ON THE SCALABILITY OF GNNS FOR MOLECULAR GRAPHS

Maciej Sypetkowski[1,*], **Frederik Wenkel**[1,2,*], **Farimah Poursafaei**[1,3], **Nia Dickson**[4], **Karush Suri**[1], **Philip Fradkin**[1,5], **Dominique Beaini**[1,2]
[*] Equal contribution, [1] Valence Labs, Montreal, [2] University of Montreal, Mila, Quebec,
[3] McGill University, Mila, Quebec, [4] NVIDIA Corporation,
[5] University of Toronto, Vector Institute
maciej.sypetkowski@recursionpharma.com, frederik@valencelabs.com

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 by studying their scaling behavior. Specifically, we analyze message-passing 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 depth, width, 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 38 tasks, outclassing previous large models. We hope that our work paves the way for an era where foundational GNNs drive pharmaceutical drug discovery.

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 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).

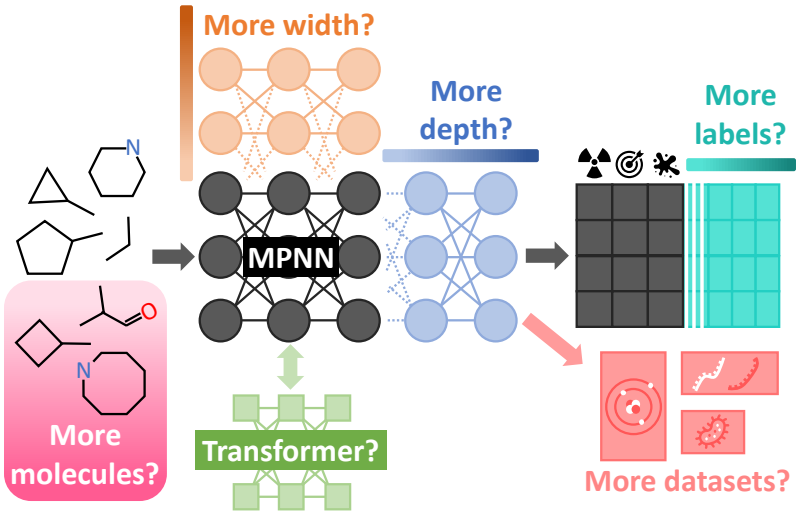


Figure 1: Summary of our GNN scaling hypotheses studied in the present work. The baseline model is presented in **dark grey**, followed by different scaling hypotheses illustrated in lighter colors. We analyze the scaling behavior of message-passing networks, graph Transformer and hybrid architectures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

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 & 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 learn 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) randomly split train and test sets for pretraining and (2) fine-tuning of pretrained models 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 depth, width, 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 scaling 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.
- We show that, by simply scaling to 1B parameters, our largest model outperforms other pretrained models, while consistently achieving parity with the state-of-the-art (SOTA) of specialized models across a wide variety of competitive downstream tasks.

More details on related work and preliminaries related to GNNs can be found in Appendix A and B.

2 METHODOLOGY

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 38 datasets from Huang et al. (2021), Polaris (Anonymous, 2024) and MoleculeNet (Wu et al., 2018). We evaluate our models in both finetuning and probing (fingerprinting) settings.

We begin by providing a detailed description of the datasets and benchmarks. We then elaborate on the choice of architectures. Finally, we discuss finetuning and probing strategies along with the results of our analysis.

2.1 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. This is a challenging approach towards learning representations with GNNs, especially as some labels are very imbalanced and sparse.

The LargeMix dataset mixture consists of 5 million molecules grouped into 5 different tasks at different graph levels, with each task having multiple labels. The diversity of this mixture of data makes this dataset suitable for pretraining large GNNs. Below is a description of the individual tasks contained within the 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 1,328 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).

2.2 FINETUNING AND PROBING BENCHMARKS

A major benefit of foundational models is that they allow to easily generalize to unseen downstream tasks through approaches like finetuning or (linear) probing. In this work we want to also study the effect of scaling of the pretrained models on the performance on downstream tasks. For downstream task evaluation we use open-source therapeutic benchmarks. For a fair and comprehensive evaluation, all models are first pretrained using a common supervised learning strategy and then finetuned (or *probed*) for molecular property prediction. The benchmarks used for evaluating are listed below.

TDC. Therapeutics Data Commons (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. This 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 tasks from either ADME (Absorption, Distribution, Metabolism, and Excretion) or DTI (Drug-Target Interaction) group for molecular property prediction (Anonymous, 2024).

MoleculeNet. This is a benchmark dataset for molecular machine learning that is built upon public datasets (Wu et al., 2018). It consists of various datasets covering different levels of molecular properties spanning from properties at the molecular level to broader impacts on the human body. There are different categories of properties including quantum mechanics, physical chemistry, biophysics, and physiology. We investigate 4 datasets that are commonly used in similar studies (such as Liu et al. (2021); Xu et al. (2021b); Liu et al. (2023b)): *BACE* (that assesses the binding outcomes of a group of inhibitors targeting β-secretase 1), *BBBP* (Blood-Brain Barrier Penetration (Martins et al., 2012) assesses if a molecule can penetrate the central nervous system), *Clintox* (toxicity of molecular compounds (Gayvert et al., 2016)), and *Sider* (Side Effect Resource (Kuhn et al., 2016) containing information about adverse drug reactions in a database of marketed drugs).

2.3 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 the MPNN++ model 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++ model, 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++. This 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., 2016; 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++. This is a hybrid model leveraging the MPNN++ inductive bias while providing the flexibility of self-attention-based modules (Ying et al., 2021a) 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++ module 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. This is an architecture identical to the GPS++, but without the MPNN++ module nor the concatenation. Instead, it relies solely on the PSEs for structural bias.

PSEs. These are important design choices 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 with respect to the expressivity of the Weisfeiler-Lehman test, a useful insight when 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.

2.4 Finetuning and Probing

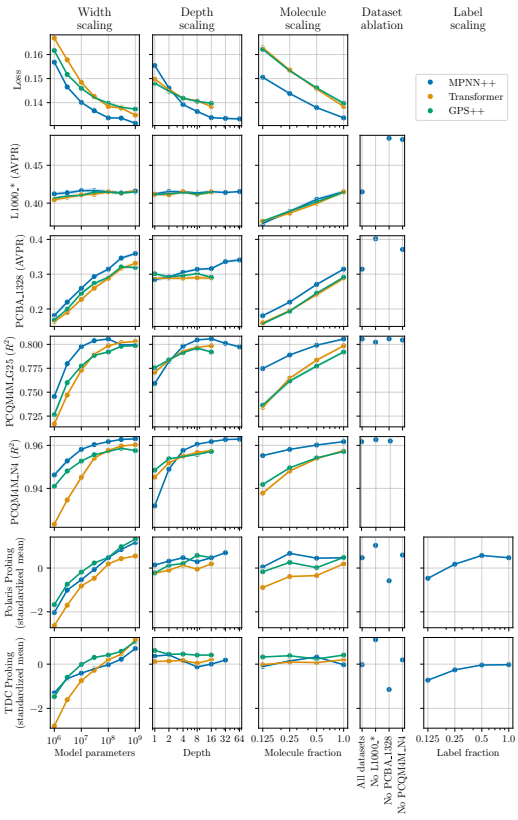
Following pretraining, we finetune our pretrained models on a range of unseen downstream tasks. While there exist no clear guidelines for finetuning GNNs, this aspect is extensively explored in this work. Notably, our evaluation considers two strategies (finetuning and probing), which both significantly benefit from increased scale of the pretrained model.

Finetuning. Since our training setup 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 discussed above 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 tasks 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 to finetuning, we employ probing using an `MLP` as a suitable strategy for obtaining general representations on downstream tasks. For 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 with respect to 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 *graph output network*, the `MLP` that directly feeds in to the task heads.

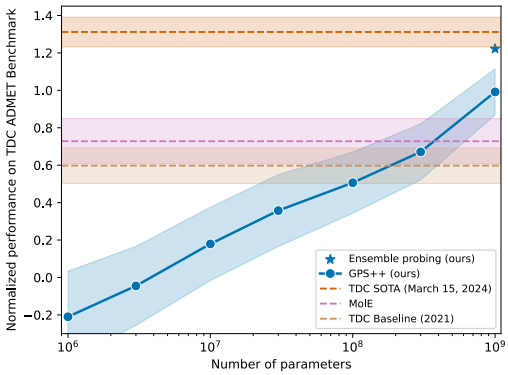
3 How Do Molecular GNNs Scale?

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.



(a)

	MPNN++											
Width Scaling												
Polaris Datasets	**admet-8-rng-SOLU-1**	**admet-8-rng-RPPB-1**	**admet-8-rng-HPPB-1**	**admet-8-rng-PERM-1**	**admet-8-rng-RCL-int-1**	**admet-8-rng-HCL-int-1**	**pkis2-kit-wt-c-1**	**pkis2-kit-wt-t-1**	**pkis2-res-wt-c-1**	**pkis2-res-wt-t-1**	**pkis2-egfr-wt-c-1**	**pkis2-egfr-wt-t-1**
Metric	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*
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	.06	.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 test 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:** Finetuning and probing performance of pretrained MPNN++ models of different width on the Polaris benchmark. **Darker green** shades denote better metric values. Larger models tend to perform better on unseen tasks. Spearman correlation values closer to 1 indicate that predictive performance correlates with larger model sizes. **(b) bottom:** Comparison of our GPS++ probing (that leverages multiple fingerprints) and the 1B Ensemble probing (that leverages fingerprints from MPNN++, Transformer and GPS++) to various baselines across TDC benchmark collection using an aggregated metric.

3.1 Scaling Trends for Pretraining

In this section, we evaluate the scaling behaviour of our models according to various parameters summarized in Figure 1, namely the architecte, width, depth, number of molecules, labels and different datasets. We use datasets from LargeMix described in Section 2.1.

Overview. Figure 2a presents the variation of architectural choices between MPNN++, Transformer and GPS++, with training curves in Appendix E, and full finetuning (and probing) results in Appendix F. Notably, all models scale favourably with the increasing scale of width (number of parameters per neuron), depth (number of GNN layers) and number of molecules (dataset size).

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 larger models for fewer epochs as the loss converged faster and more likely to exhibit 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

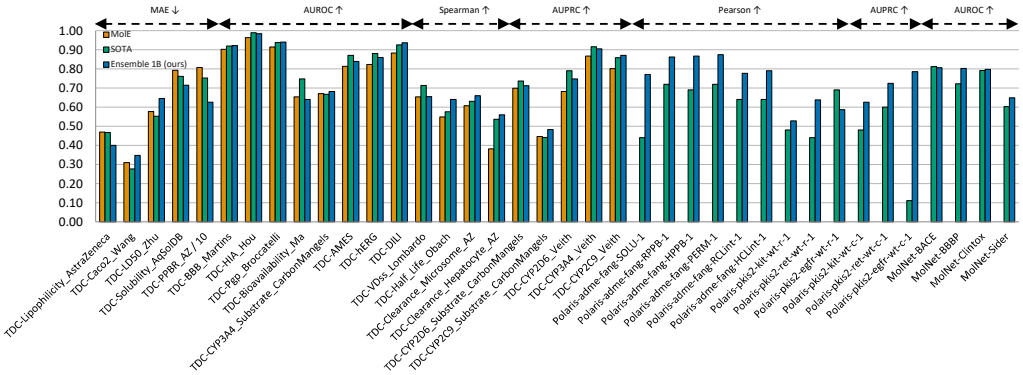


Figure 3: Comparison of our 1B Ensemble probing approach (that combines fingerprints from the MPNN++, Transformer and hybrid GPS++ model) to the SOTA across TDC, Polaris, and MoleculeNet benchmarks. For TDC, a group of 8 different models establishes SOTA across the 22 tasks. Ensemble probing establishes new SOTA on $12/22$ TDC tasks and on all but two tasks among Polaris and MoleculeNet.

12.5% improvement in test error. However, performance plateaus at around 8-16 layers for quantum datasets, while 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, with GPS++ models and Transformer benefiting more than MPNN++ 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.

Detailed scaling law analysis. We provide more detailed analysis of the observed scaling trends in terms of Equations 1 and 2 in Appendix G, observing scaling laws similar to those in other domains (Kaplan et al., 2020).

3.2 Scaling Trends on Downstream Tasks

We now evaluate scaling of models when finetuning and probing on downstream tasks. As detailed in Appendix 2.4, all weights are tuned in the case of finetuning, while the pretrained model is frozen when fingerprinting followed by probing. Due to the large number of downstream tasks spread across 38 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 (i.e., a larger model or dataset yields better downstream task performance), -1 indicating imperfect scaling (i.e., a smaller model or dataset would be 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 for MPNN++. Finally, Transformer is the only model with a *small* positive trend for depth scaling, while GPS++ and MPNN++ show close to no trend.

Width scaling. We evaluate width scaling on Polaris and TDC datasets in Figures 2b (top) and 7 (Appendix F.1). We observe linear scaling trends for MPNN++ on all Polaris datasets, with an average Spearman correlation of 0.91 for probing and 0.85 for finetuning. On TDC, a similar trends

are observed (on average 0.69 for probing and 0.72 for finetuning) with a strong correlation of >0.75 for $15/22$ datasets during probing and $17/22$ during finetuning. These results strongly indicate the benefits of larger pretrained GNNs for downstream tasks, a result consistent with prior findings in scaling of large models (Kaplan et al., 2020). Similarly, Transformer and GPS++ show strong positive scaling trends with model width.

Depth scaling. We evaluate depth scaling of MPNN++, GPS++ and Transformer models on the Polaris and TDC benchmarks in Figures 8 and 9 (Appendix F.2). For probing on Polaris, we observe weak positive trends, with average scaling Spearman correlations of 0.47, 0.55, and 0.50, respectively. We see weaker average correlations on TDC, slightly negative for MPNNN++ and GPS++ and best for Transformer with 0.27. However, finetuning MPNN++ achieves a respectable correlation of 0.33. While some datasets strongly benefit from deeper networks, others strongly deteriorate with no clear pattern observable for TDC. We conjecture that the degradation is related to oversmoothing as described in Appendix C. Certain molecular properties can be well predicted only from small local substructures, hence eliminating the need for long-range interactions that deeper networks enable.

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 their 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 11 (Appendix F.3). For MPNN++, we observe average Spearman correlations of 0.28 and 0.32 when probing and finetuning on TDC, respectively. Contrarily to their stronger trends on the pretraining tasks, Transformer and GPS++ have lower correlations during probing of 0.13 and 0.15. In the case of Polaris, only average correlation of Transformers stands out at 0.73, however reaching worse peak performance per task compared to the less correlated MPNN++ and GPS++. The globally weak positive trends come from the variation across the downstream tasks, with many strong correlations and a few strong negative correlations.

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 13 (Appendix F.4), we observe a Spearman correlation of 0.57 on Polaris and 0.54 on TDC between the ratio of training labels and the performance, with only a few negative correlations. In the finetuning regime, this number lowers to 0.37 on TDC. This makes *label scaling* the second-best strategy for improving the model's downstream performance.

We further conducted a study to determine the importance of the pretraining data in two ways. Firstly, we repeat pretraining of the models without specific pretraining datasets (*dataset ablation*). Secondly, we probe models specifically from certain task head MLPs compared to the base GNN models (*task-head ablations*).

Dataset ablation. Observing the dataset ablations in Figure 14 and 15 (Appendix F.5), we see that PCBA_1328 is the most important pretraining dataset for downstream task performance while L1000_* actually hurts the performance on certain tasks. It will therefore prove beneficial to pretrain without the L1000_* datasets as we will see later.

Task-head ablations. We further tested the effect of probing from different layers of the task heads rather than the graph output network. Results are shown in Table 16 (Appendix F.6). While overall, the graph output network leads to best performance and correlation, the representation after the first layer of the PCBA_1328 task head performs strikingly well for some tasks, possibly due to synergies from pretraining on a bio assay dataset. This suggests probing approaches using combinations of fingerprints could further improve results. On the other hand, the layers from the PCQM4M_G25 dataset perform poorly, which is intuitive as this pretraining task is dissimilar to the downstream task.

3.3 Comparison to Other Models

We now present two further optimized probing approaches that enhance performance on the various downstream tasks considered above, while additionally considering the MoleculeNet benchmark (Wu et al., 2018) here. For each benchmark, we compare to SOTA performance as of *March 15, 2024* and to the MolE model (Méndez-Lucio et al., 2022) if results are available. MolE is a particularly relevant baseline as, similar to our approach, it can be applied to many downstream tasks without major modifications, while SOTA across the benchmarks is naturally established by different approaches depending on the specific task.

Multi-fingerprint probing. Our previous task-head ablation study suggested that different fingerprints may be optimal for probing depending on the downstream task. As a result we further experimented with probing on combinations of multiple fingerprints extracted at different layers of the pretrained model. Additionally, based on our observation in the dataset ablation, we pretrained new models without the L1000_* pretraining tasks. This lead to performance improvements across all scales. In Figure 2b (bottom), we report an aggregated metric across the TDC benchmark collection (average standardized score[1]) for our GPS++ model. We observe a strictly positive scaling trend with the 1B parameter model clearly outperforming TDC Baseline (the normalized score across the best models per task reported in Huang et al. (2021)) and the MolE foundational model (Méndez-Lucio et al., 2022), a gold standard for molecular property prediction.

Ensemble probing. We also found that downstream task performance can be further enhanced by ensembling fingerprints from multiple pretrained models. Our ensemble probing approach extracts fingerprints from 1B MPNN++, Transformer and GPS++ models, followed by an `MLP`. Figure 3 compares this model across the TDC, Polaris and MoleculeNet benchmark collections to the current SOTA and MolE. Ensemble probing yields the strongest downstream task results, outperforming MolE in $19/22$ TDC tasks and establishing SOTA performance on $12/22$ TDC tasks. This makes our ensemble probing approach the model with the most SOTA entries in the TDC leaderboard followed by MapLight+GNN (Notwell & Wood, 2023) that established SOTA on 5 TDC tasks and 7 other methods that are SOTA for at least one TDC task.[2] Finally, when comparing the ensemble probing to previous best methods on the Polaris and MoleculeNet benchmarks, we observe that our model is significantly better (often by large margins). We primarily attribute the large-scale success to width scaling up to 1B parameters, suggesting us to scale even further.

We also report the aggregated score of the 1B ensemble probing in Figure 2b (bottom), which almost reaches the aggregated SOTA performance, which is remarkable recalling again that this score is a derived from the best scoring method per task of the benchmark collection, while we use a single method for all tasks.

4 Conclusion

In this paper, we studied the scalability of GNNs including message-passing networks, graph Transformers and hybrid architectures on the largest public collection of 2D molecules for molecular property prediction. We showed major performance gains from the growing amount of parameters, data and compute, both on the original test set and downstream tasks. Importantly, our models benefit tremendously from the increasing scale of width, number of molecules and number of labels. More importantly, we demonstrate a consistent performance improvement on downstream tasks with the our largest models consistently producing results on par with SOTA. We hope that our work paves the way for the development of foundational GNNs that accelerate pharmaceutical advancements and drug discovery.

4.1 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.

[1]The standardized score of a model M for a specific task T is derived as follows. Let $s_T(m)$ denote the score of a model m for task T and let $\mathcal{S}_T \coloneqq \{s_T(m) : m \in \mathcal{M}_T\}$, where $\mathcal{M}_T$ denotes the set of all methods that have been applied to task T. The aggregated score is defined as $\text{s\_agg}_T(M) \coloneqq \text{sgn}(s_T) \cdot (s_T(M) - \text{mean}(\mathcal{S}_T)) / \text{std}(\mathcal{S}_T)$, where $\text{sgn}(s_T)$ is the polarity of the metric, i.e., positive if "higher is better" and negative if "lower is better".

[2]We report scores from the TDC leaderboards extracted on March 15, 2024. SOTA on TDC is established by a group of 8 different models, namely Chemprop-RDKit (Yang et al., 2019), MapLight, MapLight+GNN (Notwell & Wood, 2023), BaseBoosting (Huang12 et al.), CFA (Quazi et al., 2023), SimGCN, ZairaChem (Oselusi et al., 2024) and ContextPred (Zhou et al., 2020).

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.

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*, pp. 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)*, pp. 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.

Kaitlyn M Gayvert, Neel S Madhukar, and Olivier Elemento. A data-driven approach to predicting successes and failures of clinical trials. *Cell chemical biology*, 23(10):1294–1301, 2016.

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*, pp. 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*, pp. 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*, pp. 770–778, 2016.

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.

David Huang12, Sauhaarda Raunak Chowdhuri13, Alex Li13, Ayush Agrawal14, Kameron Gano15, and Andy Zhu. A unified system for molecular property predictions: Oloren chemengine and its applications.

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.

Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079, 2016.

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.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021.

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*, pp. 1–8, 2023.

Ines Filipa Martins, Ana L Teixeira, Luis Pinheiro, and Andre O Falcao. A bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling*, 52 (6):1686–1697, 2012.

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.

Oscar Méndez-Lucio, Christos Nicolaou, and Berton Earnshaw. Mole: A molecular foundation model for drug discovery. *arXiv preprint arXiv:2211.02657*, 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, pp. 4602–4609, 2019.

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

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

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.

James H Notwell and Michael W Wood. Admet property prediction through combinations of molecular fingerprints. *arXiv preprint arXiv:2310.00174*, 2023.

OpenAI. Gpt-4 technical report, 2023.

Samson O Oselusi, Phumuzile Dube, Adeshina I Odugbemi, Kolajo A Akinyede, Tosin L Ilori, Elizabeth Egieyeh, Nicole RS Sibuyi, Mervin Meyer, Abram M Madiehe, Gerald J Wyckoff, et al. The role and potential of computer-aided drug discovery strategies in the discovery of novel antimicrobials. *Computers in biology and medicine*, pp. 107927, 2024.

Jure Pražnikar. Scaling laws of graphs of 3d protein structures. *Journal of bioinformatics and computational biology*, 19(03):2050050, 2021.

Mohammed Quazi, Christina Schweikert, D Frank Hsu, Tudor Oprea, et al. Enhancing admet property models performance through combinatorial fusion analysis. 2023.

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*, pp. 21–66. Springer International Publishing, 2023b. ISBN 9783031371967. doi: 10.1007/978-3-031-37196-7_2.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

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*, 2021a.

Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pp. 11548–11558. PMLR, 2021b.

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*, pp. 5833–5834, 2023.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8): 3370–3388, 2019.

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*, pp. 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.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

A 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 that 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 & 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) evaluates the power law fit of growing graph sizes utilizing random walk sampling and exploration. Similarly, Pražnikar (2021) studies 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.

B PRELIMINARIES

B.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 u, v. 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 $\mathbf{x}_i^l \in \mathbb{R}^d$ 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 $e_{uv}^l \in \mathbb{R}^d$ are concatenated into the edge feature matrix $E^l = [e_{uv}^l : (u, v) \in \mathcal{E}]^\top$.

B.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 $y \in \mathbb{R}^N$ 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 $\theta \in \mathbb{R}^B$ where $B \gg |\mathcal{D}|$. 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 $\alpha \in \mathbb{R}$ 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, $|\mathcal{D}_C|$ denotes the critical size of the dataset and $\beta \in \mathbb{R}$ is a scalar constant.

B.3 PERFORMANCE METRICS

We provide detailed explanations for the metrics used for evaluating the performance of different tasks throughout this work.

- **Pearson.** The Pearson correlation coefficient measures the strength and direction of the linear relationship between two datasets. It ranges from -1 to $+1$, where 0 indicates no correlation, -1 indicates a perfect negative linear relationship, and $+1$ indicates a perfect positive linear relationship.
- **Spearman.** The Spearman correlation coefficient is a nonparametric measure of the monotonic relationship between two datasets. Similar to other correlation coefficients, Spearman's correlation varies between -1 and $+1$, with 0 indicating no monotonic relationship. Correlations of -1 or $+1$ imply a perfect monotonic relationship between the two datasets.
- **AUROC.** The ROC curve demonstrates the trade-off between the true-positive rate and the false-positive rate at different thresholds. The area under the ROC curve (AUROC) expresses how good a model is regardless of the chosen threshold.
- **AUPRC.** This metric is useful particularly when dealing with imbalanced datasets and it summarizes the area under the precision-recall curve at various thresholds.
- **MAE.** Absolute error is the magnitude of the difference between a prediction and the true value of an observation. Mean Absolute Error (MAE) calculates the average of these

absolute errors for a group of predictions and observations, providing a measure of the overall error magnitude for the group.

C 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 behaviors 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.

D Additional Related Work

D.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.

D.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. (2021a) 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.

E TRAINING CURVES OF PRETRAINING MODELS

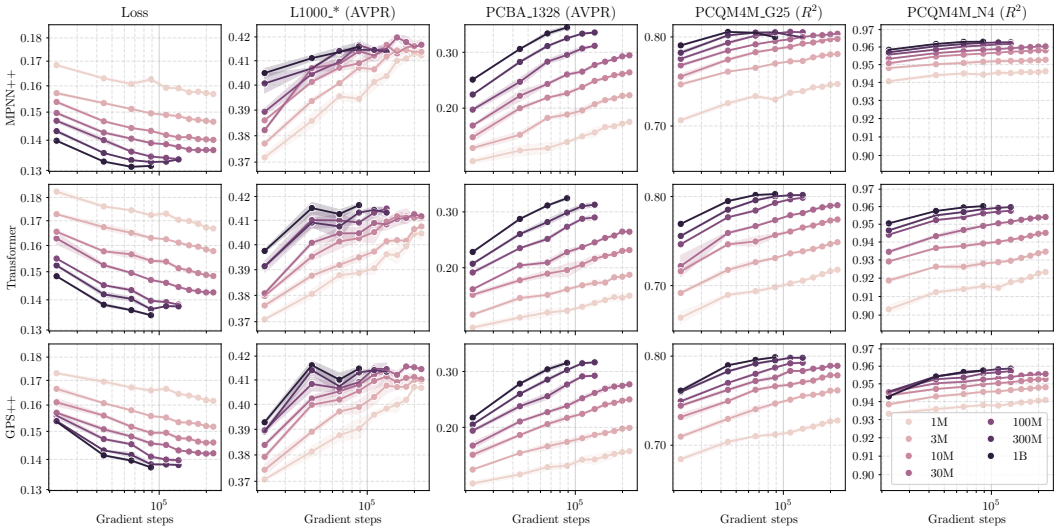


Figure 4: Model performance on the test set throughout training for MPNN++, Transformer, and GPS++ architectures with width scaling. Different colors represent models with varying number of parameters.

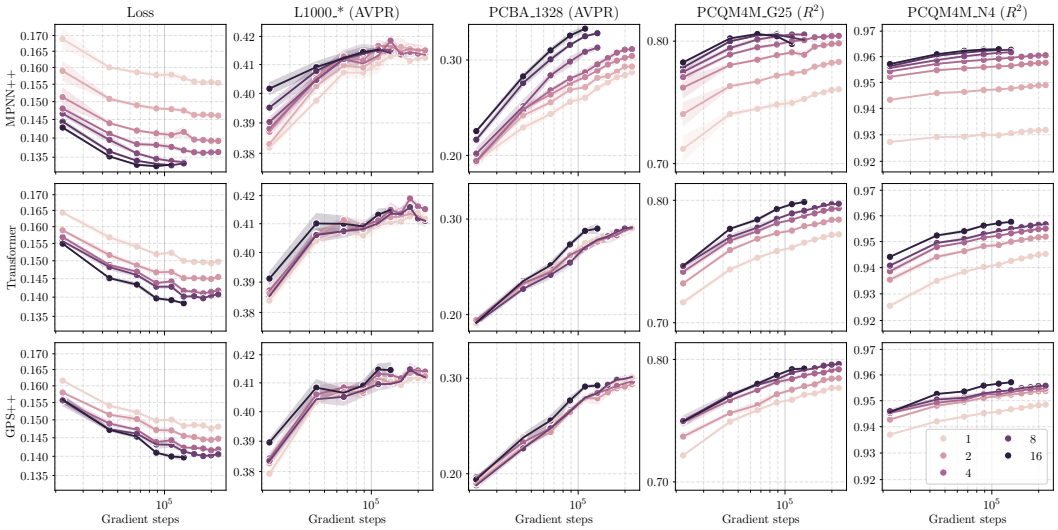


Figure 5: Model performance on the test set throughout training for MPNN++, Transformer, and GPS++ architectures with depth scaling. Different colors represent models with varying number of network layers.

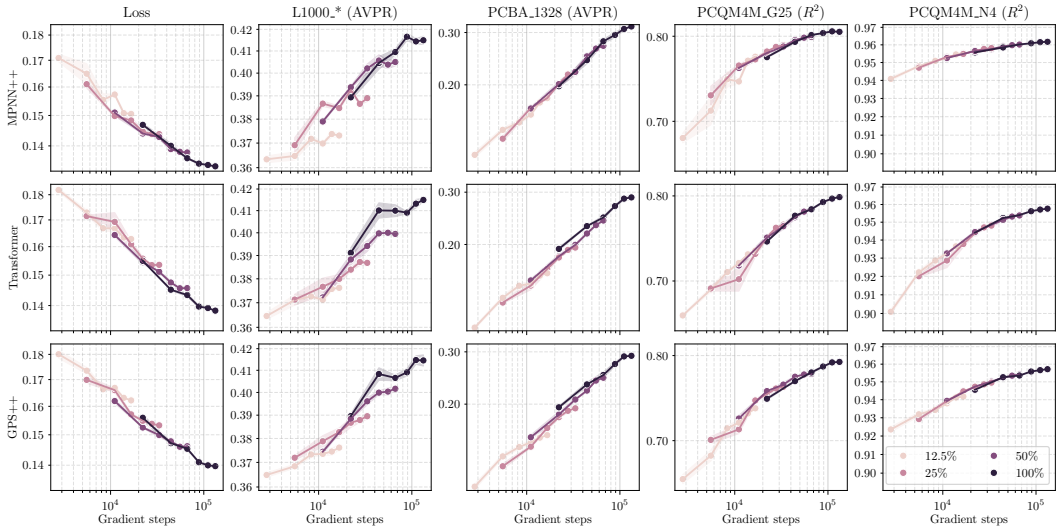


Figure 6: Model performance on the test set throughout training for MPNN++, Transformer, and GPS++ architectures with molecule scaling. Different colors represent models with varying fraction of molecules used for training.

F Additional Results on Downstream Tasks

F.1 Width Scaling

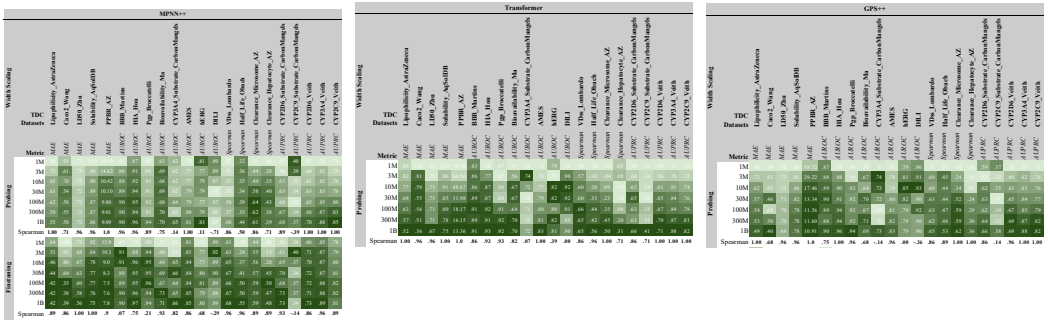


Figure 7: Comparison of probing and finetuning for MPNN++ **(left)**, Transformer **(center)**, and hybrid GPS++ **(right)** across different model sizes on the TDC benchmark. **Darker green** shades denote higher/desirable metric values. Average Spearman correlations for MPNN++, Transformer and GPS++ models show improving scaling behavior with increasing number of parameters across the TDC benchmark. The average Spearman correlation between width and performance for probing is 0.69, 0.82 and 0.73, respectively, and 0.72 when finetuning MPNN++, effectively showing that model size plays an important role in predictive performance.

F.2 DEPTH SCALING

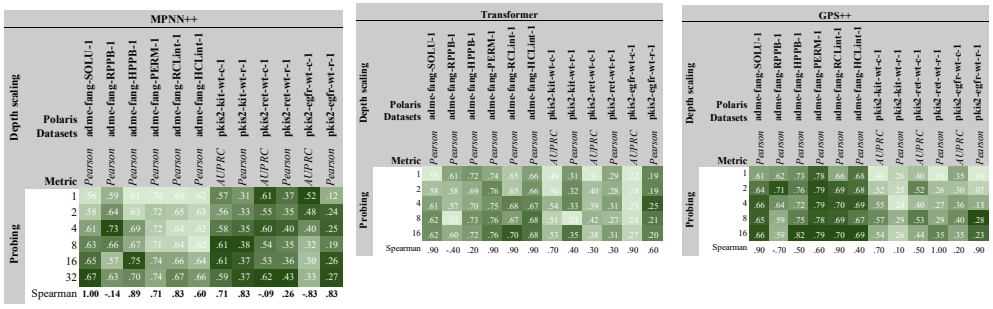


Figure 8: Comparison of probing and finetuning for different model depths on the Polaris benchmark with MPNN++ **(left)**, Transformer **(center)**, and hybrid GPS++ **(right)**. **Darker green** shades denote higher/desirable metric values. Average Spearman correlation between depth and performance is 0.47, 0.55, and 0.50, respectively. Probing shows positive scaling trend with increasing depth across the Polaris benchmark.

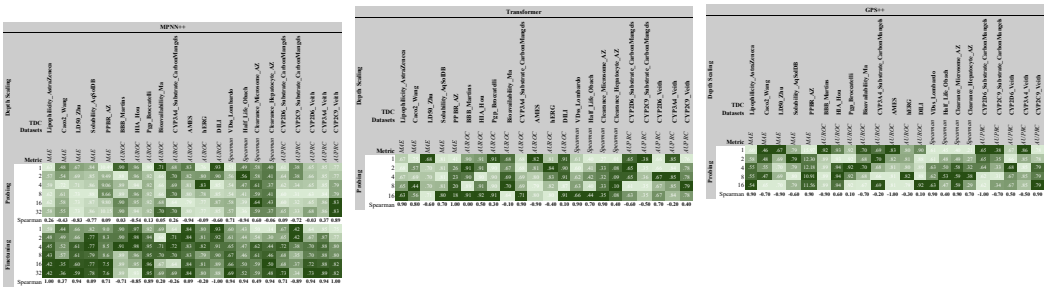


Figure 9: Comparison of probing and finetuning for MPNN++ **(left)**, Transformer **(center)**, and GPS++ **(right)** models across different model depths on the TDC benchmark. **Darker green** shades denote higher/desirable metric values. Average Spearman correlation between depth and performance is -0.11, 0.27 and -0.03, respectively, and 0.33 for finetuned MPNN++. While performance mostly increases with depths up to 8, larger depths often show signs of saturation.

F.3 MOLECULE SCALING

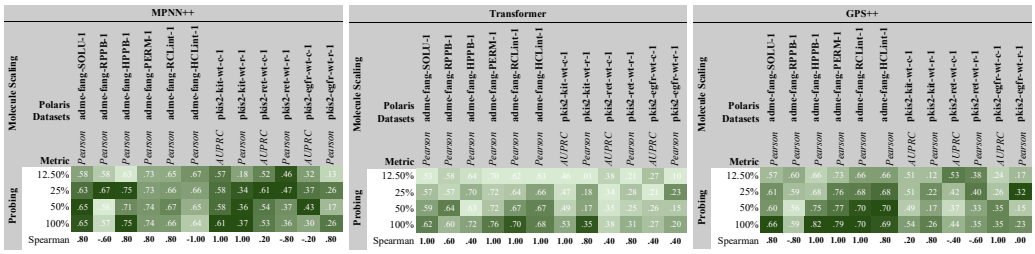


Figure 10: Scaling behavior of probed 100M parameter models across different dataset molecule fractions on the Polaris benchmark with MPNN++ **(left)**, Transformer **(center)**, and hybrid GPS++ **(right)**. **Darker green** shades denote higher/desirable metric values. Average Spearman correlation between molecule fraction and probing performance is 0.30, 0.73, and 0.40, respectively. Models show consistent improvement in performance with the increasing size of datasets.

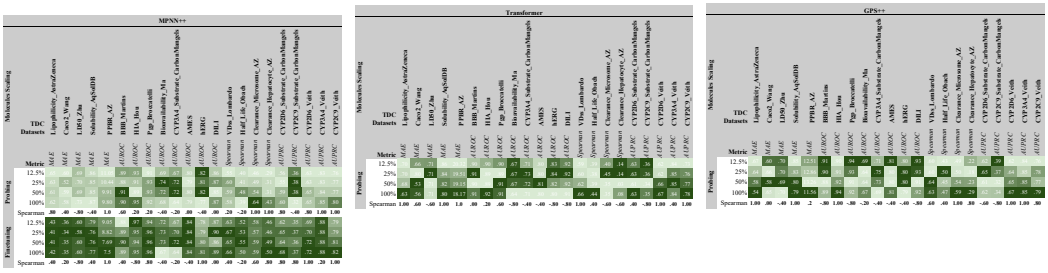


Figure 11: Comparison of probing and finetuning for MPNN++ **(left)**, Transformer **(center)** and GPS++ **(right)** models across different dataset sizes on the TDC benchmark. **Darker green** shades denote higher/desirable metric values. Average Spearman correlation between molecule fraction and probing performance is 0.28, 0.13 and 0.15, respectively, and 0.32 for finetuned MPNN++. 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.

F.4 Label Scaling

MPNN++

Label 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
	Metric	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*
Probing	12.50%	.55	.49	.50	.65	.57	.55	.53	.37	.44	.42	.30	.16
	25%	.64	.63	.70	.69	.62	.61	.52	.41	.49	.38	.33	.22
	50%	.66	.56	.73	.74	.66	.64	.59	.36	.46	.45	.39	.29
	100%	.65	.57	.75	.74	.66	.64	.61	.37	.53	.36	.30	.26
	Spearman	**.80**	**.40**	**1.00**	**.80**	**1.00**	**1.00**	**.80**	**-.60**	**.80**	**-.40**	**.40**	**.80**

Figure 12: Performance of MPNN++ probed models across different label fractions in the Polaris benchmark. **Darker green** shades denote higher/desirable metric values. Average Spearman correlation between label fraction and performance is 0.57.

MPNN++

Label scaling	TDC Datasets	Lipophilicity_AstraZeneca	Caco2_Wang	LD50_Zhu	Solubility_AqSolDB	PPBR_AZ	BBB_Martins	HIA_Hou	Pgp_Broccatelli	Bioavailability_Ma	CYP3A4_Substrate_CarbonMangels	AMES	hERG	DILI	VDss_Lombardo	Half_Life_Obach	Clearance_Microsome_AZ	Clearance_Hepatocyte_AZ	CYP2D6_Substrate_CarbonMangels	CYP2C9_Substrate_CarbonMangels	CYP2D6_Veith	CYP3A4_Veith	CYP2C9_Veith
	Metric	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*Spearman*	*Spearman*	*Spearman*	*Spearman*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*
Probing	12.5%	.70	.55	.73	.88	10.27	.82	.93	.91	.68	.67	.79	.83	.82	.63	.44	.34	.27	.63	.33	.62	.83	.78
	25%	.63	.55	.74	.92	9.89	.89	.94	.90	.67	.68	.80	.80	.85	.54	.42	.59	.33	.62	.33	.62	.83	.78
	50%	.59	.50	.74	.88	9.79	.89	.95	.91	.70	.71	.79	.80	.85	.55	.37	.53	.34	.62	.32	.65	.84	.81
	100%	.62	.58	.73	.87	9.80	.90	.95	.92	.68	.64	.79	.77	.87	.58	.39	.64	.43	.60	.32	.65	.85	.80
	Spearman	**.80**	**-.51**	**1.00**	**1.00**	**.8**	**.40**	**1.00**	**.80**	**.80**	**-.20**	**.80**	**-.80**	**1.00**	**1.00**	**.40**	**.40**	**1.00**	**-.80**	**.40**	**.80**	**1.00**	**.80**
Finetuning	12.5%	.44	.39	.62	.80	8.32	.89	.96	.94	.68	.70	.82	.81	.88	.65	.44	.60	.46	.72	.38	.70	.86	.79
	25%	.43	.42	.63	.79	7.91	.89	.96	.94	.70	.70	.84	.81	.90	.66	.46	.60	.43	.68	.39	.70	.87	.81
	50%	.42	.39	.61	.78	8.01	.89	.96	.95	.72	.74	.83	.82	.89	.67	.47	.59	.47	.72	.36	.72	.88	.83
	100%	.42	.35	.60	.77	7.5	.89	.95	.96	.67	.64	.84	.81	.89	.66	.50	.59	.50	.68	.37	.72	.88	.82
	Spearman	**.80**	**.80**	**.80**	**1.00**	**.8**	**-.80**	**-.20**	**.80**	**-.20**	**-.40**	**.40**	**.60**	**.40**	**.80**	**1.00**	**-1.00**	**.80**	**-.40**	**-.60**	**1.00**	**1.00**	**.80**

Figure 13: Comparison of MPNN++ probing and finetuning 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 MPNN++ and 0.37 for finetuned MPNN++. Finetuned models scale better when compared to probed models. Increasing label fractions do not deteriorate model performance.

F.5 DATASET ABLATION

MPNN++													
Dataset 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	pki2-kit-wt-c-1	pki2-kit-wt-r-1	pki2-ret-wt-c-1	pki2-ret-wt-r-1	pki2-egfr-wt-c-1	pki2-egfr-wt-r-1
	Metric	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*
Probing	No L1000_*	.67	.75	.75	.76	.68	.68	.61	.36	.59	.50	.51	.24
	No PCBA_1328	.45	.43	.48	.62	.56	.53	.65	.42	.56	.37	.28	.11
	No PCQM4M_N4	.64	.59	.75	.74	.66	.64	.56	.36	.71	.44	.36	.21
	Baseline	.65	.57	.75	.74	.66	.64	.61	.37	.53	.36	.30	.26

Figure 14: Comparison of probed 100M parameter MPNN++ models on the Polaris benchmark tasks (in columns) after pretrained **without** certain pretraining datasets (in rows). **Darker green** shades denote higher/desirable metric values. We observe that removing PCBA_1328 significantly hurts downstream performance across almost all tasks, while removing the L1000 leads to noticeable improvements on most tasks.

MPNN++																								
Dataset Scaling	TDC Datasets	Lipophilicity_AstraZeneca	Caco2_Wang	LD50_Zhu	Solubility_AqSolDB	PPBR_AZ	BBB_Martins	HIA_Hou	Pgp_Broccatelli	Bioavailability_Ma	CYP3A4_Substrate_CarbonMangels	AMES	hERG	DILI	VDss_Lombardo	Half_Life_Obach	Clearance_Microsome_AZ	Clearance_Hepatocyte_AZ	CYP2D6_Substrate_CarbonMangels	CYP2C9_Substrate_CarbonMangels	CYP2D6_Veith	CYP3A4_Veith	CYP2C9_Veith
	Metric	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*Spearman*	*Spearman*	*Spearman*	*Spearman*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*
Probing	No L1000_*	.52	.39	.73	.85	[illegible]	.89	.93	.93	.72	.76	.81	.81	.91	.61	.57	.58	[illegible]	.71	.42	.70	.88	.85
	No PCBA_1328	[illegible]	.62	[illegible]	.88	9.96	.87	[illegible]	[illegible]	.69	.66	[illegible]	[illegible]	.82	.31	[illegible]	[illegible]	.36	.60	[illegible]	[illegible]	[illegible]	[illegible]
	NoPCQM4M_N4	.60	.55	.73	.90	10.37	.89	.95	.90	.70	.71	.79	.80	.87	.57	.34	.62	.39	.64	.33	.67	.85	.81
	Baseline	.62	.58	.73	.87	9.80	.90	.95	.92	.68	.64	.79	.77	.87	.58	.39	.64	.43	.60	.37	.65	.85	.80

Figure 15: Performance of probed 100M parameter MPNN++ models on TDC benchmark tasks (in columns) after pretrained **without** certain datasets (in rows). **Darker green** shades denote higher/desirable metric values. We observe that removing PCBA_1328 significantly hurts downstream performance across almost all tasks, while removing the L1000 leads to noticeable improvements on most tasks.

F.6 Task-Head Ablation

MPNN++ — TDC

Metric	Lipophilicity_AstraZeneca (MAE)	Caco2_Wang (MAE)	LD50_Zhu (MAE)	Solubility_AqSolDB (MAE)	PPBR_AZ (MAE)	BBB_Martins (AUROC)	HIA_Hou (AUROC)	Pgp_Broccatelli (AUROC)	Bioavailability_Ma (AUROC)	CYP3A4_Substrate_CarbonMangels (AUROC)	AMES (AUROC)	hERG (AUROC)	DILI (AUROC)	VDss_Lombardo (Spearman)	Half_Life_Obach (Spearman)	Clearance_Microsome_AZ (Spearman)	Clearance_Hepatocyte_AZ (Spearman)	CYP2D6_Substrate_CarbonMangels (AUPRC)	CYP2C9_Substrate_CarbonMangels (AUPRC)	CYP2D6_Veith (AUPRC)	CYP3A4_Veith (AUPRC)	CYP2C9_Veith (AUPRC)
graph_output_nn	**.62**	**.58**	.73	**.87**	9.8	.90	.95	.92	**.68**	.64	**.79**	.77	**.87**	.58	**.39**	**.64**	**.43**	.60	.32	.65	.85	.80
PCBA_1328 Head-1	.65	1.09	**.73**	.88	74.0	**.90**	.85	**.93**	.68	.65	.78	.83	.87	**.62**	.04	-.13	-.21	**.67**	.34	**.68**	**.85**	**.81**
L1000_MCF7 Head-1	.72	3.82	.79	1.00	81.2	.84	.47	.89	.59	.62	.77	.83	.81	.52	-.05	-.18	-.02	.51	.25	.63	.83	.77
L1000_VCAP Head-1	.73	3.74	.77	1.01	80.5	.85	.63	.87	.58	.62	.77	**.84**	.82	.52	-.06	-.04	-.01	.48	.29	.62	.82	.76
PCQM4M_G25 Head-1	.92	2.06	.81	1.13	71.9	.81	.58	.83	.48	**.65**	.74	.75	.86	.38	-.03	-.01	.04	.49	.30	.52	.75	.70
PCBA_1328 Head-2	.81	.87	.86	1.05	**9.1**	.89	.63	.89	.57	.61	.77	.74	.80	.52	.31	.59	.30	.60	**.34**	.65	.84	.78
L1000_MCF7 Head-2	.72	.71	.78	.98	10.8	.85	**.96**	.89	.62	.59	.77	.81	.81	.53	.34	.60	.33	.57	.33	.63	.83	.79
L1000_VCAP Head-2	.73	.63	.78	.98	12.0	.88	.92	.89	.59	.63	.78	.80	.80	.59	.37	.57	.28	.62	.33	.62	.82	.77
PCQM4M_G25 Head-2	1.01	2.46	.95	1.33	30.8	.77	.50	.83	.44	.56	.72	.70	.70	.30	-.06	.17	-.03	.42	.31	.41	.69	.63

MPNN++ — Polaris

Metric	adme-fang-SOLU-1 (Pearson)	adme-fang-RPPB-1 (Pearson)	adme-fang-HPPB-1 (Pearson)	adme-fang-PERM-1 (Pearson)	adme-fang-RCLint-1 (Pearson)	adme-fang-HCLint-1 (Pearson)	pkis2-kit-wt-c-1 (AUPRC)	pkis2-kit-wt-r-1 (Pearson)	pkis2-ret-wt-c-1 (AUPRC)	pkis2-ret-wt-r-1 (Pearson)	pkis2-egfr-wt-c-1 (AUPRC)	pkis2-egfr-wt-r-1 (Pearson)
graph_output_nn	.66	.31	.49	**.76**	**.66**	.65	.55	.34	**.53**	.16	.13	.11
PCBA_1328 Head-1	**.69**	.43	**.52**	.75	.65	**.66**	.48	.05	.29	.15	.31	.08
L1000_MCF7 Head-1	.49	**.48**	.23	.73	.59	.59	.49	-.01	.28	.10	.16	.17
L1000_VCAP Head-1	.50	.45	.17	.73	.52	.56	.41	.10	.17	.18	.12	.06
PCQM4M_G25 Head-1	.11	.24	.05	.57	.37	.44	.29	.34	.18	.31	.12	.15
PCBA_1328 Head-2	.41	.32	.20	.46	.48	.42	.46	**.37**	.34	**.38**	**.31**	**.26**
L1000_MCF7 Head-2	.55	.22	.04	.69	.61	.58	**.56**	.25	.41	.31	.21	.07
L1000_VCAP Head-2	.57	.26	.22	.69	.58	.56	.55	.35	.44	.27	.21	.04
PCQM4M_G25 Head-2	.02	-.14	-.24	.46	.30	.37	.25	.35	.22	.26	.17	.17

Figure 16: Performance of probed 100M parameter MPNN++ models on TDC benchmark (left) and Polaris benchmark (right), with the tasks in columns, when probing from different task heads (in rows). **Darker green** shades denote higher/desirable metric values, and **bold/underline** indicates the best value in a given column. We observe that the *graph_output_nn* (e.g. the hidden representation that is fed to all task-heads) is overall the best choice for probing. We hypothesize this is because it captures and compresses the combined information from the various pretraining tasks. The PCBA_1328 task-head is also a good choice due to its proximity to the considered downstream tasks. The PCQM4M_G25 task-head is the least useful as the tasks are fundamentally different from the downstream tasks.

F.7 The TDC Benchmark – Data leakage

Considering that the pretraining 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.

G Scaling Law Details

We explore the power law fit which governs the scaling behavior of our GNNs using Equations 1 and 2 to compute the power law constants and identify data and parameter requirements. For metrics with higher desirable values (such as AUROC or R2), fractions inside the exponents are reversed.

Table 1 presents values of alpha (Equation 1) based on model parameters and tasks considered in our downstream task experiments on the Polaris benchmark (Figure 2b top). We choose $|\theta_c| = 1B$

and fix our final training error values corresponding to this model's performance. On average, we see $\alpha \approx 0.081$ for probing and $\alpha \approx 0.098$ for finetuning. These relationships hold across 6 orders of magnitude in $|\theta|$ indicating that finetuning behavior scales logarithmically with the number of trainable parameters. It is also worth noting that (Kaplan et al., 2020) obtain similar $\alpha \approx 0.076$ indicating that our power law fit lies within the same parameter budget.

Table 2 compares values of beta for pretraining our models as presented in Figure 2a. We choose $|\mathcal{D}_c| = 5M$ to be 100 % of molecular data. As observed, $\beta \approx 0.110$ for MPNN++, $\beta \approx 0.106$ for Transformer and $\beta \approx 0.106$ for GPS++ for the overall test error. This holds for datasets up to 5M molecules. Notably, all $\beta > 0$ indicate that, given a computational budget, larger GNNs scale favorably. We again note that Kaplan et al. (2020) have similar power law fits of $\beta \approx 0.095$ albeit with a 10^{13} token corpus. This allows us to draw two conclusions. Firstly, GNNs continue to demonstrate optimal scaling with limited datasets. Secondly, our 5M molecules (10M node features) dataset is sufficient to demonstrate scaling behavior equivalent to a 10^{13} token language corpus, up to the 1B parameters regime.

Method	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
Probing	0.055	0.053	0.041	0.034	0.049	0.053	0.083	0.147	0.064	0.078	0.112	0.206
Finetuning	0.055	0.300	0.057	0.028	0.039	0.047	0.053	0.212	0.037	0.115	0.062	0.179

Table 1: Power law constants for different downstream tasks from Polaris benchmark when varying the number of parameters (Figure 2b top).

Model	Loss	L1000 (AVPR)	PCBA_1328 (AVPR)	PCQM4M_G25 (R2)	PCQM4M_N4 (R2)
MPNN++	0.110	0.047	0.061	0.011	0.002
Transformer	0.106	0.047	0.067	0.012	0.001
GPS++	0.106	0.047	0.067	0.012	0.001

Table 2: Power law constants for pretraining different architectures when varying dataset sizes (Figure 2a).