A Metric for the Balance of Information in Graph Learning

Alex O. Davies[1], Nirav S. Ajmeri[1], Telmo de Menezes e Silva Filho[2]

[1]School of Computer Science, University of Bristol
[2]School of Engineering Mathematics and Technology, University of Bristol
alexander.davies@bristol.ac.uk, nirav.ajmeri@bristol.ac.uk, telmo.silvafilho@bristol.ac.uk

Abstract

Graph learning on molecules makes use of information from both the molecular structure and the features attached to that structure. Much work has been conducted on biasing either towards structure or features, with the aim that bias bolsters performance. Identifying which information source a dataset favours, and therefore how to approach learning that dataset, is an open issue. Here we propose Noise-Noise Ratio Difference (NNRD), a quantitative metric for whether there is more useful information in structure or features. By employing iterative noising on features and structure independently, leaving the other intact, NNRD measures the degradation of information in each. We employ NNRD over a range of molecular tasks, and show that it corresponds well to a loss of information, with intuitive results that are more expressive than simple performance aggregates. Our future work will focus on expanding data domains, tasks and types, as well as refining our choice of baseline model.

Introduction

Graphs are an intuitive way to represent data in many fields of industry and science, including chemistry (Gilmer et al. 2017), infrastructure planning (Khodayar et al. 2019), biology (Li, Huang, and Zitnik 2022) and social network analysis (Davies, Ajmeri, and Silva Filho 2022). Here we focus on molecular graphs, which have seen a great deal of research and use (Gómez-Bombarelli et al. 2018; Khemchandani et al. 2020; Popova et al. 2019). Tasks on molecular graphs typically consist of learning to predict a property, for example solvation energy, from a set of molecular graphs.

Molecular graphs are defined by both structure and features. Graph learning can then be framed as the process of extracting useful information from graphs for a given target. Structure contains atoms and bonds, defining the connections between atoms within the graph. Features then give information about individual atoms and bonds.

The degree to which Graph Neural Network (GNN) models rely on either structural or feature information during learning is highly varied. Some works target structural learning specifically, aiming to improve performance by biasing towards the information encoded in patterns between nodes (Chen, Coskunuzer, and Gel 2021; Horn et al. 2021). Other

works make the point that the optimal solution may not include graph structure at all (Bechler-Speicher et al. 2024). There are currently no metrics for whether a dataset and model combination leans towards structure or feature information, leading to complicated design processes for users.

Here we propose and demonstrate Noise-Noise Ratio Difference (NNRD), a metric for assessing how strongly a dataset relies on either feature-based or structure-based information, with clear applications for chemistry and other fields. Following an intuitive noising process along either features or structure, NNRD gives a bounded and understandable value with which data publishers or users can tailor their approaches from the outset.

Background and Related Work

GNNs, or more specifically Message Passing Neural Networks (MPNNs), parametrise message passing and aggregation, allowing deep-learning techniques to be applied on graph data. For a dataset with features $X \in R^{|V| \times D}$ and similar for edges, structure as an edgelist $E : \{(v_1, v_2), \dots\}$, and graph labels y, graphs are $G : \{X, E\}$. For a molecule, nodes are atoms and edges are bonds, with features giving information on individual atoms and bonds. When a given model aims to learn some mapping $f(G) \to y$, in reality this is $f(X, E) \to y$. The usual graph learning assumption is that topology E and features X must be considered together for information to be useful. In other words their combination has mutual information with a target y, $I(y; X, E)$.

Wu et al. (2020) explore the expressive power of GNNs, introducing the information bottleneck principle. Alon and Yahav (2021) perform similar analysis. The core thesis of both works is that a compromise often must be struck between different information sources in graph learning. More recent works have elaborated on the same theme, often emphasising that structural information can be difficult to effectively incorporate (Wu et al. 2023). Other works instead show that node information can be lost during GNN use, with original feature similarities distorted during message passing and aggregation (Jin et al. 2021). In a similar vein some works explicitly detach features and structure, with performance benefits, despite the loss of information from their inter-relation (Wang et al. 2024).

Clear from the literature is that the balance between structural and feature information, and how to effectively bal-
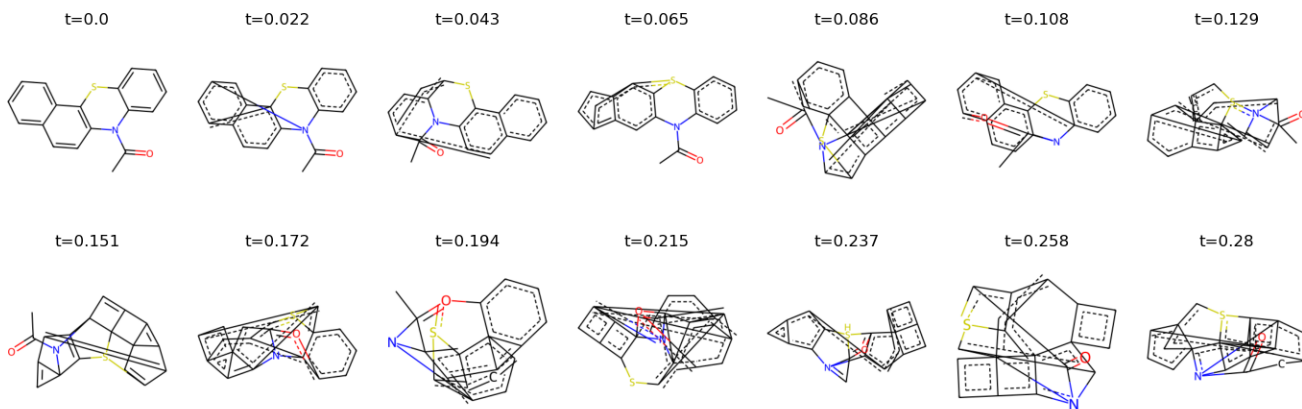
Figure 1: A molecule undergoing structure noise through edge removal and addition. Each noise step is applied on the original molecule, meaning that these examples are not sequential.

ance the use of both, is an open issue. While both issues are present for molecular datasets, they are also present on a diverse range of graph domains. Here we address the former concern, aiming to provide a single metric that describes the balance between useful information sources. Such a metric will allow chemists and other users to iterate more easily and quickly over graph learning implementations.

NNRD: Noise-Noise Ratio Difference

We define NNRD here in the domain-agnostic case. Assume that in the absence of features, structural information is still present, i.e. $I(y; E) \geq 0$, as well as that features contain their own useful information, $I(y; X) \geq 0$. Further we can assume some imbalance between features and structure, $I(y; X) \neq I(y; E)$.

Here we propose a way of investigating this imbalance between feature information and structure information. By degrading the useful information in either (or both of) X and E, the degree to which performance relies on one or the other should be apparent. A caveat here is that degrading either will presumably also degrade the information from their interdependence.

Consider some destructive noising process $N_t(\cdot)$, that degrades the useful information in either X or E. $N_0(x) = x$, with no useful information after $N_T(\cdot)$. Some imperfect model produces predictions $f(X, E) \to \hat{y}$. Performance is some $h(y, \hat{y})$ between real and predicted values. We can then sample $h(y, f(N_X, N_E) \to \hat{y})$, the useful information in both X and E, and whether performance requires both to be present, should be observable.

We propose Noise-Noise Difference Ratio (NNRD), a metric for how much a given task relies on data from features compared to structure. Here we denote $h_X(t)$ the performance of a given model on a task with no structural noise and feature noise at some noising step t. Similarly $h_E(t)$ is performance at structural noise step t with no feature noise.

t and $h_X(t), h_E(t)$ should monotonically increase to-

gether, as information for downstream tasks is necessarily removed by our noising functions. If equal information is in structure and features, they will descend at the same rate. If not, one will fall faster than the other.

From this we can define NNRD for performance metrics:

$$\text{NNRD} = \log\left(\frac{1}{|T|} \sum_t \frac{h_X(t)}{h_E(t)}\right) \quad (1)$$

Here we expect h to increase with increasing performance. NNRD is 0 when features and structure contain equal information, > 0 when structure contains more information, and < 0 when features contain more information. In the same manner, when an error is tracked instead of a performance metric, the ratio term is simply inverted. This makes direct comparison between datasets with different performance targets, for example ROC-AUC and RMSE, possible.

Experiments

We measure NNRD over the graph-level tasks from the Open Graph Benchmark (OGB) (Hu et al. 2020). These are a mix of binary classification, multi-class binary classification, and single-target regression datasets. As some multi-task datasets have missing values for different tasks, we train and report metrics by masking task-wise.

We use three Graph Isomorphism Network (GIN) layers (Xu et al. 2019) as our model, given their guarantees on expressivity, with a hidden dimension of 100 and trained with a learning rate of 0.001. We train each model and noised dataset five times, measuring the mean performance across ten increasing noise levels.

We use random edge removal/addition for structure noise, with an example shown in Figure 1. At t, we remove $|E_r| = p_t \cdot |V|$ (with $0 \leq p_t \leq 1$) edges at random from the graph, $E' = E \setminus E_r$. We then add an equal number of randomly chosen edges $|E_a| = p_t \cdot |V|$, for a final noised edgelist $E'' = E' \cup E_a$. Where edge features are present, they are transferred from removed edges onto newly added edges.

(a) BACE   (b) BBBP   (c) CLINTOX

(d) TOX21   (e) HIV   (f) SIDER

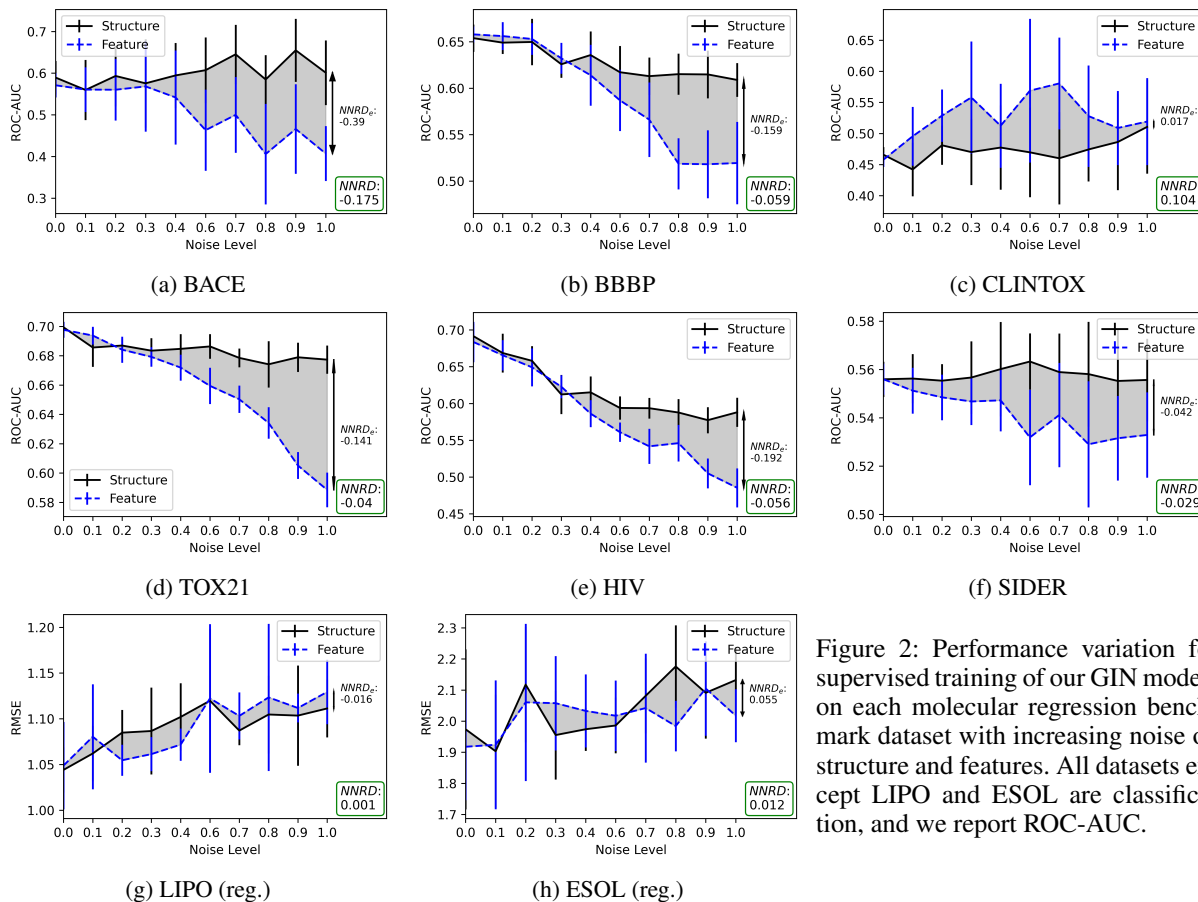(g) LIPO (reg.)   (h) ESOL (reg.)

Figure 2: Performance variation for supervised training of our GIN models on each molecular regression benchmark dataset with increasing noise on structure and features. All datasets except LIPO and ESOL are classification, and we report ROC-AUC.

This is very close to the procedure taken by graph diffusion models as forward noising processes (Vignac et al. 2023). At $t = T$, $p_t = 1$, and the structure is a random graph with the same density as the original.

For feature noise we employ random feature permutation. At t, we completely permute the features of a proportion p_t of nodes across the dataset. Let node (or edge) features across a whole dataset be $X \in \{0, 1\}^{N \times D}$ with N points and D features. At t we randomly select $N_f = p_t \cdot N$ node to permute. The features of the selected nodes are then randomly permuted, across the whole dataset, removing the useful feature information from the selected nodes, while maintaining marginal distributions.

Results

We show visualisations of increasing structure and feature noise, and its influence on model performance, in Figure 2. Here we show NNRD, and also NNRD_e, which is calculated only at the extreme noise level $t = T$. Here we see a variety of trends, varying by datasets. On the bbbp and tox21 datasets there is a reliance on feature-based information, with scores dropping a large margin as feature noise increases, but structure noise having little effect.

On other datasets, such as the esol dataset, performance seems to decrease at an equal rate when structure and feature noise increase. On the bace dataset, interestingly, the performance of the model actually increases with structure noise. This may be due to additional long-range connections from edge swapping.

Notably, only on the clintox dataset do we see a preference for structural information over feature information. Oddly performance actually increases as more noise is applied to features. Performance over structure noises is close to constant, as on other datasets. The likely conclusion here is that the model is simply unable to learn the dataset usefully enough to accurately predict the test set.

In Table 1 we record quantitative metrics. Here the same

| Dataset | t=0 | $t = T$ | | NNRD |
		Structure	Feature	
bace	0.575	0.601	0.407	-0.175
bbbp	0.658	0.609	0.520	-0.059
clintox	0.466	0.511	0.519	0.104
tox21	0.700	0.677	0.589	-0.040
hiv	0.684	0.588	0.485	-0.056
sider	0.556	0.556	0.533	-0.029
lipo	1.05	1.11	1.13	0.001
esol	1.94	2.13	2.02	0.012

Table 1: Results from training over molecular datasets.

trends as discussed from Figure 2 are present, although the simple aggregate metrics we report are of a lower fidelity than the visualisations. Crucially, in some cases, these aggregates are be misleading.

Taking the esol dataset as an example, the performance achieved at maximum noise values would indicate that structural information plays a more significant part than feature information. However, from Figure 2 we can see that in-fact they increase at very similar rates, indicating a near-balance in useful information contained by each. NNRD instead reports a value very close to zero, indicating near-parity between the information in structure and features.

Discussion

While NNRD corresponds with the visualisations in Figure 2, there are some cases where it fails. The critical example here is the clintox dataset, where the test set deviates significantly from the train set, and performance on the test set is consistently worse-than-random as a result. That said, this dataset is constructed as a challenging benchmark, and on other datasets NNRD is a useful quantitative measure.

An obvious point is that NNRD, in its current form, varies on a per-model basis. This is a difficult obstacle to overcome, in that there is not currently a de-facto best model to use across all datasets. Instead, by fixing the model used, NNRD could be reported on a per-model basis. That said, the same trends would likely be present with other models.

We envision use-cases mainly in data publishing. At the time of publishing, NNRD can be reported for a few fixed models, giving an indication to users of whether to design for structure or feature information.

Conclusion & Future Work

We have proposed a metric Noise-Noise Ratio Difference (NNRD) to evaluate whether the useful information on a given graph task weighs towards structure or features. By implementing additive noise functions over structure and features, and making use of OGB datasets, we have demonstrated the efficacy of NNRD. Our future work will follow several key directions:

Wider range of datasets By including datasets from a wider range of domains, types and tasks, instead of only molecule graphs here, we can establish whether NNRD is also useful outside of chemistry.

Model selection We plan to refine and tailor the models we use to calculate NNRD. Though a one-size-fits-all method is unlikely, a selection of different graph model approaches - for example linear models with random kernels - might provide adequate coverage.

References

Alon, U.; and Yahav, E. 2021. On the Bottleneck of Graph Neural Networks and its Practical Implications. ArXiv:2006.05205.

Bechler-Speicher, M.; Amos, I.; Gilad-Bachrach, R.; and Globerson, A. 2024. Graph Neural Networks Use Graphs When They Shouldn't. ArXiv:2309.04332.

Chen, Y.; Coskunuzer, B.; and Gel, Y. R. 2021. Topological relational learning on graphs. *Advances in Neural Information Processing Systems*, 34: 27029–27042.

Davies, A.; Ajmeri, N.; and Silva Filho, T. 2022. Realistic synthetic social networks with graph neural networks. *arXiv Pre-Print*, 1(1): 1–12. Tex.arxivid: 2212.07843.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. *In Proceedings of the 34th International Conference on Machine Learning, (ICML) 2017*, 3: 2053–2070. ISBN: 9781510855144 Publisher: International Machine Learning Society (IMLS) tex.arxivid: 1704.01212.

Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; and Aspuru-Guzik, A. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2): 268–276. Publisher: American Chemical Society tex.arxivid: 1610.02415.

Horn, M.; De Brouwer, E.; Moor, M.; Moreau, Y.; Rieck, B.; and Borgwardt, K. 2021. Topological graph neural networks. *ICLR 2022 - 10th International Conference on Learning Representations*. Publisher: International Conference on Learning Representations, ICLR tex.arxivid: 2102.07835.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J.; Barzilay, R.; Battaglia, P.; Bengio, Y.; Bronstein, M.; Günnemann, S.; Hamilton, W.; Jaakkola, T.; Jegelka, S.; Nickel, M.; Re, C.; Song, L.; Tang, J.; Welling, M.; and Zemel, R. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems*, 33: 22118–22133.

Jin, W.; Derr, T.; Wang, Y.; Ma, Y.; Liu, Z.; and Tang, J. 2021. Node Similarity Preserving Graph Convolutional Networks. ArXiv:2011.09643.

Khemchandani, Y.; O'Hagan, S.; Samanta, S.; Swainston, N.; Roberts, T. J.; Bollegala, D.; and Kell, D. B. 2020. DeepGraphMolGen, a multi-objective, computational strategy for generating molecules with desirable properties: a graph convolution and reinforcement learning approach. *Journal of cheminformatics*, 12(1). Publisher: J Cheminform.

Khodayar, M.; Wang, J.; Wang, Z.; Khodayar, M.; Wang, J.; and Wang, Z. 2019. Deep generative graph distribution learning for synthetic power grids. *arXiv*, arXiv:1901.09674. Tex.arxivid: arXiv:1901.09674.

Li, M. M.; Huang, K.; and Zitnik, M. 2022. Graph representation learning in biomedicine and healthcare. *Nature Biomedical Engineering*, 6(12): 1353–1369. Publisher: Nature Research.

Popova, M.; Shvets, M.; Oliva, J.; Isayev, O.; Popova, M.; Shvets, M.; Oliva, J.; and Isayev, O. 2019. MolecularRNN: Generating realistic molecular graphs with optimized properties. *arXiv*, arXiv:1905.13372. Tex.arxivid: arXiv:1905.13372.

Vignac, C.; Krawczuk, I.; Siraudin, A.; Wang, B.; Cevher, V.; and Frossard, P. 2023. DiGress: Discrete Denoising dif-

fusion for graph generation. In *The eleventh international conference on learning representations*.

Wang, J.; Guo, J.; Sun, Y.; Gao, J.; Wang, S.; Yang, Y.; and Yin, B. 2024. DGNN: Decoupled Graph Neural Networks with Structural Consistency between Attribute and Graph Embedding Representations. ArXiv:2401.15584.

Wu, F.; Li, S.; Wu, L.; Radev, D.; and Li, S. Z. 2023. Discovering and Explaining the Representation Bottleneck of Graph Neural Networks from Multi-order Interactions. ArXiv:2205.07266.

Wu, T.; Ren, H.; Li, P.; and Leskovec, J. 2020. Graph Information Bottleneck. In *Advances in Neural Information Processing Systems*, volume 33, 20437–20448. Curran Associates, Inc.

Xu, K.; Jegelka, S.; Hu, W.; and Leskovec, J. 2019. How powerful are graph neural networks? In *7th international conference on learning representations, ICLR 2019*. International Conference on Learning Representations, ICLR. Tex.arxivid: 1810.00826.