On the Scalability of GNNs for Molecular Graphs

Anonymous Authors ANONYMOUS EMAIL
Anonymous address

Reviewed on OpenReview: *https://openreview.net/forum?id=XXXX*

Abstract

Scaling deep learning models has been at the heart of recent revolutions in language modelling and image generation. Practitioners have observed a strong relationship between model size, dataset size, and performance. However, structure-based architectures such as Graph Neural Networks (GNNs) are yet to show the benefits of scale mainly due to the lower efficiency of sparse operations, large data requirements, and lack of clarity about the effectiveness of various architectures. We address this drawback of GNNs by studying their scaling behavior. Specifically, we analyze message-passing neural networks, graph transformers, and hybrid architectures on the largest public collection of 2D molecular graphs. For the first time, we observe that GNNs benefit tremendously from the increasing scale of width, depth, number of molecules, number of labels, and the diversity in the pretraining datasets, resulting in a 30.25% improvement when scaling to 1 billion parameters and 28.98% improvement when increasing size of dataset to eightfold. We further demonstrate strong finetuning scaling behavior on 34 tasks, outclassing previous large models. We hope that our work will pave the way for an era where foundational GNNs drive pharmaceutical drug discovery.

Keywords: Graph Neural Networks, Scaling Laws, Molecular Biology.

1 Introduction

Recent successes in language modelling (OpenAI, 2023; Touvron et al., 2023) and image generation (Ramesh et al., 2021; Rombach et al., 2022) are driven by the increasing amount of training data and computational resources. Across different domains, practitioners have observed a direct relationship between model parameter count and performance on novel tasks (Kaplan et al., 2020). In natural language processing, large transformer based models have demonstrated impressive generalization capabilities utilizing a causal autoregressive objective (Radford et al., 2019). In the meantime, image generation has undergone incredible leaps with large models trained utilizing pixel level unsupervised objectives.

While data power law scaling behavior has been tremendously beneficial in language and image domains, its practical impact on molecular reasoning and drug discovery has remained limited. This is a direct consequence of complex scientific tasks requiring reasoning regarding the underlying structure of the data (Bubeck et al., 2023). In the past, molecular property prediction approaches have made use of graph-based methods, as these allow us to reason about the structure and interaction of different components of a molecule. Molecules are naturally represented as graphs, where the nodes represent atoms and edges correspond to covalent bonds between the atoms.

Graph Neural Networks (GNNs) have emerged as a promising way of learning molecular representations (Liu et al., 2023a; Galkin et al., 2023). GNN architectures are equipped with

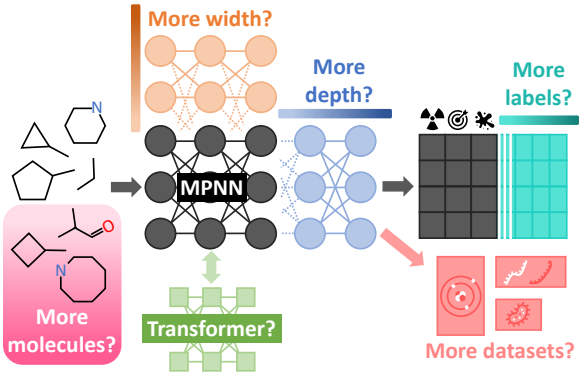


Figure 1: Summary of our GNN scaling hypotheses studied in the present work. The baseline model is presented in **dark grey**, followed by different scaling hypotheses illustrated in lighter colors. We analyze the scaling behavior of message-passing networks, graph transformer and hybrid architectures with respect to the increasing scale of width, depth, number of molecules, number of labels and diversity of datasets.

the flexibility of learning molecular structure while building general, powerful representations over graphs utilizing backpropagation. These representations have been utilized in various paradigms such as reasoning about drug properties (Stärk et al., 2022a), target binding interactions (Stärk et al., 2022b), retrosynthesis of reagents (Liu et al., 2020), ligand-based docking (Corso et al., 2023) and in-silico experiment design (Wang et al., 2023b).

Despite the growing applicability of GNNs in molecular tasks, the lack of supervised data has significantly hindered our ability to proportionally scale model sizes. It remains unclear whether graph-based architectures hold the promise of scale, similar to the paradigms of language and unsupervised image generation.

Learning molecular properties with GNNs presents its own set of unique challenges. First, multiple different GNN architectures are being actively researched. These include convolution (Kipf and Welling, 2017), message passing architectures (Beaini et al., 2021), graph transformers (Ying et al., 2021b) and hybrid architectures (Rampášek et al., 2022; Masters et al., 2022). These approaches have shown recent progress, but their applicability to practical applications remains an open question (Rong et al., 2020).

Second, the commonly used self-supervised training techniques do not transfer well when applied to molecular graphs; e.g., retrieving masked bonds and atoms is not informative. This is primarily due to large data requirements and the fact that graphs are limited in capturing domain-specific aspects such as chemical interactions and biological compositions (Liu et al., 2022). Other methods such as GPSE (Liu et al., 2023b) solely learns the graph structure, thus providing a better positional encoding for another GNN.

Lastly, public datasets have insufficient high-quality data for effective GNN architecture training. While recent attempts have been made to expand open-source datasets (Beaini et al., 2024), their extensions towards multi-task settings remain an open question.

We aim to address these limitations and provide a concrete understanding of the required data and architectures to build foundational models for molecular graphs. Specifically, we want to answer the question: *How do graph-based architectures scale in multi-task settings of large molecular graphs?*

As summarized in Figure 1, we aim to answer the above question by studying the scaling behavior of 2D molecular GNNs under different settings of width, depth, number of molecules, number of labels, and the diversity in datasets. We analyze message-passing networks, graph transformers, and hybrid architectures on the largest public collection of 2D molecular graphs. All models are tested in 2 different settings; (1) pre-training performance for randomly split train and test sets and (2) fine-tuning of pre-trained models on downstream tasks on standard benchmarks.

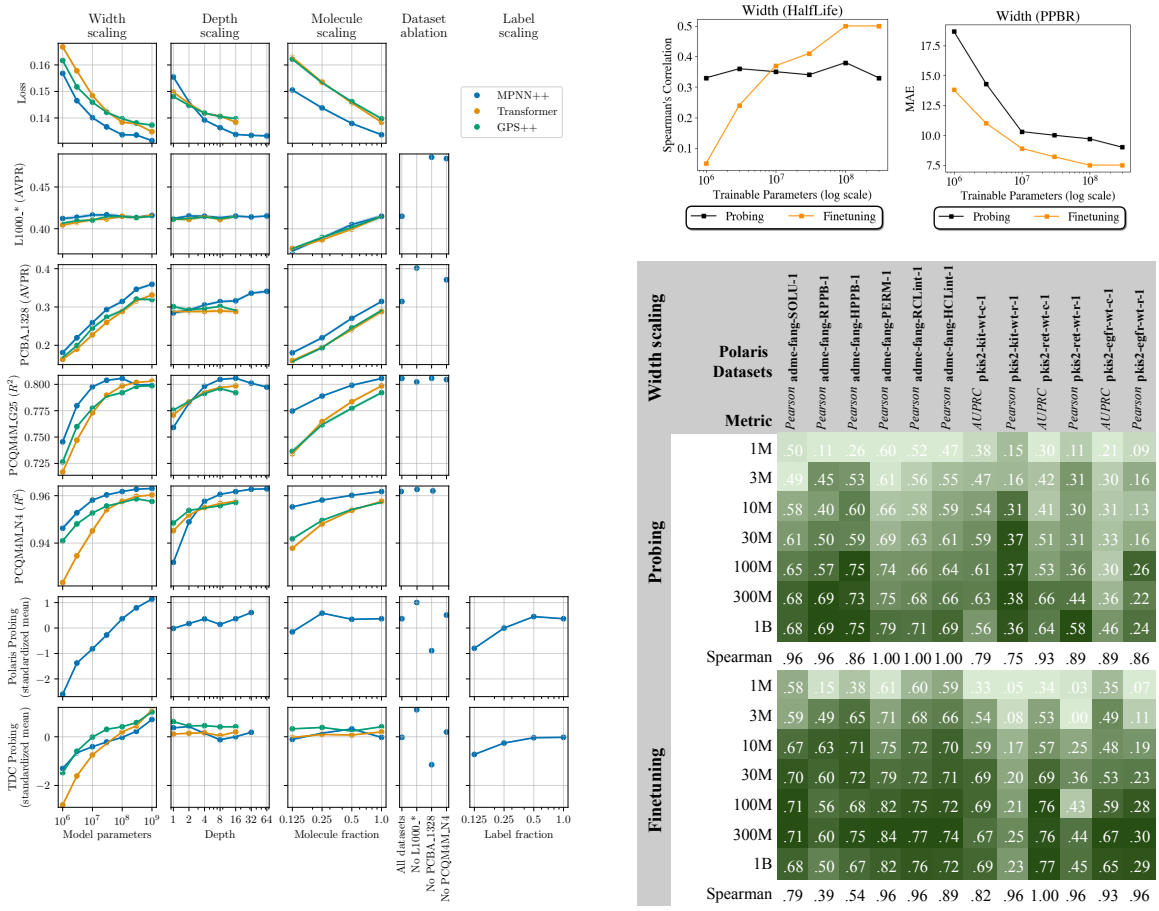
Our work aims to provide a proper understanding of how different GNN architectures scale for molecular GNNs and how it affects their performance in various settings. Our main contributions are as follows:

- We study the scaling behavior of 2D molecular GNNs under varied settings of width, depth, number of molecules, number of labels, the diversity in dataset and the architectural choice.
- We show that our largest 1 billion parameter models continue to scale with constant gains in molecular property prediction. To the best of our knowledge, this is the first work to demonstrate the continuous scaling behavior of molecular GNNs.
- We show that supervised pretraining over molecular graphs provides a rich fingerprint embedding, useful for MLP probing, and more expressive as we scale the model and datasets.
- We provide an in-depth analysis of scaling trends across different probing and finetuning strategies. Specifically, we observe that model width and number of labels are the most important factors driving finetuning performance.
- Finally, we show that, simply by scaling the width until 1B parameters, our largest model outperforms other pre-trained models, while consistently achieving parity with the state-of-the-art specialized models across a wide variety of tasks.

2 How Do Molecular GNNs Scale?

Our study aims to answer the question *How do molecular GNNs scale?* We begin by studying GNNs in the multi-task supervised pretraining setup. Since our analysis consists of multiple tasks on a large scale, we utilize the `Graphium` library (Beaini et al., 2024). Due to the absence of a unified consensus on the best architecture for molecular GNNs, we focus our efforts on three specific models. We select MPNN++ (Masters et al., 2022) which improves quantum prediction over the MPNN (Gilmer et al., 2017), Graph Transformers (Müller et al., 2023), and Hybrid GPS++ (Masters et al., 2022) along with the use of positional encodings. Finally, we evaluate our models on a range of public benchmarks with 34 datasets from Huang et al. (2021) and Polaris (Anonymous, 2024). Our study trains models in both finetuning and probing (fingerprinting) settings.

In the supplementary material, we further provide a detailed description of the datasets and benchmarks used for pre-training and finetuning (Sections A.1 and A.2) and of the choices of architectures (Section A.3). Finally, we discuss the utilized finetuning and probing strategies in Section A.4.



(a)

	Polaris Datasets	adme-fang-SOLU-1	adme-fang-RPPB-1	adme-fang-HPPB-1	adme-fang-PERM-1	adme-fang-RCLint-1	adme-fang-HCLint-1	pkis2-kit-wt-c-1	pkis2-kit-wt-r-1	pkis2-ret-wt-c-1	pkis2-ret-wt-r-1	pkis2-egfr-wt-c-1	pkis2-egfr-wt-r-1
	Metric	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*
Width scaling													
Probing	1M	.50	.11	.26	.60	.52	.47	.38	.15	.30	.11	.21	.09
	3M	.49	.45	.53	.61	.56	.55	.47	.16	.42	.31	.30	.16
	10M	.58	.40	.60	.66	.58	.59	.54	.31	.41	.30	.31	.13
	30M	.61	.50	.59	.69	.63	.61	.59	.37	.51	.31	.33	.16
	100M	.65	.57	.75	.74	.66	.64	.61	.37	.53	.36	.30	.26
	300M	.68	.69	.73	.75	.68	.66	.63	.38	.66	.44	.36	.22
	1B	.68	.69	.75	.79	.71	.69	.56	.36	.64	.58	.46	.24
	Spearman	.96	.96	.86	1.00	1.00	1.00	.79	.75	.93	.89	.89	.86
Finetuning	1M	.58	.15	.38	.61	.60	.59	.33	.05	.34	.03	.35	.07
	3M	.59	.49	.65	.71	.68	.66	.54	.08	.53	.00	.49	.11
	10M	.67	.63	.71	.75	.72	.70	.59	.17	.57	.25	.48	.19
	30M	.70	.60	.72	.79	.72	.71	.69	.20	.69	.36	.53	.23
	100M	.71	.56	.68	.82	.75	.72	.69	.21	.76	.43	.59	.28
	300M	.71	.60	.75	.84	.77	.74	.67	.25	.76	.44	.67	.30
	1B	.68	.50	.67	.82	.76	.72	.69	.23	.77	.45	.65	.29
	Spearman	.79	.39	.54	.96	.96	.89	.82	.96	1.00	.96	.93	.96

(b)

Figure 2: **(a)** Effect of scaling different design choices (columns) to model evaluation performance (rows). The *standardized mean* is calculated as mean of standardized scores for every task in a dataset group, i.e. a mean and standard deviation per task were calculated based on all our models in this study (signs of tasks with *lower is better* metrics were flipped). **(b)** – **top** Scaling trend for two selected downstream tasks from the TDC benchmark collection. **(b)** – **bottom** Finetuning and probing performance of pretrained MPNNs across widths on the Polaris benchmark. **Darker green** shades denote better performance. Larger models tend to perform better across different metrics on unseen tasks. Spearman correlation values closer to 1 indicate that predictive performance correlates with larger model sizes.

2.1 Scaling Trends during Pretraining

In this section, we evaluate the scaling behaviour of our models according to various parameters summarized in Figure 1, namely the architectural choice, width, depth, number of molecules, labels and different datasets. We analyze our models on datasets from LargeMix described in Section A.1. For additional results and experiments of our study, please refer to the supplementary material.

Overview. Figure 2a presents the variation of architectural choices between MPNN++, GPS++ (Hybrid) and Transformer, with training curves and full fine-tuning results in the supplemental material (Sections G and H). Notably, all models scale favourably with the increasing scale of width (number of parameters), depth (number of GNN layers) and number of molecules (dataset size). Models are not significantly affected by the number of labels, a positive outcome indicating weight sharing between labels. In general, all models follow monotonic trends, with differences partly occurring due to hyperparameter choices.

MPNN vs Transformer. MPNNs are found to be more parameter efficient as they perform better with small width and depth compared to Transformers. MPNNs are also data efficient as they perform significantly better when sub-sampling the quantum PCQM4M_* datasets, although molecular scaling is similar for biological datasets. Transformers being data-hungry is consistent with recent works in other domains such as natural language and computer vision (Radford et al., 2018; Alayrac et al., 2022; Galkin et al., 2023). The hybrid GPS++ architecture seems to benefit from the MPNN expressivity in low-parameter regimes, while also exhibiting a similar molecular-scaling to the transformer in low-data regimes. Finally, we notice that MPNNs are unsurprisingly more affected by depth scaling and improve (contrarily to transformers) as their receptive field depends on the number of layers.

Width scaling. As seen in the first column of Figure 2a, increasing the width has a significant impact on model performance across all tasks. Further, we trained our models for fewer epochs as they were more likely to experience overfitting on the PCQM4M_* tasks.

Depth scaling. Similar to width, depth of GNN models plays an important role in the dataset fit during test time. Deeper models with larger layers capture intricate aspects of the data resulting in 12.5 % improvement in test error. However, performance plateaus at around 8-16 layers for quantum datasets, but could be mitigated by larger datasets. For PCBA_1328, the performance continues to increase.

Molecule scaling. Unsurprisingly, the number of molecules in the training set correlates strongly with the performance of all models. Contrary to width and depth, molecule scaling is consistent across all models and test sets, although Transformer benefit more on quantum tasks. For instance, increasing the dataset size by eight-fold (12.5 % to 100 %) yields a significant 33.33 % improvement in model performance in the case of the hybrid GPS++ model.

2.2 Scaling Trends on Downstream Tasks

We now evaluate scaling of models when finetuning and probing on downstream tasks. As detailed in Section A.4, both approaches make use of the head layer trimming strategy to trim and append new task heads. In the case of finetuning, all weights are tuned, while for fingerprinting the pretrained model is held fixed. Due to the large number of tasks spread across the 34 tasks, we limit our evaluation to probing for most experiments, except for MPNN++ where we also finetune the model.

In order to summarize scaling trends, we compute the Spearman's rank correlation coefficient (Schober et al., 2018) between model performance on a given metric and the scale of model/data used. The correlation is given by a value in the range $[-1, 1]$, with a value of 1 indicating perfect scaling (larger model/dataset is preferred), -1 indicating imperfect

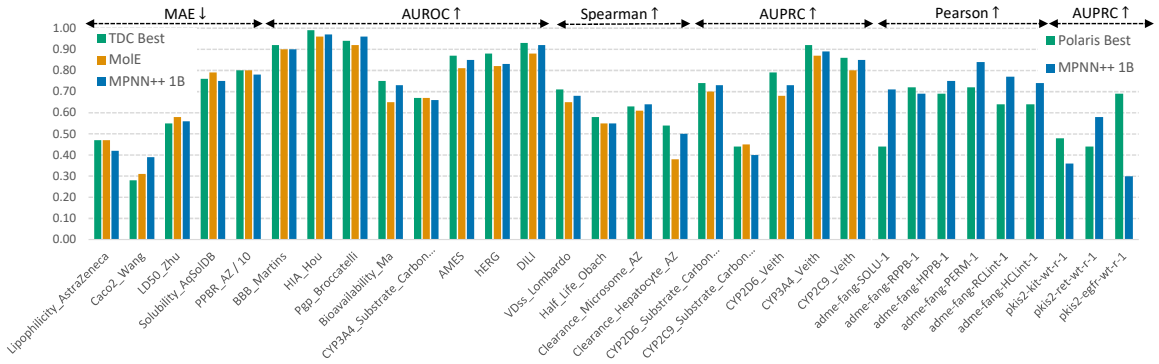


Figure 3: Comparison of our 1B MPNN++ model to the SOTA across TDC **(left)** and Polaris **(right)** ADMET benchmarks.

scaling (smaller model/dataset is preferred) and 0 indicating no correlation. We note that this evaluation scheme, although statistical, aims to answer the question *What kind of design decisions are necessary to build a foundational model for molecular representations?*

MPNN vs Transformer. For probing on downstream tasks, we study the effect of architecture choices of width, depth, and number of molecules. We find that transformers benefit from increased width on downstream tasks compared to GPS++ and MPNN++. Surprisingly, despite the number of molecules having a stronger impact on all model's performance, it only slightly impacts the downstream performance of all models, with a small benefit to MPNN++. Finally, Transformer is the only model with a *small* positive trend for depth scaling, while GPS++ and MPNN++ have close to no trend.

Width scaling. We evaluate width scaling on Polaris and TDC datasets in Figures 2b (bottom) and Figure 14 (Section H.6 in supplemental material). We observe linear scaling trends on all Polaris datasets, with an average spearman correlation of 0.91 during probing and 0.85 during finetuning. On TDC, a similar trend is observed with a strong correlation of > 0.75 for 15/22 datasets during probing and 17/22 during finetuning (using the MPNN++). These results strongly indicate the benefits of larger pre-trained GNNs for downstream tasks, a result consistent with prior findings in scaling of large models (Kaplan et al., 2020). Additionally, we note that finetuning mostly leads to even better performance at different scales when compared to naive probing.

Depth scaling. We evaluate the scaling of depth of MPNN models on TDC and Polaris benchmarks in Figures 8 and 16. On average, we observe a weak positive trend, with a scaling spearman correlation of 0.47 on Polaris, no significant trend of -0.11 on TDC during probing, but a positive trend with 0.33 correlation when finetuning. While some datasets strongly benefit from deeper networks, others strongly deteriorate. Most TDC datasets are randomly affected. We conjecture that degradation with depth is related to the oversmoothing issue described in supplemenary material (Section E). Certain molecular properties can be well predicted only from small local substructures, hence eliminating the need for long-range interactions that deeper networks enable. We further find that transformers scale better with depth having probing correlations of 0.26.

Molecule scaling. In this setting, we randomly sub-sample a number of molecules in the training set by 12.5%, 25% and 50% to study its effect on downstream tasks. Surprisingly,

probing and finetuning performance does not correlate strongly with the amount of molecules in the training set, as reported in Figures 10 and 17. We observe spearman correlations of 0.28 and 0.32 when probing and finetuning on TDC respectively. In the case of Polaris, mean correlation is 0.3. The globally weak positive trend depends on the downstream dataset, with many strong correlations and a few strong negative correlations. Contrarily to their stronger trends on the pre-training tasks, transformer and GPS++ have lower correlations during probing (0.13 and 0.14, respectively).

Label scaling. We now study the effect of target labels by randomly sub-sampling the number of labels of each dataset in the training set by 12.5%, 25% and 50%. In Figures 12 and 18, we observe large spearman correlations of 0.56 on Polaris and 0.54 on TDC between the ratio of training labels and the performance, and only a few negative correlations. In the finetuning regime, this number lowers to 0.37 on TDC. These stronger correlations put *label scaling* as the second-best strategy for improving the model's downstream performance.

Dataset importance. We further conducted a study to determine the importance of models in two ways. Firstly, we re-train models without specific datasets. Secondly, we probe models specifically from certain task head MLPs compared to the base GNN model. Observing the dataset ablations in Figure 19, we see that PCBA_1328 is the most beneficial to a model's performance while L1000_* could actually hinder the performance on certain tasks. Observing task head probing in Figure 20, we see that probing from the common part of the GNN outperformed all MLP task-heads most of the time. Further, we note that the PCBA_1328 head is again the most beneficial, possibly due to synergies from pretraining on a bio assay dataset, while the PCQM4M_G25 head is found to be the worst. This is expected since the downstream tasks are dissimilar from pretraining quantum datasets.

2.3 Comparison to other models

We now compare our largest best-performing model to strong existing baselines. In Figure 3, we notice that our MPNN++ consistently outperforms the MolE foundational model (Méndez-Lucio et al., 2022), a gold standard for molecular property prediction. Furthermore, MPNN++ performs on-par with state-of-the-art methods on majority of TDC tasks. Finally, when comparing the MPNN++ to previous best metrics on the Polaris benchmark, we note that our model is significantly better by large margins. We primarily attribute the large-scale success of our model to purely scaling its width only up to 1B parameters, despite the current trends suggesting us to scale further.

3 Conclusion

In this paper, we studied the scalability of GNN models including message-passing networks, graph transformers and hybrid architectures on the largest public collection of 2D molecules for the tasks of molecular property prediction. We showed major performance gains from the growing amount of parameters, data and compute, both on the original test set and on downstream finetuning. Importantly, our models benefit tremendously from the increasing scale of width, number of molecules and number of labels. Our largest 1 billion parameter models, including MPNN++, Transformer, and GPS++, continue to scale favourably resulting in peak 60 % improvement across precision and correlation metrics. More importantly, we demonstrate a consistent performance improvement on downstream property prediction

tasks. Finetuned 1B parameter models consistently produce results on par with specialized state-of-the-art methods. We hope that our work paves the way for the development of foundational GNNs and new architectures with applications in pharmaceutical advancements and drug discovery.

3.1 Future Work

While our study demonstrates the benefits of increasing number of parameters far greater than prior work, there are still orders of magnitude before we reach a general-purpose foundational model of molecules. Our analysis is restricted to the effect of number of parameters and molecules during pretraining and finetuning stages. Future work would aim to uncover additional aspects of GNN training such as the increasing complexity of aggregation functions and their effect on scaling properties.

References

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. *arXiv preprint arXiv:2301.03728*, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Anonymous. Polaris: An industry-led benchmarking initiative. 2024.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*, 2021.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Dominique Beaini, Saro Passaro, Vincent Létourneau, Will Hamilton, Gabriele Corso, and Pietro Liò. Directional graph networks. In *International Conference on Machine Learning*, pages 748–758. PMLR, 2021.

Dominique Beaini, Shenyang Huang, Joao Alex Cunha, Gabriela Moisescu-Pareja, Oleksandr Dymov, Samuel Maddrell-Mander, Callum McLean, Frederik Wenkel, Luis Müller, Jama Hussein Mohamud, et al. Towards foundational models for molecular learning on large-scale multi-task datasets. *ICLR 2024*, 2024.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws. *arXiv preprint arXiv:2210.14891*, 2022.

Kaidi Cao, Phitchaya Mangpo Phothilimthana, Sami Abu-El-Haija, Dustin Zelle, Yanqi Zhou, Charith Mendis, Jure Leskovec, and Bryan Perozzi. Learning large graph property prediction via graph segment training. *arXiv preprint arXiv:2305.12322*, 2023.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829, June 2023.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking, 2023.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv*, 2023.

Fernando Diaz and Michael Madaio. Scaling laws do not scale. *arXiv preprint arXiv:2307.03201*, 2023.

Gbètondji JS Dovonon, Michael M Bronstein, and Matt J Kusner. Setting the record straight on transformer oversmoothing. *arXiv preprint arXiv:2401.04301*, 2024.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. *CoRR*, abs/2110.07875, 2021.

Christopher Fifty, Jure Leskovec, and Sebastian Thrun. In-context learning for few-shot molecular property prediction. *arXiv preprint arXiv:2310.08863*, 2023.

Elias Frantar, Carlos Riquelme, Neil Houlsby, Dan Alistarh, and Utku Evci. Scaling laws for sparsely-connected foundation models. *arXiv preprint arXiv:2309.08520*, 2023.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. *arXiv preprint arXiv:2310.04562*, 2023.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

Jonathan Godwin, Michael Schaarschmidt, Alexander Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. Simple gnn regularisation for 3d molecular property prediction & beyond. *arXiv preprint arXiv:2106.07971*, 2021.

Peter Grindrod. Scaling laws for properties of random graphs that grow via successive combination. 03 2022.

Zhichun Guo, Chuxu Zhang, Wenhao Yu, John Herr, Olaf Wiest, Meng Jiang, and Nitesh V Chawla. Few-shot graph learning for molecular property prediction. In *Proceedings of the web conference 2021*, pages 2559–2567, 2021.

William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016b.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer, 2021.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021.

Pražnikar J. Scaling laws of graphs of 3d protein structures. *J Bioinform Comput Biol.*, 2021.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Cheng-Hao Liu, Maksym Korablyov, Stanisław Jastrzebski, Paweł Włodarczyk-Pruszyński, Yoshua Bengio, and Marwin HS Segler. Retrognn: Approximating retrosynthesis by graph neural networks for de novo drug design. *arXiv preprint arXiv:2011.13042*, 2020.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829*, 2023a.

Renming Liu, Semih Cantürk, Olivier Lapointe-Gagné, Vincent Létourneau, Guy Wolf, Dominique Beaini, and Ladislav Rampášek. Graph positional and structural encoder, 2023b.

Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and S Yu Philip. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5879–5900, 2022.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), September 2022. ISSN 1477-4054. doi: 10.1093/bib/bbac409.

Yizhen Luo, Kai Yang, Massimo Hong, Xing Yi Liu, and Zaiqing Nie. Molfm: A multimodal molecular foundation model, 2023.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pages 1–8, 2023.

Dominic Masters, Josef Dean, Kerstin Klaser, Zhiyi Li, Sam Maddrell-Mander, Adam Sanders, Hatem Helal, Deniz Beker, Ladislav Rampášek, and Dominique Beaini. Gps++: An optimised hybrid mpnn/transformer for molecular property prediction. *arXiv preprint arXiv:2212.02229*, 2022.

Michael Moret, Lukas Friedrich, Francesca Grisoni, Daniel Merk, and Gisbert Schneider. Generative molecular design in low data regimes. *Nature Machine Intelligence*, 2(3): 171–180, 2020.

Michael Moret, Irene Pachon Angona, Leandro Cotos, Shen Yan, Kenneth Atz, Cyrill Brunner, Martin Baumgartner, Francesca Grisoni, and Gisbert Schneider. Leveraging molecular structure and bioactivity with chemical language models for de novo drug design. *Nature Communications*, 14(1):114, 2023.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to graph transformers. *arXiv preprint arXiv:2302.04181*, 2023.

Oscar Méndez-Lucio, Christos Nicolaou, and Berton Earnshaw. Mole: a molecular foundation model for drug discovery, 2022.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978, 2023.

OpenAI. Gpt-4 technical report, 2023.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ICML 2021*, 2021.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

Roshan M Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *bioRxiv*, 2020. doi: 10.1101/2020.12.15.422761.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CVPR 2022*, 2022.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.

James E Saal, Scott Kirklin, Muratahan Aykol, Bryce Meredig, and Christopher Wolverton. Materials design and discovery with high-throughput density functional theory: the open quantum materials database (oqmd). *Jom*, 65:1501–1509, 2013.

Patrick Schober, Christa Boer, and Lothar A Schwarte. Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia*, 126(5):1763–1768, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò. 3d infomax improves gnns for molecular property prediction. *ICML 2022*, 2022a.

Hannes Stärk, Octavian-Eugen Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. *ICML 2022*, 2022b.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*, 2021.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Serbulent Unsal, Heval Atas, Muammer Albayrak, Kemal Turhan, Aybar C Acar, and Tunca Doğan. Learning functional properties of proteins with language models. *Nature Machine Intelligence*, 4(3):227–245, 2022.

Henrike Veith, Noel Southall, Ruili Huang, Tim James, Darren Fayne, Natalia Artemenko, Min Shen, James Inglese, Christopher P Austin, David G Lloyd, et al. Comprehensive characterization of cytochrome p450 isozyme selectivity across chemical libraries. *Nature biotechnology*, 27(11):1050–1055, 2009.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR 2018*, 2017.

Patrick Walters. We need better benchmarks for machine learning in drug discovery, 2023. 2024-01-18.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023a.

Yuyang Wang, Zijie Li, and Amir Barati Farimani. *Graph Neural Networks for Molecules*, page 21–66. Springer International Publishing, 2023b. ISBN 9783031371967. doi: 10.1007/978-3-031-37196-7_2.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ICLR*, 2019.

Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. *ICLR*, 2021.

Minkai Xu, Meng Liu, Wengong Jin, Shuiwang Ji, Jure Leskovec, and Stefano Ermon. Graph and geometry generative modeling for drug discovery. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5833–5834, 2023.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer, 2022.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor programs vi: Feature learning in infinite-depth neural networks, 2023.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021a.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021b.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.

Yang Yuan. On the power of foundation models. In *International Conference on Machine Learning*, pages 40519–40530. PMLR, 2023.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*, 2023.

Appendix A. Dataset and Architecture Details

A.1 Datasets

We study the scaling behavior of GNNs on the LargeMix dataset mixture (Beaini et al., 2024). These datasets cover different types of molecules exhibiting variable properties. Thus, the training is done in a multi-task setting consisting of thousands of labels, some being very imbalanced and very sparse, a challenging scenario for learning representations with GNNs.

The LargeMix dataset mixture consists of 5 million molecules grouped into 5 different tasks with each task having multiple labels. A moderate number of compounds spread across multiple tasks and labels, making this dataset suitable for pretraining large GNNs. Below is a description of the individual tasks contained within LargeMix.

- **L1000_VCAP and L1000_MCF7** are two datasets of 16k and 20k molecules, respectively, with 998 graph-level classification labels corresponding to transcriptomics changes in the cell when exposed to drugs.

- **PCBA_1328** is a dataset of 1.6M molecules with 1328 binary classification labels. Each label corresponds to the activity tags of the molecules in a bioassay reported on pubchem.

- **PCQM4M_G25 and PCQM4M_N4** are two datasets of 3.8M molecules with 25 graph-level labels and 4 node-level labels. Labels are obtained using Density Functional Theory (DFT) simulations, a highly accurate quantum simulation method (Saal et al., 2013).

A.2 Finetuning and Probing Benchmarks

Foundational models benefit from downstream finetuning as a method to generalize across novel unseen tasks. We build within this regime and study the scaling behavior of GNN models during finetuning. Our finetuning evaluation consists of open-source therapeutic benchmarks. For a fair and comprehensive evaluation, all models are first pretrained using a common supervised learning strategy and then finetuned for molecular property prediction. Benchmarks used for evaluating finetuning scaling behavior are listed below.

TDC (Therapeutics Data Common) (Huang et al., 2021) is one of the common benchmarks for drug discovery. Our study focuses on 22 ADMET (Absorption, Distribution, Metabolism, Excretion and Toxicity) tasks. While TDC serves as the bedrock for open-source drug discovery evaluation, we note that it suffers from data collection and processing biases across dissimilar molecules (Walters, 2023).

Polaris: Polaris is a recent collection of benchmarks addressing concerns over previous datasets. Developed by an industry consortium of various biotech and pharmaceutical organizations, it provides access to high-quality molecular samples across various tasks. Our analysis considers 12 of the top ADMET tasks for molecular property prediction (Anonymous, 2024).

A.3 Architectures

We broadly study three types of architectures; (1) message-passing networks, (2) graph transformers and (3) hybrid models. In the case of message-passing networks, we focus on MPNN++ as it provides a suitable testbed for evaluating molecular graphs while maintaining performance across various tasks. Our graph Transformer and hybrid models make use of GPS++, which is known for its scalable nature on quantum property predictions. In addition to GNN models, we make use of Positional and Structural Encodings (PSEs) to improve the expressivity of MPNNs and introduce a soft bias into the Transformer. We discuss architectures and their design aspects below.

MPNN++ is a variation of the neural message passing architecture with edge and global features (Gilmer et al., 2017; Battaglia et al., 2018; Bronstein et al., 2021). Choosing the MPNN++ allows us to maximize architecture expressivity while minimizing the risk of overfitting on larger datasets (Masters et al., 2022). Each MPNN block makes use of sequential `Dropout` (Srivastava et al., 2014), `MLP` and `LayerNorm` (Ba et al., 2016) modules followed by a skip connection (He et al., 2016b; Srivastava et al., 2015) across node and edge features.

$$\begin{aligned}\bar{E}^l, \mathbf{X}^l &= \texttt{Dropout}(\texttt{MLP}([\mathbf{X}^l | E^l])) \\ \mathbf{X}^l &= \texttt{LayerNorm}(\texttt{Dropout}(\mathbf{X}^l)) + \mathbf{X}^l \\ E^{l+1} &= \bar{E}^l + E^l\end{aligned}$$

GPS++ is a hybrid model leveraging the MPNN++ inductive bias while providing the flexibility of self-attention-based self attention (Ying et al., 2021a) modules to allow for a rich feature extraction scheme across nodes and edges, and was empirically proven very successful (Masters et al., 2022). Here, the standard self-attention weights are biased by a structural prior $\mathcal{B}$ from the input graph. Mathematically, the GPS++ modules carries out the following computation.

$$\begin{aligned}\mathbf{X}^{l+1}, E^{l+1} &= \texttt{MPNN++}(\mathbf{X}^l, E^l) \\ \mathbf{Z}^l &= \texttt{BiasedAttn}(\mathbf{X}^{l+1}, \mathcal{B}) \\ \mathbf{X}^{l+1} &= \texttt{MLP}(\mathbf{X}^{l+1} + \mathbf{Z}^l)\end{aligned}$$

Transformer is an architecture identical to the GPS++, but without the MPNN++ module nor the concatenation. Instead, it relies solely on the PSE's for structural bias.

PSEs are an important design choice when training GNN architectures (Rampášek et al., 2022; Liu et al., 2023b), as they allow each node to understand its position and surroundings within a graph. This is essential for any graph Transformer, but it was also shown to improve the expressivity of molecular GNNs. Specifically, we use three PSE schemes. First, we use random-walk diagonals (Dwivedi et al., 2021) to allow one to decouple structural and positional representations. Learned positional encodings are used to tackle isomorphic nodes. Second, we use Laplacian eigenvectors (Beaini et al., 2021) as these form an expressive way to encode node geometries and positions. Laplacian eigenvectors provide strong theoretical guarantees against the 1- Weisfeiler-Lehman test, a useful insight in evaluating GNNs at scale. Last, we use the Laplacian eigenvalues (Kreuzer et al., 2021) as a suitable PSE

scheme to fully leverage the Laplacian spectrum. Additionally, they provide global structural information about the graph.

A.4 Finetuning and Probing

Following pretraining, we finetune and probe our base models on a range of unseen tasks. While there does not exist a clear guideline for finetuning GNNs, we explore this key paradigm. Notably, our evaluation considers two finetuning and probing strategies which present improved scaling behavior on downstream tasks.

Finetuning. Since our training distribution consists of multiple tasks and our architectures incorporate multiple task heads, we need to identify a *finetuning module*, after which the remaining pretraining architecture is removed and replaced by a newly initialized `MLP`, the *finetuning head*. As all downstream tasks reside on the graph level, our main choice is the *graph output network*, the `MLP` that processes features after being aggregated from the node to the graph level, and further feeds into the task heads for graph-level tasks. Intuitively, this layer's output representations have benefited most from pretraining on diverse data and tasks, as it feeds directly into the various task heads. We further investigate the effect of choosing layers of the tasks heads as finetuning module to potentially leverage synergies between specific pretraining and downstream tasks. As all downstream task are are on the graph level, we trim the architecture by removing parts related to node level tasks and unused task heads.

Probing and Fingerprinting. Similar finetuning, we employ probing using an `MLP` as a suitable strategy for obtaining general representations on downstream tasks. In probing, the base model is kept frozen and only the new layers are trained. This allows the training procedure to focus the gradient on newly added parameters, resulting in task-specific head layers. In the case of large model sizes, running features through the frozen base model is expensive in memory and compute. We tackle this bottleneck by caching hidden representations on disk and reusing them during probing. Since the gradient does not impact parameters of the base model, fingerprints remain unchanged yielding an optimized strategy for downstream tasks capable of parallelization across multiple inexpensive devices. In this work, similar to the finetuning setup, we extract fingerprints from the task head `MLP`s of graph-level tasks, and from the last layer of the *graph output network*, the `MLP` that directly feeds in to the task heads.

Appendix B. Related Work

Foundation Models for Molecules. Recent work has relied on foundation models as a generalist class of models for sequential modelling (Yuan, 2023; Liu et al., 2023a) as well as knowledge retrieval (Galkin et al., 2023). Within molecular drug discovery, recent works rely on structured models of ligands (Moret et al., 2020). Luo et al. (2022) and Moret et al. (2023) study a general model for protein synthesis. Rao et al. (2020) construct a self-attention driven architecture for contact prediction. Madani et al. (2023) learn to generate a family of functional proteins. Nijkamp et al. (2023) present a class of protein-pretrained language models. Similarly, Méndez-Lucio et al. (2022) study binding interactions between different assays at the molecule-molecule interaction level.

While many models focus on the design of molecules, a recent class of methods has also focused on properties of molecules (Beaini et al., 2024). Cui et al. (2023) and Luo et al. (2023) study a general architecture for predicting similar properties across different molecules such as solubility and viscosity in a way generalizes across molecular domains with a limited set of samples. Unsal et al. (2022) learn to predict functional properties of proteins by exploiting learned structure. Our study explores similar *molecular tasks* for property prediction.

Architecture Design. Recent methods in graph architecture design focus on attending to structural information across nodes (Müller et al., 2023). Of specific interest are graph transformer networks which extract node as well as edge information by composing sequential attention modules over graph readouts (Yun et al., 2019). In parallel, graph attention networks model attention weights across edges of a graph (Veličković et al., 2017).

While attention mechanisms have demonstrated modern progress, traditional architectures such as neural message passing (Gilmer et al., 2017) and 3D infomax (Stärk et al., 2022a) hold the promise of simplicity and expressivity in modelling molecular graphs. On one hand, message passing provides a rich and expressive framework for constructing representations. On the other hand, the provable lower bound of infomax results in strong convergence guarantees. Godwin et al. (2021) study regularization based on noisy nodes for the task of molecular property prediction. Provision of noise imputation in node-level features leads to simple and expressive method for tackling sparse molecular graphs. Graph bootstrapping (Thakoor et al., 2021) allows prior architectures to scale up to larger and complex graphs for representation learning. Our exploration of *different architectures* is aligned with the aforesaid works in literature, and with recent trends towards Transformers in related machine learning fields.

Scaling Laws. Recent work in model scaling has demonstrated that performant models follow a power law relationship between their parameter sizes and performance on new data samples (Kaplan et al., 2020). Additionally, this relationship holds during the finetuning stage (Hernandez et al., 2021), thus indicating a strong reliance on model parameters. Bahri et al. (2021) explain this power law fit by observing learning as moving on a smooth data manifold. Frantar et al. (2023) study the power law fit for sparsely connected models capable of downstream generalization. Notably, sparsely connected foundation models reach a computational bottleneck as a result of different sparsity structures impacting hardware usage. (Aghajanyan et al., 2023) study the power law fit for mixed modality generative models, indicating that the scaling behavior is modality agnostic across various datasets. The result hints at the generality of scaling across different domains and applications.

Caballero et al. (2022) extend the study of scaling laws towards different training regimes (such as finetuning, downstream transfer and inference) as well as different problem domains (vision, language, audio, diffusion, generative modelling, contrastive learning and reinforcement learning). The resulting functional form results in extrapolations which are empirically accurate. Diaz and Madaio (2023) present a tangential result demonstrating that dataset entities may not necessarily scale with the growing amount of parameters and computational requirements, likely due to models leaving out essential data samples. Cherti et al. (2023) study the reproducibility of scaling laws for contrastive learning scenarios at the intersection of language and vision.

While recent work on the scaling behavior of graph networks and other structural inductive biases remains absent, a few notable works aim at studying the behavior of scaling graph sizes. Grindrod (2022) evaluate the power law fit of growing graph sizes utilizing random walk sampling and exploration. Similarly, J. (2021) study the scaling behavior of macromolecule proteins with respect to their mean node degree. Our exploration of *scaling behaviors in graph networks* is motivated by the aforesaid directions.

Appendix C. Preliminaries

C.1 Graph Neural Networks

Our problem setting consists of graphs of the form $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges. Each node $i \in \mathcal{V}$ indicates the atom and each edge $(u, v) \in \mathcal{E}$ denotes a chemical bond between two atoms. Total number of atoms in the molecule are $N = |\mathcal{V}|$ while total number of edges are $M = |\mathcal{E}|$. Node features in layer l are denoted by $\mathbf{x}_i^l \in \mathbf{R}^d$ and are concatenated into an $N \times d$ representation matrix $\mathbf{X}^l = [\mathbf{x}_1^l; \mathbf{x}_2^l; ...\mathbf{x}_N^l]^\top$. Edge features $e_{uv}^l \in \mathbf{R}^d$ are concatenated into the edge feature matrix $E^l = [e_{uv}^l : (u, v) \in \mathcal{E}]^\top$.

C.2 Scaling Laws

We denote the parameters of a model as θ with the total number of trainable parameters being $|\theta|$. We consider a training dataset $\mathcal{D}$ consisting of labeled data samples $(\mathcal{G}, y) \in \mathcal{D}$. Here, $\mathcal{G}$ indicates the input graph and $y \in \mathbf{R^N}$ denotes the categorical or continuous label. Total size of the dataset is denoted as $|\mathcal{D}|$. Given the study of large foundational models, we note that $|\theta|$ is large in size and θ lies on a high dimensional manifold such that $\theta \in \mathbf{R^B}$ where $\mathbf{B} >> |\mathcal{D}|$. Recent work has shown that increasing the size of dataset $|\mathcal{D}|$ or the number of trainable parameters $|\theta|$ has a direct power law relationship on the loss function $L_\theta(|\mathcal{D}|)$ (Kaplan et al., 2020). Mathematically, we have the following,

$$L_\theta(|\mathcal{D}|) \propto (|\theta_C|/|\theta|)^\alpha \tag{1}$$

Equation 1 denotes the power-law relationship between the number of trainable parameters and the loss obtained when utilizing the parameters θ. Further, θ_C denotes the critical parameters and $\alpha \in \mathbf{R^N}$ is a scalar constant. Intuitively, as the number of parameters approaches a critical value, with every gradient step, the test loss decreases at power-law with a constant rate. A similar relationship holds for the size of datasets. Mathematically, we have the following,

$$L_\theta(|\mathcal{D}|) \propto (|\mathcal{D}_C|/|\mathcal{D}|)^\beta \tag{2}$$

Equation 2 describes the power-law relationship between the size of dataset and loss obtained when training the model on $\mathcal{D}$. Here $|\mathcal{D}_\mathcal{C}|$ denotes the critical size of the dataset and $\beta \in \mathbf{R^N}$ is a scalar constant.

Appendix D. Experimental Details

D.1 Pretraining

All models use 2-layer MLPs to encode node and edge features, respectively, followed by the core model of 16 layers of the MPNN, Transformer or Hybrid model (except for when

scaling depth). The outputs are aggregated to the graph level and node and graph level representations go through 2-layer MLPs. Finally, representations are processed by separate task heads (2-layer MLPs) specific to each pretraining task. Further, all layers use layer norm and dropout with $p = 0.1$. The encoder and model core additional have residual connections similar to He et al. (2016a).

Our hyperparameter search for all base models was conducted on all oberserved data samples with a constant model size of $10M \pm 0.1M$ parameters. For scaling on width, zero-shot scaling from μP (Yang et al., 2022) was used. For other scaling results, μP was used to scale the model with $10M$ parameters used as the base model. In the case of depth scaling, we adjusted the learning rate as suggested by depth-μP (Yang et al., 2023). We did not consider adjusting the residual connections.

Our base MPNN, Transformer and Hybrid model is trained using Adam with a base learning rate of 0.003, 0.001 and 0.001, respectively. We use 5 warm-up epochs followed by linear learning rate decay. All pretraining has been conducted with a batch size of 1024. Scaled version of the used models require advanced training strategies due to the large model size. We used mutli-gpu training (with up to 8 NVIDIA A100-SXM4-40GB GPUs) and gradient accumulation, while adjusting batch size to keep the effective batch size constant.

D.2 Finetuning and Probing

Finetuning. As outlined in Section A.4, a finetuning module is selected from one of the layers of the pretraining architecture and a newly initialized `MLP` is appended to that layer. Here, we use 2-layer `MLP`s with a hidden dimension of 256. For each experiment, when retraining this model, we set set the dropout rate to zero and train for 40 epochs using a batch size of 256 and a constant learning rate of 0.0001. To first adjust the *finetuning head* – the newly initialized `MLP` after the finetuning module – we freeze the remaining architecture for the first 10 epochs. To find a unified finetuning strategy for each pretrained model/downstream task combination, we select the best number epoch where validation performance was maximized across all seeded runs of the experiment.

Probing. Similar to finetuning, we apply a 2-layer `MLP` to the fingerprints derived from the pretrained model. We choose a hidden dimension of 128 and train for 30 epochs with a batch size of 128 and a constant learning rate of 0.0001. Further, we use the same approach as for finetuning to select a unified number of epochs for each pretrained model/downstream task combination based on validation.

Appendix E. Trade-Off Between Over-smoothing and Depth

We note that GNN architectures exhibit *over-smoothing* phenomenon, which implies that latent representations of a network become similar and coarser as the network grows in size. Prior evidence suggests that over-smoothing occurs linearly with the increasing depth of GNN networks (Hamilton, 2020; Xu et al., 2019). We observed similar behvaiors for MPNN architectures during pretraining where the performance for node-level tasks degrades significantly with very deep networks. However, it is difficult to determine without any doubt that over-smoothing is the culprit.

On another hand, over-smoothing is believed to be aleviated by graph Transformers. Recent works argue that Transformers present favorable properties which make them robust towards over-smoothing, such as the provision of embeddings and the inductive bias of attention (Dovonon et al., 2024). However, we still observe a degradation of performance with depth of our Transformer models, in contradiction with this hypothesis. Its theoretical understanding and empirical analysis remains an open question for future work.

Appendix F. Additional Related Work

F.1 Foundational Models for Molecules

Here, we present additional advancements in foundational models making use of molecular graphs. Recent works have argued that the use of high-capacity models will be a significant boon to scientific discovery tasks (Wang et al., 2023a). Of specific interest are tasks in the quantum and molecular discovery paradigms (Zhang et al., 2023) which demand domain-specific expertise such as knowledge of structure, provision of additional inductive biases and large data requirements. Towards this hypothesis, (Fifty et al., 2023) present an in-context learning framework for molecular property prediction without explicitly using a meta learning procedure. This leads to a general algorithm capable of discovering high-level structures from a pretraining sample set. Guo et al. (2021) propose a similar framework making use of few-shot learning techniques resulting in a sample-efficient learning procedure. Xu et al. (2023) present an alternative approach by modelling the full graphical structure of molecules across different property prediction tasks. Although effective, modelling the entire graph results in a computationally intensive learning procedure. Finally, Cao et al. (2023) scale up learning to larger graph sizes by segmenting graph neighborhoods on the fly. An ad-hoc partitioning procedure is employed and interleaved with the learning phase in order to accelerate learning on larger and dense graphical clusters.

F.2 Expressivity of GNNs

Prior work highlights that GNN architectures are limited in their expressivity to distinguish between graphs of similar node arrangements but different geometrical structures (Xu et al., 2019). Various works indicate this as a consequence of aggregation functions and other design factors involved in GNN training (Hamilton, 2020). On the other hand, recent work argues that only specific architectures are found robust to over-smoothing when building latent representations (Dovonon et al., 2024). For instance, graph transformers exhibit over-smoothing robustness as they utilize strong inductive biases such as attention. Xu et al. (2021) connect the limited expressivity of GNNs with their ability to extrapolate on simpler

tasks. Contrary to multi-layer networks, GNNs struggle to extrapolate on simpler tasks but show promise for improvement. Morris et al. (2019) aim to tackle over-smoothing by building higher-order GNN architectures capable of capturing intricate node characteristics in their deeper layers. Finally, Ying et al. (2018) present the differentiable pooling module capable of pooling neighboring node features which aid in reducing noise across layer representations.

Appendix G. Training Curves of Pretraining Models

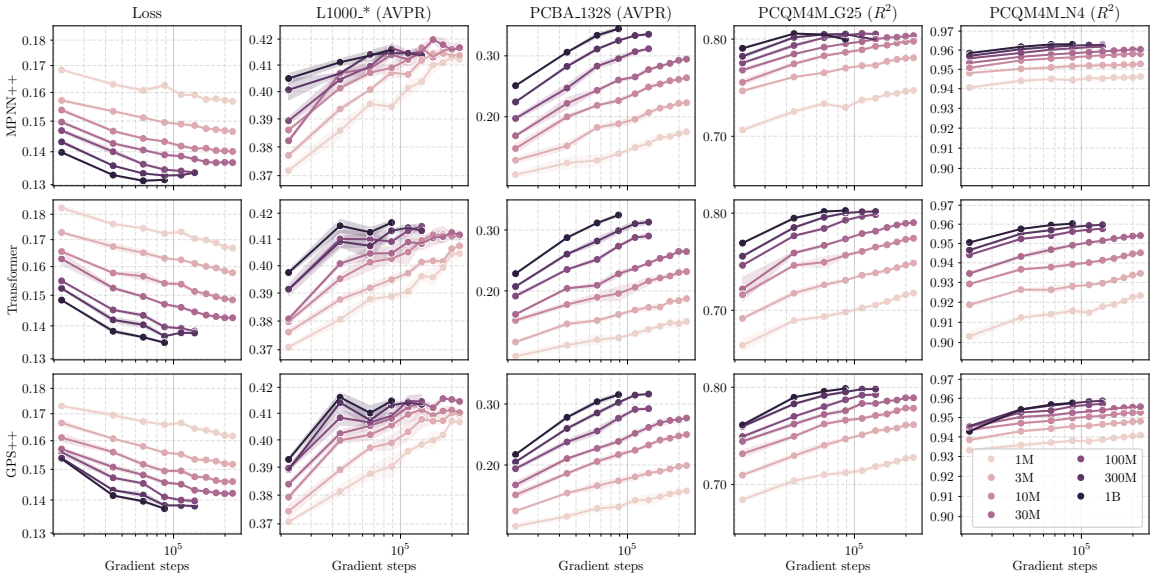


Figure 4: Model performance on the test set throughout training for MPNN++, Transformer and GPS++ architectures with width scaling. Different colors represent models with varying number of parameters.

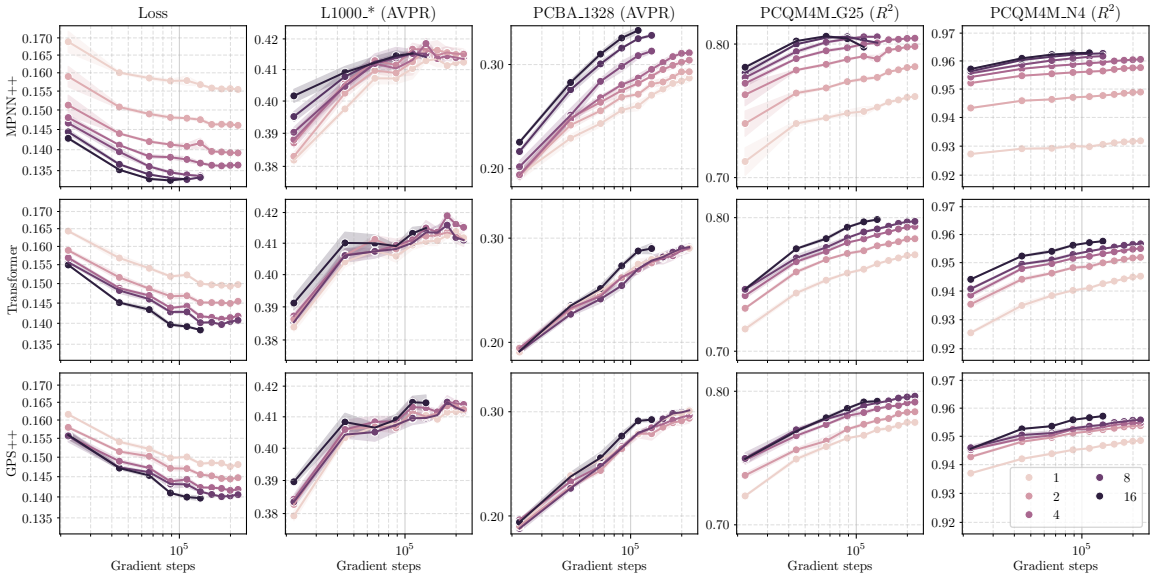


Figure 5: Model performance on the test set throughout training for MPNN++, Transformer and GPS++ architectures with depth scaling. Different colors represent models with varying number of network layers.

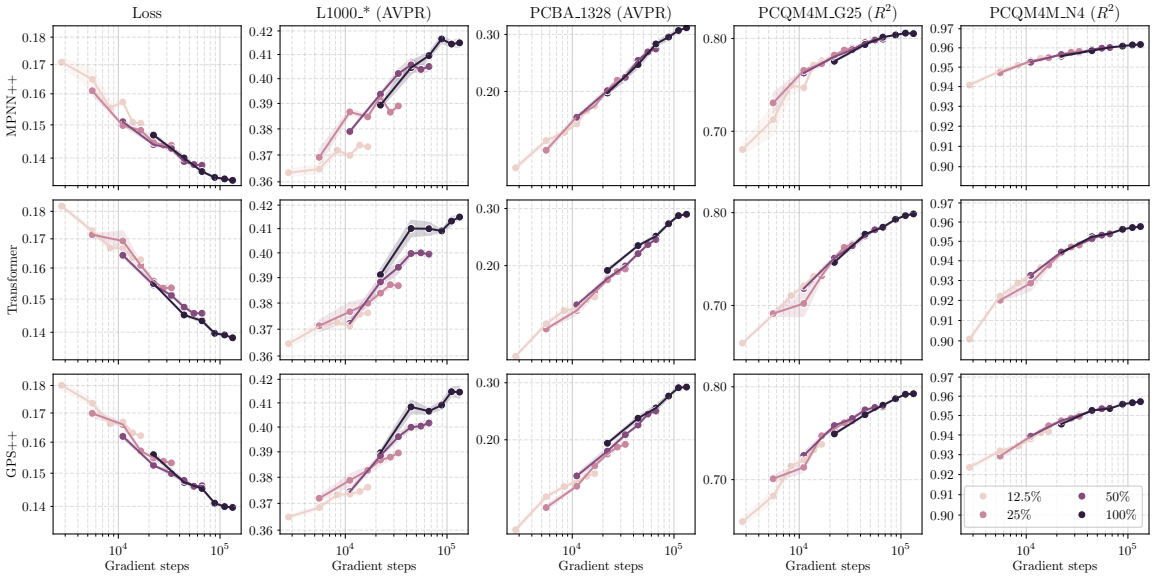


Figure 6: Model performance on the test set throughout training for MPNN++, Transformer and GPS++ architectures with molecule scaling. Different colors represent models with varying fraction of molecules used for training.

Appendix H. Additional Finetuning Results

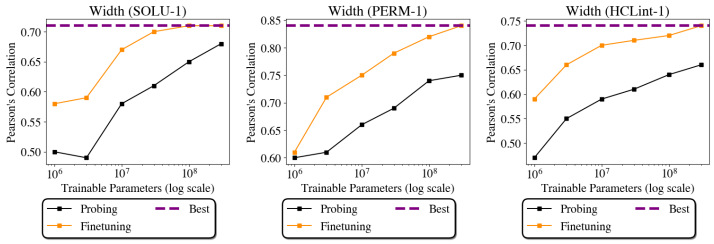


Figure 7: Comparison of probing and finetuning for different model sizes on top 3 Polaris benchmark tasks. We additionally compare to the best model reported on Polaris Hub. Compared to probing, finetuning is found to perform better at trainable parameter utilization. Larger finetuned models further outperform previous best methods on the benchmark.

H.1 The Polaris Benchmark - Depth Scaling

Depth scaling	Polaris Datasets	adme-fang-SOLU-1	adme-fang-RPPB-1	adme-fang-HPPB-1	adme-fang-PERM-1	adme-fang-RCLint-1	adme-fang-HCLint-1	plds2-klf-wt-c-1	plds2-klf-wt-r-1	plds2-ret-wt-c-1	plds2-ret-wt-r-1	plds2-egfr-wt-c-1	plds2-egfr-wt-r-1
Probing	Metric	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*
	1	.56	.59	.61	.70	.63	.62	.57	.31	.61	.37	.52	.12
	2	.58	.64	.63	.72	.65	.63	.56	.33	.55	.35	.48	.24
	4	.61	.73	.69	.72	.64	.62	.58	.35	.60	.40	.40	.25
	8	.63	.66	.67	.71	.64	.62	.61	.38	.54	.35	.32	.19
	16	.65	.57	.75	.74	.66	.64	.61	.37	.53	.36	.30	.26
	32	.67	.63	.70	.74	.67	.66	.59	.37	.62	.43	.33	.27
	64												
	Spearman	1.00	-.14	.89	.71	.83	.60	.71	.83	-.09	.26	-.83	.83

Figure 8: Comparison of probing and finetuning for different MPNN++ model depths on the Polaris benchmark. **Darker green** shades denote higher/desirable metric values. Average spearman correlation between depth and performance is 0.46. Both probing and finetuning strategies improve scaling behavior with increasing depth across the Polaris benchmark.

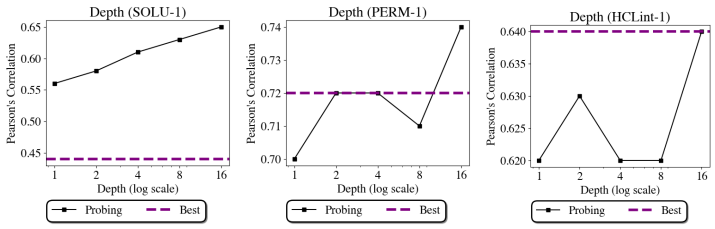


Figure 9: Comparison of probing for different depth sizes on top 3 Polaris benchmark tasks. We additionally compare to the best model reported on Polaris Hub. Probed models scale well with increasing depth and outperform previous best methods on the benchmark.

H.2 The Polaris Benchmark - Molecule Scaling

Molecule scaling	Polaris Datasets	adme-fang-SOLU-1	adme-fang-RPPB-1	adme-fang-HPPB-1	adme-fang-PERM-1	adme-fang-RCLint-1	adme-fang-HCLint-1	pkis2-kit-wt-c-1	pkis2-kit-wt-r-1	pkis2-ret-wt-c-1	pkis2-ret-wt-r-1	pkis2-egfr-wt-c-1	pkis2-egfr-wt-r-1
Probing	Metric	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*
	12.50%	.58	.58	.63	.73	.65	.67	.57	.18	.52	.46	.32	.13
	25%	.63	.67	.75	.73	.66	.66	.58	.34	.61	.47	.37	.26
	50%	.65	.56	.71	.74	.67	.65	.58	.36	.54	.37	.43	.17
	100%	.65	.57	.75	.74	.66	.64	.61	.37	.53	.36	.30	.26
	Spearman	.80	-.60	.80	.80	.80	-1.00	1.00	1.00	.20	-.80	-.20	.80

Figure 10: Scaling behavior of probed 100M parameter MPNN++ models across different dataset molecule fractions on the Polaris benchmark. **Darker green** shades denote higher/desirable metric values. Average spearman correlation between molecule fraction and performance is 0.30. Models show consistent improvement in performance with the increasing size of datasets.

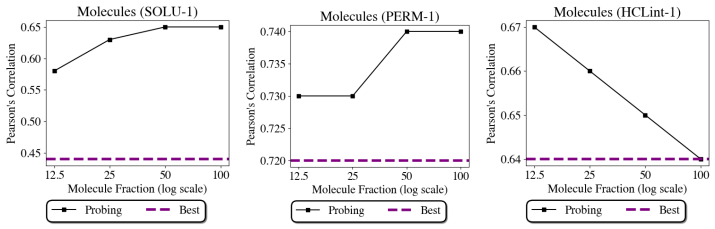


Figure 11: Comparison of probing for different molecule dataset sizes on top 3 Polaris benchmark tasks. We additionally compare to the best model reported on Polaris Hub. Probed models scale well with dataset sizes and outperform previous best methods on the benchmark.

H.3 The Polaris Benchmark - Label Scaling

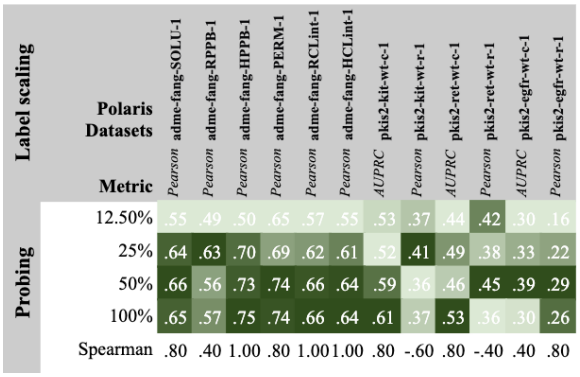


Figure 12: Performance of probed 100M parameter MPNN++ models across different label fractions in the Polaris benchmark. **Darker green** shades denote higher/desirable metric values. Average spearman correlation between labell fraction and performance is 0.56. Models continue to maintain consistent performance despite the increasing number of labels.

H.4 The Polaris Benchmark - Dataset Scaling

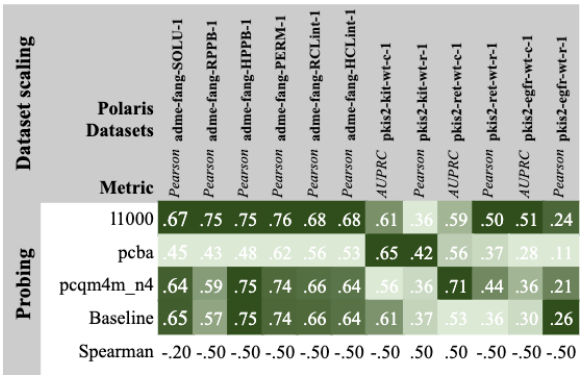


Figure 13: Comparison of probed 100M paramter MPNN++ models on the Polaris benchmark tasks (in columns) when trained on different pretraining datasets (in rows). **Darker green** shades denote higher/desirable metric values. Average spearman correlation between choice of dataset and performance is -0.31. One can note that removing the PCBA_1328 dataset significantly hinders performance across all tasks, while removing the L1000 datasets improves performance on most tasks.

H.5 The TDC Benchmark - Data leakage

Considering that the pre-training dataset is supervised, it is important to consider data-leakage as a source of experimental error. This is especially the case for the PCBA_1328 dataset.

PCBA_1328 contains only classification assays with more than 6000 molecules each, which automatically disqualifies most of TDC and all of Polaris. The TDC datasets remaining after this constraint are the 3 CYP*_Veith datasets, and the AMES dataset.

The AMES dataset is not present in PubChem, excluding it from the list of potential leaks.

Regarding the 3 CYP*_Veith datasets, they represent inibition assays against recombinant enzymes (Veith et al., 2009). The three assays from TDC, and two others from the paper, are all present and aggregated under assayID-1851. Therefore, whenever a molecule is active against any of the enzyme, the value is 1, otherwise it is 0. Therefore, there is a minor leak, although the datasets are not identical. Further, no evidence of leak was observed in terms of abnormally high performance of the model on these assays, which is expected considering that the model is learning more than 3000 labels simultaneously.

H.6 The TDC Benchmark - Width Scaling

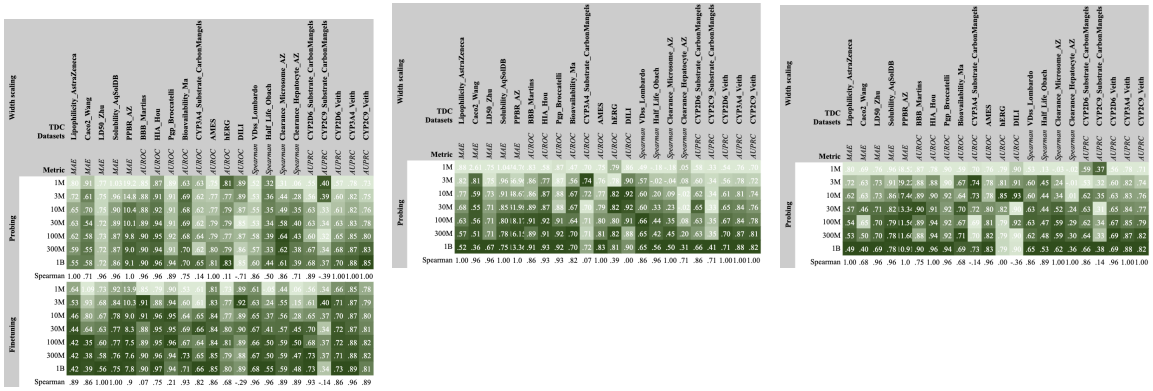


Figure 14: Comparison of probing and finetuning of width-scaled models for **(left)** MPNN++, **(center)** transformer and **(right)** hybrid models across different model sizes on the TDC benchmark. **Darker green** shades denote higher/desirable metric values. Average spearman correlations for MPNN, transformer and hybrid models are Both probing and finetuning strategies improve scaling behavior with increasing number of parameters across the TDC benchmark. The average spearman correlation between width and performance is 0.6 when probing MPNNs and 0.72 when finetuning MPNNs, effectively showing that model size plays an important role in predictive performance.

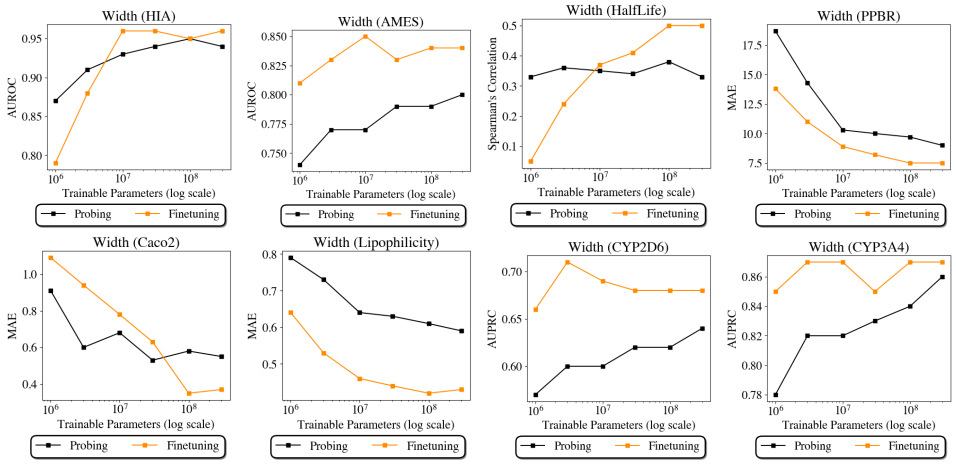


Figure 15: Comparison of probing and finetuning for different (MPNN++) model sizes on top 8 TDC benchmark tasks. Compared to probing, finetuning is found to perform better at trainable parameter utilization. This leads to an improved scaling behavior during downstream learning.

H.7 The TDC Benchmark - Depth Scaling

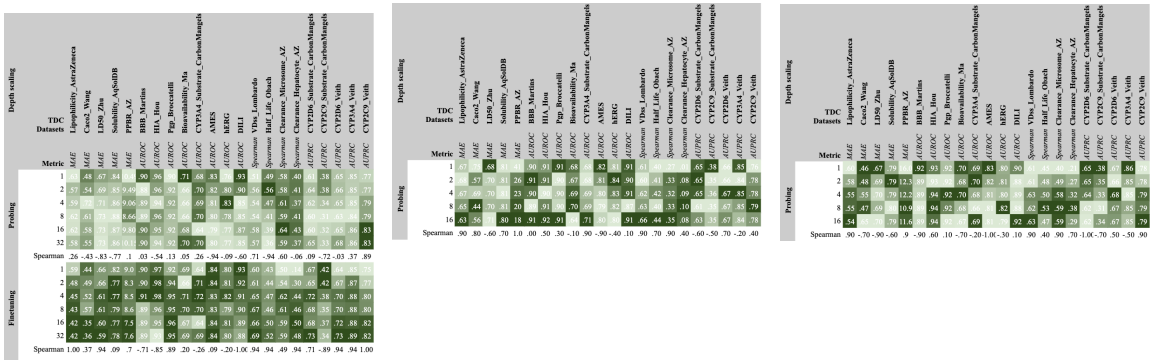


Figure 16: Comparison of probing and finetuning of depth-scaled models for **(left)** MPNN++, **(center)** transformer and **(right)** hybrid models across different model depths on the TDC benchmark. **Darker green** shades denote higher/desirable metric values. Average spearman correlation between depth and performance is -0.11 for probed MPNNs and 0.33 for finetuned MPNNs. While performance increases with depths up to 8 and 16, larger depths saturate model representations.

H.8 The TDC Benchmark - Molecule Scaling

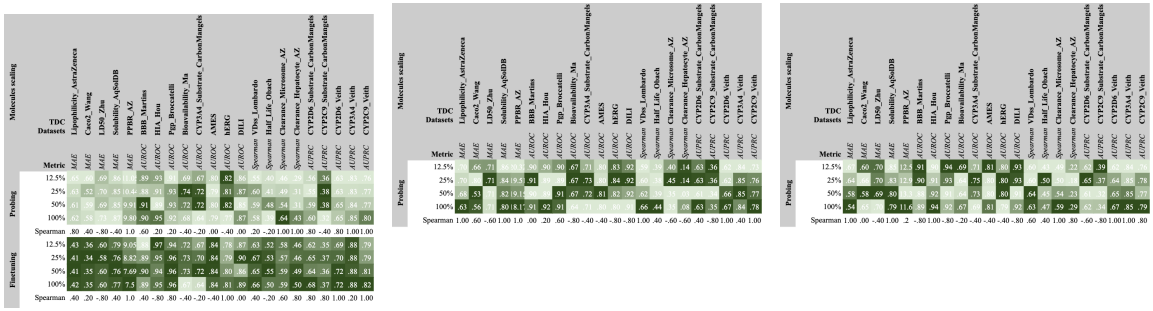


Figure 17: Comparison of probing and finetuning of molecule-scaled 100M parameter models for **(left)** MPNN++, **(center)** transformer and **(right)** hybrid models across different dataset sizes on the TDC benchmark. **Darker green** shades denote higher/desirable metric values. Average spearman correlation between molecule fraction and performance is 0.28 for probed MPNNs and 0.31 for finetuned MPNNs. Finetuned models scale better when compared to probed models. However, increasing the size of finetuning datasets leads to minor improvements beyond the 50% dataset size fraction.

H.9 The TDC Benchmark - Label Scaling

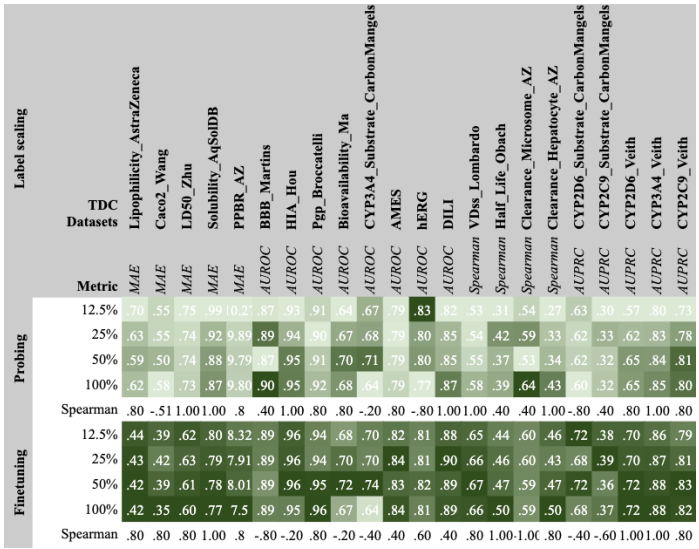


Figure 18: Comparison of probing and finetuning for 100M parameter MPNN++ across different label fractions on the TDC benchmark. **Darker green** shades denote higher/desirable metric values. Average spearman correlation between label fraction and performance is 0.54 for probed MPNNs and 0.72 for finetuned MPNNs. Finetuned models scale better when compared to probed models. Increasing label fractions do not deteriorate model performance.

H.10 The TDC Benchmark - Dataset Scaling

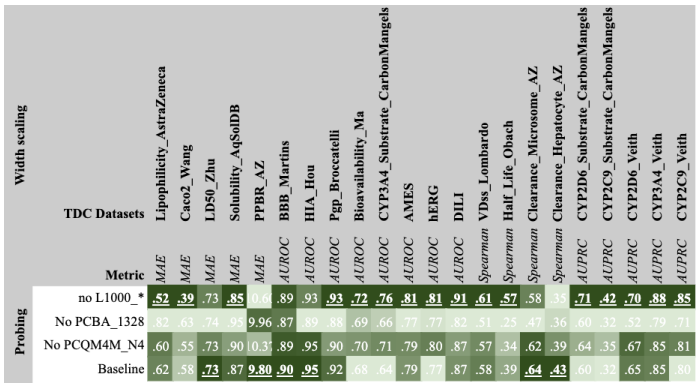


Figure 19: Performance of probed 100M parameter MPNN++ model on TDC benchmark tasks (in columns) when pretrained **without** certain datasets (in rows). **Darker green** shades denote higher/desirable metric values. One can note that removing the PCBA_1328 dataset significantly hinders performance across all tasks, while removing the L1000 datasets improves performance on most tasks.

H.11 The TDC and Polaris Benchmarks - Task-head ablations

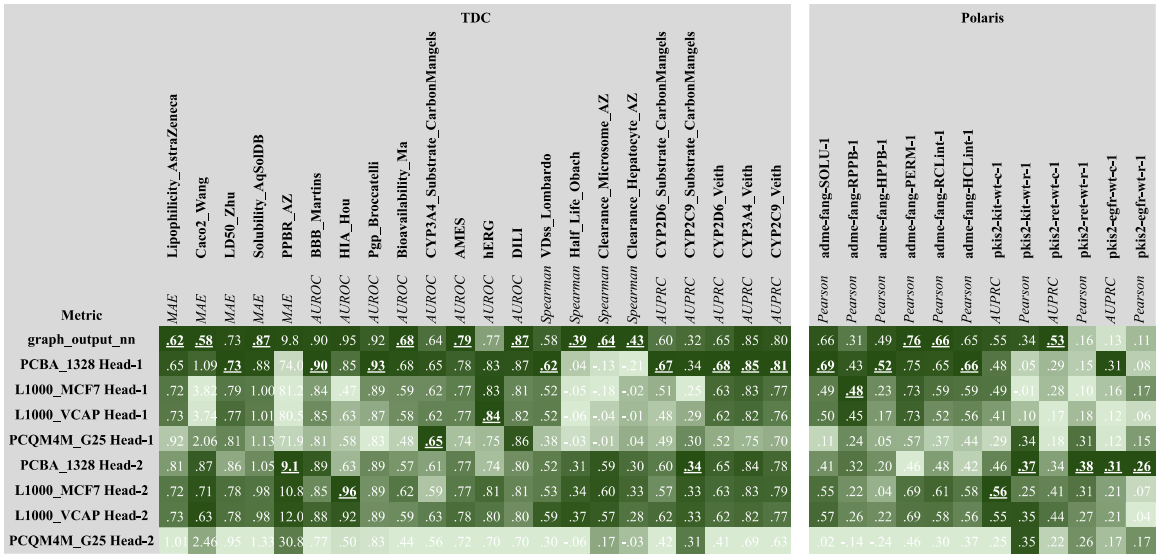


Figure 20: Performance of probed 100M MPNN++ models on TDC benchmark (left) and Polaris benchmark (right), with the tasks in columns, when probing is done from different task heads **without** certain datasets (in rows). **Darker green** shades denote higher/desirable metric values, and **bold/underline** indicates the best value in a given column. One can observe that the *graph_output_nn*, e.g. the hidden dimension that is fed to all task-heads, is generally the best choice for probing due to it containing general and compressed information. The *PCBA_1328* task-head is also a good choice due to its proximity to the downstream task. The *PCQM4M_G25* task-head is the least interesting due to it being significantly different from the downstream fine-tuning tasks.