MiniMol: A Parameter-Efficient Foundation Model for Molecular Learning

Kerstin Kläser [*1] **Błażej Banaszewski** [*1] **Samuel Maddrell-Mander** [2] **Callum McLean** [1]
Luis Müller [3] **Ali Parviz** [4 5] **Shenyang Huang** [5] **Andrew Fitzgibbon** [1]

Abstract

We propose `MiniMol`, an open-source foundation model for molecular machine learning which outperforms the best previous foundation model on 17/22 downstream tasks from the Therapeutic Data Commons (TDC) ADMET group while having ten times fewer parameters. This efficiency is achieved through the use of a graph neural network (GNN), pre-trained on about 3,300 sparsely defined graph- and node-level tasks, using a dataset of 6 million molecules and 500 million quantum and biological labels. The model learns via multi-task, multi-level supervised training to produce embeddings that generalize well to a wide range of biological tasks, and that can be efficiently used by simple Multi-Layer Perceptron (MLP) models for the downstream task, as demonstrated by our experiments.

1. Introduction

Biological machine learning often faces a paucity of data, due to time-consuming and highly specialist wet-lab processes. Traditional ML models struggle in such low-data regimes. In domains such as computer vision and natural language processing (NLP), foundation models pre-trained on large quantities of data have proved to be highly effective in low-data tasks, spurring the search for molecular foundation models (MFMs).

Work on MFMs has followed two main avenues: adapting the successful transformer architectures from NLP to operate on the SMILES representations of molecules (Honda et al., 2019; Wang et al., 2019; Méndez-Lucio et al., 2022; Ahmad et al., 2022; Taylor et al., 2022; Masters et al., 2023b); and architectures which explicitly operate on the molecular graph (Beaini et al., 2024; Ying et al., 2021; Veličković et al., 2017; Dwivedi & Bresson, 2020). While SMILES are abundant, they encode chemically vital geometric information only implicitly, so models based on SMILES strings may require more capacity and training data to represent the symmetries underlying the molecular graphs. Conversely, models such as GNNs and graph transformers may use model capacity more efficiently.

MFMs may also be categorized by how they are adapted to downstream tasks: fine-tuning or fingerprinting. In the former, all the model weights are adjusted on each downstream task, in the latter, a single frozen model processes the input molecules to generate a *fingerprint*[1], and each downstream task learns a simple MLP on its limited task data. This paper adopts the fingerprinting strategy, which is more efficient for the downstream tasks, and which can be easily packaged for use by biological ML practitioners addressing such tasks.

`MiniMol` is pre-trained on the Graphium `LargeMix` dataset with around 6 million molecules and 526 million data labels. The pre-training strategy is multi-level and multi-task (Beaini et al., 2024; Shoghi et al., 2023), wherein over 3300 sparsely defined tasks on both graph and node level are trained jointly. The key training innovation in `MiniMol` is to weight the biological and quantum tasks to maximize small-task performance.

Evaluation of MFMs should reflect the real-world scenarios in which they will be used. In this paper, we evaluate MFMs by measuring their downstream performance on the low-data applications in small molecule drug discovery. In particular, a foundation model is measured by its performance on 22 tasks from the Therapeutic Data Commons (TDC) ADMET group of datasets (Huang et al., 2021). The established ADMET leaderboard shows the performance of specialized models for each of the tasks, with a different model typically being first ranked for each task. For a foundation model, we require that the same model is used on each task, albeit with task-specific fine-tuning or an MLP "task head". The current state of the art for a single model applied to all tasks is MolE (Méndez-Lucio et al., 2022), which achieves a mean rank of 5.2 when compared against the specialized

[*]Equal contribution [1]Graphcore [2]Dayhoff Labs [3]RWTH Aachen University [4]New Jersey Institute of Technology [5]Mila - Quebec AI Institute. Correspondence to: Kerstin Kläser <kerstink@graphcore.ai>, Błażej Banaszewski <blazejb@graphcore.ai>.

[1]Molecular fingerprinting also has a long history, with examples such as ECFP (Rogers & Hahn, 2010), RDkit fingerprints (Landrum et al., 2013) and MAP4 (Capecchi et al., 2020) used for many of the same low-data downstream tasks. However, as they encode the presence of particular substructures within the molecule, performance varies across problem classes (Kim, 2021; Awale & Reymond, 2014; Probst et al., 2022). Nevertheless, fingerprints remain the baseline over which MFMs must improve.

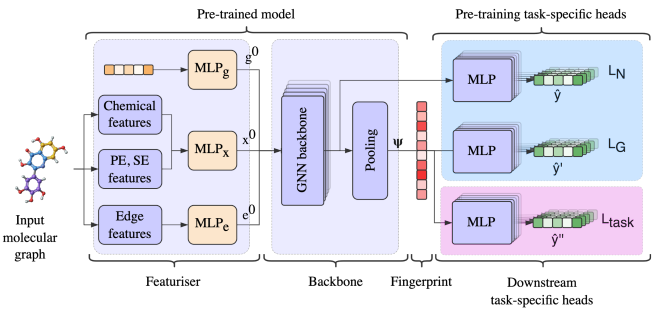

Figure 1. Schematic of the `MiniMol` architecture. An example molecule is featurized. Node feature vectors combine chemical features with positional and structural encodings. Edge features are generated using RDKit, and a random initial global vector is generated. Each initial vector is processed with an embedding MLP. The backbone consists of GINE layers, outputting the molecular fingerprint ψ after pooling. The pre-pooling output is used for pre-training on node-level tasks (L_N), e.g., PCQM4M_N4. The fingerprint ψ is used for pre-training on graph-level tasks (L_G) or as input to fine-tune downstream tasks (L_{task}), including the full set of ADMET tasks from the TDC benchmarks.

per-task models on the leaderboard. `MiniMol` achieves a mean rank of 3.4 and outperforms MolE on 17 tasks.

Our evaluation also includes a performance correlation analysis between the pre-training datasets and downstream tasks. We found, for example, that one quantum dataset often has a negative correlation with downstream tasks thus highlighting the importance of understanding the correlation between pre-training tasks and downstream tasks.

To summarize our contribution: `MiniMol` is a new molecular foundation model which is efficient both in terms of model size and in its downstream task deployment. Its efficiency comes from the use of the molecular graph structure, and its use of fingerprinting rather than fine-tuning. Our experimental paradigm and correlation analysis further give confidence in the applicability of this model, and we hope will spur the development of future models which exceed ours in both efficiency and accuracy. The pretrained `MiniMol` was made available here in a Python package of the same name.

2. Method

Here, we present our architecture for pre-training on the `LargeMix` datasets (Beaini et al., 2023), extracting fingerprints and subsequently fine-tuning to downstream tasks (see Figure 1).

Each molecule is modelled as a graph $\mathcal{G}$ with N nodes representing the atoms and M edges representing the bonds. We denote the set of edges with $\mathcal{E}$. The atom and bond features, X^0 and E^0, are generated using RDKit, providing a set of categorical and floating values. The eigenvectors and eigenvalues of the Laplacian of the graph, denoted as X^{LapVec} and X^{LapVal}, and the random walk probabilities from (Masters et al., 2023b; Rampášek et al., 2022), denoted as X^{RW}, are used as positional and structural embeddings. The input node and edge feature vectors are the concatenation of these features: nodes $X^0 = \left[X^{\text{atom}} | X^{\text{LapVec}} | X^{\text{LapVal}} | X^{\text{RW}}\right]$, and edges $E^0 = \left[E^{\text{bond}}\right]$.

A global node is added to each graph, providing an additional connection to every node. It was shown in (Li et al., 2017) that the global node dramatically improves graph-level representation. This acts both as routing between otherwise distant portions of the graph and as a readout node for the graph property. The features are initially embedded into the model dimensions using a two-layer MLP each.

Given the initial node, edge and graph embeddings we update them through multiple layers of message-passing to obtain final node embeddings $x^{\text{final}} = \texttt{GNN}(x^0, e^0, g^0)$, where `GNN` is a chosen GNN backbone (i.e. GCN, GINE, MPNN). As described in Section 3, we try three different backbone GNNs, namely GCN (Kipf & Welling, 2017), GINE (Hu et al., 2020b; Xu et al., 2019) and MPNN++ (Masters et al., 2023a) (see Appendix A.1).

Table 1. Overview of the datasets in LargeMix.

Dataset	# Molecules	# Labels	# Data Points	% of All Data Points
PCQM4M_G25	3.81M	25 (G)	93M	17%
PCQM4M_N4	3.81M	4 (N)	197.7M	37%
PCBA_1328	1.56M	1328 (G)	224.4M	41%
L1000_VCAP	15K	978 (G)	15M	3%
L1000_MCF7	12K	978 (G)	11M	2%

Pre-training. `MiniMol` is jointly pre-trained with many supervised tasks on both the graph and node levels. The `LargeMix` datasets, consisting of approximately 6M molecules and a total of 526M targets, are summarized in Table 1, more details in Appendix A.3. The total loss minimized during training is a weighted summation of each of the pre-training tasks, accounting for label sparsity per molecule. The mean absolute error (MAE) loss is used for the PCQM dataset (N4 and G25 tasks), the binary cross-entropy (BCE) loss is used for the PCBA tasks, and the hybrid cross-entropy (HCE) loss from (Beaini et al., 2023) is used for the L1000 datasets.

Following (Méndez-Lucio et al., 2022), we filter out molecules with more than 100 heavy atoms. In addition, we remove molecules in the ADMET group test sets from our pre-training data to avoid potential leakage of test labels (7% of MCF7, 4% of VCAP, 0.6% of PCBA, 0.07% of PCQM4M_G25/N4). During pre-training, we split the dataset into 92% training, 4% validation, and 4% test data.

To cover a range of GNN backbones with increasing complexity, we pre-train GCN, GINE, and MPNN++ models and subsequently evaluate their downstream performance on the TDC ADMET datasets. Each model consists of 16 GNN

Table 2. Results on downstream evaluation of MiniMol (GINE) with max pooling (see Appendix A.4 for pooling experiments) on TDC ADMET benchmarks, and comparison to the TDC leaderboard and MolE. The rank is determined for each dataset individually, on a set of 7 scores, which include the test results from the TOP5 leaderboard, MolE and MiniMol. The best result is shown in **green** and the top-3 results are highlighted in **purple**.

	TDC Dataset			Leaderboard	MolE		MiniMol (GINE)	
	Name	Size	Metric	SOTA Result	Result	Rank	Result	Rank
Absorption	Caco2 Wang	906	MAE (↓)	**0.276 ± .005**	0.310 ± .010	6	0.324 ± .012	7
	Bioavailability Ma	640	AUROC (↑)	**0.748 ± .033**	0.654 ± .028	7	0.699 ± .008	6
	Lipophilicity AZ	4,200	MAE (↓)	**0.467 ± .006**	**0.469 ± .009**	3	**0.455 ± .001**	1
	Solubility AqSolDB	9,982	MAE (↓)	**0.761 ± .025**	0.792 ± .005	5	**0.750 ± .012**	1
	HIA Hou	578	AUROC (↑)	**0.989 ± .001**	0.963 ± .019	7	**0.994 ± .003**	1
	Pgp Broccatelli	1,212	AUROC (↑)	**0.938 ± .002**	0.915 ± .005	7	**0.994 ± .002**	1
Distrib.	BBB Martins	1,975	AUROC (↑)	**0.916 ± .001**	0.903 ± .005	7	**0.923 ± .002**	1
	PPBR AZ	1,797	MAE (↓)	**7.526 ± .106**	8.073 ± .335	6	7.807 ± .188	4
	VDss Lombardo	1,130	Spearman (↑)	**0.713 ± .007**	**0.654 ± .031**	3	0.570 ± .015	7
Metabolism	CYP2C9 Veith	12,092	AUPRC (↑)	**0.859 ± .001**	0.801 ± .003	5	0.819 ± .001	4
	CYP2D6 Veith	13,130	AUPRC (↑)	**0.790 ± .001**	0.682 ± .008	6	0.718 ± .003	5
	CYP3A4 Veith	12,328	AUPRC (↑)	**0.916 ± .000**	0.867 ± .003	7	0.878 ± .001	5
	CYP2C9 Substrate	666	AUPRC (↑)	**0.441 ± .033**	**0.446 ± .062**	2	**0.481 ± .013**	1
	CYP2D6 Substrate	664	AUPRC (↑)	**0.736 ± .024**	0.699 ± .018	7	**0.726 ± .006**	2
	CYP3A4 Substrate	667	AUROC (↑)	**0.662 ± .031**	**0.670 ± .018**	1	0.644 ± .006	6
Excret.	Half Life Obach	667	Spearman (↑)	**0.562 ± .008**	**0.549 ± .024**	4	0.493 ± .002	7
	Clearance Hepatocyte	1,102	Spearman (↑)	**0.498 ± .009**	0.381 ± .038	7	**0.448 ± .006**	4
	Clearance Microsome	1,020	Spearman (↑)	**0.630 ± .010**	0.607 ± .027	6	**0.652 ± .007**	1
Toxicity	LD50 Zhu	7,385	MAE (↓)	**0.552 ± .009**	0.823 ± .019	7	**0.588 ± .010**	3
	hERG	648	AUROC (↑)	**0.880 ± .002**	0.813 ± .009	7	0.849 ± .007	6
	Ames	7,255	AUROC (↑)	**0.871 ± .002**	**0.883 ± .005**	1	0.856 ± .001	5
	DILI	475	AUROC (↑)	**0.925 ± .005**	0.577 ± .021	7	**0.944 ± .007**	1
			TDC Leaderboard Mean Rank:			5.2		**3.4**

layers with hidden dimensions adjusted such that all models have 10M $\pm 4\%$ parameters. We train each model for 100 epochs using the Adam optimizer, with a maximum learning rate of $3e^{-4}$, 5 warm-up epochs and linear learning rate decay. We present pre-training results in Appendix A.7.

Downstream tasks. For downstream tasks, we generate the global embeddings of the final layer of MiniMol from a given molecule, referred to as molecular fingerprints. These fingerprints are used as inputs to an MLP for making task-specific predictions.

3. Experimental Details

In our experiments, we pre-train MiniMol on LargeMix (Beaini et al., 2023) for various GNN backbones and subsequently fine-tune to all 22 tasks in the ADMET Group of the TDC benchmark.

Benchmarking on TDC ADMET Group. We use the ADMET group of the Therapeutics Data Commons (TDC) (Huang et al., 2021) benchmark to evaluate the downstream performance of MiniMol. Within TDC, the ADMET Benchmark Group specializes in single-instance prediction, offering a standardized suite of 22 datasets for molecular property prediction. These datasets vary in size, ranging from 475 to 13,130 molecules, with both regression and classification tasks. The datasets are categorized into Absorption, Distribution, Metabolism, Excretion, and Toxicity. To ensure a fair comparison, scaffold splits are used, with an 80:20 training/testing split. Because downstream models are small and efficient, for each dataset we search through a narrow set of hyperparameters (see Appendix A.2) to optimize an ensemble which consists of 5 models trained on different cross-validation folds (see Appendix A.5).

4. Empirical Results

We observe that pre-training performance is only marginally affected by the choice of the backbone GNN. Moreover, the pre-training performance of a given GNN backbone also varies with tasks (see Appendix A.6 for more details).

Our fine-tuning results on the ADMET group datasets of TDC show that MiniMol with GINE backbone achieves top-1 performance on 8 tasks, setting a new state-of-the-art on these datasets. Moreover, MiniMol (GINE) achieves top-3 performance on 11 tasks. Therefore, MiniMol (GINE) is shown to be a versatile model across a wide range of tasks, competing with or exceeding the perfor-

Table 3. Comparison of MiniMol to other molecular fingerprinting models using the same evaluation method as ours, including ensembles.

Model	Mean rank	>MolE	TOP1	TOP3
MiniMol	3.4	17	8	11
AGBT (Chen et al., 2021a)	5.4	10	2	4
MolFormer (Ross et al., 2022)	5.6	7	0	5
BET (Chen et al., 2021b)	6.0	7	1	2

mance of the best task-specialized architectures. We report TDC leaderboard results up until June 2024. In addition, MiniMol (GINE) outperforms MolE on 17 datasets, indicating that with only 10% of the parameters, our MiniMol approach is favourable to MolE in downstream performance across many molecular tasks. In Table 3, we compare MiniMol to other fingerprinting methods using the same evaluation downstream adaptation method as ours, so the only difference is the quality of generated fingerprints.

5. Discussion

For the fine-tuning results in Section 3, our analysis reveals that while the three GNN backbones, namely, GCN, GINE and MPNN++, all achieve similar pre-training performance, the GINE backbone shows a significant advantage when fine-tuning to downstream tasks. To give a potential explanation for this finding, recall that we adjust hidden dimensions of the different backbone GNNs to roughly align to 10M parameters. Here, the higher model complexity of MPNN++ leads to substantially smaller hidden dimension sizes than GINE. Our results thus suggest that the architectural complexity of MPNN++ is less effective in downstream performance than a simple increase in hidden dimensions. If we match the hidden dimension size of GINE in MPNN, the model would reach the size of roughly 50M parameters.

At the same time, while less complex and allowing for even larger hidden dimensions, GCN layers might be expressive enough for strong downstream performance. Specifically, the GCN omits the use of edge features and is shown to be less powerful than the 1-WL test (Xu et al., 2019) while GINE is as expressive as the 1-WL test. As such, we hypothesize that the GINE allows for a trade-off between a sufficient level of architectural complexity and a more effective use of parameter budget in terms of larger hidden dimensions translating into stronger downstream performance.

Finally, our results also reveal the general robustness of the MiniMol pipeline to the choice of backbone GNNs, where all three variants were better than MolE on the ADMET group in many tasks (see Table 2) while having significantly fewer parameters and being employed in an efficient fingerprinting pipeline, as opposed to fine-tuning all weights.

To explore which pre-training datasets impact downstream task performance, we performed a correlation analysis.

Table 4. Correlation analysis (Spearman's rho) between pre-training validation and downstream performance. The green colour indicates a beneficial correlation and the red indicates a detrimental correlation. Results with a p-value over 0.1 are blank.

Dataset	Metric	MCF	VCAP	PCBA	G25	N4
		AUROC	AUROC	AUROC	MAE	MAE
Caco2 Wang	MAE	0.590	0.651	0.718		
Bioavailability Ma	AUROC					
Lipophilicity AZ	MAE	0.568	0.539	0.627	-0.389	
Solubility AqSolDB	MAE	0.588	0.7	0.704		
HIA Hou	AUROC	0.603	0.548	0.645	-0.337	
Pgp Broccatelli	AUROC		0.361		-0.387	
BBB Martins	AUROC	0.583	0.378	0.483	-0.492	
PPBR AZ	MAE					
VDss Lombardo	Spearman		0.343			
CYP2C9 Veith	AUPRC	0.649	0.711	0.829		0.551
CYP2D6 Veith	AUPRC	0.641	0.487	0.704		0.585
CYP3A4 Veith	AUPRC	0.649	0.713	0.818		0.608
CYP2C9 Subst.	AUPRC		-0.377	-0.445		-0.586
CYP2D6 Subst.	AUPRC					
CYP3A4 Subst.	AUROC		0.409			
Half Life Obach	Spearman		0.503	0.498		
Clearance Hepato.	Spearman					
Clearance Micro.	Spearman					
LD50 Zhu	MAE	0.543	0.522	0.617		0.342
hERG	AUROC		0.57	0.453		
AMES	AUROC	0.591	0.486	0.643	-0.628	0.528
DILI	AUROC	0.49	0.416	0.454		
Sum		6.496	7.959	7.749	-2.232	2.028

Spearman's rho coefficients (Sedgwick, 2014) were calculated for each pair of pre-training and downstream metrics (see Table 4). A p-value threshold of 0.1 was used. Overall, improved pre-training metrics correlate with downstream performance. However, while the node-level quantum data is positively correlated with the downstream biological tasks, the graph-level quantum data is not. This suggests that quantum data may be more helpful in learning the node representations within the backbone than the graph-level fingerprints, which is an interesting avenue for future exploration.

6. Conclusion

In this work, we propose a novel parameter-efficient foundation model for molecular learning called MiniMol. MiniMol is pre-trained on over 3,300 biological and quantum tasks on graph- and node-level molecules and subsequently evaluated on the ADMET group of the TDC benchmark. MiniMol outperforms the previous state-of-the-art foundation model on ADMET, MolE, with only 10M parameters, 10 $\times$ fewer than MolE. In addition, fine-tuning with MLPs on the fingerprints extracted from pre-trained MiniMol, allows for efficient fine-tuning, and a correlation analysis gives insight into how to utilize pre-training datasets for downstream biological tasks.

We have recently become aware of concurrent work, MolGPS (Sypetkowski et al., 2024), an ensemble of 1B parameter models, that beats MolE in 18 (vs 17 for MiniMol) ADMET group tasks and is top-ranked in 12 (MiniMol is 8), although it is not clear whether it is open source. This provides a data point at the other end of the efficiency/accessibility spectrum for MFMs, encouraging future work in this important area.

References

Ahmad, W., Simon, E., Chithrananda, S., Grand, G., and Ramsundar, B. ChemBERTa-2: Towards Chemical Foundation Models, September 2022. URL http://arxiv.org/abs/2209.01712. arXiv:2209.01712 [cs, q-bio].

Awale, M. and Reymond, J.-L. Atom pair 2d-fingerprints perceive 3d-molecular shape and pharmacophores for very fast virtual screening of zinc and gdb-17. *Journal of chemical information and modeling*, 54(7):1892–1907, 2014.

Beaini, D., Huang, S., Cunha, J. A., Li, Z., Moisescu-Pareja, G., Dymov, O., Maddrell-Mander, S., McLean, C., Wenkel, F., Müller, L., Mohamud, J. H., Parviz, A., Craig, M., Koziarski, M., Lu, J., Zhu, Z., Gabellini, C., Klaser, K., Dean, J., Wognum, C., Sypetkowski, M., Rabusseau, G., Rabbany, R., Tang, J., Morris, C., Koutis, I., Ravanelli, M., Wolf, G., Tossou, P., Mary, H., Bois, T., Fitzgibbon, A., Banaszewski, B., Martin, C., and Masters, D. Towards Foundational Models for Molecular Learning on Large-Scale Multi-Task Datasets, October 2023. URL http://arxiv.org/abs/2310.04292. arXiv:2310.04292 [cs].

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

Capecchi, A., Probst, D., and Reymond, J.-L. One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. *Journal of cheminformatics*, 12(1):1–15, 2020.

Chen, D., Gao, K., Nguyen, D. D., et al. Algebraic graph-assisted bidirectional transformers for molecular property prediction. *Nat Commun*, 12:3521, 2021a. doi: 10.1038/s41467-021-23720-w.

Chen, D., Zheng, J., Wei, G.-W., and Pan, F. Extracting predictive representations from hundreds of millions of molecules. *The Journal of Physical Chemistry Letters*, 12(44):10793–10801, 2021b. doi: 10.1021/acs.jpclett.1c03058.

Dwivedi, V. P. and Bresson, X. A Generalization of Transformer Networks to Graphs, December 2020. URL https://arxiv.org/abs/2012.09699v2.

Honda, S., Shi, S., and Ueda, H. R. SMILES Transformer: Pre-trained Molecular Fingerprint for Low Data Drug Discovery, November 2019. URL http://arxiv.org/abs/1911.04738. arXiv:1911.04738 [cs, stat].

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020a.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for Pre-training Graph Neural Networks, February 2020b. URL http://arxiv.org/abs/1905.12265. arXiv:1905.12265 [cs, stat].

Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., and Leskovec, J. Ogb-lsc: A large-scale challenge for machine learning on graphs. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Advances in neural information processing systems*, 2021.

Kim, S. Exploring chemical information in pubchem. *Current protocols*, 1(8):e217, 2021.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. 2017.

Landrum, G. et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8:31, 2013.

Li, J., Cai, D., and He, X. Learning Graph-Level Representation for Drug Discovery, September 2017. URL http://arxiv.org/abs/1709.03741. arXiv:1709.03741 [cs, stat].

Masters, D., Dean, J., Klaser, K., Li, Z., Maddrell-Mander, S., Sanders, A., Helal, H., Beker, D., Fitzgibbon, A., Huang, S., Rampášek, L., and Beaini, D. Gps++: Reviving the art of message passing for molecular property prediction. *Transactions on Machine Learning Research*, 2023a.

Masters, D., Dean, J., Klaser, K., Li, Z., Maddrell-Mander, S., Sanders, A., Helal, H., Beker, D., Fitzgibbon, A., Huang, S., Rampášek, L., and Beaini, D. GPS++: Reviving the Art of Message Passing for Molecular Property Prediction, February 2023b. URL http://arxiv.org/abs/2302.02947. arXiv:2302.02947 [cs].

Méndez-Lucio, O., Nicolaou, C., and Earnshaw, B. Mole: a molecular foundation model for drug discovery. *arXiv preprint arXiv:2211.02657*, 2022.

Méndez-Lucio, O., Nicolaou, C., and Earnshaw, B. MolE: a molecular foundation model for drug discovery, November 2022. URL http://arxiv.org/abs/2211.02657. arXiv:2211.02657 [cs, q-bio].

Probst, D., Schwaller, P., and Reymond, J.-L. Reaction classification and yield prediction using the differential reaction fingerprint DRFP. *Digital Discovery*, 1(2):91–97, 2022. doi: 10.1039/D1DD00006C. URL https://pubs.rsc.org/en/content/articlelanding/2022/dd/d1dd00006c. Publisher: Royal Society of Chemistry.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a General, Powerful, Scalable Graph Transformer. Technical Report arXiv:2205.12454, arXiv, May 2022. URL http://arxiv.org/abs/2205.12454. arXiv:2205.12454 [cs] type: article.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50 (5):742–754, 2010.

Ross, J., Belgodere, B., Chenthamarakshan, V., et al. Large-scale chemical language representations capture molecular structure and properties. *Nat Mach Intell*, 4:1256–1264, 2022. doi: 10.1038/s42256-022-00580-7.

Sedgwick, P. Spearman's rank correlation coefficient. *Bmj*, 349, 2014.

Shoghi, N., Kolluru, A., Kitchin, J. R., Ulissi, Z. W., Zitnick, C. L., and Wood, B. M. From Molecules to Materials: Pre-training Large Generalizable Models for Atomic Property Prediction, October 2023. URL http://arxiv.org/abs/2310.16802. arXiv:2310.16802 [cs].

Subramanian, A., Narayan, R., Corsello, S. M., Peck, D. D., Natoli, T. E., Lu, X., Gould, J., Davis, J. F., Tubelli, A. A., Asiedu, J. K., et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171 (6):1437–1452, 2017.

Sypetkowski, M., Wenkel, F., Poursafaei, F., Dickson, N., Suri, K., Fradkin, P., and Beaini, D. On the scalability of gnns for molecular graphs, 2024.

Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravia, E., Poulton, A., Kerkez, V., and Stojnic, R. Galactica: A large language model for science. *CoRR*, abs/2211.09085, 2022.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph Attention Networks, October 2017. URL https://arxiv.org/abs/1710.10903v3.

Wang, S., Guo, Y., Wang, Y., Sun, H., and Huang, J. SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction. BCB '19, pp. 429–436, New York, NY, USA, September 2019. Association for Computing Machinery. ISBN 978-1-4503-6666-3. doi: 10.1145/3307339.3342186. URL https://doi.org/10.1145/3307339.3342186.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How Powerful are Graph Neural Networks? *arXiv:1810.00826 [cs, stat]*, February 2019. URL http://arxiv.org/abs/1810.00826. arXiv: 1810.00826.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do Transformers Really Perform Bad for Graph Representation?, November 2021. URL http://arxiv.org/abs/2106.05234. arXiv:2106.05234 [cs].

A. Appendix

A.1. MPNN architecture

In what follows, we describe the MPNN architecture in (Masters et al., 2023b) in detail. Here, the embeddings are incrementally updated with each MPNN layer in the model as:

$$x^{\ell+1},\, e^{\ell+1},\, g^{\ell+1} = \texttt{MPNN}(x^{\ell},\, e^{\ell},\, g^{\ell}) \tag{1}$$

The edge embedding is updated by concatenation of the edge feature with the node features at each end of the bond, with the global features. This is processed with the edge MLP and then summed with the skip connection, shown in 2.

$$\bar{e}^{\ell}_{uv} = \texttt{MLP}_{\text{edge}}\left(\left[x^{\ell}_{u} \,|\, x^{\ell}_{v} \,|\, e^{\ell}_{uv} \,|\, g^{\ell}\right]\right) \tag{2}$$

The node embedding, shown in eq.3 concatenates the node features with the summed edge features of all edges connected (senders and receiver) and the global features before passing this vector through an MLP and finally adding the skip connection.

$$\bar{x}^{\ell}_{i} = \texttt{MLP}_{\text{node}}\left(\left[x^{\ell}_{i} \left| \sum_{(u,i)\in\mathcal{E}} \bar{e}^{\ell}_{ui} \right| \sum_{(i,v)\in\mathcal{E}} \bar{e}^{\ell}_{iv} \left| \sum_{(u,i)\in\mathcal{E}} x^{\ell}_{u} \right| g^{\ell}\right]\right) \tag{3}$$

The global node is concatenated with the sum of all node and edge features in the graph (eq. 4).

$$\bar{g}^{\ell} = \left[g^{\ell} \left| \sum_{j\in\mathcal{V}} \bar{x}^{\ell}_{j} \right| \sum_{(u,v)\in\mathcal{E}} \bar{e}^{\ell}_{uv}\right] \tag{4}$$

Where the final components are computed with skip-connections as:

$$x^{\ell+1}_{i} = \bar{x}^{\ell}_{i} + x^{\ell}_{i}; \quad e^{\ell+1}_{uv} = \bar{e}^{\ell}_{uv} + e^{\ell}_{uv}; \quad g^{\ell+1} = \bar{g}^{\ell} + g^{\ell}; \tag{5}$$

This is represented diagrammatically in Fig. 2.

A.2. Hyperparameter selection

We select hyperparameters for our fine-tuning as follows. We compute a hyperparameter sweep over the maximum number of epochs; the learning rate; the dropout rate and whether to use none, batch or layer normalization in the task head. Optionally, we sweep over the width and depth of the task head MLP. Each configuration is run on the same random seed. Following the instructions provided by TDC[2], we use the provided scaffold splits for our train/validation splits via the method `get_train_valid_split` and take the benchmark test split also provided by TDC. Then, for each dataset, we select the hyperparameters resulting from the model with the smallest validation loss and subsequently re-run this model on k random seeds. Here, we distinguish between two sweep configurations. In the first configuration, we only sweep over the learning rate $\in \{0.001, 0.0005, 0.0003, 0.0001, 5e^{-5}\}$ and set the number of epochs to 25, dropout to 0.1, hidden dimension to 1024 and the number of layers to 3. In the second configuration, we set the number of epochs to 25 and sweep over whether or not to use a skip connection, the learning rate $\in \{0.0005, 0.0003, 0.0001\}$, the hidden dimension $\in \{512, 1024, 2048\}$, the number of layers $\in \{3, 4\}$, dropout $\in \{0.0, 0.1\}$, the number of warmup epochs $\in \{0, 5\}$ and the learning rate schedule $\in \{\text{constant}, \text{linear}, \text{cosine}\}$.

A.3. Pre-training datasets

PCQM4M_G25_N4. This dataset contains 3.8M molecules from the PCQM dataset (Hu et al., 2021), from the OGB-LSC challenge. The dataset consists of quantum chemistry calculations for 25 molecular graph-level properties, and 4 node-level properties per atom, resulting in about 400M labelled data points.

PCBA. This dataset contains 1.5M molecules from the OGBG-PCBA dataset (Hu et al., 2020a). This bioassay dataset, derived experimentally from high-throughput screening methods, details the impact of the molecules on living cells across 1328 sparse labels. This results in about 100M labelled data points.

[2]Available at https://tdcommons.ai/benchmark/overview/

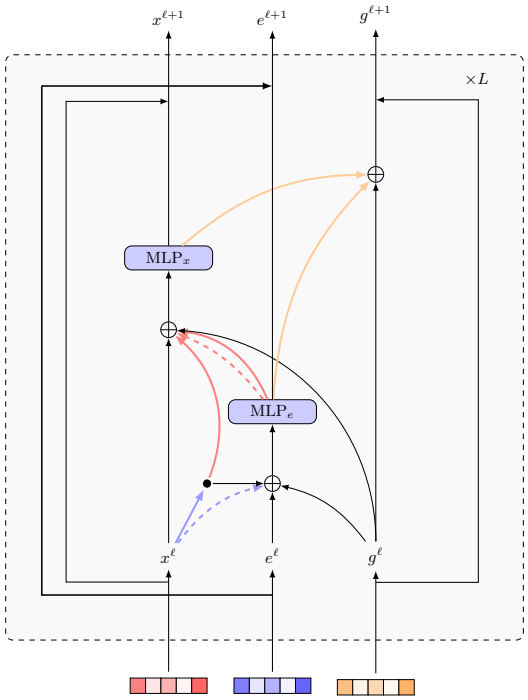

Figure 2. Example of the MPNN block architecture given in Eq.1. The edge update in Eq.2 gathers the nodes and edges before passing through the MLP first, then this output is used for the node update in Eq.3, gathering all connected node features and updated edge features. The global update in Eq.4 connects all nodes and edges, before finally the skip connections in Eq.5.

L1000_VCAP and `L1000_MCF7`. These datasets contain 26k molecules from the L1000 dataset (Subramanian et al., 2017) which details the change to gene expression profiles and cellular processes when exposed to the molecules in the dataset across about 1000 labels and 26M data points.

These diverse labels from fundamental quantum chemistry properties to macro-scale cellular impact encourage a single general representation of the molecule suitable for downstream tasks. The combined `LargeMix` contains multiple task labels per molecule. The datasets only partially overlap thus requiring the model to generalize across domains from sparse labels on molecules.

A.4. Experimentation with pooling methods

We evaluated three different pooling strategies when going from the node level to graph level representation and summarized our findings in A.4.

	Mean rank	MoIE	TOP1	TOP3
sum	3.9	16	4	10
mean	3.5	16	6	11
max	3.4	17	8	11

A.5. Downstream evaluation strategy

The strategy used for evaluating and ensembling the models is explained in the form of pseudo-code below.

While building the ensemble, the best epoch is selected based on validation loss, and to distinguish which ensemble to select for testing (e.g. while choosing one out of the sweep), the ensemble's mean validation metric is used. Final test scores are derived from the top ensemble, with error bars reported from five trials.

Since our fingerprinting approach permits fast evaluation of downstream predictors, we conduct extensive hyperparameter sweeping across all tasks. For CPU-only runs, training a downstream model only takes 1 to 10 minutes per model per dataset.

Algorithm 1 Downstream evaluation strategy

Input: a set of hyperparameter combinations HPs, number of repetitions n_neps, number of cross-validation folds n_folds
for each h_i in HPs **do**
  for rep_i in $range(n\_reps)$ **do**
    select $seed$
    for $fold_i$ in $range(n\_folds)$ **do**
      train a model on $fold_i$
      save best model based on val loss
    end for
    build ensemble of n_folds models
    evaluate on ensemble
    save val and test scores
  end for
  calculate mean and std of val and test scores across all n_reps models
end for

A.6. `MiniMol` backbone choice

Table 5 presents the mean performance of three GNN architectures (GCN, GINE, MPNN++) across all 22 downstream tasks from TDC ADMET. GINE demonstrates a significant empirical advantage as the GNN backbone for downstream tasks.

Table 5. The effect of specific GNN architectures in the backbone of the fingerprinting model on the downstream performance. The rank is determined for each dataset individually, on a set of 7 scores, which include the test results from the TOP5 TDC leaderboard, MolE and `MiniMol`. Here, all models used sum pooling, whereas our best model uses max pooling.

`MiniMol` backbone	Mean Rank	# Top1 Results	# Top3 Results
MPNN++	4.5	3	6
GCN	4.3	4	8
GINE	3.9	4	10

A.7. LargeMix Results

Table 6 presents the performance of three GNN architectures (GCN, GINE, MPNN++) across various pertaining datasets.

Table 6. Results for GNN 10M baselines on LARGEMIX dataset. We report performance metrics on the test set for each dataset in LARGEMIX separately. The best scores per metric per dataset are marked in bold.

Dataset	Metric	Model GCN	Model GINE	Model MPNN
PCQM4M_G25	MAE ↓	0.218	0.208	**0.200**
	Pearson ↑	0.884	0.889	**0.892**
	R^2 ↑	0.790	0.799	**0.803**
PCQM4M_N4	MAE ↓	0.025	0.022	**0.021**
	Pearson ↑	0.975	0.979	**0.980**
	R^2 ↑	0.952	0.959	**0.961**
PCBA_1328	CE ↓	0.033	0.033	0.033
	AUROC ↑	0.777	**0.784**	0.782
	AP ↑	0.286	**0.302**	0.287
L1000_VCAP	CE ↓	**0.061**	**0.061**	**0.061**
	AUROC ↑	0.500	**0.514**	0.500
	AP ↑	0.504	0.504	**0.506**
L1000_MCF7	CE ↓	0.059	**0.058**	0.059
	AUROC ↑	**0.533**	0.531	0.519
	AP ↑	0.513	**0.516**	0.514