0.017$	$37.510 \pm 1.416$	$5.068 \pm 0.126$
SWAN	$-1.671 \pm 0.007$	$34.208 \pm 0.578$	$4.840 \pm 0.045$
sssp			
A-DGN	$-2.566 \pm 0.089$	$4.425 \pm 0.879$	$1.176 \pm 0.140$
DRew	$-2.386 \pm 0.001$	$6.589 \pm 0.015$	$1.279 \pm 0.011$
GCNII	$-2.213 \pm 0.177$	$10.369 \pm 3.575$	$2.128 \pm 0.429$
GCN	$-2.217 \pm 0.033$	$9.743 \pm 0.757$	$2.102 \pm 0.094$
GIN	$-2.138 \pm 0.090$	$11.868 \pm 2.689$	$2.234 \pm 0.271$
GPS	<b><math>-3.115 \pm 0.040</math></b>	<b><math>1.255 \pm 0.113</math></b>	<b><math>0.472 \pm 0.050</math></b>
GraphCON	$-1.488 \pm 0.000$	$52.104 \pm 0.016$	$5.734 \pm 0.011$
PH-DGN	$-2.616 \pm 0.317$	$4.656 \pm 3.013$	$1.323 \pm 0.485$
SWAN	$-2.782 \pm 0.205$	$2.905 \pm 1.556$	$0.896 \pm 0.232$

## 89 D Hyperparameter Selection

90 Tables 2 to 11 report the hyperparameter search space and the best values selected for each task (diam,  
 91 ecc, sssp, and ECHO-Chem) across the different GNN architectures we considered. For details on  
 92 specific hyperparameters, we refer the reader to the original papers. Each table includes the name of  
 93 the hyperparameter, its search range or categorical options, and the optimal value obtained for each  
 94 task, as identified through hyperparameter tuning on the validation set.

95 Another strong evidence supporting the long-range nature of the ECHO benchmark, implicitly comes  
 96 from our hyperparameter optimization process. Specifically, Bayesian Optimization consistently  
 97 selected configurations with a large number of GNN layers. This suggests that, even without explicit  
 98 guidance, the hyperparameter optimization procedure identifies deeper models as necessary to  
 99 minimize validation error, further reinforcing the notion that the task demands long-range information  
 100 propagation.

Table 2: Hyperparameters and their best values across tasks for A-DGN.

Hyperparameter	Search interval	diam	sssp	ecc	ECH0-Chem
Number of layers	$[1 - 40]$	27	28	40	34
Hidden dimension	$[16 - 256]$	68	65	45	130
Learning rate	$[10^{-5}, 10^{-2}]$	0.00101	0.00229	0.00473	0.00072
Weight decay	$[10^{-8}, 10^{-3}]$	0.00098	0.00000	0.00003	0.00001
$\epsilon$	$[0.001, 0.5]$	0.19254	0.32934	0.10560	0.25667
$\gamma$	$[0.001, 0.5]$	0.41827	0.46803	0.21252	0.19499
Graph convolution	NaiveAggr, GCN	NaiveAggr	NaiveAggr	NaiveAggr	GCN

Table 3: Hyperparameters and their best values across tasks for DRew.

Hyperparameter	Search interval	diam	sssp	ecc	ECH0-Chem
Number of layers	$[1 - 4]$	4	4	4	4
k-hop	$[1 - 10]$	10	10	10	10
Hidden dimension	$[16 - 256]$	249	78	119	232
Learning rate	$[10^{-5}, 10^{-2}]$	0.00037	0.00797	0.00126	0.00036
Weight decay	$[10^{-8}, 10^{-3}]$	0.00068	0.00011	0.00003	0.0
Employ delay	True, False	False	False	False	True

Table 4: Hyperparameters and their best values across tasks for GCNII.

Hyperparameter	Search interval	diam	sssp	ecc	ECH0-Chem
Number of layers	$[10 - 40]$	32	39	30	37
Hidden dimension	$[16 - 256]$	81	40	64	33
Learning rate	$[10^{-5}, 10^{-2}]$	0.00260	0.00032	0.00005	0.00345
Weight decay	$[10^{-8}, 10^{-3}]$	0.00000	0.00009	0.00009	0.00002
$\alpha$	$[0.0, 0.9]$	0.70544	0.07902	0.04742	0.17158

Table 5: Hyperparameters and their best values across tasks for GCN.

Hyperparameter	Search interval	diam	sssp	ecc	ECH0-Chem
Number of layers	$[1 - 40]$	26	40	26	8
Hidden dimension	$[16 - 256]$	48	42	40	109
Learning rate	$[10^{-5}, 10^{-2}]$	0.00007	0.00004	0.00023	0.00079
Weight decay	$[10^{-8}, 10^{-3}]$	0.00007	0.00009	0.00002	0.00002

Table 6: Hyperparameters and their best values across tasks for GIN.

Hyperparameter	Search interval	diam	sssp	ecc	ECH0-Chem
Number of layers	$[10 - 40]$	29	34	25	11
Hidden dimension	$[16 - 256]$	58	170	78	197
Learning rate	$[10^{-5}, 10^{-2}]$	0.00002	0.00003	0.00006	0.00002
Weight decay	$[10^{-8}, 10^{-3}]$	0.00003	0.00036	0.00091	0.00069

Table 7: Hyperparameters and their best values across tasks for GINE.

Hyperparameter	Search interval	diam	sssp	ecc	ECH0-Chem
Number of layers	$[1 - 40]$	–	–	–	22
Hidden dimension	$[16 - 256]$	–	–	–	85
Learning rate	$[10^{-5}, 10^{-2}]$	–	–	–	0.00014
Weight decay	$[10^{-8}, 10^{-3}]$	–	–	–	0.00004

Table 8: Hyperparameters and their best values across tasks for GPS.

Hyperparameter	Search interval	diam	sssp	ecc	ECHO-Chem
Number of layers	[1 – 40]	17	26	17	36
Hidden dimension	[16 – 256]	40	56	162	216
Learning rate	$[10^{-5}, 10^{-2}]$	0.00004	0.00031	0.00034	0.00005
Weight decay	$[10^{-8}, 10^{-3}]$	0.00015	0.00029	0.00007	0.00005

Table 9: Hyperparameters and their best values across tasks for GraphCON.

Hyperparameter	Search interval	diam	sssp	ecc	ECHO-Chem
Number of layers	[1 – 40]	37	25	19	35
Hidden dimension	[16 – 256]	63	72	151	253
Learning rate	$[10^{-5}, 10^{-2}]$	0.00088	0.00013	0.00007	0.00066
Weight decay	$[10^{-8}, 10^{-3}]$	0.00038	0.00001	0.00001	0.00043
$\epsilon$	[0.001, 1.0]	0.57880	0.95470	0.98433	0.72111

Table 10: Hyperparameters and their best values across tasks PH-DGN.

Hyperparameter	Search interval	diam	sssp	ecc	ECHO-Chem
Number of layers	[1 – 40]	17	37	21	14
Hidden dimension	[16 – 256]	28	66	120	103
Learning rate	$[10^{-5}, 10^{-2}]$	0.00150	0.00178	0.00037	0.00033
Weight decay	$[10^{-8}, 10^{-3}]$	0.00054	0.00082	0.00081	0.00063
$\epsilon$	[0.001, 1.0]	0.34977	0.16491	0.36140	0.68993
$\alpha$	[0.01, 1.0]	0.47190	0.90892	0.63323	0.87607
$\beta$	[0.01, 1.0]	0.70474	0.92918	0.99675	0.91251
$p$ conv mode	NaiveAggr, GCN	GCN	GCN	GCN	GCN
$q$ conv mode	NaiveAggr, GCN	GCN	GCN	GCN	NaiveAggr
Doubled dimension	True, False	False	False	True	True
Final state	$p, q, pq$	$pq$	$p$	$pq$	$pq$
Dampening mode	param+ MLP4ReLU DGNReLU	param+	DGNReLU	param	param+
External mode	MLP4Sin, DGNtanh	MLP4Sin	MLP4Sin	MLP4Sin	MLP4Sin

Table 11: Hyperparameters and their best values across tasks for SWAN.

Hyperparameter	Search interval	diam	sssp	ecc	ECHO-Chem
Number of layers	[1 – 40]	28	40	32	38
Hidden dimension	[16 – 256]	167	97	195	163
Learning rate	$[10^{-5}, 10^{-2}]$	0.00040	0.00107	0.00086	0.00063
Weight decay	$[10^{-8}, 10^{-3}]$	0.00057	0.00011	0.00010	0.00016
$\epsilon$	[0.001, 1.0]	0.54847	0.45462	0.07451	0.38229
$\gamma$	[0.001, 0.5]	0.41480	0.28342	0.45928	0.07794
$\beta$	[-1.0, 1.0]	0.34233	-0.20976	0.37682	-0.36245
Graph convolution	AntiSymNaiveAggr (ASNA) BoundedGCNConv (BGC) BoundedNaiveAggr (BNA)	ASNA	BNA	BNA	ASNA
Attention	True, False	True	False	False	False

## E Additional dataset information

Table 12 provides a summary of the input and target features used in the ECHO-Synth and ECHO-Chem datasets. Figures 3 and 4 report detailed statistics on the structural properties of the graphs in the two datasets, including distributions of the number of nodes, number of edges, average node degree, and graph diameter. Additionally, Figures 5a and 5b illustrate the correlation between the number of nodes and the graph diameter, highlighting structural differences between real and synthetic data. These insights support the design choices for model evaluation across diverse graph regimes.

Table 12: Summary of dataset properties.

Dataset	Node Features	Edge Features	Target
Synthetic	Random scalar, source indicator for sssp	None	diam, sssp, ecc
ECHO-Chem	Atomic number, distance from center of mass	Bond type, bond length	Partial charges

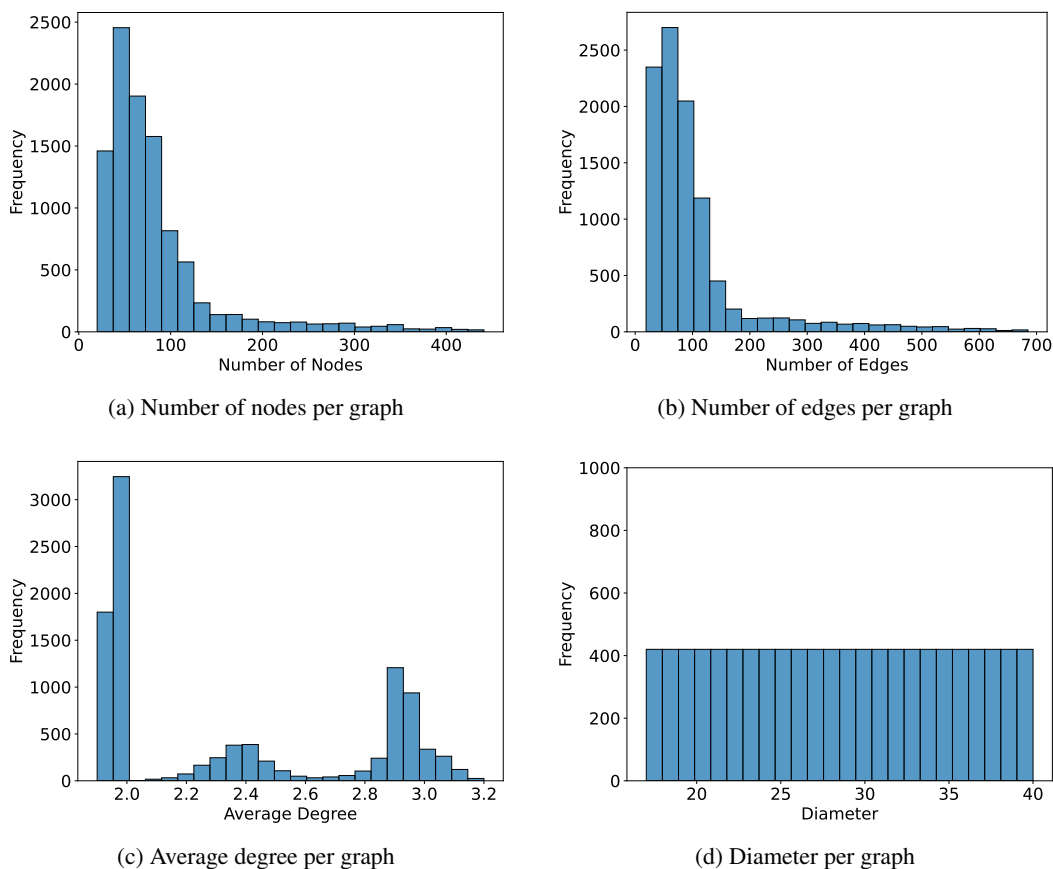


Figure 3: Statistics of the ECHO-Synth dataset.

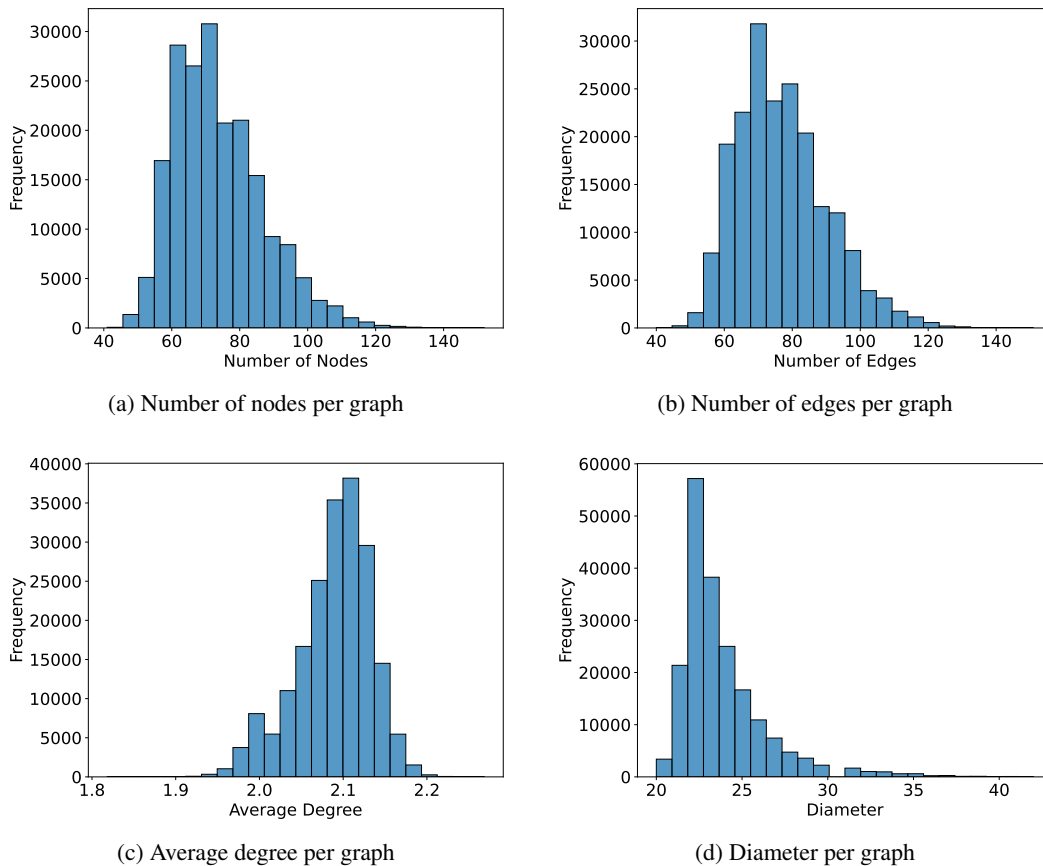


Figure 4: Statistics of the ECHO-Chem dataset.

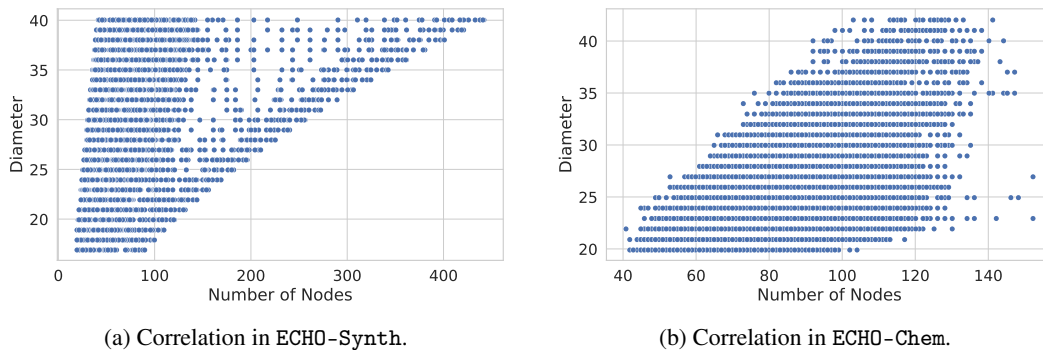


Figure 5: Correlation between number of nodes and graph diameter in ECHO-Synth and ECHO-Chem.

## 108 F Runtimes

109 To assess the computational efficiency and predictive performance of all models, we report both  
 110 training and inference runtimes measured on a NVIDIA H100 GPU, as well as the mean absolute  
 111 error (MAE) across tasks in the ECHO benchmark (see Table 13). Training time is measured as the  
 112 average per-epoch duration over 10 epochs, while inference time is computed as the average forward  
 113 pass duration over 10 independent runs on the test set, using a batch size of 512. The three metrics  
 114 correspond to the best hyperparameter configuration selected for each model. This comprehensive  
 115 evaluation allows a direct comparison of models not only in terms of accuracy but also with respect to  
 116 their scalability and practical deployability. We note that DRew’s reported runtime does not include



the preprocessing step, which involves computing the Floyd–Warshall algorithm [5], a procedure with cubic time complexity in the number of nodes.

Table 13 highlights that while transformer-based models like GPS achieve strong performance on long-range tasks, particularly on the real-world ECHO–Chem dataset, they do so at the cost of significantly higher computational overhead. In contrast, architectures such as SWAN and A-DGN strike a more favorable balance between efficiency and accuracy, suggesting the potential of non-dissipative DE-GNNs in overcoming the limitations of standard message passing.

Table 13: Training and inference runtime (in seconds, mean  $\pm$  standard deviation) on the ECHO Benchmark measured on a NVIDIA H100 GPU. Training time refers to the average time per epoch computed over 10 epochs. Inference time refers to the forward pass on the test set, computed over 10 independent runs. In both cases the batch size is set to 512. For each task, the reported values correspond to the best configuration of each model as selected during model selection. DRew’s reported runtime does not include the preprocessing step, which involves computing the Floyd–Warshall algorithm, a procedure with cubic time complexity in the number of nodes.

Metric	Model	diam	sssp	ecc	ECHO–Chem
Training (s)	A-DGN	1.430 $\pm$ 0.100	1.460 $\pm$ 0.130	1.710 $\pm$ 0.070	38.010 $\pm$ 0.430
Inference (s)		0.028 $\pm$ 0.002	0.018 $\pm$ 0.001	0.027 $\pm$ 0.001	0.809 $\pm$ 0.001
MAE		1.151 $\pm$ 0.038	1.176 $\pm$ 0.140	4.981 $\pm$ 0.037	8.47 $\pm$ 0.05
Training (s)	DRew	1.920 $\pm$ 0.050	1.880 $\pm$ 0.060	1.760 $\pm$ 0.100	31.090 $\pm$ 1.140
Inference (s)		0.100 $\pm$ 0.001	0.043 $\pm$ 0.001	0.057 $\pm$ 0.002	0.636 $\pm$ 0.007
Pre-processing (s)		48.108 $\pm$ 0.943	48.108 $\pm$ 0.943	48.108 $\pm$ 0.943	610.303 $\pm$ 1.629
MAE		1.243 $\pm$ 0.047	1.279 $\pm$ 0.011	<b>4.651<math>\pm</math>0.020</b>	8.37 $\pm$ 0.06
Training (s)	GCNII	1.700 $\pm$ 0.050	1.830 $\pm$ 0.020	1.620 $\pm$ 0.110	37.570 $\pm$ 0.500
Inference (s)		0.071 $\pm$ 0.002	0.062 $\pm$ 0.001	0.059 $\pm$ 0.001	0.463 $\pm$ 0.005
MAE		2.005 $\pm$ 0.093	2.128 $\pm$ 0.429	5.241 $\pm$ 0.030	9.26 $\pm$ 0.14
Training (s)	GCN	1.480 $\pm$ 0.060	1.790 $\pm$ 0.060	1.450 $\pm$ 0.100	16.590 $\pm$ 0.300
Inference (s)		0.048 $\pm$ 0.001	0.065 $\pm$ 0.003	0.046 $\pm$ 0.002	0.139 $\pm$ 0.001
MAE		3.832 $\pm$ 0.262	2.102 $\pm$ 0.004	5.233 $\pm$ 0.034	12.31 $\pm$ 2.04
Training (s)	GIN	1.410 $\pm$ 0.220	1.370 $\pm$ 0.060	1.340 $\pm$ 0.040	17.960 $\pm$ 0.460
Inference (s)		0.020 $\pm$ 0.001	0.019 $\pm$ 0.001	0.016 $\pm$ 0.001	0.122 $\pm$ 0.001
MAE		1.630 $\pm$ 0.161	2.234 $\pm$ 0.271	4.869 $\pm$ 0.092	13.29 $\pm$ 0.12
Training (s)	GINE	N/A	N/A	N/A	30.240 $\pm$ 1.140
Inference (s)		N/A	N/A	N/A	0.164 $\pm$ 0.001
MAE		N/A	N/A	N/A	8.15 $\pm$ 0.09
Training (s)	GPS	9.720 $\pm$ 0.070	14.580 $\pm$ 0.210	11.960 $\pm$ 0.050	689.420 $\pm$ 1.040
Inference (s)		4.536 $\pm$ 0.006	7.026 $\pm$ 0.001	6.235 $\pm$ 0.076	97.952 $\pm$ 0.433
MAE		2.160 $\pm$ 0.098	<b>0.472<math>\pm</math>0.050</b>	4.758 $\pm$ 0.021	<b>5.65<math>\pm</math>0.12</b>
Training (s)	GraphCON	0.990 $\pm$ 0.120	0.920 $\pm$ 0.040	0.940 $\pm$ 0.190	13.570 $\pm$ 0.780
Inference (s)		0.006 $\pm$ 0.001	0.004 $\pm$ 0.001	0.006 $\pm$ 0.001	0.066 $\pm$ 0.003
MAE		2.969 $\pm$ 0.189	5.734 $\pm$ 0.011	5.474 $\pm$ 0.001	15.20 $\pm$ 0.05
Training (s)	PH-DGN	2.840 $\pm$ 0.060	4.480 $\pm$ 0.060	3.010 $\pm$ 0.060	46.710 $\pm$ 0.780
Inference (s)		0.180 $\pm$ 0.011	0.375 $\pm$ 0.002	0.299 $\pm$ 0.006	1.005 $\pm$ 0.021
MAE		1.627 $\pm$ 0.398	1.323 $\pm$ 0.485	5.068 $\pm$ 0.126	7.92 $\pm$ 0.07
Training (s)	SWAN	2.330 $\pm$ 0.120	2.130 $\pm$ 0.050	2.090 $\pm$ 0.110	81.590 $\pm$ 0.700
Inference (s)		0.203 $\pm$ 0.002	0.099 $\pm$ 0.001	0.168 $\pm$ 0.001	3.822 $\pm$ 0.048
MAE		<b>1.121<math>\pm</math>0.070</b>	0.896 $\pm$ 0.232	4.840 $\pm$ 0.045	8.79 $\pm$ 0.06

## G Chemical Simulation Technical Information

This appendix provides detailed information on the computational pipeline used to derive partial atomic charges in the ECHO–Chem dataset. The pipeline comprises three primary stages: (i) 3D structure generation from SMILES, (ii) quantum chemical computation of partial charges, and (iii) geometry optimization.

### 3D Structure Generation from SMILES.

Since subsequent charge optimization steps require pre-optimized 3D coordinates, all structures were geometry-optimized prior to simulation using Open Babel [28] and its Python interface, Pybel [29]. Initial molecular geometries were generated from SMILES strings using the General AMBER Force Field (GAFF) [15]. GAFF was chosen over alternatives such as MMFF94 [17] due to its favorable trade-off between accuracy and computational cost, and its strong performance in predicting both energies and geometries. The optimization procedure involved 100 steps of coarse minimization followed by 500 steps of local refinement for each molecule. The SMILES strings were converted into 3D conformers, which were then minimized to yield low-energy structures. These structures were exported in SDF format for subsequent compatibility. The average time required for 3D structure generation per molecule—considering only those satisfying the ECHO-Chem dataset diameter criteria—was  $562 \pm 124$  ms.

### Quantum Chemical Computations with ORCA.

Table 14: Mean time required for computation of partial charges of a single molecule with different configuration of the ORCA tool. Variance is computed over 30 random molecules from the ECHO-Chem dataset. \*Denotes the chosen basis set for DFT computation.

Method	Setting	Times (s)
HF-3c	LooseSCF	$10.4 \pm 1.3$
HF-3c	TightSCF	$28.1 \pm 4.3$
B3LYP def2-TZVP* (DFT)	LooseSCF	$146.5 \pm 10.1$
B3LYP def2-TZVP* (DFT)	TightSCF	$634.5 \pm 21.3$

To compute partial atomic charges, we employed the ORCA quantum chemistry software suite (version 6.0.1) [24, 25, 26]. All calculations were performed using the composite HF-3c method, a computationally efficient approximation to Hartree-Fock theory that incorporates computational corrections for accuracy and reliability. This combination makes HF-3c highly suitable for large-scale quantum computations, balancing physical fidelity with computational tractability. We provide a summary of required times for computation with both full DFT computations and HF methods in Table 14. Under this configuration, the average runtime for a single quantum chemical calculation was  $10.4 \pm 1.3$  seconds per molecule requiring  $\approx 3$  weeks of computational time on our hardware configuration. Simulation were run exploiting full thread parallelism provided by ORCA.

#### Self-Consistent Field (SCF) Convergence Settings.

To further accelerate the simulations, we adopted the LooseSCF setting in ORCA, which applies relaxed convergence thresholds to the SCF procedure. This configuration is particularly advantageous in high-throughput scenarios where full precision may be unnecessary. The tolerances govern convergence with respect to energy changes, gradient norms, density matrix stability, and integral thresholds.

**Charge Extraction.** Atomic partial charges were extracted using the Mulliken population analysis method [23]. These charges were used as supervision signals in our dataset generation pipeline.

## H Hardware resources

All quantum chemistry simulations were conducted on a dual-socket Intel Xeon 6780E machine with a total of 288 physical cores (144 cores per socket, 1 thread per core). Each socket is equipped with 108MiB of L3 cache, for a combined 216MiB of shared L3 cache, along with 288MiB of L2 and 27MiB of L1 (data + instruction) cache across the system. The CPUs support AVX2 and FMA instruction sets, enabling efficient linear algebra operations, which are critical for electronic structure methods.

The machine is configured with two NUMA nodes, each associated with one of the sockets. Each NUMA node has over 500GiB of local RAM for a total of approximately 1TiB of RAM. The high memory capacity and bandwidth are critical for quantum chemistry workloads, particularly those using density functional theory (DFT) or correlated wavefunction methods, which require extensive memory for large basis sets and integral evaluations.

The large number of physical cores allowed us to parallelize over both molecular batches and internal basis function evaluations, providing efficient scaling for density functional theory (DFT) and semi-empirical calculations.

For model training and inference, we used a separate compute node equipped with 8 NVIDIA H100 GPUs.

## I Limitations

Although we believe that our work has a positive impact, it is important to note that the generation of ECHO-Chem required substantial quantum simulation time ( $\approx 3$  weeks), making future extensions or reprocessing expensive. This might hinder widespread replication or adaptation of the dataset without access to comparable computational resources. Also, ECHO currently evaluates long-range propagation on static graphs. This excludes temporal graphs [13], which are prevalent in areas such as social networks or biological systems. Extending the benchmark to dynamic settings remains an important future direction.

## References

- [1] Jacob Bamberger, Benjamin Gutteridge, Scott le Roux, Michael Bronstein, and Xiaowen Dong. On Measuring Long-Range Interactions in Graph Neural Networks. In *Proceedings of the 42th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 13–19 Jul 2025.
- [2] Ali Behrouz and Farnoosh Hashemi. Graph mamba: Towards learning on graphs with state space models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 119–130, 2024.
- [3] Chen Cai, Truong Son Hy, Rose Yu, and Yusu Wang. On the connection between MPNN and graph transformer. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 3408–3430. PMLR, 23–29 Jul 2023.
- [4] Yun Young Choi, Sun Woo Park, Minhoo Lee, and Youngho Woo. Topology-informed graph transformer. In Sharvaree Vadgama, Erik Bekkers, Alison Pouplin, Sekou-Oumar Kaba, Robin Walters, Hannah Lawrence, Tegan Emerson, Henry Kvinge, Jakub Tomczak, and Stephanie Jegelka, editors, *Proceedings of the Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM)*, volume 251 of *Proceedings of Machine Learning Research*, pages 20–34. PMLR, 29 Jul 2024.
- [5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
- [6] Yuhui Ding, Antonio Orvieto, Bobby He, and Thomas Hofmann. Recurrent distance filtering for graph representation learning. In *Forty-first International Conference on Machine Learning*, 2024.
- [7] Vijay Prakash Dwivedi, Ladislav Rampásek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long Range Graph Benchmark. In *Advances in Neural Information Processing Systems*, volume 35, pages 22326–22340. Curran Associates, Inc., 2022.
- [8] Moshe Eliasof, Alessio Gravina, Andrea Ceni, Claudio Gallicchio, Davide Bacciu, and Carola-Bibiane Schönlieb. Graph Adaptive Autoregressive Moving Average Models. In *Forty-second International Conference on Machine Learning*, 2025.
- [9] Federico Errica, Henrik Christiansen, Viktor Zaverkin, Takashi Maruyama, Mathias Niepert, and Francesco Alesiani. Adaptive message passing: A general framework to mitigate oversmoothing, oversquashing, and underreaching. *arXiv preprint arXiv:2312.16560*, 2024.

- [10] Simon Geisler, Arthur Kosmala, Daniel Herbst, and Stephan Günnemann. Spatio-spectral graph neural networks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 49022–49080. Curran Associates, Inc., 2024.
- [11] Lorenzo Giusti, Teodora Reu, Francesco Ceccarelli, Cristian Bodnar, and Pietro Liò. Cin++: Enhancing topological message passing. *arXiv preprint arXiv:2306.03561*, 2023.
- [12] Daniel Glickman and Eran Yahav. Diffusing graph attention. *arXiv preprint arXiv:2303.00613*, 2023.
- [13] Alessio Gravina and Davide Bacciu. Deep Learning for Dynamic Graphs: Models and Benchmarks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2024.
- [14] Alessio Gravina, Moshe Eliasof, Claudio Gallicchio, Davide Bacciu, and Carola-Bibiane Schönlieb. On oversquashing in graph neural networks through the lens of dynamical systems. In *The 39th Annual AAAI Conference on Artificial Intelligence*, 2025.
- [15] Stefan Grimme, Jens Antony, Stephan Ehrlich, and Helge Krieg. A consistent and accurate ab initio parametrization of density functional dispersion correction (dft-d) for the 94 elements h-pu. *J. Chem. Phys.*, 132:154104, 2010.
- [16] Benjamin Gutteridge, Xiaowen Dong, Michael M Bronstein, and Francesco Di Giovanni. Drew: Dynamically rewired message passing with delay. In *International Conference on Machine Learning*, pages 12252–12267. PMLR, 2023.
- [17] Thomas A. Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of Computational Chemistry*, 17(5-6):490–519, 1996.
- [18] Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann Lecun, and Xavier Bresson. A generalization of ViT/MLP-mixer to graphs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 12724–12745. PMLR, 23–29 Jul 2023.
- [19] Simon Heilig, Alessio Gravina, Alessandro Trenta, Claudio Gallicchio, and Davide Bacciu. Port-Hamiltonian Architectural Bias for Long-Range Propagation in Deep Graph Networks. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [20] Guixiang Ma, Vy A. Vo, Theodore L. Willke, and Nesreen K. Ahmed. Augmenting recurrent graph neural networks with a cache. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD ’23, page 1608–1619, New York, NY, USA, 2023. Association for Computing Machinery.
- [21] Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23321–23337. PMLR, 23–29 Jul 2023.
- [22] Gaspard Michel, Giannis Nikolentzos, Johannes F. Lutzeyer, and Michalis Vazirgiannis. Path neural networks: Expressive and accurate graph neural networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 24737–24755. PMLR, 23–29 Jul 2023.
- [23] R. S. Mulliken. Electronic Population Analysis on LCAO–MO Molecular Wave Functions. I. *The Journal of Chemical Physics*, 23(10):1833–1840, October 1955.
- [24] Frank Neese. Software update: the orca program system, version 5.0. *WIREs Comput. Mol. Sci.*, 12(1):e1606, 2022.
- [25] Frank Neese. The shark integral generation and digestion system. *J. Comput. Chem.*, 44:381–396, 2023.

- 266 [26] Frank Neese, Frank Wennmohs, Ute Becker, and Christoph Riplinger. The ORCA quantum  
267 chemistry program package. *The Journal of Chemical Physics*, 152(22):224108, June 2020.
- 268 [27] Nhat Khang Ngo, Truong Son Hy, and Risi Kondor. Multiresolution graph transformers and  
269 wavelet positional encoding for learning long-range and hierarchical structures. *The Journal of*  
270 *Chemical Physics*, 159(3), July 2023.
- 271 [28] Noel M. O’Boyle, Michael Banck, Craig A. James, Chris Morley, Tim Vandermeersch, and  
272 Geoffrey R. Hutchison. Open Babel: An open chemical toolbox. *Journal of Cheminformatics*,  
273 3(1):33, October 2011.
- 274 [29] Noel M. O’Boyle, Chris Morley, and Geoffrey R. Hutchison. Pybel: A Python wrapper for the  
275 OpenBabel cheminformatics toolkit. *Chemistry Central Journal*, 2(1):5, March 2008.
- 276 [30] Ladislav Rampásek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and  
277 Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in*  
278 *Neural Information Processing Systems*, 35, 2022.
- 279 [31] Behzad Shirzad, Amir M. Rahmani, and Marzieh Aghaei. Exphormer: Sparse attention for  
280 graphs. *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.
- 281 [32] Jan Tönshoff, Martin Ritzert, Eran Rosenbluth, and Martin Grohe. Where did the gap go?  
282 reassessing the long-range graph benchmark. In *The Second Learning on Graphs Conference*,  
283 2023.
- 284 [33] Chloe Wang, Oleksii Tsepa, Jun Ma, and Bo Wang. Graph-mamba: Towards long-range graph  
285 sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*, 2024.