On the Scalability of GNNs for Molecular Graphs

Maciej Sypetkowski[*]
Valence Labs, Montreal
maciej@valencelabs.com

Frederik Wenkel[*]
Valence Labs, Montreal
Université de Montréal, Mila Quebec
frederik@valencelabs.com

Farimah Poursafaei
Valence Labs, Montreal
McGill University, Mila Quebec

Nia Dickson
NVIDIA Corporation

Karush Suri
Valence Labs, Montreal

Philip Fradkin
Valence Labs, Montreal
University of Toronto, Vector Institute

Dominique Beaini
Valence Labs, Montreal
Université de Montréal, Mila Quebec

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 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 networks, graph Transformers, and hybrid architectures on the largest public collection of 2D molecular graphs for supervised pretraining. For the first time, we observe that GNNs benefit tremendously from the increasing scale of depth, width, number of molecules and associated labels. A major factor is the diversity of the pretraining data that comprises thousands of labels per molecule derived from bio-assays, quantum simulations, transcriptomics and phenomic imaging. We further demonstrate strong finetuning scaling behavior on 38 highly competitive downstream tasks, outclassing previous large models. This gives rise to *MolGPS*, a new graph foundation model that allows to navigate the chemical space, outperforming the previous state-of-the-arts on 26 out the 38 downstream tasks. 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 [47, 65] and image generation [52, 55] 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 [27]. In natural language processing, large Transformer-based models have demonstrated impressive generalization capabilities utilizing a causal autoregressive objective [51]. In the meantime, image generation has undergone incredible leaps with large models trained utilizing pixel level unsupervised objectives.

[*]Equal contribution; order determined alphabetically.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

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 [10]. 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.

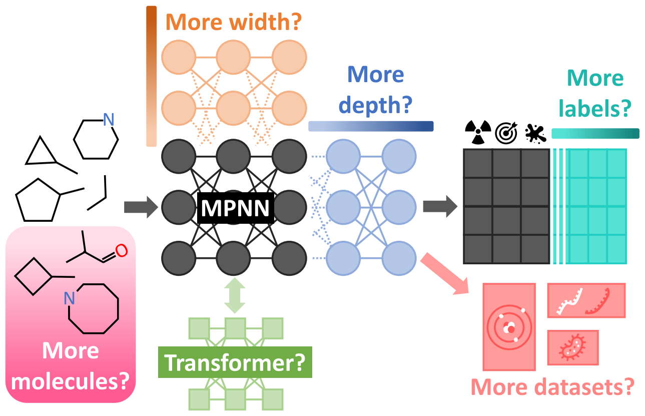


Figure 1: Summary of our GNN scaling hypotheses studied in the present work. The baseline model is presented in **dark grey**, followed by different scaling hypotheses illustrated in lighter colors. We analyze the scaling behavior of message-passing networks, graph Transformers and hybrid architectures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

Graph Neural Networks (GNNs) have emerged as a promising way of learning molecular representations [33, 17]. 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 [61], target binding interactions [62], retrosynthesis of reagents [32], ligand-based docking [13] and in-silico experiments [70].

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 graph-convolutions [29], message passing architectures [6], graph Transformers [80] and hybrid architectures [53, 39]. These approaches have shown recent progress, but their applicability to practical applications remains an open question [56].

Second, 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 [36]. Other methods such as GPSE [34] 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 [7], 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 supervised multi-task training on 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. The models are tested in 2 different settings: (1) randomly split train and test sets for pretraining and (2) finetuning/probing of pretrained models on standard benchmarks.

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

- 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 3 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 (and quality) of labels are the most important factors driving finetuning performance.
- Finally, we propose *MolGPS*, a foundation model derived from our findings on how to best scale molecular GNNs. MolGPS constitutes the most dominant model across the presented benchmarks to date, establishing state-of-the-art (SOTA) on 26/38 highly competitive downstream tasks.

2 Preliminaries

2.1 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 $\mathbf{x}_i^l \in \mathbb{R}^{d_n}$ and are concatenated into an $N \times d_n$ representation matrix $\mathbf{X}^l = [\mathbf{x}_1^l; \mathbf{x}_2^l; ...\mathbf{x}_N^l]^\top$. Edge features $e_{uv}^l \in \mathbb{R}^{d_e}$ are concatenated into the $M \times d_e$ edge feature matrix $E^l = [e_{uv}^l : (u, v) \in \mathcal{E}]^\top$.

2.2 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 $L_\theta(|\mathcal{D}|)$ [27]. 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 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.

3 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 [7]. 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++ [39] which improves quantum prediction over the MPNN [18], Graph Transformers [44], and Hybrid

GPS++ [39] along with the use of positional encodings. Finally, we evaluate our models on a range of public benchmarks with 38 datasets from TDC [25], Polaris[2], and MoleculeNet [71]. 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.

3.1 Datasets

We study the scaling behavior of GNNs primarily on the LargeMix dataset mixture [7]. 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. While most of our experiments are conducted with LargeMix, we later also explore an additional data derived from phenomic cell imaging that highlights the potential of this emerging data modality.

LargeMix. This 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 [28].
- **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 [57].

Phenomics. To exemplify how incorporating additional data can enhance LargeMix and improve the obtained pretrained models, we experimented with an additional data type that comes from phenomic imaging. The dataset contains $\sim$6k labels for $\sim$500k molecules that were derived from phenomic imaging [8] of cells perturbed with either a dose of a compound or a gene knockout. We conducted a similarity analysis between the obtained images (represented by vector embeddings Kraus et al. [30]) subject to a compound perturbation on one side, and images subject to a gene perturbation on the other side. The pretraining task is to predict for each compound if it has a phonemically visible similarity to a gene knockout (indicating a biological relationship).

3.2 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 [25] 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 [68].

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.

[2]PolarisHub: https://polarishub.io/

MoleculeNet. This is a benchmark dataset for molecular machine learning that is built upon public datasets [71]. 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 [35, 74, 34].

3.3 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 [18, 5, 9]. Choosing the MPNN++ allows us to maximize architecture expressivity while minimizing the risk of overfitting on larger datasets [39]. Each MPNN block makes use of sequential `Dropout` [59], `MLP` and `LayerNorm` [3] modules followed by a skip connection [23, 60] across node and edge features:

$$\bar{E}^l, \mathbf{X}^l = \texttt{Dropout}(\texttt{MLP}([\mathbf{X}^l|E^l])); \quad \mathbf{X}^l = \texttt{LayerNorm}(\texttt{Dropout}(\mathbf{X}^l)) + \mathbf{X}^l; \quad E^{l+1} = \bar{E}^l + E^l$$

GPS++. This is a hybrid model leveraging the MPNN++ inductive bias while providing the flexibility of self-attention-based modules [79] to allow for a rich feature extraction scheme across nodes and edges, and was empirically proven very successful [39]. 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:

$$\mathbf{X}^{l+1}, E^{l+1} = \texttt{MPNN++}(\mathbf{X}^l, E^l); \quad \mathbf{Z}^l = \texttt{BiasedAttn}(\mathbf{X}^{l+1}, \mathcal{B}); \quad \mathbf{X}^{l+1} = \texttt{MLP}(\mathbf{X}^{l+1} + \mathbf{Z}^l)$$

Transformer. This model is identical to GPS++, but without the MPNN++ module and concatenation. Instead, it relies solely on positional and structural encodings (PSEs) for structural bias.

PSEs. These are important design choices when training GNN architectures [53, 34], 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 [15] to allow one to decouple structural and positional representations. Learned positional encodings are used to tackle isomorphic nodes. Second, we use Laplacian eigenvectors [6] 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 [31] as a suitable PSE scheme to fully leverage the Laplacian spectrum. Additionally, they provide global structural information about the graph.

3.4 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 the 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 strategy for solving new 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 probing head layers. In the case of large model sizes, extracting embeddings (so-called *fingerprints*) from the frozen base model is expensive with respect to memory consumption and compute. We tackle this bottleneck by caching fingerprints 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 into the task heads.

4 Experiments

4.1 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 architectural choice, width, depth, number of molecules, labels, and different datasets. We analyze our models on datasets from LargeMix described in Section 3.1. For detailed results and experiments of our study, please refer to the supplementary material.

Overview. Figure 2 presents the variation of architectural choices between MPNN++, Transformer and GPS++, with training curves in Appendix D, and full finetuning (and probing) results in Appendix E. 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).

MPNN++ vs Transformer. MPNN++ models are more parameter efficient as they perform better with small width and depth compared to Transformers. They are also data efficient as they perform significantly better for the quantum PCQM4M_* tasks when sub-sampling the datasets, although smaller differences are observed for the remaining (biological) datasets. Transformers being "data-hungry" is consistent with recent works in domains such as natural language and computer vision [50, 2, 17]. The hybrid GPS++ benefits 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 MPNN++ models are more affected by depth, which is unsurprising considering that, contrarily to Transformers, their receptive field depends on the number of layers.

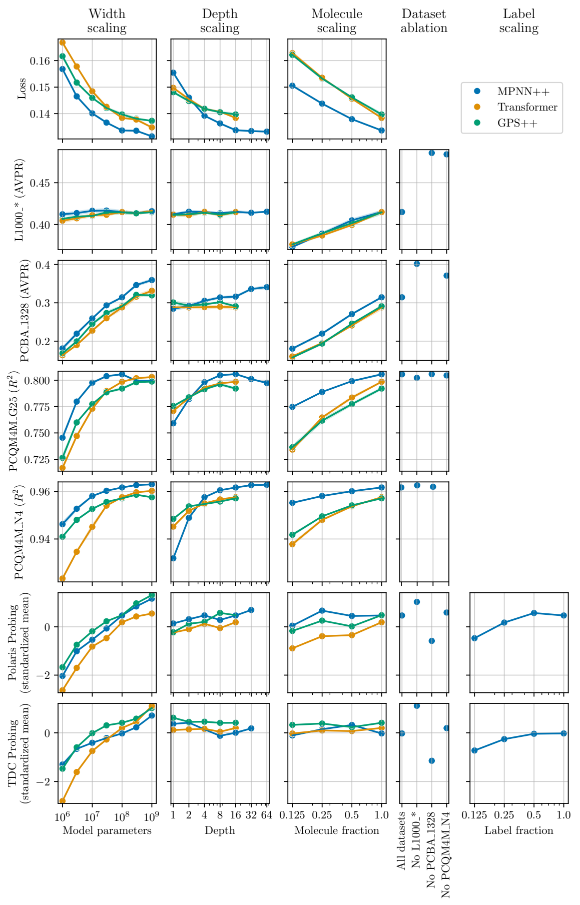


Figure 2: Effect of scaling different scaling types (columns) to test performance (rows). The *standardized mean* is calculated as mean of standardized scores for every task in a dataset group, i.e., a mean and standard deviation per task were calculated based on all our models in this study (signs of tasks with *lower is better* metrics were flipped).

Width scaling. As seen in the first column of Figure 2, 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 in 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. 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 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 [27].

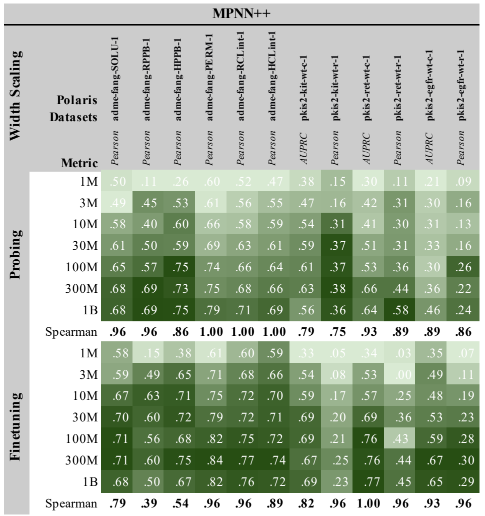
4.2 Scaling Trends on Downstream Tasks

We now evaluate scaling of models when finetuning and probing on downstream tasks. As detailed in Section 3.4, 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.

To summarize scaling trends, we compute the Spearman's rank correlation coefficient [58] 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 an important 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 more from increased width on downstream tasks compared to GPS++ and MPNN++ as seen in Figure 2 (bottom two columns). 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 (Figure 2).



		MPNN++											
Width 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	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**

Figure 3: 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.

Width scaling. We evaluate width scaling on Polaris and TDC datasets in Figure 3 (and Figure 9 in the appendix). 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 scaling studies [27]. Similarly, Transformer and GPS++ show strong positive width scaling trends.

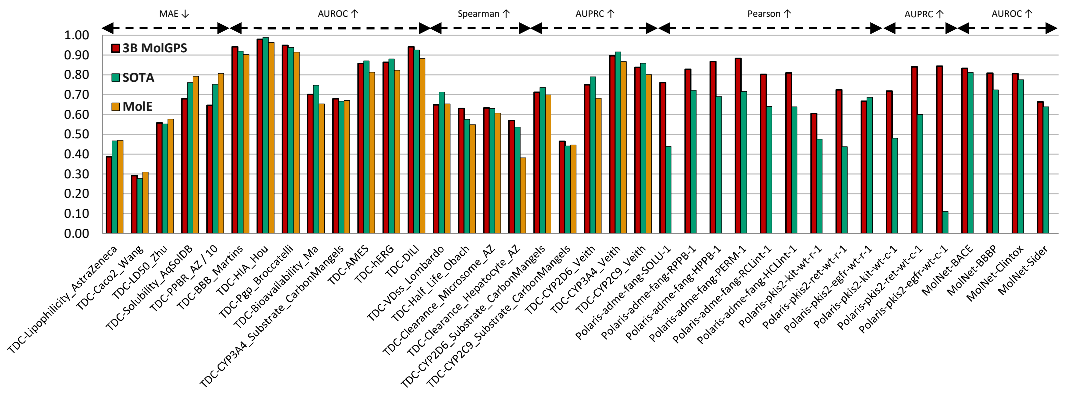


Figure 4: Comparison of our MolGPS foundation model (that combines fingerprints from the MPNN++, Transformer and hybrid GPS++ model) to the SOTA across TDC, Polaris, and MoleculeNet benchmarks. SOTA refers to the maximum value for each dataset. MolGPS establishes new SOTA on $11/22$ TDC tasks and on all but one task among Polaris and MoleculeNet.

Depth scaling. We evaluate the scaling of depth of MPNN++, GPS++ and Transformer models on the Polaris and TDC benchmarks in Figures 10 and 11 (Appendix E.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, being 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 the TDC datasets. We conjecture that degradation with depth is related to the oversmoothing issue 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 12 and 13 (Appendix E.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 14 and 15 (Appendix E.4), we observe a large 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. These stronger correlations put *label scaling* as the second-best strategy for improving the model's downstream performance.

Dataset ablation. 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*).

Observing the dataset ablations in Figure 16 and 17 (Appendix E.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 later to pretrain without L1000_*.

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 Figure 18 (Appendix E.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 bioassays. 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.

Probing vs. finetuning. So far, we have considered finetuning and probing side-by-side, establishing both as effective strategies. However, given the relatively similar performance and the significantly higher computational cost of finetuning, we find probing to be more advantageous. Another major benefit of probing is the ability to leverage multi-level information from the pretrained GNN as investigated in our task head ablation study above. We recall that our pretraining is based on a supervised multi-task learning approach. As a result, different task heads capture task-specific information, while earlier layers that feed into the task heads carry more general information. When combining several fingerprints, we can think of taking into account knowledge from several "experts".

4.3 Towards a Final Foundation Model

We now explain how the above findings can be pieced together to develop *MolGPS*, a powerful graph foundation model. Apart from scaling the model width, we found two other design choices with a major impact on the performance for the various downstream tasks. We report results on the MoleculeNet benchmark [71] here in addition to the previously used TDC and Polaris benchmarks.

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 choose probing (instead of finetuning) and further experiment with combinations of multiple fingerprints extracted at different layers of a pretrained model, which improves performance on downstream tasks. Moreover, performance can be further enhanced by combining fingerprints from multiple pretrained models.

Pretraining without L1000. Additionally, based on our observation in the dataset ablation, we pretrained new models without the L1000_* tasks, which leads to performance improvements across all scales. We hypothesize this is due to the challenging signal-to-noise ratio for those particular tasks, as also pointed out in the literature [64].

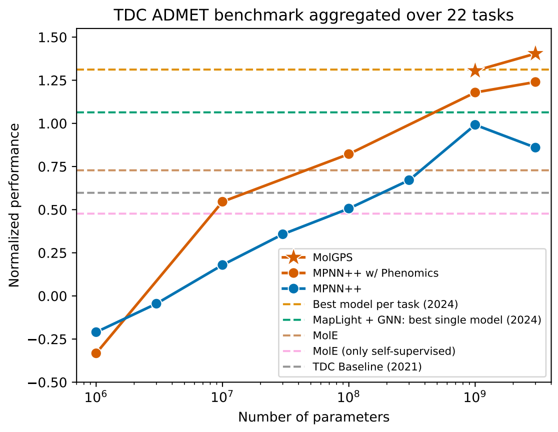


Figure 5: Comparison of our MPNN++ probing (that leverages multiple fingerprints; with and without additional phenomics pretraining) and MolGPS (that leverages fingerprints from MPNN++, Transformer and GPS++) to various baselines across TDC benchmark collection.

In Figure 5, we present multi-fingerprint probing results of the MPNN++ model. We report the normalized performance[3] across the TDC benchmark collection. We observe a strictly positive scaling trend up to 1B parameters, clearly outperforming TDC Baseline (the normalized score across the best models per task reported in [25]) and the MolE foundational model [43], a gold standard for molecular property prediction (including a model variant with *only* self-supervised pretraining). The 1B MPNN++ only slightly trails MapLight+GNN [46], the best-performing *single* model on the TDC benchmark. Surprisingly, we were unable to further scale the model to the 3B parameter regime, likely due to limitations of our pretraining data mix.

Integrating phenomics into pretraining. To improve our pretraining data mix, we experimented with an additional data type for model pretraining. The dataset contains 6k labels for 500k molecules that were derived from phenomic imaging [8] of cells perturbed with either a dose of a compound or a gene knockout. The pretraining task is to predict for each compound if it has a phenomically visible similarity to a gene knockout (indicating a biological relationship).

[3]The normalized performance of a model on a benchmark collection is calculated by first deriving the z-scores of the model's performance per task relative to the leaderboard. We flip the sign of z-scores for tasks that are "the lower the better" and then average the z-score across the tasks.

Adding phenomics data to LargeMix (without L1000), improved our downstream task performance across the board. Comparing scaling trends in Figure 5, MPNN++ with phenomics exhibits a significant vertical upwards shift compared to the original MPNN++. Notably, we were also able to extend our scaling study to the 3B parameter regime. While we were previously unable to extend the scaling trend, MPNN++ with phenomics maintains a positive scaling trend. These findings highlight the vast potential of our method as more data from different modalities becomes available.

MolGPS. We introduce a final graph foundation model that leverages the various findings of this paper. MolGPS inherits many architecture design choices from the **G**eneral, **P**owerful, **S**calable Graph Transformer method [53] and can be used to *navigate* the molecular space. Our 1B and 3B MolGPS combines fingerprints from MPNN++, Transformer, and GPS++ models (of scale 1B and 3B, respectively) that have been pretrained *with* phenomics data and *without* L1000, followed by a specialized probing `MLP`. Figure 4 compares this model across the TDC, Polaris and MoleculeNet benchmark collections to the current SOTA for each task and to MolE. MolGPS yields by far the strongest downstream task results, outperforming MolE in $21/22$ TDC tasks and establishing SOTA performance on $11/22$ TDC tasks. This makes MolGPS the model with the most SOTA entries in the TDC leaderboard followed by MapLight+GNN [46] that established SOTA on 5 TDC tasks and 7 other methods that are SOTA for at least one TDC task.[4] Similarly, compared to previous best methods on the Polaris and MoleculeNet benchmarks, we observe that our model is significantly better (often by large margins) for all but one downstream task. We primarily attribute the large-scale success to width scaling up to 3B parameters and the integration of phenomics data for pretraining. Figure 5 shows the normalized performance of MolGPS for the TDC benchmark, where it performs comparable to the best model per task for 1B parameters and clearly outperforms that baseline for 3B parameters. This is remarkable recalling that this score is derived from the best scoring method *per task* of the benchmark collection, while we use a single method for all tasks. Further comparison of MolGPS to foundation models that rely on self-supervised pretraining can be found in Appendix F.

5 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 3B parameter models, including MPNN++, Transformer, and GPS++, continue to scale favourably. More importantly, we demonstrate a consistent performance improvement on downstream property prediction tasks via finetuning and probing as we scale model and data size. Finally, we derive MolGPS, a powerful foundational model based on a multi-fingerprint probing approach that can be used to navigate the chemical space, establishing state-of-the-art on 26 out of 38 highly competitive downstream tasks. We hope that our work paves the way for the development of foundational GNNs and new architectures with applications in pharmaceutical drug discovery.

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. It will also be important to bridge current limitations for training large GNNs for molecules related to the expensive graph featurization and fast data loading.

Broader Impact. We foresee positive impacts of GNNs in areas of drug discovery, pharmaceutical advancements and tackling rare diseases by studying their molecular configurations. On the other hand, such models could also be used for harmful purposes such as developing chemical weapons and biohazards. We note that the usage of GNN models for such applications is less likely.

[4]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 [78], MapLight, MapLight+GNN [46], BaseBoosting [26], CFA [49], SimGCN, ZairaChem [48] and ContextPred [85]. For MoleculeNet, we use the same split as GraphMVP [35], thus report the best results from their table.

References

[1] A. Aghajanyan, L. Yu, A. Conneau, W.-N. Hsu, K. Hambardzumyan, S. Zhang, S. Roller, N. Goyal, O. Levy, and L. Zettlemoyer. Scaling laws for generative mixed-modal language models. *arXiv preprint arXiv:2301.03728*, 2023.

[2] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*, 2021.

[5] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[6] D. Beaini, S. Passaro, V. Létourneau, W. Hamilton, G. Corso, and P. Liò. Directional graph networks. In *International Conference on Machine Learning*, pages 748–758. PMLR, 2021.

[7] D. Beaini, S. Huang, J. A. Cunha, G. Moisescu-Pareja, O. Dymov, S. Maddrell-Mander, C. McLean, F. Wenkel, L. Müller, J. H. Mohamud, et al. Towards foundational models for molecular learning on large-scale multi-task datasets. *ICLR 2024*, 2024.

[8] M.-A. Bray, S. Singh, H. Han, C. T. Davis, B. Borgeson, C. Hartland, M. Kost-Alimova, S. M. Gustafsdottir, C. C. Gibson, and A. E. Carpenter. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nature protocols*, 11(9): 1757–1774, 2016.

[9] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[10] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[11] K. Cao, P. M. Phothilimthana, S. Abu-El-Haija, D. Zelle, Y. Zhou, C. Mendis, J. Leskovec, and B. Perozzi. Learning large graph property prediction via graph segment training. *arXiv preprint arXiv:2305.12322*, 2023.

[12] D. Chen, Y. Zhu, J. Zhang, Y. Du, Z. Li, Q. Liu, S. Wu, and L. Wang. Uncovering neural scaling laws in molecular representation learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[13] G. Corso, H. Stärk, B. Jing, R. Barzilay, and T. Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking, 2023.

[14] G. J. Dovonon, M. M. Bronstein, and M. J. Kusner. Setting the record straight on transformer oversmoothing. *arXiv preprint arXiv:2401.04301*, 2024.

[15] V. P. Dwivedi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson. Graph neural networks with learnable structural and positional representations. *CoRR*, abs/2110.07875, 2021.

[16] C. Fifty, J. Leskovec, and S. Thrun. In-context learning for few-shot molecular property prediction. *arXiv preprint arXiv:2310.08863*, 2023.

[17] M. Galkin, X. Yuan, H. Mostafa, J. Tang, and Z. Zhu. Towards foundation models for knowledge graph reasoning. *arXiv preprint arXiv:2310.04562*, 2023.

[18] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[19] J. Godwin, M. Schaarschmidt, A. Gaunt, A. Sanchez-Gonzalez, Y. Rubanova, P. Veličković, J. Kirkpatrick, and P. Battaglia. Simple gnn regularisation for 3d molecular property prediction & beyond. *arXiv preprint arXiv:2106.07971*, 2021.

[20] Z. Guo, C. Zhang, W. Yu, J. Herr, O. Wiest, M. Jiang, and N. V. Chawla. Few-shot graph learning for molecular property prediction. In *Proceedings of the web conference 2021*, pages 2559–2567, 2021.

[21] W. L. Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] D. Hernandez, J. Kaplan, T. Henighan, and S. McCandlish. Scaling laws for transfer, 2021.

[25] K. Huang, T. Fu, W. Gao, Y. Zhao, Y. Roohani, J. Leskovec, C. W. Coley, C. Xiao, J. Sun, and M. 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.

[26] D. Huang12, S. R. Chowdhuri13, A. Li13, A. Agrawal14, K. Gano15, and A. Zhu. A unified system for molecular property predictions: Oloren chemengine and its applications.

[27] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models, 2020.

[28] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–D1380, 2023.

[29] T. N. Kipf and M. 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.

[30] O. Kraus, K. Kenyon-Dean, S. Saberian, M. Fallah, P. McLean, J. Leung, V. Sharma, A. Khan, J. Balakrishnan, S. Celik, et al. Masked autoencoders for microscopy are scalable learners of cellular biology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11757–11768, 2024.

[31] D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, and P. Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34: 21618–21629, 2021.

[32] C.-H. Liu, M. Korablyov, S. Jastrzębski, P. Włodarczyk-Pruszyński, Y. Bengio, and M. H. Segler. Retrognn: Approximating retrosynthesis by graph neural networks for de novo drug design. *arXiv preprint arXiv:2011.13042*, 2020.

[33] J. Liu, C. Yang, Z. Lu, J. Chen, Y. Li, M. Zhang, T. Bai, Y. Fang, L. Sun, P. S. Yu, et al. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829*, 2023.

[34] R. Liu, S. Cantürk, O. Lapointe-Gagné, V. Létourneau, G. Wolf, D. Beaini, and L. Rampášek. Graph positional and structural encoder, 2023.

[35] S. Liu, H. Wang, W. Liu, J. Lasenby, H. Guo, and J. Tang. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021.

[36] Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, and S. Y. Philip. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5879–5900, 2022.

[37] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), Sept. 2022. ISSN 1477-4054. doi: 10.1093/bib/bbac409.

[38] A. Madani, B. Krause, E. R. Greene, S. Subramanian, B. P. Mohr, J. M. Holton, J. L. Olmos Jr, C. Xiong, Z. Z. Sun, R. Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pages 1–8, 2023.

[39] D. Masters, J. Dean, K. Klaser, Z. Li, S. Maddrell-Mander, A. Sanders, H. Helal, D. Beker, L. Rampášek, and D. Beaini. Gps++: An optimised hybrid mpnn/transformer for molecular property prediction. *arXiv preprint arXiv:2212.02229*, 2022.

[40] M. Moret, L. Friedrich, F. Grisoni, D. Merk, and G. Schneider. Generative molecular design in low data regimes. *Nature Machine Intelligence*, 2(3):171–180, 2020.

[41] M. Moret, I. Pachon Angona, L. Cotos, S. Yan, K. Atz, C. Brunner, M. Baumgartner, F. Grisoni, and G. Schneider. Leveraging molecular structure and bioactivity with chemical language models for de novo drug design. *Nature Communications*, 14(1):114, 2023.

[42] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. 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.

[43] O. Méndez-Lucio, C. Nicolaou, and B. Earnshaw. Mole: a molecular foundation model for drug discovery, 2022.

[44] L. Müller, M. Galkin, C. Morris, and L. Rampášek. Attending to graph transformers. *arXiv preprint arXiv:2302.04181*, 2023.

[45] E. Nijkamp, J. A. Ruffolo, E. N. Weinstein, N. Naik, and A. Madani. Progen2: exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978, 2023.

[46] J. H. Notwell and M. W. Wood. Admet property prediction through combinations of molecular fingerprints. *arXiv preprint arXiv:2310.00174*, 2023.

[47] OpenAI. Gpt-4 technical report, 2023.

[48] S. O. Oselusi, P. Dube, A. I. Odugbemi, K. A. Akinyede, T. L. Ilori, E. Egieyeh, N. R. Sibuyi, M. Meyer, A. M. Madiehe, G. 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*, page 107927, 2024.

[49] M. Quazi, C. Schweikert, D. F. Hsu, T. Oprea, et al. Enhancing admet property models performance through combinatorial fusion analysis. 2023.

[50] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[51] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[52] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. *ICML 2021*, 2021.

[53] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35: 14501–14515, 2022.

[54] R. M. Rao, J. Meier, T. Sercu, S. Ovchinnikov, and A. Rives. Transformer protein language models are unsupervised structure learners. *bioRxiv*, 2020. doi: 10.1101/2020.12.15.422761.

[55] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. *CVPR 2022*, 2022.

[56] Y. Rong, Y. Bian, T. Xu, W. Xie, Y. Wei, W. Huang, and J. Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.

[57] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton. Materials design and discovery with high-throughput density functional theory: the open quantum materials database (oqmd). *Jom*, 65:1501–1509, 2013.

[58] P. Schober, C. Boer, and L. A. Schwarte. Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia*, 126(5):1763–1768, 2018.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15 (1):1929–1958, 2014.

[60] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[61] H. Stärk, D. Beaini, G. Corso, P. Tossou, C. Dallago, S. Günnemann, and P. Liò. 3d infomax improves gnns for molecular property prediction. *ICML 2022*, 2022.

[62] H. Stärk, O.-E. Ganea, L. Pattanaik, R. Barzilay, and T. Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. *ICML 2022*, 2022.

[63] S. Thakoor, C. Tallec, M. G. Azar, M. Azabou, E. L. Dyer, R. Munos, P. Veličković, and M. Valko. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*, 2021.

[64] X. Tong, N. Qu, X. Kong, S. Ni, K. Wang, L. Zhang, Y. Wen, S. Zhang, X. Li, and M. Zheng. Transigen: Deep representation learning of chemical-induced transcriptional profile. *bioRxiv*, pages 2023–11, 2023.

[65] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[66] H. Veith, N. Southall, R. Huang, T. James, D. Fayne, N. Artemenko, M. Shen, J. Inglese, C. P. Austin, D. G. Lloyd, et al. Comprehensive characterization of cytochrome p450 isozyme selectivity across chemical libraries. *Nature biotechnology*, 27(11):1050–1055, 2009.

[67] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *ICLR 2018*, 2017.

[68] P. Walters. We need better benchmarks for machine learning in drug discovery, 2023. 2024-01-18.

[69] H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[70] Y. Wang, Z. Li, and A. Barati Farimani. *Graph Neural Networks for Molecules*, page 21–66. Springer International Publishing, 2023. ISBN 9783031371967. doi: 10.1007/978-3-031-37196-7_2.

[71] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2): 513–530, 2018.

[72] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *ICLR*, 2019.

[73] K. Xu, M. Zhang, J. Li, S. S. Du, K.-i. Kawarabayashi, and S. Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. *ICLR*, 2021.

[74] M. Xu, H. Wang, B. Ni, H. Guo, and J. Tang. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pages 11548–11558. PMLR, 2021.

[75] M. Xu, M. Liu, W. Jin, S. Ji, J. Leskovec, and S. 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.

[76] G. Yang, E. J. Hu, I. Babuschkin, S. Sidor, X. Liu, D. Farhi, N. Ryder, J. Pachocki, W. Chen, and J. Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer, 2022.

[77] G. Yang, D. Yu, C. Zhu, and S. Hayou. Tensor programs vi: Feature learning in infinite-depth neural networks, 2023.

[78] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[79] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

[80] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

[81] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.

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

[83] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

[84] X. Zhang, L. Wang, J. Helwig, Y. Luo, C. Fu, Y. Xie, M. Liu, Y. Lin, Z. Xu, K. Yan, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*, 2023.

[85] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. 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 [82, 33] as well as knowledge retrieval [17]. Within molecular drug discovery, recent works rely on structured models of ligands [40]. Luo et al. [37] and Moret et al. [41] study a general model for protein synthesis. Rao et al. [54] construct a self-attention driven architecture for contact prediction. Madani et al. [38] learn to generate a family of functional proteins. Nijkamp et al. [45] present a class of protein-pretrained language models. Similarly, Méndez-Lucio et al. [43] 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 [7]. Our study follows up by exploring similar *molecular tasks* for property prediction.

We elaborate more on 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 [69]. Of specific interest are tasks in the quantum and molecular discovery paradigms [84] 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. [16] 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. [20] propose a similar framework making use of few-shot learning techniques resulting in a sample-efficient learning procedure. Xu et al. [75] 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. [11] 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.

Architecture Design. Recent methods in graph architecture design focus on attending to structural information across nodes [44]. Of specific interest are graph Transformer networks which extract node as well as edge information by composing sequential attention modules over graph readouts [83]. In parallel, graph attention networks model attention weights across edges of a graph [67].

While attention mechanisms have demonstrated modern progress, traditional architectures such as neural message passing [18]. On one hand, message passing provides a rich and expressive framework for constructing representations. Godwin et al. [19] 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 [63] 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 [27]. Additionally, this relationship holds during the finetuning stage [24], thus indicating a strong reliance on model parameters. Bahri et al. [4] explain this power law fit by observing learning as moving on a smooth data manifold. Aghajanyan et al. [1] 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, which can be extended to the study of scaling laws towards different training regimes (such as finetuning, downstream transfer and inference) and different problem domains (vision, language, audio, generative modelling, etc.) Our exploration of *scaling behaviors in graph networks* is motivated by the aforesaid directions.

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 [72]. Various works indicate this as a consequence of aggregation functions and other design factors involved in GNN training [21]. On the other hand, recent work argues that only specific architectures are found robust to over-smoothing when building latent representations [14]. For instance, graph Transformers exhibit over-smoothing robustness as they utilize strong inductive biases such as attention. Xu et al. [73] 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. [42] 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. [81] present the differentiable pooling module capable of pooling neighboring node features which aid in reducing noise across layer representations.

B Experimental Details

B.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 GPS++ (except for when scaling depth). To be able to tackle graph-level tasks, the outputs are aggregated to the graph level, e.g., by summing them up across the atoms of a molecule. Then, node and graph level representations go through separate 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 the design in He et al. [22].

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 [76] 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 [77]. We did not consider adjusting the residual connections.

Our base MPNN++, Transformer and hybrid GPS++ models are 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 multi-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. Most models were trained on sigle gpus but our 300M and 1B parameter models used 4 and 8 gpus, respectively.

B.2 Finetuning and Probing

Finetuning. As outlined in Section 3.4, 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 `MLPs` with a hidden dimension of 256. For each experiment, when retraining this model, we 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 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.

We finally note that all downstream task experiments were conducted on single cpus.

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.
- **Average Standardized Score.** 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 $\mathcal{S}_T := \{s_T(m) : m \in \mathcal{M}_T\}$, where $\mathcal{M}_T$ denotes the set of all methods that have been applied to task T. The aggregated score is defined as $\text{s\_agg}_T(M) := \text{sgn}(s_T) \cdot (s_T(M) - \text{mean}(\mathcal{S}_T))\,/\text{std}\,(\mathcal{S}_T)$, where $\text{sgn}(s_T)$ is the polarity of the metric, i.e., positive if "higher is better" and negative if "lower is better".

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 [21, 72]. 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 [14]. 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 Training Curves of Pretraining Models

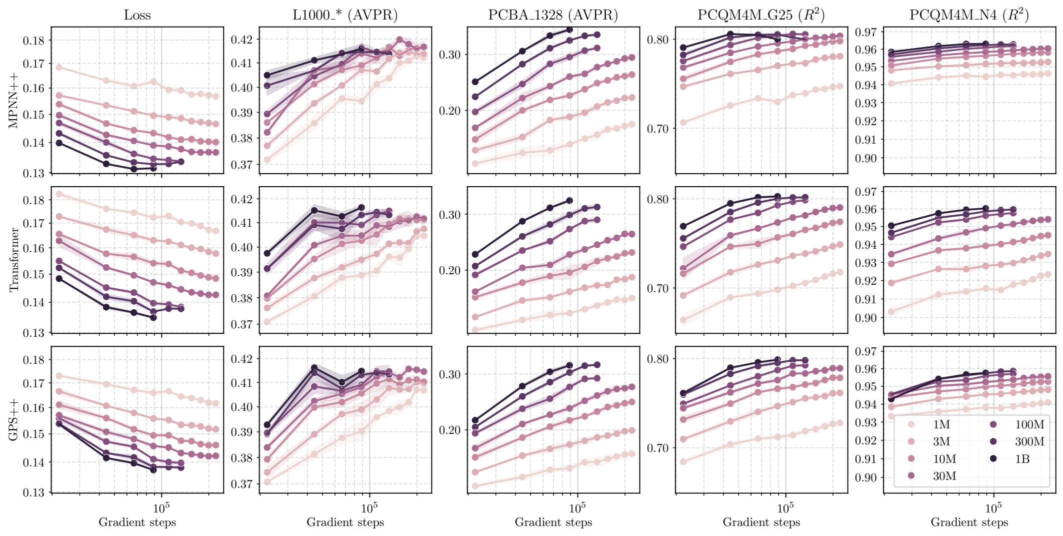


Figure 6: Model performance on the test set throughout training for MPNN++, Transformer, and GPS++ architectures with width scaling. Different colors represent models with varying number of parameters.

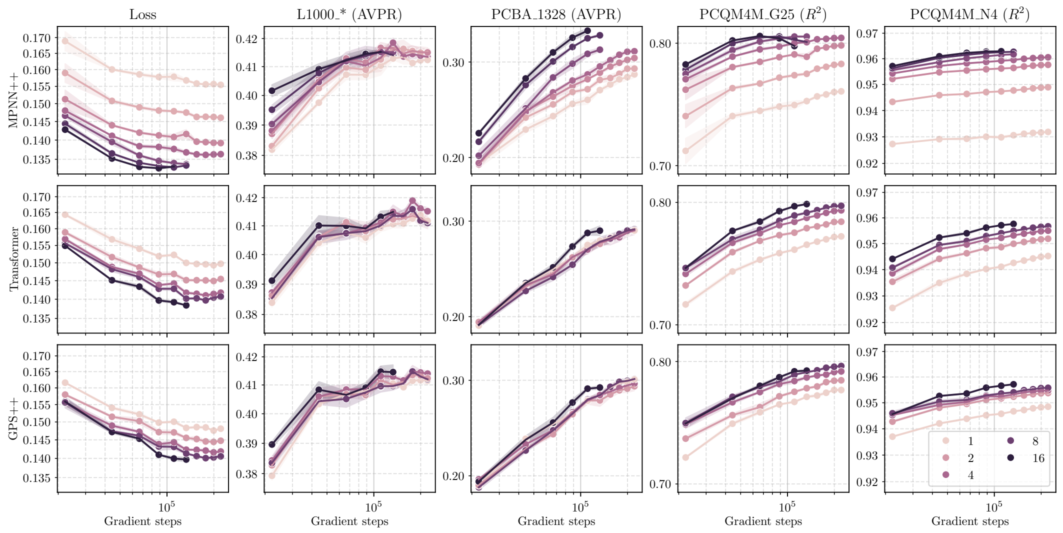


Figure 7: Model performance on the test set throughout training for MPNN++, Transformer, and GPS++ architectures with depth scaling. Different colors represent models with varying number of network layers.

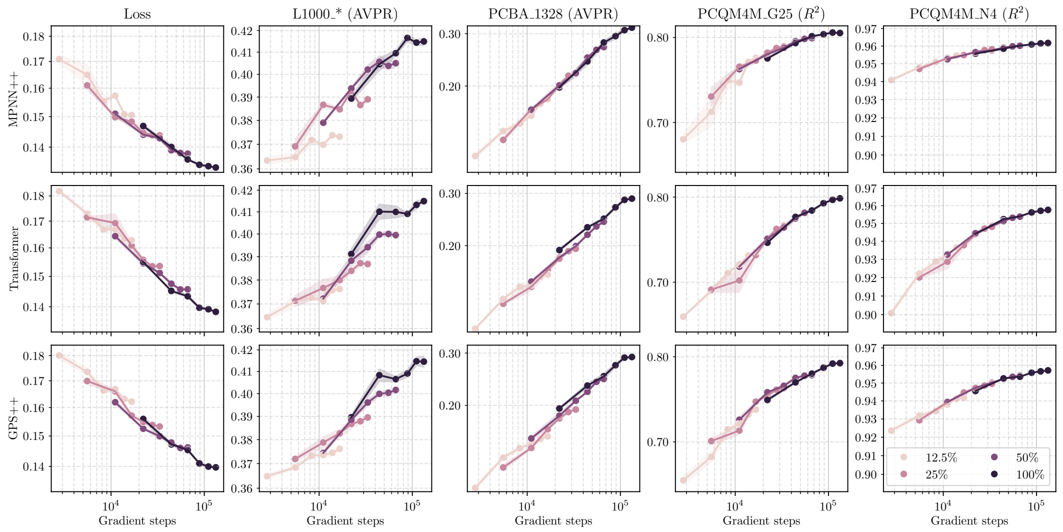


Figure 8: Model performance on the test set throughout training for MPNN++, Transformer, and GPS++ architectures with molecule scaling. Different colors represent models with varying fraction of molecules used for training.

E Additional Results on Downstream Tasks

E.1 Width Scaling

MP																	
Width Scaling	TDC Datasets	Lipophilicity_AstraZeneca	Caco2_Wang	LD50_Zhu	Solubility_AqSolDB	PPBR_AZ	BBB_Martins	HIA_Hou	Pgp_Brocatelli	Bioavailability_Ma	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*	*Spearman*	*Spearman*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*
Probing	1M	.80	.91	.77	1.03	19.19	.85	.87	.89	.63	.31	.06	.55	.40	.57	.78	.73
	3M	.72	.61	.75	.96	14.82	.88	.91	.91	.69	.44	.28	.56	.39	.60	.82	.75
	10M	.65	.70	.75	.90	10.42	.88	.92	.91	.68	.49	.35	.63	.33	.61	.82	.76
	30M	.63	.54	.72	.89	10.10	.89	.94	.91	.69	.58	.40	.63	.34	.63	.83	.78
	100M	.62	.58	.73	.87	9.80	.90	.95	.92	.68	.64	.43	.60	.32	.65	.85	.80
	300M	.59	.55	.72	.87	9.01	.90	.94	.91	.70	.62	.38	.67	.34	.68	.87	.83
	1B	.55	.58	.72	.86	9.09	.90	.96	.94	.70	.61	.39	.68	.37	.70	.88	.85
	Spearman	**1.00**	**.71**	**.96**	**.96**	**1.0**	**.96**	**.96**	**.89**	**.75**	**.86**	**.71**	**.89**	**-.39**	**1.00**	**1.00**	**1.00**
Finetuning	1M	.64	1.09	.73	.92	13.9	.85	.79	.90	.53	.44	.06	.56	.34	.66	.85	.78
	3M	.53	.93	.68	.84	10.3	.91	.88	.94	.60	.55	.15	.61	.40	.71	.87	.79
	10M	.46	.80	.67	.78	9.0	.91	.96	.95	.64	.56	.28	.65	.37	.70	.87	.80
	30M	.44	.64	.63	.77	8.3	.88	.95	.95	.69	.57	.45	.70	.34	.72	.87	.81
	100M	.42	.35	.60	.77	7.5	.89	.95	.96	.67	.59	.50	.68	.37	.72	.88	.82
	300M	.42	.38	.58	.76	7.6	.90	.96	.94	.73	.59	.47	.73	.37	.71	.88	.82
	1B	.42	.39	.56	.75	7.8	.90	.97	.94	.71	.59	.48	.73	.34	.73	.89	.81
	Spearman	**.89**	**.86**	**1.00**	**1.00**	**.9**	**.07**	**.75**	**.21**	**.93**	**.89**	**.89**	**.93**	**-.14**	**.86**	**.96**	**.89**

Transformer																							
Width 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	1M	.88	2.61	.75	1.04	74.76	.83	.58	.87	.47	.70	.75	.79	.86	.49	-.18	-.18	.05	.58	.33	.54	.76	.70
	3M	.82	.81	.75	.96	66.96	.86	.77	.87	.56	.74	.76	.78	.90	.57	-.02	-.04	.08	.60	.34	.56	.78	.72
	10M	.77	.59	.73	.91	48.67	.86	.87	.88	.67	.72	.77	.82	.92	.60	.20	.09	-.02	.62	.34	.61	.81	.74
	30M	.68	.55	.71	.85	31.98	.89	.87	.88	.67	.70	.79	.82	.92	.60	.33	.23	-.02	.65	.33	.65	.84	.76
	100M	.63	.56	.71	.80	18.17	.91	.92	.91	.64	.71	.80	.80	.91	.66	.44	.35	.08	.63	.35	.67	.84	.78
	300M	.57	.51	.71	.78	16.15	.89	.91	.92	.70	.71	.81	.82	.88	.65	.42	.45	.20	.63	.35	.70	.87	.81
	1B	.52	.36	.67	.75	13.36	.91	.93	.92	.70	.72	.83	.81	.90	.65	.56	.50	.31	.66	.41	.71	.88	.82
	Spearman	**1.00**	**.96**	**.96**	**1.00**	**1.0**	**.86**	**.93**	**.93**	**.82**	**.07**	**1.00**	**.39**	**.00**	**.86**	**.96**	**1.00**	**.71**	**.86**	**.71**	**1.00**	**1.00**	**1.00**

GPS++																							
Width Scaling	TDC Datasets	Lipophilicity_AstraZeneca	Caco2_Wang	LD50_Zhu	Solubility_AqSolDB	PPBR_AZ	BBB_Martins	HIA_Hou	Pgp_Brocatelli	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	1M	.80	.69	.76	.96	58.52	.87	.78	.90	.59	.70	.76	.79	.90	.53	.13	-.03	-.02	.59	.37	.56	.78	.71
	3M	.72	.63	.73	.91	29.22	.88	.88	.90	.67	.74	.78	.81	.91	.60	.45	.24	-.01	.53	.32	.60	.82	.74
	10M	.62	.63	.73	.86	17.46	.89	.90	.92	.64	.73	.78	.85	.93	.60	.44	.34	.01	.62	.35	.63	.83	.76
	30M	.57	.46	.71	.82	13.34	.90	.91	.92	.70	.72	.80	.82	.90	.63	.44	.52	.24	.63	.31	.65	.84	.77
	100M	.54	.65	.70	.79	11.56	.89	.94	.92	.67	.69	.81	.79	.92	.63	.47	.59	.29	.62	.34	.67	.85	.79
	300M	.53	.50	.70	.78	11.60	.88	.94	.92	.71	.70	.82	.79	.90	.62	.48	.59	.30	.64	.33	.69	.87	.82
	1B	.49	.40	.69	.78	10.91	.90	.96	.94	.69	.73	.83	.79	.90	.65	.53	.62	.36	.66	.38	.69	.88	.82
	Spearman	**1.00**	**.68**	**.96**	**.96**	**1.0**	**.75**	**1.00**	**.96**	**.68**	**-.14**	**.96**	**.00**	**-.36**	**.86**	**.89**	**1.00**	**1.00**	**.86**	**.14**	**.96**	**1.00**	**1.00**

Figure 9: **Width Scaling:** of probing and finetuning for MPNN++ **(left)**, Transformer **(center)**, and hybrid GPS++ **(right)** across different model sizes on the TDC benchma green shades denote higher/desirable metric values. Average Spearman correlations for MPNN++, Transformer and GPS++ models show improving sc r with increasing number of parameters across the TDC benchmark. The average Spearman correlation between width and performance for probing is nd 0.73, respectively, and 0.72 when finetuning MPNN++, effectively showing that model size plays an important role in predictive performance.

https://doi.org/10.52202/079017-0626

E.2 Depth Scaling

MP				
Polaris Datasets	adme-fang-SOLU-1	adme-fang-RPPB-1	adme-fang-HPPB-1	adme-fang-PERM-1
Metric	*Pearson*	*Pearson*	*Pearson*	*Pearson*
1	.56	.59	.61	.70
2	.58	.64	.63	.72
4	.61	.73	.69	.72
8	.63	.66	.67	.71
16	.65	.57	.75	.74
32	.67	.63	.70	.74
Spearman	**1.00**	**-.14**	**.89**	**.71**

Polaris Datasets	pkis2-ret-wt-c-1	pkis2-ret-wt-r-1	pkis2-egfr-wt-c-1	pkis2-egfr-wt-r-1
Metric	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*
1	.61	.37	.52	.12
2	.55	.35	.48	.24
4	.60	.40	.40	.25
8	.54	.35	.32	.19
16	.53	.36	.30	.26
32	.62	.43	.33	.27
Spearman	**-.09**	**.26**	**-.83**	**.83**

Transformer												
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*
1	.55	.61	.72	.74	.65	.66	.49	.31	.31	.29	.22	.19
2	.58	.58	.69	.76	.65	.66	.50	.32	.40	.28	.24	.19
4	.61	.57	.70	.75	.68	.67	.54	.33	.39	.31	.23	.25
8	.62	.51	.73	.76	.67	.68	.51	.23	.42	.27	.24	.21
16	.62	.60	.72	.76	.70	.68	.53	.35	.38	.31	.27	.20
Spearman	.90	-.40	.20	.90	.90	.90	.70	.40	.30	.30	.90	.60

GPS++												
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*
1	.61	.62	.73	.78	.66	.68	.49	.26	.40	.19	.35	.03
2	.64	.71	.76	.79	.69	.68	.52	.25	.52	.26	.30	.07
4	.66	.64	.72	.79	.70	.69	.55	.24	.40	.27	.36	.13
8	.65	.59	.75	.78	.69	.67	.57	.29	.53	.29	.40	.28
16	.66	.59	.82	.79	.70	.69	.54	.26	.44	.35	.35	.23
Spearman	.90	-.70	.50	.60	.90	.40	.70	.10	.50	1.00	.20	.90

Figure 10: **Depth Scaling:** n of probing for different model depths on the Polaris benchmark with MPNN++ **(left)**, Transformer **(center)**, and hybrid GPS++ **(right)**. **Darker gr** enote higher/desirable metric values. Average Spearman correlation between depth and performance is 0.47, 0.55, and 0.50, respectively. Probing show ling trend with increasing depth across the Polaris benchmark.

MP

Depth Scaling	TDC Datasets	Lipophilicity_AstraZeneca	Caco2_Wang	LD50_Zhu	Solubility_AqSolDB	PPBR_AZ	BBB_Martins	HIA_Hou	Pgp_Broccatelli	Bioavailability_Ma
	Metric	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*AUROC*	*AUROC*	*AUROC*	*AUROC*
Probing	1	.63	.48	.67	.84	10.49	.90	.96	.90	.71
	2	.57	.54	.69	.85	9.49	.88	.96	.92	.66
	4	.59	.72	.71	.86	9.06	.89	.94	.92	.66
	8	.62	.61	.73	.88	8.66	.89	.96	.92	.66
	16	.62	.58	.73	.87	9.80	.90	.95	.92	.68
	32	.58	.55	.73	.86	10.15	.90	.94	.92	.70
	Spearman	**0.26**	**-0.43**	**-0.83**	**-0.77**	**0.09**	**0.03**	**-0.54**	**0.13**	**0.05**
Finetuning	1	.59	.44	.66	.82	9.0	.90	.97	.92	.69
	2	.48	.49	.66	.77	8.3	.90	.98	.94	.66
	4	.45	.52	.61	.77	8.5	.91	.98	.95	.71
	8	.43	.57	.61	.79	8.6	.89	.96	.95	.70
	16	.42	.35	.60	.77	7.5	.89	.95	.96	.67
	32	.42	.36	.59	.78	7.6	.89	.93	.95	.69
	Spearman	**1.00**	**0.37**	**0.94**	**0.09**	**0.71**	**-0.71**	**-0.85**	**0.89**	**0.20**

	Clearance_Microsome_AZ	Clearance_Hepatocyte_AZ	CYP2D6_Substrate_CarbonMangels	CYP2C9_Substrate_CarbonMangels	CYP2D6_Veith	CYP3A4_Veith	CYP2C9_Veith
	Spearman	*Spearman*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*
	.58	.40	.61	.38	.65	.85	.77
	.58	.41	.64	.38	.66	.85	.77
	.61	.37	.62	.34	.65	.85	.79
	.59	.41	.60	.31	.63	.84	.79
	.64	.43	.60	.32	.65	.86	.83
	.59	.37	.65	.33	.68	.86	.83
4	**0.60**	**-0.06**	**0.09**	**-0.72**	**-0.03**	**0.37**	**0.89**
	.50	.14	.67	.42	.64	.85	.75
	.54	.30	.65	.42	.67	.87	.77
	.62	.44	.72	.38	.70	.88	.80
	.61	.46	.68	.35	.70	.88	.80
	.59	.50	.68	.37	.72	.88	.82
	.59	.48	.73	.34	.73	.89	.82
4	**0.49**	**0.94**	**0.71**	**-0.89**	**0.94**	**0.94**	**1.00**

Transformer

Depth 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	1	.67	.75	.68	.81	41	.90	.91	.91	.68	.68	.82	.81	.91	.61	.40	.27	.01	.65	.38	.66	.85	.76
	2	.68	.57	.70	.81	26	.91	.91	.90	.67	.68	.81	.84	.90	.60	.41	.33	.08	.65	.35	.66	.84	.78
	4	.67	.69	.70	.81	23	.90	.90	.90	.69	.69	.80	.83	.91	.62	.42	.32	.09	.65	.36	.67	.85	.78
	8	.65	.44	.70	.81	20	.89	.91	.90	.70	.69	.79	.82	.87	.63	.40	.33	.10	.61	.35	.67	.85	.79
	16	.63	.56	.71	.80	18	.91	.92	.91	.64	.71	.80	.80	.91	.66	.44	.35	.08	.63	.35	.67	.84	.78
	Spearman	**0.90**	**0.80**	**-0.60**	**0.70**	**1.00**	**0.00**	**0.50**	**0.30**	**-0.10**	**0.90**	**-0.90**	**-0.40**	**0.10**	**0.90**	**0.70**	**0.90**	**0.40**	**-0.60**	**-0.50**	**0.70**	**-0.20**	**0.40**

GPS++

Depth 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	1	.60	.46	.67	.79	15.97	.92	.93	.92	.70	.69	.83	.80	.90	.61	.45	.40	.21	.65	.38	.67	.86	.78
	2	.58	.48	.69	.79	12.30	.89	.93	.92	.68	.70	.82	.81	.88	.61	.48	.49	.27	.65	.35	.66	.85	.78
	4	.55	.55	.70	.79	12.18	.89	.94	.92	.70	.68	.81	.80	.89	.63	.50	.58	.32	.64	.33	.68	.85	.79
	8	.55	.47	.69	.80	10.91	.89	.94	.92	.68	.66	.81	.82	.88	.62	.53	.59	.38	.62	.31	.67	.85	.79
	16	.54	.65	.70	.79	11.56	.89	.94	.92	.67	.69	.81	.79	.92	.63	.47	.59	.29	.62	.34	.67	.85	.79
	Spearman	**0.90**	**-0.70**	**-0.90**	**-0.60**	**0.90**	**-0.90**	**0.60**	**0.10**	**-0.70**	**-0.20**	**-1.00**	**-0.30**	**0.10**	**0.90**	**0.40**	**0.90**	**0.70**	**-1.00**	**-0.70**	**0.50**	**-0.50**	**0.90**

Figure 11: **Depth Scaling:** … n of probing and finetuning for MPNN++ **(left)**, Transformer **(center)**, and GPS++ **(right)** models across different model depths on the TDC benchm… **green** shades denote higher/desirable metric values. Average Spearman correlation between depth and probing performance is -0.11, 0.27 and -0.03, res… d 0.33 for finetuned MPNN++. While performance mostly increases with depths up to 8, larger depths often show signs of saturation.

https://doi.org/10.52202/079017-0626

E.3 Molecule Scaling

MP				
Polaris Datasets	adme-fang-SOLU-1	adme-fang-RPPB-1	adme-fang-HPPB-1	adme-fang-PERM-1
Metric	*Pearson*	*Pearson*	*Pearson*	*Pearson*
12.50%	.58	.58	.63	.73
25%	.63	.67	.75	.73
50%	.65	.56	.71	.74
100%	.65	.57	.75	.74
Spearman	**.80**	**-.60**	**.80**	**.80**

Polaris Datasets	pkis2-ret-wt-c-1	pkis2-ret-wt-r-1	pkis2-egfr-wt-c-1	pkis2-egfr-wt-r-1
Metric	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*
12.50%	.52	.46	.32	.13
25%	.61	.47	.37	.26
50%	.54	.37	.43	.17
100%	.53	.36	.30	.26
Spearman	**.20**	**-.80**	**-.20**	**.80**

Transformer												
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*
12.50%	.53	.58	.64	.70	.62	.63	.46	.01	.38	.21	.27	.10
25%	.57	.57	.70	.72	.64	.66	.47	.18	.34	.28	.21	.23
50%	.59	.64	.63	.72	.67	.67	.49	.17	.35	.25	.26	.15
100%	.62	.60	.72	.76	.70	.68	.53	.35	.38	.31	.27	.20
Spearman	**1.00**	**.60**	**.40**	**1.00**	**1.00**	**1.00**	**1.00**	**.80**	**.40**	**.80**	**.40**	**.40**

GPS++												
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*
12.50%	.57	.60	.66	.73	.66	.66	.51	.12	.53	.38	.24	.17
25%	.61	.59	.68	.76	.68	.68	.51	.22	.42	.40	.26	.32
50%	.60	.56	.75	.77	.70	.70	.49	.17	.37	.33	.35	.15
100%	.66	.59	.82	.79	.70	.69	.54	.26	.44	.35	.35	.23
Spearman	**.80**	**-.80**	**1.00**	**1.00**	**1.00**	**.80**	**.20**	**.80**	**-.40**	**-.60**	**1.00**	**.00**

Figure 12: **Molecule Scal** behavior of probed 100M parameter models across different dataset molecule fractions on the Polaris benchmark with
MPNN++ **(left)**, Transforr and hybrid GPS++ **(right)**. **Darker green** shades denote higher/desirable metric values. Average Spearman correlation
between molecule fraction performance is 0.30, 0.73, and 0.40, respectively. Models show consistent improvement in performance with the increasing
size of datasets.

Molecules Scaling	TDC Datasets	Lipophilicity_AstraZeneca	Caco2_Wang	LD50_Zhu	Solubility_AqSolDB	PPBR_AZ	BBB_Martins	HIA_Hou	Pgp_Brocatelli	Bioavailability_Ma
	Metric	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*AUROC*	*AUROC*	*AUROC*	*AUROC*
Probing	12.5%	.65	.60	.69	.86	11.05	.89	.93	.91	.69
	25%	.63	.52	.70	.85	10.44	.88	.91	.93	.74
	50%	.61	.59	.69	.85	9.91	.91	.89	.93	.72
	100%	.62	.58	.73	.87	9.80	.90	.95	.92	.68
	Spearman	**.80**	**.40**	**-.80**	**-.40**	**1.0**	**.60**	**.20**	**.20**	**-.40**
Finetuning	12.5%	.43	.36	.60	.79	9.05	.88	.97	.94	.72
	25%	.41	.34	.58	.76	8.82	.89	.95	.96	.73
	50%	.41	.35	.60	.76	7.69	.90	.94	.96	.73
	100%	.42	.35	.60	.77	7.5	.89	.95	.96	.67
	Spearman	**.40**	**.20**	**-.80**	**.40**	**1.0**	**.40**	**-.80**	**.80**	**-.40**

Clearance_Microsome_AZ	Clearance_Hepatocyte_AZ	CYP2D6_Substrate_CarbonMangels	CYP2C9_Substrate_CarbonMangels	CYP2D6_Veith	CYP3A4_Veith	CYP2C9_Veith
Spearman	*Spearman*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*
.46	.29	.56	.36	.63	.83	.76
.49	.31	.55	.38	.63	.83	.77
.54	.31	.59	.38	.65	.84	.77
.64	.43	.60	.32	.65	.85	.80
1.00	**1.00**	**.80**	**-.40**	**.80**	**1.00**	**1.00**
.58	.46	.62	.35	.69	.88	.79
.57	.46	.65	.37	.70	.88	.79
.59	.49	.64	.36	.72	.88	.81
.59	.50	.68	.37	.72	.88	.82
.60	**.80**	**.80**	**.80**	**1.00**	**.20**	**1.00**

Transformer

Molecules Scaling	TDC Datasets	Lipophilicity_AstraZeneca	Caco2_Wang	LD50_Zhu	Solubility_AqSolDB	PPBR_AZ	BBB_Martins	HIA_Hou	Pgp_Brocatelli	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	.66	.71	.86	20.32	.90	.90	.90	.67	.71	.80	.83	.92	.59	.39	.40	.14	.63	.36	.62	.84	.73
	25%	.70	.80	.71	.84	19.51	.91	.89	.89	.67	.73	.80	.84	.92	.60	.38	.45	.14	.63	.36	.62	.85	.76
	50%	.68	.53	.71	.82	19.15	.90	.89	.91	.67	.72	.81	.82	.92	.62	.39	.35	.03	.61	.34	.66	.85	.77
	100%	.63	.56	.71	.80	18.17	.91	.92	.91	.64	.71	.80	.80	.91	.66	.44	.35	.08	.63	.35	.67	.84	.78
	Spearman	**1.00**	**.60**	**-.60**	**1.00**	**1.0**	**.00**	**.20**	**.60**	**-.80**	**-.40**	**-.40**	**-.80**	**-.80**	**1.00**	**.40**	**-.60**	**-.60**	**.40**	**-.80**	**1.00**	**.40**	**1.00**

GPS++

Molecules Scaling	TDC Datasets	Lipophilicity_AstraZeneca	Caco2_Wang	LD50_Zhu	Solubility_AqSolDB	PPBR_AZ	BBB_Martins	HIA_Hou	Pgp_Brocatelli	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%	.67	.60	.70	.85	12.51	.91	.90	.94	.69	.71	.81	.80	.93	.60	.43	.49	.22	.62	.39	.62	.84	.76
	25%	.64	.66	.70	.83	12.86	.90	.91	.93	.64	.75	.80	.80	.93	.60	.50	.50	.18	.65	.37	.64	.85	.78
	50%	.58	.58	.69	.80	13.26	.88	.92	.91	.64	.73	.80	.80	.91	.64	.45	.54	.23	.61	.32	.65	.85	.77
	100%	.54	.65	.70	.79	11.56	.89	.94	.92	.67	.69	.81	.79	.92	.63	.47	.59	.29	.62	.34	.67	.85	.79
	Spearman	**1.00**	**.00**	**-.40**	**1.00**	**.2**	**-.80**	**1.00**	**-.80**	**-.20**	**-.40**	**-.40**	**-.40**	**-.80**	**.60**	**.40**	**1.00**	**.80**	**-.60**	**-.80**	**1.00**	**1.00**	**.80**

Figure 13: **Molecule Scali** son of probing and finetuning for MPNN++ **(left)**, Transformer **(center)** and GPS++ **(right)** models across different dataset sizes on the TDC benchma **green** shades denote higher/desirable metric values. Average Spearman correlation between molecule fraction and probing performance is 0.28, 0.13 spectively, and 0.32 for finetuned MPNN++. Finetuned models scale better when compared to probed models. However, increasing the size of finetu s leads to minor improvements beyond the 50% dataset size fraction.

E.4 Label Scaling

Label Scaling	MPNN++												
	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 14: **Label Scaling:** 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 probing performance is 0.57.

Label scaling	MPNN++											
	TDC Datasets	Lipophilicity_AstraZeneca	Caco2_Wang	LD50_Zhu	Solubility_AqSolDB	PPBR_AZ	BBB_Martins	HIA_Hou	Pgp_Broccatelli	Bioavailability_Ma	CYP3A4_Substrate_CarbonMangels	AMES
	Metric	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*	*AUROC*
Probing	12.5%	.70	.55	.75	.99	10.27	.87	.93	.91	.64	.67	.79
	25%	.63	.55	.74	.92	9.89	.89	.94	.90	.67	.68	.79
	50%	.59	.50	.74	.88	9.79	.87	.95	.91	.70	.71	.79
	100%	.62	.58	.73	.87	9.80	.90	.95	.92	.68	.64	.79
	Spearman	**.80**	**-.51**	**1.00**	**1.00**	**.8**	**.40**	**1.00**	**.80**	**.80**	**-.20**	**.80**
Finetuning	12.5%	.44	.39	.62	.80	8.32	.89	.96	.94	.68	.70	.82
	25%	.43	.42	.63	.79	7.91	.89	.96	.94	.70	.70	.84
	50%	.42	.39	.61	.78	8.01	.89	.96	.95	.72	.74	.83
	100%	.42	.35	.60	.77	7.5	.89	.95	.96	.67	.64	.84
	Spearman	**.80**	**.80**	**.80**	**1.00**	**.8**	**-.80**	**-.20**	**.80**	**-.20**	**-.40**	**.40**

Label scaling	MPNN++											
	TDC Datasets	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	*AUROC*	*AUROC*	*Spearman*	*Spearman*	*Spearman*	*Spearman*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*	*AUPRC*
Probing	12.5%	.83	.82	.53	.31	.54	.27	.63	.30	.57	.80	.73
	25%	.80	.85	.54	.42	.59	.33	.62	.33	.62	.83	.78
	50%	.80	.85	.55	.37	.53	.34	.62	.32	.65	.84	.81
	100%	.77	.87	.58	.39	.64	.43	.60	.32	.65	.85	.80
	Spearman	**-.80**	**1.00**	**1.00**	**.40**	**.40**	**1.00**	**-.80**	**.40**	**.80**	**1.00**	**.80**
Finetuning	12.5%	.81	.88	.65	.44	.60	.46	.72	.38	.70	.86	.79
	25%	.81	.90	.66	.46	.60	.43	.68	.39	.70	.87	.81
	50%	.82	.89	.67	.47	.59	.47	.72	.36	.72	.88	.83
	100%	.81	.89	.66	.50	.59	.50	.68	.37	.72	.88	.82
	Spearman	**.60**	**.40**	**.80**	**1.00**	**-1.00**	**.80**	**-.40**	**-.60**	**1.00**	**1.00**	**.80**

Figure 15: **Label Scaling:** 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.

E.5 Dataset Ablation

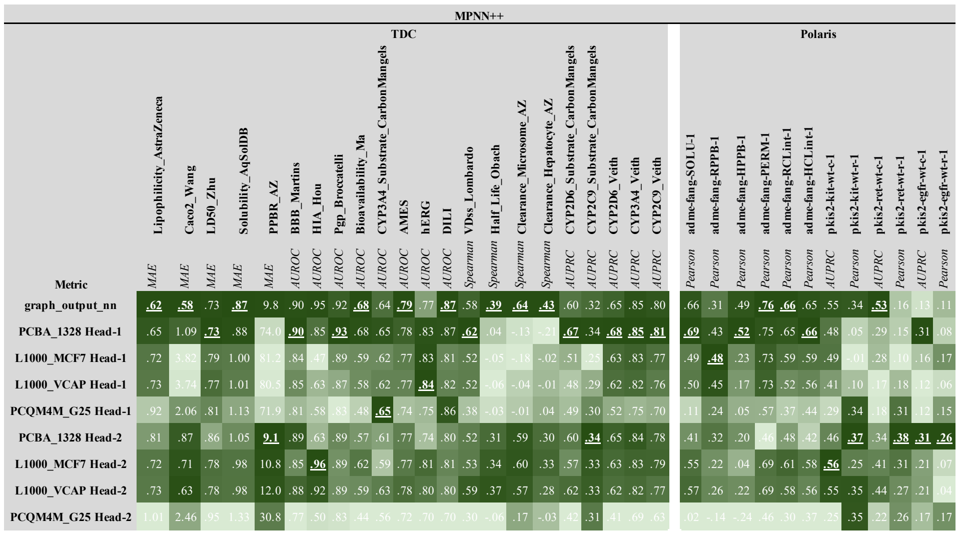
Dataset Scaling	MPNN++												
	Polaris Datasets	adme-fang-SOLU-1	adme-fang-RPPB-1	adme-fang-HPPB-1	adme-fang-PERM-1	adme-fang-RCLint-1	adme-fang-HCLint-1	pkis2-kit-wt-c-1	pkis2-kit-wt-r-1	pkis2-ret-wt-c-1	pkis2-ret-wt-r-1	pkis2-egfr-wt-c-1	pkis2-egfr-wt-r-1
	Metric	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*	*AUPRC*	*Pearson*
Probing	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 16: **Dataset Ablation:** 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.

Dataset Scaling	MPNN++																						
	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**	10.60	.89	.93	**.93**	**.72**	**.76**	**.81**	**.81**	**.91**	**.61**	**.57**	.58	.35	**.71**	**.42**	**.70**	**.88**	**.85**
	No PCBA_1328	.82	.63	.74	.95	9.96	.87	.89	.88	.69	.66	.77	.77	.82	.51	.25	.47	.36	.60	.32	.52	.79	.71
	NoPCQM4M_N4	.60	.55	.73	.90	10.37	.89	.95	.90	.70	.71	.79	.80	.87	.57	.34	.62	.39	.64	.35	.67	.85	.81
	Baseline	.62	.58	**.73**	.87	**9.80**	**.90**	**.95**	.92	.68	.64	.79	.77	.87	.58	.39	**.64**	**.43**	.60	.32	.65	.85	.80

Figure 17: **Dataset Ablation:** 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.

E.6 Task-Head Ablation



MPNN++

TDC

	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*
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

Polaris

	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*
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 18: **Task-Head Ablation:** 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.

E.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 [28], excluding it from the list of potential leaks. Regarding the 3 CYP*_Veith datasets, they represent inibition assays against recombinant enzymes [66]. 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.

Table 1: Comparison of MolGPS variants to the self-supervised GraphMVP model on the MoleculeNet dataset (test AUROC). Note that we only consider the datasets that were not part of our pretraining.

Method	BACE	BBBP	Clintox	Sider
GraphMVP	0.812	0.724	0.775	0.639
1B MolGPS w/o Phenomics	0.806	**0.802**	0.797	0.649
1B MolGP	**0.828**	0.8	**0.809**	**0.67**
3B MolGPS	**0.832**	**0.809**	**0.807**	**0.666**

F Comparison of MolGPS to Unsupervised Methods

Compared to unsupervised pretraining approaches, it is important to remark that the scale of pretraining data is not directly comparable to that of our supervised pretraining approach, i.e, billions of molecules in some cases. We note that previous works in the GNN literature [12] found interesting scaling trends despite scaling to even less than the 5M graphs that we have aggregated in the LargeMix dataset.

In our supervised context, the data scale does not only depend on the number of molecules seen during pretraining. Instead, each molecule should be considered in conjunction with the set of pretraining labels. PCBA_1328 considers more than 1k different labels per molecule (albeit with high sparsity) and PCQM4M comes with 25 graph-level tasks and 4 node-level tasks (for each node of the 4M molecules). We point to our label scaling study that shows the impact of reduced data diversity by removing labels (e.g., Figure 2) for convincing evidence for this trend. The performance gains from incorporating Phenomics data into the pretraining data mix also support this claim, where the addition of 500k molecules with 6k highly informative labels per graph (Figure 5) leads to strong performance gains.

Here, we provide further comparison to unsupervised pretraining approaches (Table 1 and 2). We observe that the unsupervised MolE model variants [43] clearly underperform the standard MolE model that leverages both unsupervised and supervised pretraining (Table 2), which is in turn outperformed by the listed MolGPS variants by large margins. Even our smaller 100M MPNN++ models (that are of comparable parametric size to MolE) both outperform the self-supervised variants (see Figure 5).

In Table 1, we compare to the self-supervised GraphMVP [35] model on MoleculeNet. MolGPS without Phenomics pretraining outperforms GraphMVP on 3/4 tasks, while the model variants that were pretrained with additional Phenomics data outperform across all tasks.

Table 2: Comparison of MolGPS variants to the self-supervised variants of MolE and the standard MolE (supervised + self-supervised pretraining) on TDC benchmark collection. Supervised pretraining significantly improves MolE performance as seen in the normalized performance (left-most column). MolGPS models outperform all MolE models by a large margin, exhibiting the best performance in all but two tasks.

		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**
Method	Norm.	MAE	MAE	MAE	MAE	MAE	AUROC	AUROC	AUROC	AUROC	AUROC	AUROC	AUROC	AUROC	Spear.	Spear.	Spear.	Spear.	AUPRC	AUPRC	AUPRC	AUPRC	AUPRC
MolE-FuncEnv (unsupervised)	0.47	0.46	0.355	0.597	0.799	8.57	0.895	0.951	0.873	0.638	0.612	0.831	**0.871**	0.89	0.622	0.579	0.567	0.373	**0.715**	0.411	0.678	0.857	0.759
MolE-AtomEnv (unsupvervised)	0.477	0.464	0.471	0.582	0.81	8.191	0.895	0.949	0.871	0.683	0.633	0.832	0.844	0.883	0.596	0.518	0.531	0.367	0.706	0.429	0.665	0.865	0.773
MolE	0.728	0.469	0.31	**0.577**	0.792	8.073	0.903	0.963	0.915	0.654	0.67	0.813	0.823	0.883	**0.654**	0.549	0.607	0.381	0.699	0.446	0.682	0.867	0.801
1B MolGPS w/o Phenomics	1.222	0.4	0.347	0.645	0.714	**6.249**	0.922	**0.984**	0.941	0.64	**0.681**	0.8389	0.86	**0.937**	**0.655**	**0.64**	**0.659**	**0.56**	0.712	**0.483**	**0.747**	**0.905**	**0.871**
1B MolGPS	**1.305**	**0.391**	**0.288**	0.589	**0.706**	6.497	**0.939**	0.975	**0.947**	**0.686**	**0.681**	**0.85**	**0.868**	0.933	0.649	**0.632**	**0.649**	0.527	0.700	**0.474**	0.741	0.898	0.832
3B MolGPS	**1.404**	**0.386**	**0.292**	**0.557**	**0.679**	**6.464**	**0.941**	**0.98**	**0.948**	**0.701**	0.68	**0.857**	0.864	**0.942**	0.649	0.631	0.633	**0.57**	**0.713**	0.464	**0.75**	**0.9**	**0.838**

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 3 presents values of α (Equation 1) based on model parameters and tasks considered in our downstream task experiments on the Polaris benchmark (Figure 3). 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. [27] obtain similar $\alpha \approx 0.076$ indicating that our power law fit lies within the same parameter budget.

Table 4 compares values of β (Equation 2) for pretraining our models as presented in Figure 2. 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. [27] 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.

Table 3: Power law constants (α in Equation 1) for different downstream tasks from Polaris benchmark when varying the number of parameters (Figure 3).

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 4: Power law constants (β in Equation 2) for pretraining different architectures when varying dataset sizes (Figure 2).

Model	Global Loss	L1000 (AVPR)	PCBA (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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main contributions introduced in the Abstract and Section 1-Introduction, are investigated and discussed throughout the paper. In short, we investigate how GNNs benefit from the increasing scale of depth, width, number of molecules, number of labels, and diversity in the pretraining datasets. These scaling trends are extensively investigated and the outcomes are reflected in Section 3. Additionally, we presented MolGPS, a graph foundation model, whose details and performance are fully presented in Section 4.3.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss some of the limitations of this work as potential directions for the future research in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: This paper primarily presents empirical results. The observed scaling trends of our work in the theoretical context of prior work in Section G with references providing the theoretical context.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: For pretraining, we use datasets and code from the literature [7]. The code can be found at `https://github.com/datamol-io/graphium`, while the data can be found at `https://zenodo.org/records/10797794`. We further provide the experimental details throughout the main text and in Section B, covering both pretraining and finetuning/probing.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use the LargeMix dataset [7], which can be found at `https://zenodo.org/records/10797794`. Links to the datasets for finetuning and probing can be found referring to the corresponding references in the paper. The experiments can be reproduced by using the code associated to [7] (`https://github.com/datamol-io/graphium`) together with the instructions in the main text and Section B of the appendix.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: For pretraining, we use the train/test proposed in [7] and similar follow the splitting conventions of the benchmark collections for finetuning/probing. Hyperparameters are discussed in the main text and in Section B of the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We conducted several iterations of each experiments and reported the average results over all runs. In most cases, we have also demonstrated the standard deviations (for example, Figure 5 (Bottom)), or as the correlations across the performance of different models. However, there are some cases that we decided to omit error bars for improving the readability of the figures and tables. Moreover, we mainly study the trends where correlation metrics do not integrate naturally with error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The omputational resources that were used are reported in Section B in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and made sure to adhere to them.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both positive and negative impacts of this work in Section 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not propose new data. We have discussed the impact of our work in Section 5. We believe that our proposed model and approach for harmful purposes is considerably less likely and it is not a direct outcome of our work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have use publicly available data [7] that are available here. These data are shared under `Creative Commons Attribution Non Commercial Share Alike 4.0 International` license. In addition, we built upon the `Graphium` library [7] whose license is based on Apache 2.0 (The Graphium License can be found here).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: We do not propose any new asset.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This question is not applicable to this work.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This question is not applicable to this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-0626