nature communications

Article

https://doi.org/10.1038/s41467-024-53751-y

MolE: a foundation model for molecular graphs using disentangled attention

Received: 11 March 2024

Accepted: 18 October 2024

Published online: 12 November 2024

Check for updates

Oscar Méndez-Lucio [1], Christos A. Nicolaou [1,2] & Berton Earnshaw [1]

Models that accurately predict properties based on chemical structure are valuable tools in the chemical sciences. However, for many properties, public and private training sets are typically small, making it difficult for models to generalize well outside of the training data. Recently, this lack of generalization has been mitigated by using self-supervised pretraining on large unlabeled datasets, followed by finetuning on smaller, labeled datasets. Inspired by these advances, we report MolE, a Transformer architecture adapted for molecular graphs together with a two-step pretraining strategy. The first step of pretraining is a self-supervised approach focused on learning chemical structures trained on ~842 million molecular graphs, and the second step is a massive multi-task approach to learn biological information. We show that finetuning models that were pretrained in this way perform better than the best published results on 10 of the 22 ADMET (absorption, distribution, metabolism, excretion and toxicity) tasks included in the Therapeutic Data Commons leaderboard (c. September 2023).

Machine learning has been successfully applied to chemical sciences for many decades[1]. In particular, molecular property prediction has been critical in successfully advancing material and drug discovery projects[2]. Nonetheless, a major challenge in this area still is to represent a molecule in a way that is compatible with machine learning algorithms with minimum information loss. Initially, molecules were represented in terms of their physicochemical properties (e.g., partition coefficient) or information that can be obtained from the molecular formula such as molecular weight or number of heteroatoms[3]. While this approach was successful for the first quantitative structure-activity relationship (QSAR) studies[4], it used only global properties of the molecule, and not the chemical structure itself. With time, molecules were described in more sophisticated ways using molecular fingerprints such as MACCS keys[5] and Extended Connectivity Fingerprints (ECFPs)[6] among others. These molecular fingerprints encode substructures of the molecules either in the form of preset chemical groups or as atom environments. Despite their successful use in numerous QSAR applications, molecular fingerprints fail to preserve the complete molecular graph topology especially when using a small fingerprint length[6].

Following recent advances in natural language modeling, it was noted that molecules could be used directly as input for predictive models in the form of SMILES[7,8], a string-based representation developed to store and search molecular structures in a fast and easy way. SMILES have been used as inputs for deep learning architectures such as recurrent neural networks (RNNs)[9] and Transformers[10–14], though they suffer from the fact that molecules do not have unique SMILES representations. Other types of string-based representations have been proposed[15], e.g., Self-Referencing Embedded Strings (SELFIES)[16], which encode the molecular graph in the form of a Chomsky type-2 free context grammar appropriate for deep learning applications. An alternative is to use a graph representation of the molecule where nodes represent atoms and edges represent bonds. Such an approach is compatible with graph neural networks (GNNs), which have been extensively used for molecular property prediction[17,18]. Typically, GNNs aggregate the local information of each node with that of its neighboring atoms, and this information is then aggregated into a single molecular representation used to predict specific properties. Despite the fact that GNNs could provide the most natural way for learning representations of molecules that perform well in property prediction tasks, they also suffer from some drawbacks, e.g. in each layer of the GNN, an atom can only aggregate information from its nearest neighbors.

[1]Recursion, Salt Lake City, UT, USA. [2]Present address: Novo Nordisk Research Center, Lexington, MA, USA. e-mail: oscar.mendez-lucio@recursion.com; berton.earnshaw@recursion.com

A strategy frequently employed for learning meaningful representations in language involves training foundation models. These are models typically trained on extensive unlabeled datasets via self-supervised training, which can subsequently be finetuned for various downstream tasks[19–21]. This strategy has also been used to train foundation models for chemistry using SMILES[11,14]. Conversely, pretraining strategies for molecular graphs are not as straightforward and only a few attempts have been reported on relatively limited data[22–24]. In particular, Hu et al. proposed the Context Prediction approach[22] where the task consists of encoding part of the molecule using a GNN and matching the resulting embedding with the embedding of the rest of the molecule (referred as context graph) using negative sampling, and trained on 2 million molecules. In this paper we report a foundation model for chemistry trained on the molecular graphs of ~842 million molecules using self-supervised pretraining. We refer to this model as MolE, short for Molecular Embeddings. In particular, MolE learns molecular embeddings, at the atomic environment level, directly from a molecular graph using a transformer[25]. Specifically, we modified the disentangled attention in DeBERTa[26] to account for relative atom positions in a molecular graph. We also describe a self-supervised pretraining strategy for graphs in which each atom predicts its atom environment, i.e. the atom type and connectivity of all neighboring atoms. Using the ADMET tasks defined in the Therapeutic Data Commons (TDC)[27], we show that MolE is capable of being finetuned on small datasets to achieve top performance. This work is of relevance for chemical sciences where large amounts of unlabeled molecular structures are available but the size of labeled datasets is usually very small.

Results

A transformer model for molecular graphs

Model inputs. Contrary to SMILES-based models, in which characters composing the SMILES string are used as tokens, MolE directly works with graphs by providing both atom identifiers as input tokens and graph connectivity as the relative position information. Atom identifiers are calculated by hashing different atomic properties (i.e., Daylight atomic invariants) into a single integer[6]. In particular, this hash contains the following information: number of neighboring heavy atoms, number of neighboring hydrogen atoms, valence minus the number of attached hydrogens, atomic charge, atomic mass, attached bond types, and ring membership. Atom identifiers (also known as atom environments of radius 0) were computed using the Morgan algorithm[28] as implemented in RDKit[29]. In addition to tokens, MolE also takes graph connectivity information as input which is an important inductive bias since it encodes the relative position of atoms in the molecular graph. In this case, the graph connectivity is given as a topological distance matrix d where d_{ij} corresponds to the length of the shortest path over bonds separating atom i from atom j.

Model architecture. MolE uses a Transformer[25] as its base architecture, which also has been applied to graphs previously[30,31]. The performance of transformers can be attributed in large part to the extensive use of the self-attention mechanism. In standard transformers, the input tokens are embedded into queries, keys and values $Q, K, V \in R^{N\times d}$, which are used to compute self-attention as:

$$A = \frac{QK^T}{\sqrt{d}} \tag{1}$$

$$H_0 = \mathrm{softmax}(A)V \tag{2}$$

where $H_0 \in R^{N\times d}$ are the output hidden vectors after self-attention, and d is the dimension of the hidden space. In order to explicitly carry positional information through each layer of the transformer, MolE uses the disentangled self-attention from DeBERTa[26]:

$$a_{ij} = Q_i^c K_j^{cT} + Q_i^c K_{i,j}^{pT} + K_j^c Q_{j,i}^{pT} \tag{3}$$

$$A = \frac{a}{\sqrt{3d}} \tag{4}$$

$$H_0 = \mathrm{softmax}(A)V^c \tag{5}$$

where $Q^c, K^c, V^c \in R^{N\times d}$ are context queries, keys and values that contain token information (used in standard self-attention), and $Q_{i,j}^p, K_{i,j}^p \in R^{N\times d}$ are the position queries and keys that encode the relative position of the ith atom with respect to the jth atom. The use of disentangled attention makes MolE invariant with respect to the order of the input atoms.

Pretraining strategy. As mentioned earlier, self-supervised pretraining can effectively transfer information from large unlabeled datasets to smaller datasets with labels. Here we present a two-step pretraining strategy as shown in Fig. 1. The first step is a self-supervised approach to learn chemical structure representation. For this we use a BERT-like approach[32] in which each atom is randomly masked with a probability of 15%, from which 80% of the selected tokens are replaced by a mask token, 10% replaced by a random token from the vocabulary, and 10% are not changed. Different from BERT, the prediction task is not to predict the identity of the masked token, but to predict the corresponding atom environment (or functional atom environment[6]) of radius 2, meaning all atoms that are separated from the masked atom by two or less bonds. It is important to keep in mind that we used different tokenization strategies for inputs (radius 0) and labels (radius 2) and that input tokens do not contain overlapping data of neighboring atoms to avoid information leakage. This incentivizes the model to aggregate information from neighboring atoms while learning local molecular features. MolE learns via a classification task where each atom environment of radius 2 has a predefined label, contrary to the Context Prediction approach[22] where the task is to match the embedding of atom environments of radius 4 to the embedding of context atoms (i.e., surrounding atoms beyond radius 4) via negative sampling. The second step uses a graph-level supervised pretraining with a large labeled dataset. As proposed by Hu et al.[22], combining node- and graph-level pretraining helps to learn local and global features that improve the final prediction performance. More details regarding the pretraining steps can be found in the Methods section.

Achieving high performance on downstream tasks. MolE was pretrained using an ultra-large database of ~842 million molecules from ZINC20[33] and ExCAPE-DB[34], employing a self-supervised scheme (with an auxiliary loss) followed by a supervised pretraining with ~456K molecules (see Methods section for more details). We assess the quality of the molecular embedding by finetuning MolE on a set of downstream tasks. In this case, we use a set of 22 ADMET tasks included in the Therapeutic Data Commons (TDC) benchmark[27]. This benchmark is composed of 9 regression and 13 binary classification tasks on datasets that range from hundreds (e.g, DILI with 475 compounds) to thousands of compounds (such as CYP inhibition tasks with ~13,000 compounds). An advantage of using this benchmark is that it provides a standardized way to compare model performance (using the mean and standard deviation of 5 independent runs). As of September 2023, there have been ~15 different methods officially evaluated on this benchmark, including models using precomputed fingerprints (e.g., RDKit or Morgan Fingerprints), convolutional neural networks using SMILES, and different versions of graph neural networks such as ChemProp[18].

https://doi.org/10.1038/s41467-024-53751-y

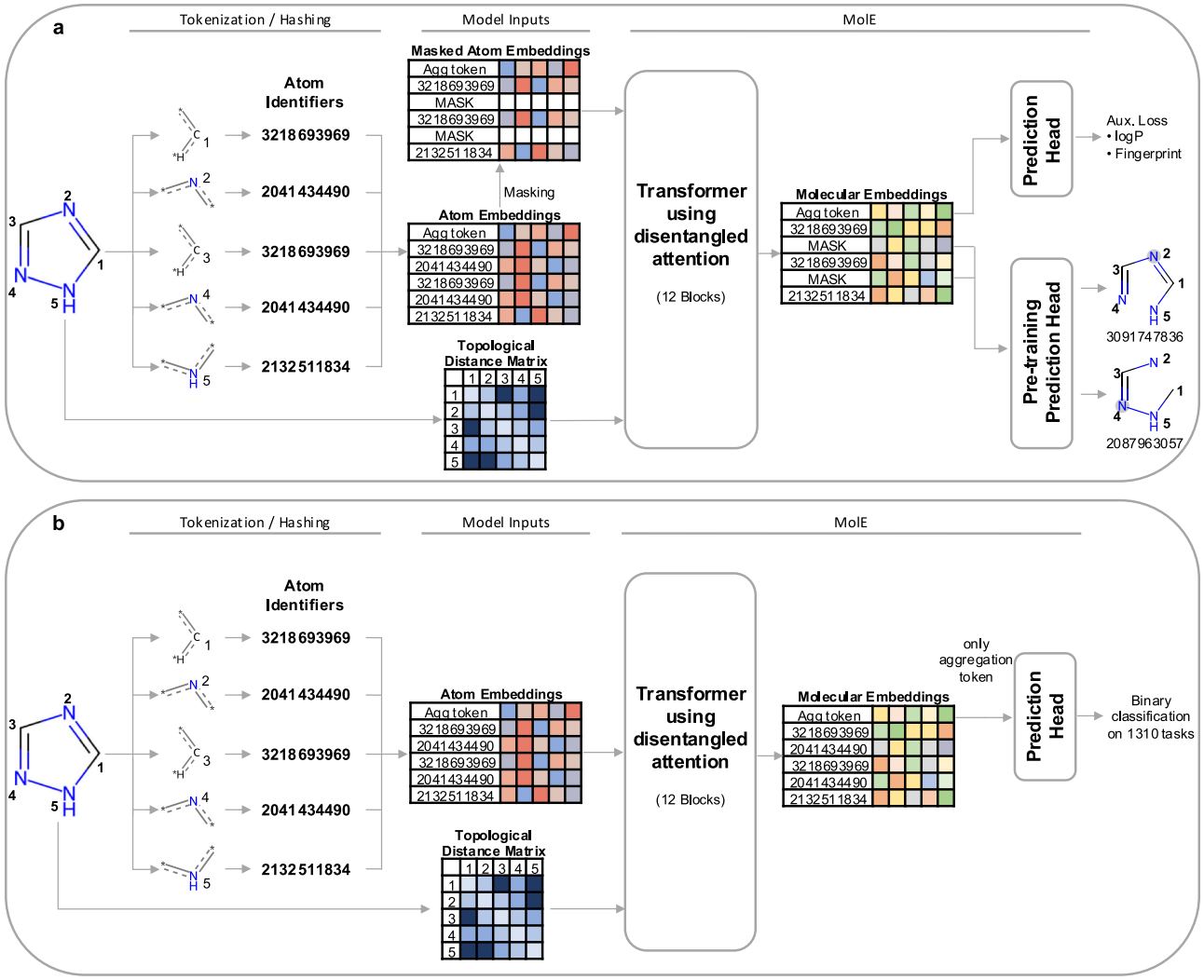

Fig. 1 | Pretraining and finetuning approaches used in this study. a A self-supervised approach in which an input atom is masked and the task is to predict the corresponding atom environment of radius 2, i.e. the masked atom plus all the neighboring atoms separated by no more than two bonds. Note that in this particular example the two masked tokens correspond to the same atom identifier (2041434490), but the atom environment associated with each is different. **b** Supervised approach in which embeddings of individual tokens are aggregated into an aggregation token which is fed into a prediction head.

Table 1 lists the result of MolE on the TDC benchmark achieving state-of-the-art performance on 10 of the 22 tasks (September 2023) and is the second best model on 4 tasks. More specifically, it was the best model on 6 regression and 4 classification (mainly CYP inhibition) tasks. Note that after including MolE's results, the next best model, ZairaChem, achieves top performance on only 5 of the 22 tasks. Not surprisingly, MolE achieves top results on tasks with larger datasets, such as those predicting CYP inhibition. Nonetheless, it also achieves top performance on some tasks with only a few hundred training examples, such as predicting half-life and CYP substrates.

Understanding MolE performance through ablation studies

Using the TDC ADMET tasks described previously, we conducted ablation studies to understand the impact of various architectural and pretraining choices on model performance. Table 2 provides a summary of the results, while Supplementary Tables S1–S5 provide a detailed view. In order to minimize the large amount of training required to complete these ablation studies, we performed the self-supervised training step on the GuacaMol dataset[35] (~1.2 million compounds) while maintaining the same supervised strategy.

Effect of disentangled attention. An important structural choice of MolE is the use of disentangled attention[26]. This feature uses relative positional embeddings to inform the model about the location of each atom in the molecule making it invariant to the order of the atoms. Removing disentangled attention has a significant impact on performance, similar to the impact of removing positional embeddings from standard bidirectional transformers[36,37]. Supplementary Fig. S1 shows the training loss and test accuracy for both scenarios of self-supervised pretraining. As expected, self-supervised pretraining using disentangled attention achieves high accuracy (of 0.96) during the masked modeling task while not using this attention results in an accuracy of 0.12. We also evaluate MolE without disentangled attention on the TDC benchmark where it performed worse in 19 out of 22 tasks compared to the model with disentangled attention (Supplementary Table S1).

Effect of pretrainings. Supplementary Table S1 shows the performance of MolE on the TDC ADMET tasks 1) without any pretraining and 2) with supervised pretraining only. Remarkably, training MolE on each individual task without any pretraining already exhibits better performance on 4 of the 22 tasks (PPBR, VDss, Half life and DILI) compared to early models benchmarked on TDC by Huang et al.[27]. This suggests that just the use of transformers with disentangled attention already

https://doi.org/10.1038/s41467-024-53751-y

Table 1 | Comparison between the best models reported in the Therapeutic Data Commons (TDC) leaderboard (as of September 2023) and finetuned MolE

				Best in TDC Leaderboard (September 2023)		Best in TDC Leaderboard (June 2024)		MolE
	Dataset	Metric	Size	Current best model	Result	Current best model	Result	Result
Absorption	Caco2	MAE	906	BaseBoosting	0.285 ± 0.005	MapLight	0.276 ± 0.005	0.329 ± 0.008
	HIA	AUROC	578	RFStacker	0.988 ± 0.002	MapLight + GNN	0.989 ± 0.001	0.984 ± 0.005
	Pgp	AUROC	1212	ZairaChem	0.935 ± 0.006	MapLight + GNN	0.938 ± 0.002	0.93 ± 0.005
	Bioavailability	AUROC	640	SimGCN	0.748 ± 0.033	SimGCN	0.748 ± 0.033	0.64 ± 0.046
	Lipophilicity	MAE	4200	Chemprop-RDKit	0.467 ± 0.006	Chemprop-RDKit	0.467 ± 0.006	**0.406** ± **0.009**
	Solubility	MAE	9982	Chemprop-RDKit	0.762 ± 0.020	Chemprop-RDKit	0.761 ± 0.025	0.776 ± 0.019
Distribution	BBB	AUROC	1975	Lantern RADR	0.962 ± 0.003	CFA	0.920 ± 0.006	0.903 ± 0.003
	PPBR	MAE	1797	Chemprop	7.811 ± 0.163	MapLight + GNN	7.526 ± 0.106	**7.229** ± **0.168**
	VDss	Spearman	1130	Basic ML	0.627 ± 0.010	MapLight + GNN	0.713 ± 0.007	**0.644** ± **0.013**
Metabolism	CYP2D6 inhibition	AUPRC	13,130	Chemprop-RDKit	0.672 ± 0.008	MapLight + GNN	0.790 ± 0.001	**0.679** ± **0.006**
	CYP3A4 inhibition	AUPRC	12,328	ZairaChem	0.875 ± 0.002	MapLight + GNN	0.916 ± 0.000	**0.876** ± **0.002**
	CYP2C9 inhibition	AUPRC	12,092	ZairaChem	0.786 ± 0.004	MapLight + GNN	0.859 ± 0.001	0.782 ± 0.001
	CYP2D6 substrate	AUPRC	664	ZairaChem	0.685 ± 0.029	ContextPred	0.736 ± 0.024	**0.692** ± **0.017**
	CYP3A4 substrate	AUROC	667	CNN (DeepPurpose)	0.662 ± 0.031	CFA	0.667 ± 0.019	**0.692** ± **0.019**
	CYP2C9 substrate	AUPRC	666	ZairaChem	0.441 ± 0.033	ZairaChem	0.441 ± 0.033	0.409 ± 0.014
Excretion	Half life	Spearman	667	Euclia ML model	0.547 ± 0.032	CFA	0.576 ± 0.025	**0.578** ± **0.032**
	Clearance microsome	Spearman	1102	RFStacker	0.625 ± 0.002	MapLight + GNN	0.630 ± 0.010	**0.632** ± **0.008**
	Clearance hepatocyte	Spearman	1020	Basic ML	0.440 ± 0.003	CFA	0.536 ± 0.020	**0.456** ± **0.027**
Toxicity	hERG	AUROC	648	SimGCN	0.874 ± 0.014	MapLight + GNN	0.880 ± 0.002	0.835 ± 0.018
	Ames	AUROC	7255	ZairaChem	0.871 ± 0.002	ZairaChem	0.871 ± 0.002	0.834 ± 0.015
	DILI	AUROC	475	ZairaChem	0.925 ± 0.005	ZairaChem	0.925 ± 0.005	0.852 ± 0.022
	LD50	MAE	7385	BaseBoosting KyQVZ6b2	0.552 ± 0.009	BaseBoosting KyQVZ6b2	0.552 ± 0.009	0.602 ± 0.016

MolE shows state-of-the-art results in 10 of the 22 tasks in TDC (bold and underlined values). This table shows the mean and standard deviation of 5 independent training runs.
MAE Mean Absolute Error, *AUROC* Area Under the Receiver Operating Characteristic Curve, *AUPRC* Area Under the Precision-Recall Curve.

Table 2 | Number of tasks on which various MolE ablations achieve best performance on the Therapeutic Data Commons (TDC) leaderboard (c. September 2023) for 22 ADMET (absorption, distribution, metabolism, excretion and toxicity) tasks

Pretraining	Self-supervised label	No auxiliary loss	logP as auxiliary loss	FP as auxiliary loss
None	–	0	–	–
Only supervised	–	2	–	–
Only self-supervised (~1.2 Million)	AtomEnvs	2	2	2
	FunctionalEnvs	4	2	3
Self-supervised (~1.2 Million) + Supervised	AtomEnvs	7	7	2
	FunctionalEnvs	6	4	6
Only self-supervised (~842 Million)	AtomEnvs	1	–	3
Self-supervised (842 Million) + Supervised	AtomEnvs	6	–	10

See Supplementary Information for detailed results.

positively impacts performance despite training only on small, task-specific datasets. Unfortunately, none of these models performed better than the best models on the TDC leaderboard (c. September 2023). Supervised pretraining alone improves performance on the TDC tasks, however the improvement over the baseline is only marginal, suggesting that supervised pretraining alone is not enough to learn transferable representations.

In Supplementary Table S2 we display the results for self-supervised pretraining. Here we consider three approaches: vanilla Masked Token Modeling (MTM), MolE and MolE-FE. While all approaches use atom environments of radius 0 as input, the prediction labels are different in each of them. The vanilla MTM task predicts the identity of the masked token used as input (i.e., predicts atom environments of radius 0), MolE task predicts atom environments of radius 2, while MolE-FE predicts functional atom environments[6] of radius 2. Though high accuracy (>98%) on the validation set was obtained with either strategy, MolE-FE performed slightly better on the benchmark tasks, outperforming previous models[27,38] on the leaderboard (c. September 2023) on 4 of 22 tasks, whereas MolE did so on only 2 tasks (see Table 2). However, adding supervised pretraining after self-supervised pretraining greatly improved the performance of MolE, achieving top results on 7 of the 22 tasks. MolE-FE results also improved (from 4 to 6 top results) when adding supervised pretraining. Interestingly, vanilla MTM was the worst performer of the three strategies. Two possible reasons for this are: 1) it seems to be an easier pretraining task due to the small vocabulary size (207 tokens) compared to predicting atom environments (~140,000 tokens) and 2) MolE and MolE-FE indirectly include the vanilla MTM task since they need to predict the identity of the masked atom in order to select the correct atom environment of radius 2.

Effect of auxiliary tasks. We also investigated the addition of the following auxiliary tasks during self-supervised pretraining as a way of possibly learning more meaningful chemical representations: learning the partition coefficient (logP) or learning a binary fingerprint of the molecule. For logP, we add both an additional token (referred to as an aggregation token) to the input and a prediction head to its encoded output and minimize the error of the logP prediction at the same time as the masked-token task (Fig. 1a). We calculate logP using RDKit[29].

Table 2 and Supplementary Table S3 show results for both MolE and MolE-FE trained with logP as auxiliary loss. In general, only a marginal decrease of performance was observed for the self-supervised version of MolE-FE, which in this case is the best performer in only 2 tasks instead of 4. Interestingly, using this auxiliary loss did not change the performance of MolE with either pretraining strategy.

For fingerprint learning, we framed this as a multitask binary classification problem where the task is to decide whether each atom environment of radius 2 in the vocabulary is present in the molecule or not. In this way, we force the model to aggregate the information of all atom environments in the molecule in a single vector that can be used as a starting point for downstream tasks. As can be seen in Table 2 and Supplementary Table S4, this auxiliary task again resulted in equal or lower performance for both environments and both pretraining strategies. Our hypothesis is that fingerprint learning is a very complex task due to the numerous outputs and requires a larger number of training examples before providing any benefit.

Effect of data size. In our ongoing exploration, we also analyzed the influence of extending pretraining to significantly larger datasets: two datasets randomly sampled from ZINC with 10 and 100 million molecules each and the full ZINC dataset containing ~842 million molecules. This was motivated by the fact that larger training sets tend to improve model generalization. For clarity, we have included the results of this extended pretraining in Supplementary Fig. S3 and Supplementary Tables S5 and S6. Note that performance improvements with larger training sets were significant, improving model performance across many tasks. The performance of MolE was particularly impressive when trained on full ZINC, showing improvements in 10 tasks, outpacing the previously top-performing models referenced in the TDC leaderboard (c. September 2023). These results emphasize the benefits realized from pretraining on larger and more diverse datasets, giving the model a more comprehensive understanding of chemistry. As a matter of fact, the model was exposed to approximately 60 million different Bemis-Murcko scaffolds when trained on the full ZINC20, with the most recurrent ones being the pyridine, cyclohexane, and thiophene rings (found in 2.3, 1.6 and 0.99 million substances respectively)[33]. Nonetheless, considering that chemical diversity is a difficult concept to measure, the effect of diversity falls outside the scope of this current study. There were a few tasks where the performance did not significantly improve or even slightly decreased. Such instances might be attributed to factors such as the complexity of the task, the nature of the chemical structures involved, or other limitations. A deeper analysis is necessary to understand these specific cases, but the overall results suggest that increasing the amount of pretraining data substantially improves model performance.

MolE embeddings are a meaningful representation of molecular graphs

An important feature of MolE is its ability to generate meaningful molecular embeddings. In order to demonstrate this, we subjected the molecular embeddings to both intrinsic and extrinsic tests[39]. The intrinsic evaluation measures the quality of the embeddings independent of its predictive functionality. This test largely concentrates on assessing the topological or functional relationships between molecular embeddings, similar to how syntactic or semantic relationships are assessed in a word embedding, and can be considered a more generalized evaluation of the ability of embeddings to capture structural attributes of molecules. In contrast, the extrinsic analysis inspects the efficacy of the embeddings in downstream tasks, making this assessment more computationally demanding yet meaningful for particular tasks, though less interpretable. Since there is no single evaluation that thoroughly examines model embeddings, it is recommended to employ multiple metrics[39].

In this study, we employed an intrinsic test centered around similarity, specifically neighbors variation[40]. For this, we compute the molecular embeddings of ~79 K compounds from the GuacaMol test set. For each compound embedding, we located the k-nearest neighbors (where k is 5, 10, 15, 25, 50, 100) using cosine similarity. Then, we established the overlap of k-nearest neighbors to those identified using Morgan fingerprints (radius 2)[6,28] or RDKitFP[29] combined with Tanimoto similarity, a widely recognized method for evaluating chemical similarity[3]. The distribution of neighborhood overlap across all molecules is shown in Fig. 2 as boxplots. Overall, observations indicate that closer neighborhoods (i.e., $k = 5$, 10) are more conserved and the overlap decreases when considering more distant neighbors. Moreover, MolE embeddings have a limited overlap with Morgan fingerprints, conserving a median of approximately 20% of neighbors for the 5 or 10 nearest neighbors, a number that decreases for more distant neighbors. This number is considerably lower when compared to the existing overlap between the two baselines, Morgan and RDKitFP, sharing a median of around 40% neighbors at $k = 5$. A similar comparison was conducted using MolE embeddings of the model solely pretrained using the self-supervised approach. Notably, these embeddings demonstrate a substantial overlap with Morgan fingerprints, sharing a median of roughly 60% of neighbors at $k = 5$, surpassing the overlap between Morgan and RDKitFP. The similar behavior of MolE self-supervised embeddings and Morgan fingerprints can be attributed to the fact that both are based on atom environments of radius 2, and also reaffirms the success of the self-supervised approach in learning chemical information. Altogether, these results demonstrate that embeddings do capture information about the chemical structure, and this remains true regardless of their performance in prediction tasks.

The extrinsic evaluation of molecular embeddings was executed using the TDC tasks described above. In this case, MolE embeddings are being used as input features for training an XGBoost model. This procedure does not update the embeddings, allowing a proper evaluation of their quality for the particular prediction task. The outcome of this evaluation is detailed in Supplementary Table S7, where the best performers in 12 of the 22 tasks were models trained with MolE embeddings, as opposed to those trained using embeddings from the self-supervised-only MolE, Morgan fingerprints (radius 2) or RDKitFP. It is noteworthy that XGBoost trained with MolE embeddings also outperformed TDC leaderboard models for the hepatocyte clearance regression task. These results imply that the supervised pretraining makes MolE embeddings more biologically significant, and may partially explain the discrepancy between these embeddings and Morgan or RDKitFP fingerprints in the intrinsic evaluation, since the latter only contain information regarding the molecular structure.

Figure 2b offers a UMAP representation of MolE embedding space. Each point represents the embedding of an atom environment present in the ~79K compounds from the GuacaMol test set. For simplicity, we are just showcasing the atom environments centered around a heteroatom. The selected examples show cases where different subgraphs, anticipated to have similar biological impacts (like bioisosteres), are positioned closely in the embedding space. Overall, it is crucial to note that results found in this section demonstrate that MolE (self-supervised only) embeddings capture chemical information similar to Morgan fingerprints, and that MolE embeddings contain a degree of biological information that enhances their performance in TDC benchmark tasks. However, one should not interpret these findings as suggesting that these embeddings will provide a superior molecular representation across all possible tasks, whether for similarity search or predictive capabilities.

Discussion

In this paper we report MolE, which uses a transformer with disentangled attention (i.e., DeBERTa) to predict chemical and biological

https://doi.org/10.1038/s41467-024-53751-y

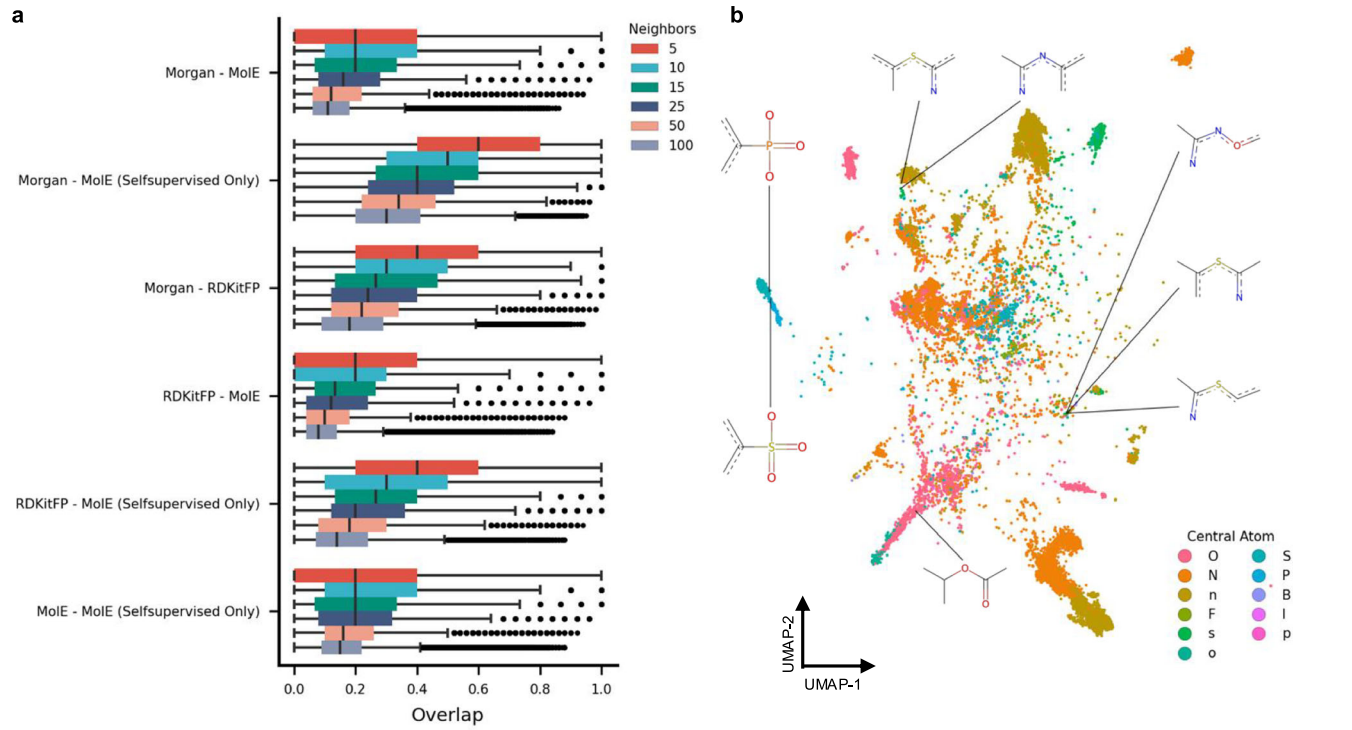

Fig. 2 | Results from evaluating MolE embeddings. a Evaluation of molecular embeddings on a neighbor variation test. These boxplots represent the distribution of neighborhood overlap across all molecules ($n = 79{,}568$) for different molecular encodings. The closer the overlap is to 1, the more k-nearest neighbors are shared between the two encoding methods. Morgan fingerprints of radius 2 show high neighborhood overlap with embeddings from MolE pretrained solely on the self-supervised task. The centerline of the boxplot represents the median; the bounds of the box represent the first and third quartile and the whiskers the 1.5 interquartile rage (IQR). **b** U-map representation of the MolE atomic embeddings for environments centered on heteroatoms. It is interesting to see that different subgraphs with similar biological effects (e.g. bioisosteres) lay close in the embedding space.

properties directly from molecular graphs. The specific contributions in this paper are:

- We showed that transformers with disentangled attention can directly be used on molecular graphs when they are represented by atom environments of radius 0 and relative positional embeddings.
- We proposed a self-supervised approach for molecular graphs in which the task is to learn atom environments of radius greater than 0 from atom environments of radius 0, which only include information about a single atom and all bonds attached to it.
- By using a two-step pretraining approach – self-supervised learning followed by supervised learning – we were able to train models that performed better than previously reported approaches on 10 of the 22 tasks included in the Therapeutic Data Commons leaderboard as of September 2023 and in 5 of the 22 tasks by June 2024 (Table 1).

We hypothesize that learning atom environments forces the model to aggregate the local chemical groups that will be used for prediction. Learning an embedding of atom environments and how to aggregate them into a molecular embedding can help to solve some problems of classical fingerprints such as sparsity and clashes when using bit vectors. Interestingly, this self-supervision approach is not limited to transformers since it can easily be used to pretrain GNN. The effect of data diversity during the self-supervision is still to be determined since we only used drug-like molecules. Nonetheless we expect that larger and more diverse datasets can only improve current performance of the model. Overall we consider this work as an initial step towards a foundation model for chemical property prediction.

Methods

Datasets

The self-supervised pretraining was done using ~842 million molecules from ZINC20[33] and ExCAPE-DB[34] and validated on set of ~44 million molecules. For ablation studies, the self-supervised pretraining was done using the GuacaMol[35] training set containing ~1.2 million compounds and the GuacaMol[35] test set of 79K molecules for validation. It is worth mentioning that only molecules with no more than 100 heavy atoms were used, and we removed from the training set all molecules included in TDC test sets to avoid information leakage. All remaining SMILES were transformed into molecular graphs using RDKit from which distance matrices and atom environments were calculated (radius 0 to be used as input and radius 2 as labels). Atom environment identifiers were aggregated into two vocabularies, one used for input and one for labels. The input vocabulary consists of 207 tokens corresponding to all atom environments of radius 0 present in the 1.2 million molecules in GuacaMol, plus the ~880 million molecules in ZINC20[33]. Similarly the vocabulary used for labels contains ~141K atom environments or ~114K functional atom environments (Table 3). These were selected taking the 90K most frequent atom environments or functional environments from GuacaMol training set plus the 90K most frequent form ZINC20 and removing those that appear in less than 3 molecules.

The supervised pretraining was done using ~456,000 molecules with activity data on 1310 readouts from ChEMBL[41] which was curated following the protocol proposed by Mayr et al.[42] and which was used

Table 3 | Summary of tasks and input/output vocabularies used for supervised and self-supervised pretraining approaches

	Self-supervised pretraining		Supervised pretraining	
Label type	Input (Radius 0)	Output (Radius 2)	Molecules	Tasks
Atom Envs	207	~141K	~456K	1310
Functional Envs	207	~114K	~456K	1310

Table 4 | List of values used during hyperparameter optimization of XGBoost models

Hyperparameter name	Values
Gamma	0, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6, 51.2, 102.4, 200
Learning rate	0.01, 0.03, 0.06, 0.1, 0.15, 0.2, 0.25, 0.300000012, 0.4, 0.5, 0.6, 0.7
Maximum depth	5, 6, 7, 8, 9, 10, 11, 12, 13, 14
Number of estimators	50, 65, 80, 100, 115, 130, 150
Alpha	0, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6, 51.2, 102.4, 200
Lambda	0, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6, 51.2, 102.4, 200

for pretraining by Hu et al.[22]. An important difference with the approach of Hu et al.[22] is that we did not use the complete dataset for supervised pretraining, but instead we removed ~9900 molecules that were present in the test sets of the TDC benchmark. This avoids information leakage since in some cases the overlap between the TDC test sets and the dataset used for supervised pretraining could reach more than 80% of the molecules (Supplementary Table S8). Removing these compounds from pretraining stages makes MolE models suitable to be fairly benchmarked in TDC.

Training

MolE uses the DeBERTa[26] base configuration (12 transformer layers with 12 attention heads each) with a prediction head connected to the output of an aggregation token composed of a two-layer MLP with GELU[43] and dropout[44] layers in between (Supplementary Fig. S4). Self-supervised pretraining was carried out for 420,000 steps using a batch size of 512 molecules distributed across 8 GPUs (making an effective batch size of 4096 compounds). Learning rate was linearly increased to 2×10^{-4} during the first 10,000 steps, followed by a linear decaying learning rate schedule. Supervised approach was pretrainied for 60,000 steps using a batch size of 512 molecules in a single GPU. In this case we used a learning rate of 5×10^{-6} with the same schedule as the self-supervised training. Gradient norms were clipped at 1.0 and no weight decay was used.

For finetuning, only the weights of the prediction head were randomly initialized. Models were trained for 100 epochs using a batch size of 32 molecules. The model was evaluated on the validation set every 5 epochs and only the weights from the best-performing model according to these validation metrics were retained for further evaluation in the test set. We ran hyperparameter optimization to find the best learning rate (1e−5, 8e−6, 5e−6, 3e−6, 1e−6, 5e−7) and dropout rate (0, 0.1, 0.15 in the prediction head) with a 5-fold cross validation using the folds provided in the TDC benchmark datasets. During training, the learning rate was linearly increased during the first 10% of the training steps, and then kept constant after that.

Benchmark

MolE models were evaluated using the ADMET benchmark group from the Therapeutic Data Commons (TDC)[27]. This benchmark provides datasets that have been previously standardized and divided into training and test sets (80%/20% using scaffold splitting) to fairly evaluate molecular property prediction models. It is composed of 22 different classification and regression tasks for properties relevant to drug discovery. For example, cell permeability (Caco-2), Human Intestinal Absorption (HIA), and p-glycoprotein inhibition (Pgp), together with other physicochemical properties, can give a good estimate of how much of the drug will be absorbed by the body. Properties like volume of distribution (VDss), the plasma protein binding rate (PPBR), and the Blood-Brain Barrier (BBB) give us an idea of how the drug will be distributed across the body. Knowing whether a molecule inhibits or is substrate for a particular cytochrome (CYP) isoform indicates possible biotransformations that can affect the time the drug remains in the body, which is measured by half-life and clearance rate. Finally, knowing whether a molecule is cardiotoxic (hERG), genotoxic (Ames), or hepatotoxic (DILI) is of great importance to get a safe drug into the clinic. More information about each of these tasks is listed in Table 1.

TDC maintains a leaderboard of the performance of different models on these tasks. These models provide a standard for performance comparison purposes since they use different architectures and encoding strategies, e.g., pre-calculated descriptors such as Morgan or RDKit 2D fingerprints[29], CNNs trained using SMILES, and different variations of graph-based approaches such as NeuralFP[17], GCNs[45], AttentiveFP[46], and others. It also includes models pretrained with different strategies e.g., AttrMasking and ContextPred[22].

XGBoost models

XGBoost models were used to evaluate the quality of molecular embeddings in extrinsic prediction tasks. For each model, hyperparameter search was performed using Bayesian optimization across the values listed in Table 4. This process finds the set of hyperparameters that minimizes the error during cross validation. For this, a gaussian process model was used as an optimizer and updated for 120 iterations. A final model was trained with the best hyperparameters and then evaluation on the test splits provided by TDC.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The data that support the findings of this study is completely public and available in the following links: • ZINC: https://zinc.docking.org/ • GuacaMol: https://figshare.com/projects/GuacaMol/56639 • Therapeutic Data Commons (TDC): https://tdcommons.ai/ Source data are provided with this paper.

Code availability

The code to use the model reported in this study is be available under the Attribution-NonCommercial 4.0 International License (CC-BY-NC 4.0) in https://github.com/recursionpharma/mole_public[47,48].

References

1. Martin, Y. C. Hansch analysis 50 years on. *WIREs Comput. Mol. Sci.* **2**, 435–442 (2012).
2. Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
3. Willett, P., Barnard, J. M. & Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **38**, 983–996 (1998).
4. Hansch, C. & Fujita, T. p-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *J. Am. Chem. Soc.* **86**, 1616–1626 (1964).
5. Durant, J. L., Leland, B. A., Henry, D. R. & Nourse, J. G. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.* **42**, 1273–1280 (2002).
6. Rogers, D. & Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **50**, 742–754 (2010).

7. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).
8. Weininger, D., Weininger, A. & Weininger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Comput. Sci.* **29**, 97–101 (1989).
9. Winter, R., Montanari, F., Noé, F. & Clevert, D. A. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.* **10**, 1692–1701 (2019).
10. Chithrananda, S. & Ramsundar, B. ChemBERTa: Utilizing Transformer-Based Attention for Understanding Chemistry. Preprint at https://arxiv.org/abs/2010.09885 (2020).
11. Ahmad, W., Simon, E., Chithrananda, S., Grand, G. & Ramsundar, B. ChemBERTa-2: Towards chemical foundation models. Preprint at http://arxiv.org/abs/2209.01712 (2022).
12. Wang, S., Guo, Y., Wang, Y., Sun, H. & Huang, J. 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*, 429–436 (Association for Computing Machinery, 2019).
13. Fabian, B. et al. Molecular representation learning with language models and domain-relevant auxiliary tasks. Preprint at https://arxiv.org/abs/2011.13230 (2020).
14. Ross, J. et al. Large-scale chemical language representations capture molecular structure and properties. *Nat. Mach. Intell.* **4**, 1256–1264 (2022).
15. O'Boyle, N. & Dalke, A. DeepSMILES: an adaptation of SMILES for use in machine-learning of chemical structures. Preprint at https://chemrxiv.org/engage/chemrxiv/article-details/60c73ed6567dfe7e5fec388d (2018).
16. Krenn, M., Häse, F., Nigam, A. K., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn. Sci. Technol.* **1**, 045024 (2020).
17. Duvenaud, D. et al. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems* (eds Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R.) (NeurIPS, 2015).
18. Yang, K. et al. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **59**, 3370–3388 (2019).
19. Brown, T. B. et al. Language Models are Few-Shot Learners. Preprint at http://arxiv.org/abs/2005.14165 (2020).
20. Zhang, S. et al. OPT: Open Pre-trained Transformer Language Models. Preprint at http://arxiv.org/abs/2205.01068 (2022).
21. Chowdhery, A. et al. PaLM: Scaling Language Modeling with Pathways. Preprint at http://arxiv.org/abs/2204.02311 (2022).
22. Hu, W. et al. Strategies for pre-training Graph Neural Networks. Preprint at http://arxiv.org/abs/1905.12265 (2019).
23. Wang, Y., Wang, J., Cao, Z. & Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **4**, 279–287 (2022).
24. Beaini, D. et al. Towards Foundational Models for Molecular Learning on Large-Scale Multi-Task Datasets. Preprint at http://arxiv.org/abs/2310.04292 (2023).
25. Vaswani, A. et al. Attention Is All You Need. Preprint at http://arxiv.org/abs/1706.03762 (2017).
26. He, P., Liu, X., Gao, J. & Chen, W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. Preprint at http://arxiv.org/abs/2006.03654 (2020).
27. Huang, K. et al. Artificial intelligence foundation for therapeutic science. *Nat. Chem. Biol.* **18**, 1033–1036 (2022).
28. Morgan, H. L. The generation of a unique machine description for chemical structures-A technique developed at chemical abstracts service. *J. Chem. Doc.* **5**, 107–113 (1965).
29. Landrum, G. A. RDKit:Open-source cheminformatics. https://doi.org/10.5281/zenodo.7671152 (Zenodo, 2023).
30. Liu, X., Ye, K., van Vlijmen, H. W. T., IJzerman, A. P. & van Westen, G. J. P. DrugEx v3: scaffold-constrained drug design with graph transformer-based reinforcement learning. *J. Cheminform.* **15**, 24 (2023).
31. Ying, C. et al. Do transformers really perform badly for graph representation? *Adv. Neural Inf. Process. Syst.* **34**, 28877–28888 (2021).
32. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Preprint at http://arxiv.org/abs/1810.04805 (2018).
33. Irwin, J. J. et al. ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *J. Chem. Inf. Model.* **60**, 6065–6073 (2020).
34. Sun, J. et al. ExCAPE-DB: An integrated large scale dataset facilitating Big Data analysis in chemogenomics. *J. Cheminform.* **9**, 1–9 (2017).
35. Brown, N., Fiscato, M., Segler, M. H. S. & Vaucher, A. C. GuacaMol: Benchmarking Models for De Novo Molecular Design. *J. Chem. Inf. Model.* **59**, 1096–1108 (2018).
36. Haviv, A., Ram, O., Press, Ofir, Izsak, P. & Levy, O. Transformer Language Models without Positional Encodings Still Learn Positional Information. Preprint at http://arxiv.org/abs/2203.16634 (2022).
37. Lasri, K., Lenci, A. & Poibeau, T. Word Order Matters when you Increase Masking. Preprint at http://arxiv.org/abs/2211.04427 (2022).
38. Tian, H., Ketkar, R. & Tao, P. ADMETboost: a web server for accurate ADMET prediction. *J. Mol. Model.* **28**, 408 (2022).
39. Wang, B., Wang, A., Chen, F., Wang, Y. & Kuo, C.-C. J. Evaluating word embedding models: methods and experimental results. *APSIPA Trans. Signal Inf. Proces.* **8**, e19 (2019).
40. Pierrejean, B. & Tanguy, L. Towards qualitative word embeddings evaluation: Measuring neighbors variation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop* (Association for Computational Linguistics, 2018).
41. Gaulton, A. et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **40**, D1100–7 (2012).
42. Mayr, A. et al. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem. Sci.* **9**, 5441–5451 (2018).
43. Hendrycks, D. & Gimpel, K. Gaussian Error Linear Units (GELUs). Preprint at http://arxiv.org/abs/1606.08415 (2016).
44. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
45. Kipf, T. N. & Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. Preprint at http://arxiv.org/abs/1609.02907 (2016).
46. Xiong, Z. et al. Pushing the Boundaries of Molecular Representation for Drug Discovery with the Graph Attention Mechanism. *J. Med. Chem.* **63**, 8749–8760 (2020).
47. Méndez-Lucio, O., Nicolaou, C. & Earnshaw, B. *MolE: A Foundation Model for Molecular Graphs Using Disentangled Attention (Code)*. https://doi.org/10.5281/ZENODO.13891642 (Zenodo, 2024).
48. Méndez-Lucio, O., Nicolaou, C. & Earnshaw, B. *MolE: A Foundation Model for Molecular Graphs Using Disentangled Attention [Source Code]*. https://doi.org/10.24433/CO.2928188.v1 (CodeOcean, 2024).

Acknowledgements

Authors thank Jake Schmidt, Kian Kenyon-Dean and Estefania Barreto-Ojeda for their engineering support. To all the HPC team at Recursion, especially Alexander Timofeyev, Brent Gawryluik and Joshua Fryer for keeping BioHive computer cluster in optimal conditions to run all the

training jobs in this work. Thibault Varin, Sarah Karbalaeikhani and Maria Elena Garcia Ochagavia for their feedback as early users and to all the people that helped to improve and test this project.

Author contributions

O.M.L. and B.E. conceived the idea and planned the project. O.M.L. wrote the code and performed the experiments. O.M.L., C.N., and B.E. wrote the paper and had insightful discussions that helped to improve the initial idea.

Competing interests

O.M.L., C.N., and B.E. are or were employees of Recursion when developing this project.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-024-53751-y.

Correspondence and requests for materials should be addressed to Oscar Méndez-Lucio or Berton Earnshaw.

Peer review information *Nature Communications* thanks Jinho Chang and the other anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.