Improving Deep Learning Based Molecular Fingerprints Through Informed Resampling

Noah Getz and **Pieter Feenstra** and **Patrick Timons**
getzn@mit.edu
maxfeen@mit.edu
ptimons@mit.edu

Abstract

The creation of specialized languages, like SMILES strings, has enabled the use of natural language processing based approaches across a multitude of disciplines. However, properties inherent to corpuses of natural language may not be characteristic of these specialized languages. In the instance of SMILES, the distribution of molecules in commonly used datasets are entirely meaningless with respect to any physically meaningful property. We hypothesize that minimally filtering and changing the distribution of molecules during pretraining of a transformer-based encoder will change the representation that the model learns. We confirm this through testing three different distributions and find that models pretrained on customized distributions perform better on downstream tasks and have more physically meaningful attention mechanisms.

1 Introduction

The process of drug discovery is both costly and time-consuming, with an estimated 85% of drug candidates failing during clinical trials (Sun et al., 2022). This results in substantial losses both in time and money when pursuing unfruitful candidates. The aim of machine learning in drug discovery is to reduce these inefficiencies, by either excluding unsuitable molecules early or by identifying and directing research towards promising candidates which have the potential to treat currently untreatable or incurable diseases. Two major tasks in achieving this are the accurate prediction of molecular properties and the generation of molecular structures with desirable properties.

While at first seemingly dissimilar from human language, molecular structures can also be written as one-dimensional sequences of text using a formal language called Simplified Molecular Input Line Entry System (SMILES) notation (Lin et al., 2022). Every valid SMILES string corresponds to a specific molecular structure, enabling techniques in Natural Language Processing to be leveraged to both predict the properties of molecules and generate them.

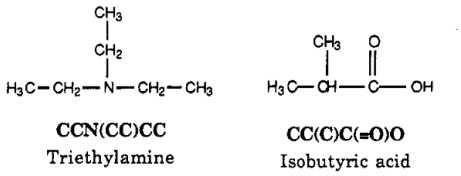

Figure 1: Examples of SMILES strings

When training a language model, our objective is to learn a distribution over sequences of tokens. For the case of human language, we can simply aim to learn the empirical distribution of a large corpus since this should approximate the probability of a human generating a given sentence. However, in the case of SMILES strings, this objective is far harder to define. As is the case in human language, there are an infinite number of possible molecules in chemical space, but unlike human language, there is no way to a priori define the probability of generating a given molecule. What would the probability of generating a molecule even mean?

While many open-source datasets of molecules exist, the distribution of molecules in such datasets is not tethered to any physical properties. For example, while the commonly used molecular datasets ChEMBL and ZINC both filter out molecules that are extremely non-drug like, the distribution of molecules in these datasets is meaningless (Gaulton et al., 2012). This poses a large issue for unsupervised learning methods, especially for encoders, since encoders may learn worse representations of molecules with biased training data. Ideally, the embedding space learned by encoders should capture physically meaningful properties such as solubility, not just whether a

SMILES string corresponds to a valid molecule.

We explore this with ChemBERTa (Weininger, 1988), a version of the RoBERTa (Liu et al., 2019) architecture that is optimized for use with SMILES strings. To determine how the underlying distribution of data used during pretraining affects the downstream performance of ChemBERTa, we pretrained a model for each of three resampled datasets corresponding to different distributions of physical properties.

We show that this simple resampling trick improves the downstream performance of ChemBERTa on property prediction tasks. We further argue that this demonstrates that when pretraining transformers using a masked language modeling objective, the underlying distribution of data has been a neglected area of research. We believe that this is especially important when training on datasets comprised of SMILES strings or other formal grammars besides human language, and should be further studied.

2 Background and Related Works

There has been significant work done in the application of transformers to drug discovery. For molecular generation, MolGPT adapted the GPT Decoder to generate SMILES sequences and thereby sample from chemical space (Bagal et al., 2022). For property prediction, models such as SMILES-BERT, MolBERT and ChemBERTa have adapted the BERT architecture to learn vector representations of SMILES strings which can then be used as inputs to Deep Neural Networks (Wang et al., 2019; Chithrananda et al., 2020; Li et al., 2021). These vector representations are called deep-learning based molecular fingerprints and in recent years have outperformed traditional rules based molecular fingerprints on a variety of downstream tasks (Baptista et al., 2022).

Bias in word embeddings is a major area of research, since embedding quality directly impacts downstream applications, such as molecular property prediction. In the traditional NLP domain, it has been shown that contextualized embeddings encode various biases such as gender and racial biases as a result of their training data (Zhao et al., 2019; Rudinger et al., 2018; Zhao et al., 2018; Kaneko and Bollegala, 2019). There have been multiple attempts to correct biases incurred during pretraining via a supervised fine-tuning phase (**?**Mazuz et al., 2023).

In the case of decoder architectures, researchers have tried to bridge this gap by incorporating an additional stage of supervised training which uses reinforcement learning techniques to reward decoders based on the desired properties of generated molecules (Mazuz et al., 2023).

To the best of our knowledge, there has been no work to alleviate this issue with BERT based models in the molecular machine learning domain. We believe this may cause learned embeddings to account for the rules of SMILES syntax and the distribution of the arbitrary training dataset but not necessarily physically meaningful properties of molecules.

One important such important property is lipophilicity - the ability of a molecule to dissolve in fatty solvents. This is measured using the log partition coefficient (LogP) which is the logarithm of the ratio between the solubility of the molecule in octanol and water. This property greatly affects how easily a molecule can be absorbed and permeated in the body (Tshepelevitsh et al., 2020). For this reason, consideration of LogP is included in Lipinski's Rule of 5, a set of 5 rules that small molecules must satisfy to be considered drug-like (Lipinski et al., 1997). More specifically, Lipinksi argues that drug-like small molecules should have a value of LogP less than 5 and ideally between 1.35 and 1.8. The importance of LogP to identifying drug candidates makes it an ideal property to focus on during our experiments.

A common method for estimating LogP was developed by Thomas Crippen and proceeds by first classifying all atoms of a molecule in terms of 68 distinct atomic types such as first degree aromatic carbons or third degree amines. Crippen made the assumption that the LogP value of a molecule could be estimated as the sum of atomic LogP contributions. Crippen then fit LogP contributions for each atom type to a large dataset of experimental LogP values, enabling these contributions to be used to quickly approximate the LogP value for other structures.

3 Methods

3.1 Dataset Selection

We trained our models using the ZINC 100K dataset which is a commonly used benchmark for computational chemistry tasks and consists of a random subset of 100K compounds from the ZINC

database of drug-like small molecules (Irwin et al., 2012). Since the ZINC 100K dataset contains structural information but not physical properties, we then computed an estimate of the LogP values in the dataset using Crippen's method of atomic LogP contributions as made available in the RDKit Python package (Wildman and Crippen, 1999; rdk, Accessed 2023). It is important to note that the calculation of a LogP values for a molecule is not as trivial as summing up atomic contributions. However, this property still offers a good surrogate for the overall distribution of a dataset with respect to LogP.

When evaluating our model, we used the Lipophilicity dataset provided by MoleculeNet (Wu et al., 2017). This commonly used benchmark consists of 4200 compounds with experimentally determined lipophilicty values at a pH of 7.4. We chose this dataset because it enables us to determine how resampling the dataset affects not only how well the embedding space captures the computationally estimated values of lipophilicity but also experimentally determined values.

3.2 Model Choice

We tested our sampling method with ChemBERTa, a transformer model based on the RoBERTa architecture (Liu et al., 2019). It is the best performing transformer model with publicly available code that is optimized for computational chemistry. As an encoder model, it learns impressive embeddings of molecules that are useful for predicting molecular properties. As such, it is a useful platform for comparing the performance of models trained on different resampling methods.

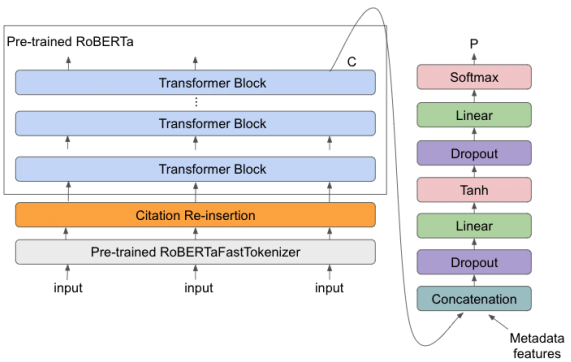

Figure 2: RoBERTa Architecture (Liu et al., 2019)

3.3 Dataset Resampling

To resample the ZINC 100k dataset according to various distributions of LogP, we first used Crippen's method to approximate the LogP values of all molecules in the ZINC 100K and MoleculeNet Lipophillicity datasets. We then divided the range of values into 50 bins to construct a discrete distribution for each dataset. During training, to select a training point we first sample a bin from the chosen discreet distribution across bins, then sample a training point uniformly from the chosen bin.

In the first case, we defined the probability of sampling from each bin of the ZINC 100K dataset to be uniform across all 50 bins. This change is motivated by the observation that in the original dataset, the model was exposed to a very narrow range of possible lipophilicy values. We aimed to see if balancing the dataset would allow the model to better learn important features of molecules with more extreme lipophilicity values and in turn cause the model to perform better on downstream tasks.

In the second case, we defined the probability of sampling from each bin of the ZINC 100K dataset to be equal to the probability of sampling from the same bin of the MoleculeNet Lipophilicity dataset. This change is motivated by attempting to reduce the shift in the distribution of data between pretraining and finetuning, which we believed would improve the performance of the model on downstream tasks.

Lastly, as a baseline we sample uniformly from the ZINC 100K dataset.

To train the model on each distribution, we altered the ChemBERTa model to use a weighted random sampler which samples molecules in the dataset according to an arbitrary probability vector.

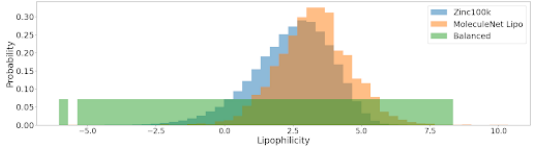

Figure 3: Distributions of Data During Pretraining

3.4 Model Training

3.4.1 Pretraining

We pretrained a version of the ChemBERTa model with three layers, three attention heads per layer, and a hidden size of 64. We used the training script provided in bert-loves-chemistry (Ahmad et al.,

2022), which is a repository of different HuggingFace models and pipelines. We pretrained with a masked language modeling objective, with default the default training parameters provided. We followed this to ensure that we are only modifying the training procedure of ChemBERTa in the sampling of data.

3.4.2 Finetuning

We fine-tuned by training a regression head with one hidden dimension of size 192 on top of the pretrained model. We then finetuned over a random train split of lipophilicity for 5 epochs, as this was when tracked metrics converged.

4 Experiments and Results

4.1 Finetuning on Property Prediction

During finetuning with the MoleculeNet Lipophilicity dataset, we recorded three evaluation metrics: mean squared error, Pearson correlation, and Spearman correlation.

Model	MSE	Pearson	Spearman
Balanced	**0.9061**	**0.6079**	**0.5583**
Matched	0.9577	0.5841	0.5422
Original	0.9714	0.5705	0.5118

Table 1: Transfer Learning (Regression on Lipophilicity)

We found that the model trained on a balanced distribution outperformed the other models in every metric recorded. Both models trained on customized distributions outperformed the original distribution, suggesting that resampling data during pretraining has a meaningful impact downstream performance. Note that Spearman correlation is typically the best suited metric for machine learning in the context of drug discovery due to its focus on ranks. Typically, it is identifying the top couple of compounds that are most important for identifying drug candidates.

These two distributions were the best-performing distributions that we tested, but we believe that further work is required to determine what distributional choices are ideal for property prediction in a drug discovery setting.

4.2 Changes in Attention

While attention is not necessarily an explanation, we aimed to see how training the model on resampled datasets changed patterns in the learned attention distribution of the model. To do so, we take advantage of the atom types defined by Crippen.

For each of the tokenized SMILES strings in the MoleculeNet dataset, we classified all tokens which correspond to atoms by their Crippen atom type. The attention distribution for every tokenized SMILES string and attention head indicates how much all pairs of tokens attend to each other. We then aggregated these attention distributions by the atom type of tokens, and averaged across all structures and attention heads to determine the average extent that tokens of every atom type attend to each other. The resulting heatmap indicates for example, the extent that every Chlorine token attends to every aromatic Carbon token. After doing this for the model trained on the balanced and original datasets, we took the difference of the coarsened heatmaps to indicate changes in patterns of attention. The difference in attention are visualized in the heatmap below (See Figure 4).

The most apparent result is that that tokens attend to themselves and other tokens of the same type significantly less when using a balanced distribution compared to the original distribution. This suggests that the model may be learning more physically meaningful contextualized embeddings of atom tokens.

Further, the vertical red bar observable in the figure indicates that when training on the rebalanced dataset, most atom types pay greater attention to aromatic carbons. An aromatic carbon is a carbon within a ring system that satisfies a set of properties related to the distribution of electrons. The number of aromatic rings in a molecule is highly positively correlated with it's lipophilicity (Ritchie and Macdonald, 2009). These changes in attention suggest that the model may be better able to identify aromatic rings and the atoms which comprise them, making it easier to predict the lipophilicity of the molecules.

5 Discussion

In this age of unlabeled data surplus, it is becoming increasing important to assess the distributional properties of our training data. Although there exist massive repositories of unlabeled data spanning many domains and modalities, the mechanisms releasing this data onto the internet for public use are not always well-understood. Thus, researchers must be careful not to treat model training as a

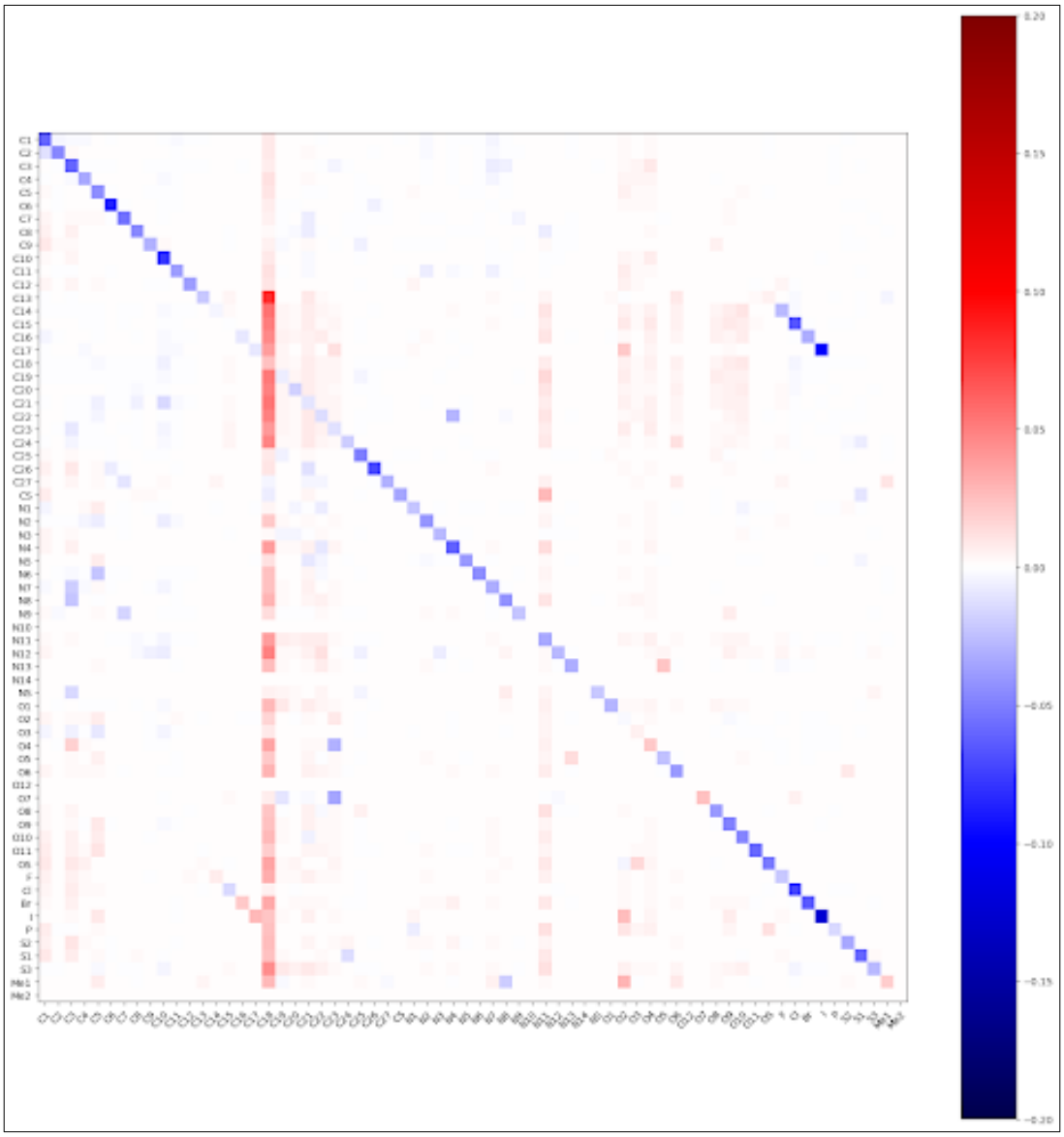

Figure 4: Difference in Attention Maps (Balanced minus Original)

black box. We show that altering the distributional properties of our training data leads to tangible impacts on model performance, bringing to light the imperative to take training data distribution into consideration.

We consider one of our major contributions to the fields of natural language processing and molecular machine learning to be utilizing computationally approximated quantities of molecules as a heuristic to intelligently alter the training distribution. With training transformer-based encoders on specialized languages in general, it may prove useful to develop heuristics with which to alter the training distributions. Additionally, we think that informed sampling during pretraining is applicable to traditional natural language processing issues. For instance, relating to the problems with gender bias associated with contextual embeddings attributed to training bias, heuristics to approximate the level of bias in a document could be useful in order to correctly weight a document's contribution to the weight update during pretraining.

Although further research will be required to validate our results and show that informed sampling during pretraining remains useful with increased number of model parameters, our results indicate the potential.

6 Impact Statement

This research can meaningfully impact the use of transformers in the context of SMILES strings. Cur-

rent and future works in this application could use a resampling method to improve their downstream performance. As such, this work can further improve the use of machine learning in drug discovery. Of course, research conducted in the applications of machine learning to drug discovery and biology in general always has inherit risk, as well as reward. For example, the acceleration of drug discovery and being able to direct the focus of research in labs can lead to breakthroughs in long-standing topics, like drugs targeted towards the treatment of Alzheimer's, Parkinson's, and cancer.

However, just because this research was intended to be focused on the applications for drug discovery does not mean that this can be its sole use. Just as we focused on improving predictions on lipophilicity, so too could somebody do this for toxicity, or other harmful molecular properties like inhibiting tumor-suppression genes.

The research of how to change models to reduce the possibility of foul-play is ongoing and a very hot topic. As datasets of small molecules and their properties become even larger, regulation should be developed regarding their accessibility to reduce the possibility for bad actors to use them to develop biological weapons.

7 Citations

7.1 References

References

Accessed 2023. RDKit: Open-source cheminformatics.

Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. 2022. Chemberta-2: Towards chemical foundation models.

V. Bagal, R. Aggarwal, P. K. Vinod, and U. D. Priyakumar. 2022. Molgpt: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62(9):2064–2076.

Delora Baptista, João Correia, Bruno Pereira, and Miguel Rocha. 2022. Evaluating molecular representations in machine learning models for drug response prediction and interpretability. *Journal of Integrative Bioinformatics*, 19(3):20220006.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. 2020. Chemberta: Large-scale self-supervised pretraining for molecular property prediction.

A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, and J. P. Overington. 2012. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(Database issue):D1100–D1107.

John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. 2012. Zinc: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768.

Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. In *Annual Meeting of the Association for Computational Linguistics*.

Juncai Li, Xiaofei Jiang, and Yulin Wang. 2021. Molbert: An effective molecular representation with bert for molecular property prediction. *Wireless Communications and Mobile Computing*, 2021:7181815.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI Open*, 3:111–132.

Christopher A. Lipinski, Franco Lombardo, Beryl W. Dominy, and Paul J. Feeney. 1997. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 23(1):3–25. In Vitro Models for Selection of Development Candidates.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

E. Mazuz, G. Shtar, and B. et al. Shapira. 2023. Molecule generation using transformers and policy gradient reinforcement learning. *Scientific Reports*, 13:8799.

Timothy J. Ritchie and Simon J.F. Macdonald. 2009. The impact of aromatic ring count on compound developability – are too many aromatic rings a liability in drug design? *Drug Discovery Today*, 14(21):1011–1020.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *North American Chapter of the Association for Computational Linguistics*.

D. Sun, W. Gao, H. Hu, and S. Zhou. 2022. Why 90% of clinical drug development fails and how to improve it? *Acta Pharmaceutica Sinica. B*, 12(7):3049–3062.

Sofja Tshepelevitsh, Sandip A. Kadam, Astrid Darnell, Johan Bobacka, Alo Rüütel, Tõiv Haljasorg, and Ivo Leito. 2020. Logp determination for highly lipophilic hydrogen-bonding anion receptor

molecules. *Analytica Chimica Acta*, 1132:123–133.

Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. 2019. Smiles-bert: Large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '19, page 429–436, New York, NY, USA. Association for Computing Machinery.

David Weininger. 1988. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28:31–36.

Scott Wildman and Gordon Crippen. 1999. Prediction of physicochemical parameters by atomic contributions. *Journal of Chemical Information and Computer Sciences*, 39:868–873.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. 2017. Moleculenet: A benchmark for molecular machine learning. *arXiv preprint arXiv:1703.00564*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. *ArXiv*, abs/1904.03310.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. In *Conference on Empirical Methods in Natural Language Processing*.

8 Code Availability

https://github.com/Noahb930/6.8611_project

A Appendix

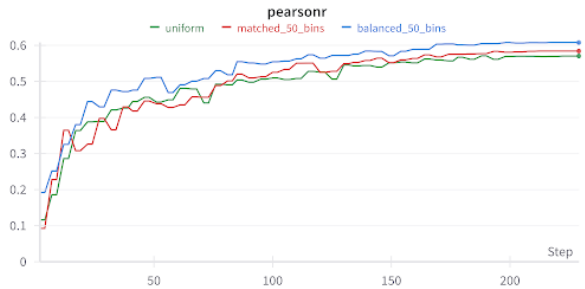

Figure 5: Pearson Correlation Coefficient During Finetuning

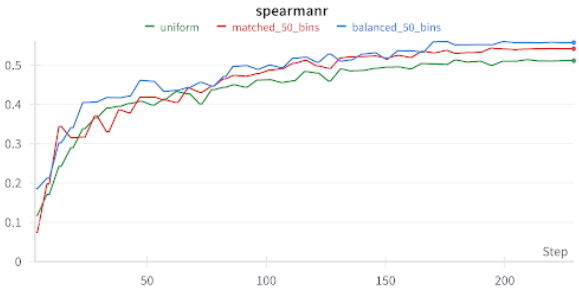

Figure 6: Spearman Correlation During Finetuning

Table 1. Atom Type Descriptions and Contributions

type	descriptions	SMARTS[a]	log *P*	obsd	MR	obsd
C1	1°, 2° aliphatic	'[CH4]', '[CH3]C', '[CH2](C)C'	0.1441	5080	2.503	2560
C2	3°, 4° aliphatic	'[CH](C)(C)C', '[C](C)(C)(C)C'	0.0000	1014	2.433	587
C3[b]	1°, 2° heteroatom	'[CH3][(N,O,P,S,F,Cl,Br,I)]' '[CH2X4](N,O,P,S,F,Cl,Br,I)]'	−0.2035	5452	2.753	1513
C4[c]	3°, 4° heteroatom	'[CH1X4][(N,O,P,S,F,Cl,Br,I)]' '[CH0X4][(N,O,P,S,F,Cl,Br,I)]'	−0.2051	2431	2.731	847
C5	C = heteroatom	'[C] = [A#X]'	−0.2783	5758	5.007	961
C6	C = C aliphatic	'[CH2] = C', '[CH1](=C)A', '[CH0](=C)(A)A','[C](=C)=C'	0.1551	1062	3.513	570
C7	acetylene, nitrile	'[CX2]#A'	0.00170	465	3.888	215
C8	1° aromatic carbon	'[CH3]c'	0.08452	1085	2.464	231
C9	1° aromatic heteroatom	'[CH3][a#X]'	−0.1444	200	2.412	9
C10	2° aromatic	'[CH2X4]a'	−0.0516	2016	2.488	247
C11	3° aromatic	'[CHX4]a'	0.1193	825	2.582	114
C12	4° aromatic	'[CH0X4]a'	−0.0967	456	2.576	36
C13[d]	aromatic heteroatom	'[cH0]−[!(C,N,O,S,F,Cl,Br,I)]'	−0.5443	30	4.041	33
C14	aromatic halide	'[c][#9]'	0.0000	350	3.257	25
C15	aromatic halide	'[c][#17]'	0.2450	1329	3.564	61
C16	aromatic halide	'[c][#35]'	0.1980	298	3.180	47
C17	aromatic halide	'[c][#53]'	0.0000	124	3.104	19
C18	aromatic	'[cH]'	0.1581	7915	3.350	926
C19	aromatic bridgehead	'[c](:a)(:a):a'	0.2955	1179	4.346	64
C20	4° aromatic	'[c](:a)(:a)-a'	0.2713	514	3.904	19
C21	4° aromatic	'[c](:a)(:a)-C'	0.1360	5050	3.509	703
C22	4° aromatic	'[c](:a)(:a)-N'	0.4619	3428	4.067	95
C23	4° aromatic	'[c](:a)(:a)-O'	0.5437	2427	3.853	167
C24	4° aromatic	'[c](:a)(:a)-S'	0.1893	851	2.673	21
C25	4° aromatic	'[c](:a)(:a) = C', '[c](:a)(:a) = N', '[c](:a)(:a) = O'	−0.8186	661	3.135	4
C26	C = C aromatic	'[C](=C)(a)A', '[C](=C)(c)a', '[CH](= C)a', '[C] = c'	0.2640	344	4.305	57
C27[e]	aliphatic heteroatom	'[CX4][!(C,N,O,P,S,F,Cl,Br,I)]'	0.2148	24	2.693	101
CS	carbon supplemental	'[#6]' not matching any basic C type	0.08129	0	3.243	0
H1	hydrocarbon	'[#1][#6]', '[#1][#1]'	0.1230	9852	1.057	3361
H2[f]	alcohol	'[#1]O[CX4]', '[#1]Oc', '[#1]O[!(C,N,O,S)]', '[#1][!C,N,O)]'	−0.2677	1744	1.395	493
H3	amine	'[#1][#7]', '[#1]O[#7]'	0.2142	4954	0.9627	216
H4	acid	'[#1]OC = [#6]', '[#1]OC = [#7]', '[#1]OC = O', '[#1]OC = S', '[#1]OO', '[#1]OS'	0.2980	622	1.805	81
HS	hydrogen supplemental	'[#1]' not matching any basic H type	0.1125	0	1.112	0
N1	1° amine	'[NH2+0]A'	−1.0190	1014	2.262	70
N2	2° amine	'[NH+0](A)A'	−0.7096	2040	2.173	81
N3	1° aromatic amine	'[NH2+0]a'	−1.0270	696	2.827	30
N4	2° aromatic amine	'[NH+0](A)a', '[NH+0](a)a'	−0.5188	1346	3.000	21
N5	imine	'[NH+0] = A', '[NH+0] = a'	0.08387	48	1.757	2
N6	substituted imine	'[N+0](=A)A', '[N+0](=A)a', '[N+0](=a)A', '[N+0](=a)a'	0.1836	1010	2.428	40
N7	3° amine	'[N+0](A)(A)A'	−0.3187	1720	1.839	99
N8	3° aromatic amine	'[N+0](a)(A)A', '[N+0](a)(a)A', '[N+0](a)(a)a'	−0.4458	492	2.819	21
N9	nitrile	'[N+0]#A'	0.01508	382	1.725	72
N10	protonated amine	'[NH3+*]', '[NH2+*]', '[NH+*]'	−1.950	189		0
N11	unprotonated aromatic	'[n+0]'	−0.3239	2819	2.202	96
N12	protonated aromatic	'[n+*]'	−1.119	104		0
N13	4° amine	'[NH0+*](A)(A)(A)A', '[NH0+*](=A)(A)A', '[NH0+*](=A)(A)a', '[NH0+*](=[#6])=[#7]'	−0.3396	1075	0.2604	75
N14	other ionized nitrogen	'[N+*]#A', '[N−*]', '[N+*](=[N−*])=N'	0.2887	17	3.359	4
NS	nitrogen supplemental	'[#7]' not matching any basic N type	−0.4806	0	2.134	0
O1	aromatic	'[o]'	0.1552	413	1.080	56
O2	alcohol	'[OH]', '[OH2]'	−0.2893	2317	0.8238	526
O3	aliphatic ether	'[O](C)C', '[O](C)[A#X]', '[O]([A#X])[A#X]'	−0.0684	2376	1.085	925
O4	aromatic ether	'[O](A)a','[O](a)a'	−0.4195	1957	1.182	130
O5	oxide	'[O]=[#8]', '[O]=[#7]', '[OX1−*][#7]'	0.0335	1272	3.367	88
O6	oxide	'[OX1−*][#16]'	−0.3339	718	0.7774	34
O7[g]	oxide	'[OX1−*][!(N,S)]'	−1.189	138	0.000	24
O8	aromatic carbonyl	'[O]=c'	0.1788	657	3.135	4
O9	carbonyl aliphatic	'[O]=[CH]C', '[O]=C(C)C', '[O]=C(C)[A#X]','[O]=[CH]N', '[O]=[CH]O','[O]=[CH2]', '[O]=[CX2]=O'	−0.1526	3163	0.000	767
O10	carbonyl aromatic	'[O]=[CH]c', '[O]=C(C)c', '[O]=C(c)c', '[O]=C(c)[a#X]', '[O]=C(c)[A#X]', '[O]=C(C)[a#X]'	0.1129	1534	0.2215	125
O11	carbonyl heteroatom	'[O]=C([A#X])[A#X]', '[O]=C([A#X])[a#X]', '[O]=C([a#X])[a#X]'	0.4833	1063	0.3890	43
O12	acid	'[O−1]C(=O)'	−1.326	187		0
OS	oxygen supplemental	'[#8]' not matching any basic O type	−0.1188	0	0.6865	0
F	fluorine	'[#9−0]'	0.4202	814	1.108	120
Cl	chlorine	'[#17−0]'	0.6895	1613	5.853	630
Br	bromine	'[#35−0]'	0.8456	366	8.927	250
I	iodine	'[#53−0]'	0.8857	137	14.02	61
Hal[h]	ionic halogens	'[#9−*]', '[#17−*]', '[#35−*]', [#53−*]' '[#53+*]'	−2.996	19		0

Figure 7: Chart of Crippen LogP Values (Wildman and Crippen, 1999)