ChemBERTa-2: Towards Chemical Foundation Models

Walid Ahmad*
Reverie Labs
walid@reverielabs.com

Elana Simon*
Reverie Labs
elana@reverielabs.com

Seyone Chithrananda
UC Berkeley
seyonec@berkeley.edu

Gabriel Grand
Reverie Labs & MIT CSAIL
gg@mit.edu

Bharath Ramsundar
Deep Forest Sciences
bharath@deepforestsci.com

Abstract

Large pretrained models such as GPT-3 have had tremendous impact on modern natural language processing by leveraging self-supervised learning to learn salient representations that can be used to readily finetune on a wide variety of downstream tasks [1]. We investigate the possibility of transferring such advances to *molecular* machine learning by building a chemical foundation model, ChemBERTa-2, using the "language" of SMILES. While labeled data for molecular prediction tasks is typically scarce, libraries of SMILES strings are readily available.

In this work, we build upon ChemBERTa [2] by optimizing the pretraining process. We compare multi-task and self-supervised pretraining by varying hyperparameters and pretraining dataset size, up to 77M compounds from PubChem. To our knowledge, the 77M set constitutes one of the largest datasets used for molecular pretraining to date. We find that with these pretraining improvements, we are competitive with existing state-of-the-art architectures on the MoleculeNet [3] benchmark suite. We analyze the degree to which improvements in pretraining translate to improvement on downstream tasks.

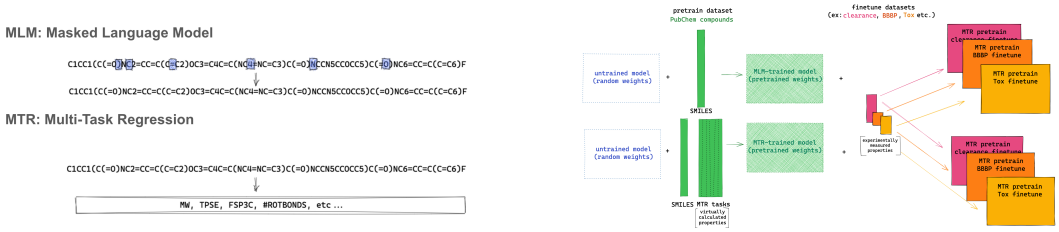
1 Motivation

Over the past few years, transformers [4, 5] have emerged as popular architectures for learning self-supervised representations of molecules from text representations. ChemBERTa [2] introduced a BERT-like transformer model that learns molecular fingerprints through semi-supervised pretraining and pretrained it on a dataset of 10M compounds. MolBERT [6] experiments with a number of different pretraining objectives on a dataset of 1.6M compounds. SMILES-BERT [7] pretrains on 18.7M compounds from Zinc.

ChemBERTa-2 is a BERT-like transformer model [8] that learns molecular fingerprints through semi-supervised pretraining of the language model. ChemBERTa-2 employs masked-language modelling (MLM) and multi-task regression (MTR) over a large corpus of 77 million SMILES strings, a well-known text representation of molecules. SMILES, is its own language, with a simple vocabulary, consisting of a series of characters representing atom and bond symbols, and very few grammar rules. [9]. ChemBERTa-2 explores the scaling hypothesis that pretraining effectively on larger datasets can yield improved performance, using the largest training dataset in molecular representation learning.

*Equal contribution

2021 ELLIS Machine Learning for Molecule Discovery Workshop.



(a) MLM vs. MTR

(b) Training pipeline

Figure 1: a) An illustration of masked language modeling (MLM) and multitask regression (MTR) pretraining tasks. b) The training pipeline implemented to achieve results in this paper.

2 Related Work

While this paper and its preceding works explore transformer-based pretraining for molecular models, a parallel line of work has explored the used of graph-based pretraining methods. SNAP [10] introduces graph pretraining methods based on node attribute masking and structural similarity. Grover [11] scales graph-transformer pretraining to a 100 million parameter model pretrained on 10M compounds. MolGNet [12] uses a message passing architecture to pretrain a 53 million parameter model on 11M compounds.

A number of recent works have explored alternative pretraining methodologies including contrastive learning [13]. Other work attempts to combine molecular graph and transformer based pretraining methodologies into a unified "dual" framework [14], or considers techniques inspired by neural machine translation by learning to translate between SMILES and InChi representations of a molecule [15]. Very recent work investigates whether large language models such as GPT-3, trained on non-chemical corpuses have learned meaningful chemistry [16].

3 Methods

ChemBERTa-2 is based on the RoBERTa [8] transformer implementation in HuggingFace [17]. We use the same training dataset of 77M unique SMILES from ChemBERTa [2]. We canonicalize and globally shuffle the SMILES to facilitate large-scale pretraining. For validation, we first set aside a fixed set of 100k compounds. We divide the remaining dataset by sampling subsets of 5M, 10M and 77M (the full set), constituting three datasets to be used across both pretraining tasks.

3.1 Pretraining Strategies and Setup

Masked Language Modeling: We adopt the masked language modeling (MLM) pretraining procedure from RoBERTa, which masks 15% of the tokens in each input string and trains the model to correctly identify them. We use a maximum vocab size of 591 tokens based on a dictionary of common SMILES characters and a maximum sequence length of 512 tokens.

Multi-task Regression: We compute a set of 200 molecular properties for each compound in our training dataset. These properties do not require any experimental measurements and can each be calculated from SMILES alone using RDKit [18]. We then train a multitask regression (MTR) architecture to predict these properties simultaneously. Because these tasks have very different scales and ranges, we mean-normalize the labels for each task prior to training.

Pretraining Setup: Models are trained on AWS EC2 instances equipped with Nvidia T4 GPUs. We set early stopping patience to equal one pass through the dataset, to ensure that for any dataset size, the model has an opportunity to see each compound at least once.

3.2 Hyperparameter Search

Most language modeling architectures have hyperparameters that are tuned on datasets comprised of written and spoken language, such as English. SMILES on the other hand, have a very different grammatical structure. To ensure an adequate assessment of ChemBERTa-v2 performance, we conduct a thorough hyperparameter search (subject to compute constraints).

We select 50 random hyperparameter configurations, varying the hidden size, number of attention heads, dropout, intermediate size, number of hidden layers, and the learning rate. Models have between 5M and 46M parameters. Each configuration is trained on each of the MLM and MTR pretraining tasks, with the 5M dataset. Using the smallest dataset size ensures that we can train until convergence (as dictated by early stopping). From pretraining results, we select five configurations, with varying validation loss values, to train on the 10M and 77M sets. Five configurations are selected for MLM and MTR, independently from one another, with the objective of evaluating how pretraining loss can affect downstream performance.

3.3 Finetuning on MoleculeNet

We evaluate our models on several regression and classification tasks from MoleculeNet [3] selected to cover a range of dataset sizes (1.5K - 8.0K examples) and medicinal chemistry applications (brain penetrability, toxicity, solubility, and on-target inhibition). These included the BACE, Clearance, Delaney, Lipophilicity, BBBP, ClinTox, HIV, Delaney, and Tox21 datasets. For datasets with multiple tasks, we selected a single representative task: the clinical toxicity (CT_TOX) task from ClinTox and the p53 stress-response pathway activation (SR-p53) task from Tox21. For each dataset, we generate an 80/10/10 train/valid/test split using the scaffold splitter from DeepChem [19]. We finetune models for up to 100 epochs with early stopping based on validation loss, and explore train-time hyperparameters via HuggingFace's built-in `optuna` optimization tooling, varying learning rate, random seed, and batch size. Finetuning task labels are normalized to have zero mean and unit standard deviation during training for regression tasks, and balanced class weights for classification tasks.

4 Results

In Table 1, we share our results on MoleculeNet benchmarks. ChemBERTa-2 configurations are able to achieve competitive results on nearly all tasks, and outperform D-MPNN (`chemprop` implementation) on 6 out of 8 tasks.

	BACE *RMSE*	**Clearance** *RMSE*	**Delaney** *RMSE*	**Lipo** *RMSE*	**BACE** *ROC*	**BBBP** *ROC*	**ClinTox** *ROC*	**SR-p53** *ROC*
D-MPNN	2.253	49.754	1.105	1.212	0.812	0.697	**0.906**	0.719
RF	**1.3178**	52.0770	1.7406	0.9621	**0.8507**	0.7194	0.7829	0.724
GCN	1.6450	51.2271	0.8851	0.7806	0.818	0.676	0.907	0.688
ChemBERTa-1						0.643	0.733	0.728
ChemBERTa-2								
MLM-5M	1.451	54.601	0.946	0.986	0.793	0.701	0.341	0.762
MLM-10M	1.611	53.859	0.961	1.009	0.729	0.696	0.349	0.748
MLM-77M	1.509	52.754	1.025	0.987	0.735	0.698	0.239	0.749
MTR-5M	1.477	50.154	0.874	0.758	0.734	**0.742**	0.552	**0.834**
MTR-10M	1.417	48.934	**0.858**	**0.744**	0.783	0.733	0.601	0.827
MTR-77M	1.363	**48.515**	0.889	0.798	0.799	0.728	0.563	0.817

Table 1: Comparison of ChemBERTa-2 pretrained on different tasks (MLM and MTR) and on different dataset sizes (5M, 10M, and 77M), vs. existing architectures on selected MoleculeNet tasks. We report ROC-AUC (↑) for classification and RMSE (↓) for regression tasks. D-MPNNs were trained with the `chemprop` [20] library. We could not benchmark easily against Grover [11] due to differences in benchmarking procedures.

4.1 Selection of Pretraining Method

On every downstream finetuning task, models pretrained on the MTR task tend to perform better than models pretrained on the MLM task. However, in our current implementation, MTR training is substantially slower than MLM, due to the increased dataset size from the 200-element label vector. To address this, we observe that MLM pretraining loss corresponds very well with MTR pretraining loss for a given architecture. In Figure 2, we can see MTR vs MLM loss for a given configuration on

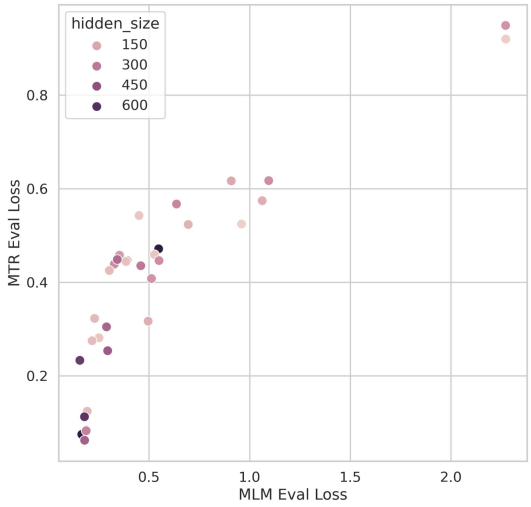


Figure 2: Comparing MLM and MTR pretrain losses

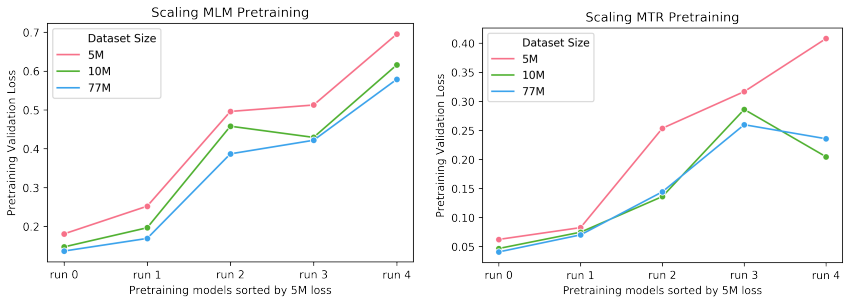


Figure 3: Pretrain losses for each of the 5 model configurations that were trained on all three datasets (5M, 10M, and 77M). MLM configurations are on the left, and MTR on the right. The configurations are sorted by their loss when training with 5M compounds along the x-axis. Note that there is considerable performance variability across runs.

the 5M dataset (where the same configurations were trained for both tasks). Thus, an architecture search can first be done via MLM pretraining, and the selected architecture(s) can then be trained on the MTR task for superior downstream performance.

In our experiments, we observe consistent improvements to pretraining loss with increased dataset size. As we see in Figure 3, training a model until convergence on 77M unique smiles instead of 5M can improve the pretraining loss by 25-35%, an observation which holds across models with varying levels of performance for both MLM and MTR pretraining.

We find that the degree to which improving performance on the pretraining tasks transfers to downstream tasks, varies by dataset. In Figure 4 we show two examples of transfer learning from pretrain tasks to finetune tasks with varying degrees of success. Improving (decreasing) pretraining loss for MLM and MTR leads to almost linear improvements (decrease) in Lipophilicity RMSE. This pattern does not hold as clearly for BACE Classification.

5 Dimension Reduction of ChemBERTa Embeddings

We used UMAP [21] to inspect the representations learned by pre-trained ChemBERTa models on the BACE and BBBP tasks, and contrast them to ECFP embeddings. We aim to see how well pre-trained language models without any further fine-tuning on MolNet benchmarks perform at clustering embeddings according to their labels. We drop large extra fragments in SMILES (using

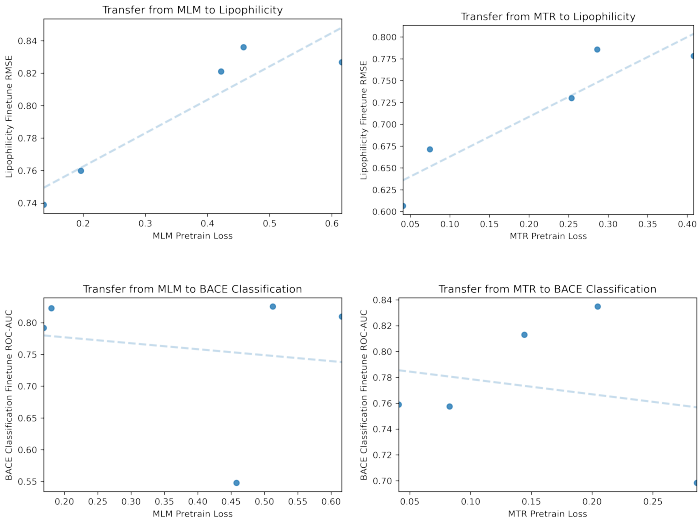


Figure 4: Finetuning performance versus pretraining loss. Left Column: MLM Pretraining, Right Column: MTR Pretraining. Top Row: Lipophilicity Finetune, RMSE (↓), Bottom Row: BACE Classification Finetune, ROC-AUC (↑). The dotted lines represent linear models fit to the datapoints.

RDKit's LargeFragmentChooser) to avoid presence of salts, which are irrelevant to blood-brain barrier permeability, before generating both transformer and ECFP embeddings.

We parameterize a UMAP model based on Jaccard distance with the following settings `metric = "jaccard", n_neighbors = 25, n_components = 2, low_memory = False, min_dist = 0.001`. We find that on average, ChemBERTa embeddings from both pre-trained masked-language and multi-task regression models are a stronger prior representation for a variety of downstream tasks to be fine-tuned on.

6 Discussion

In this work, we introduce ChemBERTa-2, an updated transformer architecture for molecular property prediction. By more deeply exploring the pretraining pipeline, we are able to achieve more competitive baseline results on downstream tasks, while extracting insights into pretraining strategies for language models. We use the more efficient MLM pretraining to select suitable hyperparameters for MTR pretraining and investigate the relationship between pretraining loss and downstream performance. Future work will benchmark against Grover and other graph based architectures and extend pretraining to larger datasets.

We also observe that certain finetuning tasks benefited greatly from pretraining improvements (either due to increased training time, increased pretraining dataset size, or varied hyperparameters) whereas others did not. The varied degree of transfer could depend on the type of task modeled (ex: solubility vs toxicity), the structural features of molecules in each dataset, the size of each dataset, or potentially even other features of the dataset we have not considered. We do not explore this phenomena in this work but note that this is an important area for future exploration. We have mostly focused on ways to improve molecular transfer learning by optimizing model pretraining procedures, but just as critical is to understand the conditions under which datasets might meaningfully benefit from pretraining.

We open source the trained models from this project. While recent work has highlighted the dual use risk [22] of chemical models, for now we believe that the challenge of synthesizing a novel molecule limits the potential harms from releasing updated models. The balance may shift in future continuation of this research, and we will continue to assess the risks of dual use for future open source releases.

We also note that the terminology of foundation model has drawn criticism due to the fact that the data large language models are trained on is typically heavily biased. We note that the data we pretrain on is drawn from more fundamental chemical calculations, so we feel that the terminology

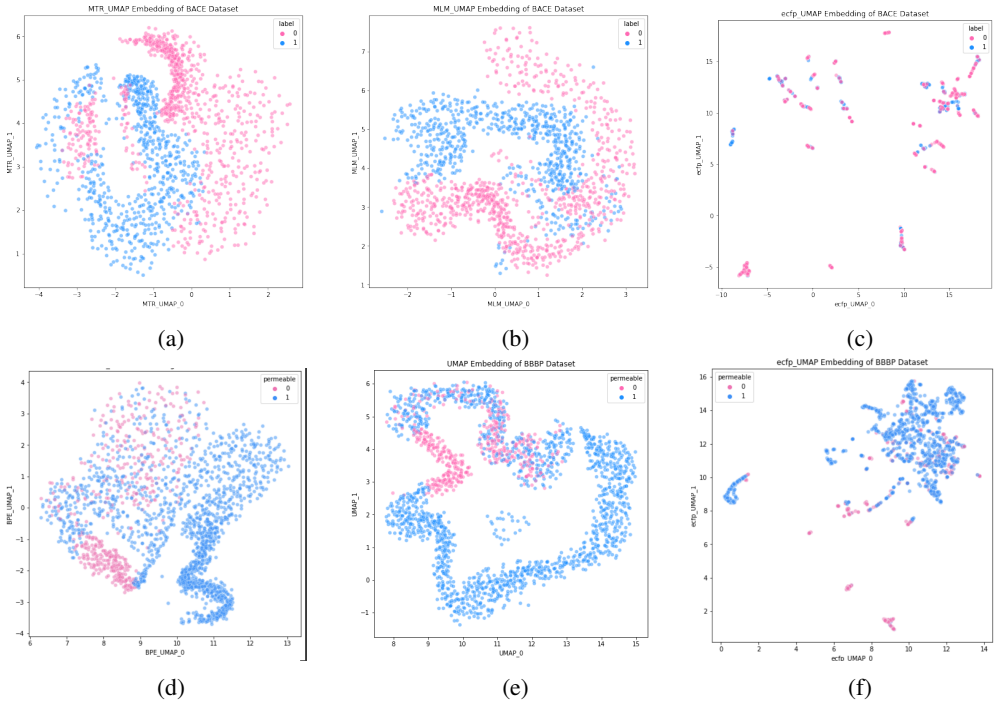


Figure 5: (a) MTR-77M embeddings fit using UMAP on BACE classification task. (b) MLM-77M embeddings fit using UMAP on BACE classification task. (c) ECFP embeddings fit using UMAP on BACE classification task. (d) MTR-77M embeddings fit using UMAP on BBBP classification task. (e) MLM-77M embeddings fit using UMAP on BBBP classification task (f) ECFP embeddings fit using UMAP on BBBP classification task

"chemical foundation model" is appropriate even if the use of the term "foundation model" may be less appropriate for models such as GPT-3.

References

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

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

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] Benedek Fabian, Thomas Edlich, Héléna Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.

[7] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. 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, pages 429–436, 2019.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

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

[10] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[11] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *arXiv preprint arXiv:2007.02835*, 2020.

[12] Pengyong Li, Jun Wang, Yixuan Qiao, Hao Chen, Yihuan Yu, Xiaojun Yao, Peng Gao, Guotong Xie, and Sen Song. Learn molecular representations from large-scale unlabeled molecules for drug discovery. *arXiv preprint arXiv:2012.11175*, 2020.

[13] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.

[14] Jinhua Zhu, Yingce Xia, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Dual-view molecule pre-training. *arXiv preprint arXiv:2106.10234*, 2021.

[15] Robin Winter, Floriane Montanari, Frank Noé, and Djork-Arné Clevert. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chemical science*, 10(6):1692–1701, 2019.

[16] Andrew D White, Glen M Hocky, Heta A Gandhi, Mehrad Ansari, Sam Cox, Geemi P Wellawatte, Subarna Sasmal, Ziyue Yang, Kangxin Liu, Yuvraj Singh, et al. Do large language models know chemistry? 2022.

[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910, 2019.

[18] Greg Landrum et al. Rdkit: Open-source cheminformatics. 2006.

[19] B Ramsundar, P Eastman, E Feinberg, J Gomes, K Leswing, A Pappu, M Wu, and V Pande. Deepchem: Democratizing deep-learning for drug discovery, quantum chemistry, materials science and biology, 2016.

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

[21] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[22] Fabio Urbina, Filippa Lentzos, Cédric Invernizzi, and Sean Ekins. Dual use of artificial-intelligence-powered drug discovery. *Nature Machine Intelligence*, 4(3):189–191, 2022.

7 Appendix

7.1 Training Tips

These are some tips and things that we found useful when running large scale pretraining experiments across different machines. 1) Due to availability of machines we train a few models on smaller machines that require decreasing the batch size to fit into memory. When doing this we decrease the learning rate accordingly to mitigate the effects of using different batch sizes. 2) When training MTR, our input data starts in CSV format. We find that using the default HuggingFace CSV data loader is notably slower than their text-based data loader so we develop a custom wrapper around

their text-based loader that parses the text into tabular format. For large-scale pretraining, this makes a big difference for efficiency. 3) To save on cost we use AWS spot instances which sometimes get interrupted. HuggingFace has a nice system for re-starting models part-way through training. Using this is critical for training models that take a long time and get interrupted periodically.