Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-026-00387-2

Original Article

Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark

Sadiq Halliru Usman [1], Sanjay Choudhary [2], Abdulrasheed Halliru Usman [3]

1. *Robotic Engineering, Vivekananda Global University, Jaipur, IND*
2. *Mechanical Engineering, Vivekananda Global University, Jaipur, IND*
3. *Polymer Chemistry, Federal University Dutsin-Ma, Dutsin-Ma, NGA*

Received: September 02, 2026 | Review began: September 09, 2026 | Review ended: October 18, 2026 | Published: November 23, 2026

© **Copyright** 2026
This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

Background

Graph neural networks (GNNs) are widely used to predict molecular properties, but comparisons with classical models depend on design choices that are rarely varied, including how a molecule is converted into a graph. We asked whether the choice between a covalent-bond graph and a dense distance-cutoff graph changes the ranking of GNNs and classical models.

Methods

We used a random subset of 10,000 molecules from the QM9 dataset and predicted the energy gap between the highest occupied and lowest unoccupied molecular orbitals. Ten repeated random splits (80% training, 10% validation, and 10% test) were shared by all models, giving 10 runs per model. Hyperparameters were chosen on validation data only, and GNNs were trained to early stopping. Three GNN architectures (a graph convolutional network, a graph isomorphism network (GIN), and GraphSAGE) used node features on bond graphs. A GIN on a dense 5-Å graph served as a control. Random forest and kernel ridge regression on atom counts or Coulomb-matrix eigenvalues were the classical baselines. We report paired differences with 95% CIs, effect sizes, and Holm-adjusted *P*-values.

Results

The mean predictor had a test mean absolute error of 1.08 eV. The best classical model (random forest on Coulomb-matrix eigenvalues) reached 0.611 eV. The three GNNs on bond graphs reached 0.289-0.304 eV, about 50-53% lower than the best classical model (Holm-adjusted $P < 0.001$ for each). The dense graph was almost complete (mean node degree, 15.2; edge density, 89.7%). On that graph, the same GIN reached 0.796 eV, similar to a random forest on atom counts (0.747 eV), and 0.506 eV worse than on the bond graph (95% CI, 0.491-0.521).

Conclusions

The apparent ranking of GNNs and classical models depended on how molecules were converted into graphs, not only on the model family. Benchmarks should report graph density and vary graph construction. These findings are limited to small molecules, one target, one data size, and GNNs that use node features only; the errors remain well above those reported for geometry-aware models, which we did not test.

Categories: Artificial Intelligence and Machine Learning in the Cloud, Graph Algorithms, Computational Neuroscience

How to cite this article:
Usman S, Choudhary S, Usman A (November 23, 2026) Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark. Cureus J Comput Sci 3 : es44389-026-00387-2. DOI https://doi.org/10.7759/s44389-026-00387-2

Keywords: graph neural networks, molecular property prediction, graph construction, random forest, benchmarking, qm9 dataset, statistical testing

Introduction

Graph neural networks (GNNs) are widely used to predict molecular properties because a molecule can be represented naturally as a graph of atoms and bonds [1,2]. Benchmark suites such as MoleculeNet have standardized the comparison of GNNs with classical machine-learning methods on datasets that include the QM9 collection of small organic molecules [3,4]. An earlier large comparison on QM9 found that graph-based representations and graph convolutions gave the lowest errors for orbital energies and the orbital energy gap, whereas methods built on hand-crafted descriptors performed better for several energetic properties [5]. Descriptors such as the eigenvalues of the Coulomb matrix remain simple and inexpensive classical baselines [6].

How a molecule is converted into a graph is a design choice that is seldom varied in such comparisons. Two choices are common. The first connects atoms that are covalently bonded. The second connects every pair of atoms closer than a distance cutoff, as is done in geometry-aware models [7,8]. For molecules as small as those in QM9, a cutoff of a few angstroms can connect almost every pair of atoms. A model that then uses only the element of each atom as a node feature receives little information beyond the composition of the molecule. A benchmark that does not report graph density, or vary the graph, could therefore misstate how GNNs compare with simpler models.

We tested two hypotheses on the task of predicting the energy gap between the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO). The first hypothesis (H1) was that, under a protocol with validation-based tuning, training to convergence, repeated random splits, and an equal number of runs per model, GNNs on covalent-bond graphs outperform random forest and kernel ridge regression. The second hypothesis (H2) was that the same GNN performs much worse on a near-complete distance-cutoff graph than on a bond graph, to the point that it resembles a composition-only baseline. These hypotheses were formulated after an earlier analysis of ours that used only the dense graph and had suggested the opposite ranking, and they were not preregistered. The experiments reported here were designed to test whether that earlier ranking was an artifact of graph construction.

Materials And Methods

Dataset and target

We used a pre-processed copy of QM9 distributed with the code of DimeNet [8] (https://github.com/gasteigerjo/dimenet). It contains 130,831 molecules with three-dimensional geometries and quantities computed with density functional theory [4,9]. The original release contains 133,885 molecules; the remainder were excluded by the preprocessing of that file, whose criteria we did not verify independently. A random subset of 10,000 molecules was drawn with a fixed seed. The target was the HOMO-LUMO gap, in electron volts.

Graph construction

Atoms were nodes, and the only node feature was a one-hot encoding of the element (hydrogen, carbon, nitrogen, oxygen, or fluorine). Two graphs were built for every molecule. In the bond graph, two atoms were connected when their distance was below 1.2 times the sum of their covalent radii. This rule was applied to the geometries and is not a chemical bond perception algorithm. In the dense graph, which reproduces the construction of our earlier analysis, two atoms were connected when they were within 5.0 Å. We report the mean node degree of both graphs, the mean edge density of the dense graph, and the fraction of molecules whose bond graph forms a single connected component.

Classical models

Two descriptor sets were used. The first contained the number of atoms of each element, the number of atoms, the number of edges of the dense graph, and the mean, standard deviation, minimum, and maximum of the interatomic distances within the cutoff (11 features). The second contained the eigenvalues of the Coulomb matrix [6] sorted in

How to cite this article:
Usman S, Choudhary S, Usman A (November 23, 2026) Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark. Cureus J Comput Sci 3 : es44389-026-00387-2. DOI https://doi.org/10.7759/s44389-026-00387-2

descending order and padded with zeros to the largest molecule size (29 atoms). Random forest regression with 300 trees [10] was trained on both sets. Kernel ridge regression with a Laplacian kernel was trained on the standardized Coulomb-matrix eigenvalues, using a random training subset of up to 5,000 molecules per run because of memory. All classical models used scikit-learn [11].

Graph neural networks

Three architectures were implemented with PyTorch [12] and PyTorch Geometric [13]: the graph convolutional network (GCN) [14], the graph isomorphism network (GIN) [15], and GraphSAGE [16]. Each used a linear embedding of the node features, a stack of message-passing layers with rectified linear activations, mean pooling over atoms, and a two-layer readout. These models use node features only and no interatomic distances. All three were trained on the bond graph. To isolate the effect of graph construction, the GIN was also trained on the dense graph as a control. Models with distance features on the edges, such as the edge-conditioned message-passing network, were not trained because they required about 28 s per epoch on our hardware, which made the full protocol infeasible.

Training and model selection

The molecules were split at random into training (80%), validation (10%), and test (10%) sets, and this was repeated 10 times with different seeds. The same 10 splits were used for every model, so every model has exactly 10 runs and all comparisons are paired. Hyperparameters of every model were chosen with the validation set of the first split only. The test sets were used only to report results. For the GNNs, we varied one factor at a time around a base configuration (learning rate 0.003, width 64, three layers): the learning rate (0.001 or 0.003), the width (64 or 128), and the depth (three or five layers). For random forest, we varied the minimum leaf size (one or three) and the fraction of features per split (1.0 or 0.5). For kernel ridge regression, we varied the regularization (0.001, 0.01, or 0.1) and the kernel width (0.01, 0.03, or 0.1) on a full grid. GNNs were trained with the Adam optimizer, a batch size of 128, learning-rate reduction on validation plateau, and early stopping on validation mean absolute error (MAE) (patience 15 epochs, maximum 120 epochs). The model from the epoch with the lowest validation error was evaluated on the test set. Targets were standardized with the training-set mean and standard deviation. Because the number of epochs was set by early stopping, width, depth, and learning rate were varied independently of training length. All experiments ran on one central processing unit core. The runs were interrupted and resumed from saved results several times; random seeds were fixed for each split, and a repeated tuning pass returned identical validation errors.

Statistical analysis

The primary metric was the MAE on the test set. We also report the coefficient of determination (R^2) and the training time per run. The training-set mean, used as a predictor, served as a reference on each split. The classical model with the lowest mean test MAE (random forest on Coulomb-matrix eigenvalues) was the reference for paired comparisons across the 10 splits. For every other model, we report the mean paired difference in test MAE with a 95% confidence interval from the *t*-distribution, the paired standardized effect size (mean difference divided by the standard deviation of the differences), and the paired *t*-test *P*-value adjusted by the Holm procedure [17] over the six comparisons. Wilcoxon signed-rank tests were run as a check; with 10 pairs, the smallest attainable two-sided *P*-value is 0.002. Because the 10 test sets overlap, the splits are not independent; we return to this in the "Discussion" section.

Results And Discussion

Graph properties

The dense graph was almost complete. For a mean molecule size of 18.0 atoms, the mean node degree was 15.2, and the mean edge density was 89.7%. The bond graph had a mean node degree of 2.07, and 99.9% of molecules (9,986 of 10,000) formed a single connected component.

How to cite this article:
Usman S, Choudhary S, Usman A (November 23, 2026) Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark. Cureus J Comput Sci 3 : es44389-026-00387-2. DOI https://doi.org/10.7759/s44389-026-00387-2

Model comparison

The mean predictor had a test MAE of 1.084 ± 0.021 eV and an R^2 of −0.001 (mean ± standard deviation over the 10 splits). Table *1* and Figure *1* show the results for all models. The three GNNs on bond graphs reached MAEs of 0.289 ± 0.011, 0.292 ± 0.014, and 0.304 ± 0.011 eV (GIN, GraphSAGE, and GCN, respectively), with R^2 values of about 0.90-0.91. The best classical models were random forest and kernel ridge regression on Coulomb-matrix eigenvalues (0.611 ± 0.015 and 0.616 ± 0.019 eV). Random forest on atom counts reached 0.747 ± 0.016 eV, and the GIN on the dense graph reached 0.796 ± 0.021 eV.

Model	Test MAE, eV (mean ± SD)	R^2	Time per run (s)	Median epochs
GIN, bond graph	0.289 ± 0.011	0.91	91	90
GraphSAGE, bond graph	0.292 ± 0.014	0.90	180	96
GCN, bond graph	0.304 ± 0.011	0.90	214	106
Random forest, Coulomb matrix	0.611 ± 0.015	0.62	35	–
Kernel ridge, Coulomb matrix	0.616 ± 0.019	0.63	2	–
Random forest, atom counts	0.747 ± 0.016	0.47	5	–
GIN, dense graph (control)	0.796 ± 0.021	0.42	104	53
Mean predictor	1.084 ± 0.021	−0.001	–	–

TABLE 1: Test-set performance over 10 repeated splits (10,000 molecules).

Note: Times are for one central processing unit core.
GIN, Graph Isomorphism Network; GCN, Graph Convolutional Network; MAE, Mean Absolute Error; SD, Standard Deviation

How to cite this article:
Usman S, Choudhary S, Usman A (November 23, 2026) Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark. Cureus J Comput Sci 3 : es44389-026-00387-2. DOI https://doi.org/10.7759/s44389-026-00387-2

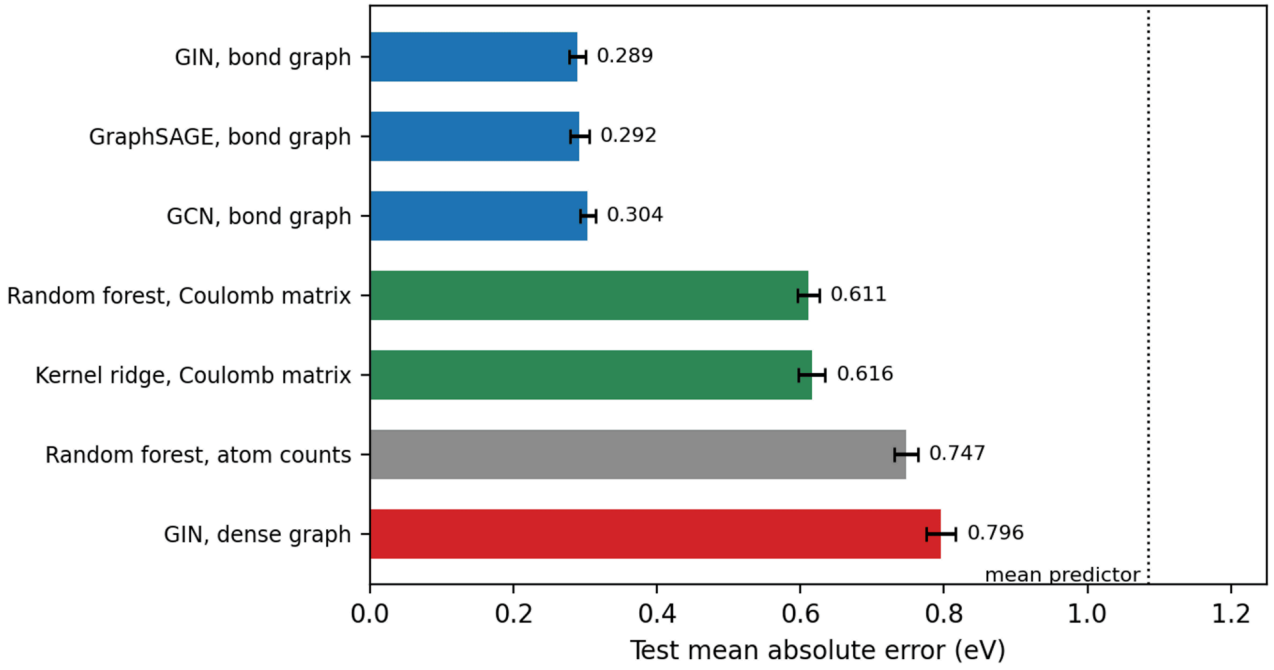


FIGURE 1: Test MAE (mean and standard deviation over 10 splits) for each model.

Note: The dotted line marks the mean predictor.
GIN, Graph Isomorphism Network; GCN, Graph Convolutional Network; MAE, Mean Absolute Error

Paired comparisons

Table 2 gives the paired comparisons with random forest on Coulomb-matrix eigenvalues. Each of the three GNNs on bond graphs had a lower MAE than the reference by 0.31-0.32 eV (for the GIN, −0.322 eV; 95% CI, −0.338 to −0.306; paired effect size, −14.4; Holm-adjusted $P < 0.001$). Kernel ridge regression did not differ from the reference (Holm-adjusted $P = 0.15$), whereas random forest on atom counts and the GIN on the dense graph were worse than the reference (Holm-adjusted $P < 0.001$ for both). Wilcoxon signed-rank tests gave the same conclusions. The effect sizes are very large relative to the spread across splits, but the *P*-values should be read with the caveat that the splits are not independent.

Model	Difference in MAE vs reference (eV)	95% CI (eV)	Paired effect size	Holm-adjusted *P*
GIN, bond graph	−0.322	−0.338 to −0.306	−14.4	<0.001
GraphSAGE, bond graph	−0.319	−0.333 to −0.305	−16.0	<0.001
GCN, bond graph	−0.308	−0.323 to −0.293	−14.7	<0.001
Kernel ridge, Coulomb matrix	+0.005	−0.002 to +0.012	0.5	0.15
Random forest, atom counts	+0.136	+0.125 to +0.148	8.5	<0.001

How to cite this article:
Usman S, Choudhary S, Usman A (November 23, 2026) Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark. Cureus J Comput Sci 3 : es44389-026-00387-2. DOI https://doi.org/10.7759/s44389-026-00387-2

GIN, dense graph (control)	+0.185	+0.171 to +0.198	9.9	<0.001

TABLE 2: Paired comparisons with random forest on Coulomb-matrix eigenvalues (reference).

Note: A negative difference means a lower error than the reference.
GIN, Graph Isomorphism Network; GCN, Graph Convolutional Network; MAE, Mean Absolute Error

Effect of graph construction

The GIN was trained with identical settings on the same 10 splits on both graphs. On the dense graph, its MAE was higher by 0.506 eV (95% CI, 0.491-0.521; paired effect size, 24.1; $P < 0.001$). On the dense graph, it performed similarly to the random forest on atom counts (0.796 vs 0.747 eV), and both were far from the bond-graph models.

Convergence, tuning, and training time

Figure *2* shows validation curves for the first split. The three bond-graph models reached a plateau, and the dense-graph model plateaued early at a much higher error. Of the 40 GNN runs, 37 stopped by early stopping before the 120-epoch limit. Three runs reached the limit (GIN on the bond graph, split 3; GCN on the bond graph, splits 5 and 6; splits counted from one). Excluding them changed the mean MAE by at most 0.003 eV (GIN, 0.289-0.292 eV; GCN, 0.304-0.301 eV). The selected hyperparameters are in Table *3*. Training times per run ranged from about 2 s (kernel ridge regression) to about 3.5 min (GCN) on one core (Table *1*).

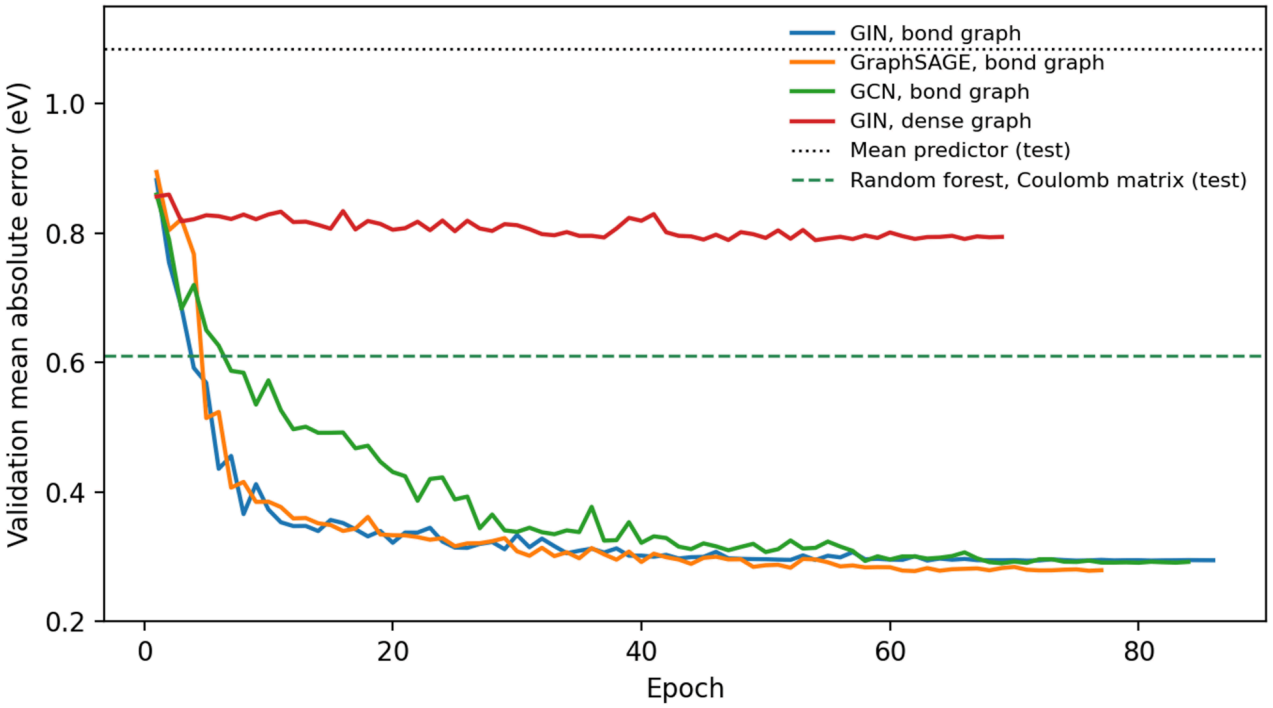


FIGURE 2: Validation MAE by epoch for the first split.

Note: Dashed and dotted lines show test MAE (mean over splits) of the best classical model and of the mean predictor.
GIN, Graph Isomorphism Network; GCN, Graph Convolutional Network; MAE, Mean Absolute Error

How to cite this article:
Usman S, Choudhary S, Usman A (November 23, 2026) Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark. Cureus J Comput Sci 3 : es44389-026-00387-2. DOI https://doi.org/10.7759/s44389-026-00387-2

Model	Learning rate	Width	Layers	Validation MAE (eV)	Range over grid (eV)
GIN, bond graph	0.003	64	3	0.269	0.269-0.302
GraphSAGE, bond graph	0.003	128	3	0.274	0.274-0.325
GCN, bond graph	0.003	128	3	0.293	0.293-0.402
GIN, dense graph (control)	0.003	64	3	0.785	0.785-0.812

TABLE 3: Selected hyperparameters of the GNNs and the validation MAE of the selected configuration on the first split.

GIN, Graph Isomorphism Network; GCN, Graph Convolutional Network; MAE, Mean Absolute Error; GNNs, Graph Neural Networks

For random forest, the selected minimum leaf size was one with a feature fraction of 1.0 on Coulomb-matrix eigenvalues, and three with 0.5 on atom counts. For kernel ridge regression, the selected regularization was 0.1, and the kernel width was 0.03.

Discussion

Both hypotheses were supported in this benchmark. On covalent-bond graphs, three GNN architectures that use only element types as node features had errors about 50-53% lower than the best classical model, which is consistent with the earlier QM9 comparison in which graph-based methods gave the lowest errors for the orbital energy gap [5]. The same GIN on a near-complete distance-cutoff graph performed like a random forest on atom counts. An analysis restricted to the dense graph would therefore have concluded that classical models outperform GNNs, which is the opposite of what the bond-graph results show. In this setting, the ranking of model families depended more on the input graph than on the model class.

A plausible explanation is that, when almost every atom is connected to every other atom and only the element is available as a node feature, message passing and pooling carry little more than the composition of the molecule. Our control is consistent with this explanation, but we did not test it directly, for example, with composition-only features fed to the same network or with intermediate cutoffs. We also did not test whether adding distances as edge features would change the dense-graph result, because models with edge features were too slow for our hardware.

The absolute accuracy of the bond-graph GNNs is limited. Their errors remain well above those reported for geometry-aware models that use interatomic distances [7,8], which we did not run, so we make no claim about the relative performance of those models or about state-of-the-art accuracy. The classical baselines are also not the strongest available; other descriptors were compared on QM9 in the earlier study [5], and kernel ridge regression here used at most 5,000 training molecules per run.

This study has further limitations. We used one data size (10,000 of 130,831 molecules), one target, and one initial random subset, so the variability we report excludes the sampling of the subset. Hyperparameters were chosen on one split with small one-factor-at-a-time grids, and tuning effort was comparable across model families but not exhaustive. The 10 repeated random splits share molecules, so their test errors are correlated and the paired *P*-values are likely to be optimistic; corrected resampling tests or nested cross-validation would be more conservative. The effect sizes are large relative to the spread across splits, which makes a change in the main conclusion unlikely, but the *P*-values should not be over-interpreted. Three GNN runs did not converge within the epoch limit, although excluding them did not change the

How to cite this article:
Usman S, Choudhary S, Usman A (November 23, 2026) Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark. Cureus J Comput Sci 3 : es44389-026-00387-2. DOI https://doi.org/10.7759/s44389-026-00387-2

results. Bond graphs were derived from a covalent-radius rule and not from chemical bond perception. QM9 contains only small molecules with five elements, so the results may not transfer to larger or more diverse molecules. Finally, all experiments ran on one central processing unit core.

Future work should add models with distance features on the edges, which requires more compute than we had, test several data sizes up to the full dataset and several targets, vary the cutoff of the distance graph, and use nested cross-validation for the statistical comparison.

Conclusions

In this benchmark, whether GNNs outperformed classical models for predicting a molecular electronic property depended on how molecules were converted into graphs. With graphs built from covalent bonds, three architectures clearly outperformed the classical baselines. With a near-complete distance-cutoff graph, the same architecture performed like a baseline that uses only atom counts. Benchmarks that compare graph-based and classical models should therefore report the density of the input graph and vary the graph construction before drawing conclusions about model families.

These conclusions are limited to small molecules, one target, one data size, and networks that use node features only, and they do not establish state-of-the-art accuracy. The code and raw results are provided so that the analysis can be repeated and extended to models with distance features and to larger data.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Sadiq Halliru Usman, Sanjay Choudhary, Abdulrasheed Halliru Usman

Acquisition, analysis, or interpretation of data: Sadiq Halliru Usman, Sanjay Choudhary, Abdulrasheed Halliru Usman

Drafting of the manuscript: Sadiq Halliru Usman, Sanjay Choudhary, Abdulrasheed Halliru Usman

Critical review of the manuscript for important intellectual content: Sadiq Halliru Usman, Sanjay Choudhary, Abdulrasheed Halliru Usman

Supervision: Sadiq Halliru Usman, Sanjay Choudhary, Abdulrasheed Halliru Usman

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue.

Conflicts of interest: In compliance with the ICMJE uniform disclosure form, all authors declare the following:

- **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work.
- **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work.
- **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

How to cite this article:
Usman S, Choudhary S, Usman A (November 23, 2026) Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark. Cureus J Comput Sci 3 : es44389-026-00387-2. DOI https://doi.org/10.7759/s44389-026-00387-2

Data Availability Statements

The datasets (and/or code) supporting this study are available from the corresponding author upon reasonable request.

References

1. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G: The graph neural network model. IEEE Transactions on Neural Networks. 2009, 20:61-80. 10.1109/TNN.2008.2005605
2. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE: Neural message passing for quantum chemistry. Proceedings of the 34th International Conference on Machine Learning. PMLR, Sydney, Australia; 2017. 70:1263-1272. 10.48550/arXiv.1704.01212
3. Wu Z, Ramsundar B, Feinberg EN, et al.: MoleculeNet: a benchmark for molecular machine learning. Chemical Science. 2018, 9:513-530. 10.1039/c7sc02664a
4. Ramakrishnan R, Dral PO, Rupp M, von Lilienfeld OA: Quantum chemistry structures and properties of 134 kilo molecules. Scientific Data. 2014, 1:140022. 10.1038/sdata.2014.22
5. Faber FA, Hutchison L, Huang B, et al.: Prediction errors of molecular machine learning models lower than hybrid DFT error. Journal of Chemical Theory and Computation. 2017, 13:5255-5264. 10.1021/acs.jctc.7b00577
6. Rupp M, Tkatchenko A, Müller KR, von Lilienfeld OA: Fast and accurate modeling of molecular atomization energies with machine learning. Physical Review Letters. 2012, 108:058301. 10.1103/PhysRevLett.108.058301
7. Schütt KT, Kindermans PJ, Sauceda HE, Chmiela S, Tkatchenko A, Müller KR: SchNet: a continuous-filter convolutional neural network for modeling quantum interactions. Advances in Neural Information Processing Systems. 2017, 30:992-1002. 10.48550/arXiv.1706.08566
8. Gasteiger J, Groß J, Günnemann S: Directional message passing for molecular graphs. International Conference on Learning Representations. 2020, 10.48550/arXiv.2003.03123
9. Ruddigkeit L, van Deursen R, Blum LC, Reymond JL: Enumeration of 166 billion organic small molecules in the Chemical Universe Database GDB-17. Journal of Chemical Information and Modeling. 2012, 52:2864-2875. 10.1021/ci300415d
10. Breiman L: Random forests. Machine Learning. 2001, 45:5-32. 10.1023/A:1010933404324
11. Pedregosa F, Varoquaux G, Gramfort A, et al.: Scikit-learn: machine learning in Python. Journal of Machine Learning Research. 2011, 12:2825-2830. 10.48550/arXiv.1201.0490
12. Paszke A, Gross S, Massa F, et al.: PyTorch: an imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems. 2019, 32:8024-8035. 10.48550/arXiv.1912.01703
13. Fey M, Lenssen JE: Fast graph representation learning with PyTorch Geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds. 2019, 10.48550/arXiv.1903.02428
14. Kipf TN, Welling M: Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations. 2017, 10.48550/arXiv.1609.02907
15. Xu K, Hu W, Leskovec J, Jegelka S: How powerful are graph neural networks?. International Conference on Learning Representations. 2019, 10.48550/arXiv.1810.00826
16. Hamilton WL, Ying R, Leskovec J: Inductive representation learning on large graphs. Advances in Neural Information Processing Systems. 2017, 30:1024-1034. 10.48550/arXiv.1706.02216
17. Holm S: A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics. 1979. 6:65-70.

How to cite this article:
Usman S, Choudhary S, Usman A (November 23, 2026) Graph Construction Determines Whether Graph Neural Networks Outperform Classical Models for Molecular Property Prediction: A Controlled Benchmark. Cureus J Comput Sci 3 : es44389-026-00387-2. DOI https://doi.org/10.7759/s44389-026-00387-2