BMC Bioinformatics

RESEARCH

Open Access

Can large language models understand molecules?

Shaghayegh Sadeghi[1*], Alan Bui[1], Ali Forooghi[1], Jianguo Lu[1] and Alioune Ngom[1]

*Correspondence:
sadeghi3@uwindsor.ca

[1] School of Computer Science, Univeristy of Windsor, Sunset Ave, Windsor, ON N9B 3P4, Canada

Abstract

Purpose: Large Language Models (LLMs) like Generative Pre-trained Transformer (GPT) from OpenAI and LLaMA (Large Language Model Meta AI) from Meta AI are increasingly recognized for their potential in the field of cheminformatics, particularly in understanding Simplified Molecular Input Line Entry System (SMILES), a standard method for representing chemical structures. These LLMs also have the ability to decode SMILES strings into vector representations.

Method: We investigate the performance of GPT and LLaMA compared to pre-trained models on SMILES in embedding SMILES strings on downstream tasks, focusing on two key applications: molecular property prediction and drug-drug interaction prediction.

Results: We find that SMILES embeddings generated using LLaMA outperform those from GPT in both molecular property and DDI prediction tasks. Notably, LLaMA-based SMILES embeddings show results comparable to pre-trained models on SMILES in molecular prediction tasks and outperform the pre-trained models for the DDI prediction tasks.

Conclusion: The performance of LLMs in generating SMILES embeddings shows great potential for further investigation of these models for molecular embedding. We hope our study bridges the gap between LLMs and molecular embedding, motivating additional research into the potential of LLMs in the molecular representation field. GitHub: https://github.com/sshaghayeghs/LLaMA-VS-GPT.

Keywords: Large language models, LLaMA, GPT, SMILES embedding

Introduction

Molecule embedding is an important task in drug discovery [1, 2], and finds wide applications in related tasks such as molecular property prediction [3–6], drug-target interaction (DTI) prediction [7–9] and drug-drug interaction (DDI) prediction [10, 11].

Molecule embedding techniques learn the features either from the molecular graphs that encode the connectivity information of a molecule structure or from the line annotations of their structures, such as the popular SMILES (simplified molecular-input line-entry system) representation [4].

Molecule embedding via SMILES strings evolve and synchronize with the advances in language modelling [12, 13], starting with static word embedding [14], to contextualized

pre-trained models [4, 15, 16]. These embedding techniques aim to capture relevant structural and chemical information in a compact numerical representation [17]. The fundamental hypothesis asserts that structurally similar molecules behave in similar ways. This enables machine learning algorithms to process and analyze molecular structures for property prediction and drug discovery tasks.

With the breakthroughs made in LLMs, one prominent question is whether LLMs can understand molecules and make inferences on molecule data? More specifically, can LLMs produce high quality semantic representations? Gua et al. [18] made a preliminary study by evaluating several chemical inference tasks using LLMs. Their study has been limited to utilizing and evaluating LLMs performance in answering SMILES-related queries. We move further by exploring the ability of these models to effectively embed SMILES has yet to be fully explored, maybe partially due to cost of API calls. Our conclusions are:

(1) LLMs do outperform traditional methods.
(2) The performance is task dependent, sometimes data dependent.
(3) Newer versions of LLMs do improve over older versions, even though they are trained on more generic tasks.
(4) We observe that embeddings from LLaMA overall outperform GPT embeddings.
(5) Another interesting observation of our research is that LLaMA and LLaMa2 are very close regarding embedding performance.

Related work

For accurate prediction of chemical properties using machine learning, leveraging molecule embeddings as input feature vectors is crucial [19]. Early molecular embedding methods such as Morgan FingerPrint (FP) [20] encode the structural information of a molecule into a fixed-length binary or integer vector with the knowledge of chemistry.

However, for a more generalized embedding, numerous studies have explored methods to embed molecular structures. While some studies focus on the graph representation of the molecular structure to encode the important topology information directly [21–23], many choose the string representation of molecules (SMILES) due to rapid advancements in natural language processing (NLP). Initial efforts in this domain utilized foundational NLP architectures like auto-encoders [24] and recurrent neural networks (RNN) to generate embeddings [19]. However, the scarcity of labelled data has shifted focus towards methods that can be pre-trained on unlabeled data, such as Mol2Vec and SPVec [14, 25].

With the increasing prominence of transformer models in natural language analysis-where they are pre-trained on extensive unsupervised data and then fine-tuned for specific tasks like classification-transformer-based models have become increasingly relevant in the SMILES language domain. For instance, SMILES-BERT [15] has inspired numerous studies to adapt the transformers framework. These studies try to modify this framework to improve their performance on SMILES strings by adapting RoBERTa (Robustly optimized BERT approach) instead of the BERT model [6] or develop domain-specific self-supervised pre-training tasks [16], or integrate the local message

passing mechanism of graph neural networks (GNNs) into BERT to enhance learning from molecular graphs [5]. Additionally, MolFormer [4] introduces a novel approach by combining molecular language with transformer encoder models, incorporating rotary positional embeddings (RoPE) from RoFormer, to produce more effective molecular embeddings [4, 26].

However, pre-training these models on millions of molecules requires substantial hardware resources. For example, MolFormer necessitates up to 16 V100 graphics processing units (GPUs) [4]. Consequently, it is computationally more feasible to use pre-trained large language models (LLMs), such as GPT [27] and LLaMA [28, 29], for generating embeddings. These models have already been trained on vast amounts of data, making them readily available for processing SMILES strings to obtain molecular embeddings without extensive hardware.

Up to our current knowledge, the application of GPT and LLaMA in chemistry has primarily been limited to utilizing and evaluating its performance in answering queries. Further exploration and implementation of LLMs for more advanced tasks within chemistry are yet to be thoroughly documented. For example, to examine how well LLMs understand chemistry, Guo et al. [18] used LLMs to assess the performance of these models on practical chemistry tasks only using queries. Their results demonstrate that GPT models are comparable with classical machine learning models when applied to chemical problems that can be transformed into classification or ranking tasks such as property prediction. However, they stop evaluating the LLM's ability to answer prompts and do not evaluate the embedding power of LLMs. Hence, inspired by many language-based methods that tried to extract molecular embedding, our study represents a pioneering effort, being the first to rigorously assess the capabilities of LLMs like GPT and LLaMA in using LLMs embedding for chemistry tasks.

LLMs

LLMs, exemplified by architectures like BERT [12], GPT [27], LLaMA [28], and LLaMA2 [29] excel at understanding context within sentences and generating coherent text. They leverage attention mechanisms and vast training data to capture contextual information, making them versatile for text generation, translation, and sentiment analysis tasks. While Word2Vec enhances word-level semantics, language models provide a deeper understanding of context and facilitate more comprehensive language understanding and generation. Pre-trained models from LLMs can transform text into dense, high-dimensional vectors, which capture contextual information and meaning. Using pre-trained LLMs offers an edge as they transfer knowledge from their vast training data, enabling the extraction of context-sensitive representations without requiring extensive task-specific data or feature engineering [30].

This work focuses on obtaining the embeddings of SMILES strings from GPT and LLaMA models to find the model that achieves the best performance. OpenAI [31] present many GPT-based embeddings including: *'text-embedding-ada-002'*, *'text-embedding-3-small'*, *'text-embedding-3-large'*. Our research used the most recent embedding model, *text-small-3-embeddings*. This model is acclaimed for being the best among available embedding models and the most affordable method available by OpenAI. Text-small-3-embeddings employs the *'cl100k-base'* token calculator to generate embeddings,

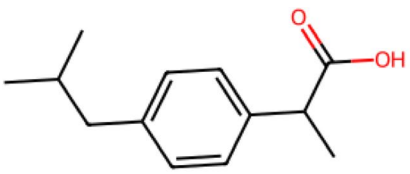
resulting in a 1536-dimensional vector representation. We input SMILES strings into this model, allowing GPT to create embeddings for each string. These embeddings serve as the feature vector for our classification tasks.

In parallel, we leveraged the capabilities of LLaMA [28] and its advanced variant, LLaMA2 [29]. These models, ranging from only 7 to 65 billion parameters, are built on the Transformers architecture. LLaMA2, an enhancement of LLaMA, benefits from training on an expanded publicly available data set. Its pre-training corpus grew by 40%, and its context length doubled to 4096 tokens. LLaMa models employ a decoder-only Transformer architecture with causal multi-headed attention in each layer. Drawing architectural inspiration from prominent language models like GPT-3 and PaLM (Pathways Language Model) [32], they incorporate features such as pre-normalization, RMSNorm, SwiGLU activation functions, and rotary positional embeddings (RoPE) [26] in every transformer layer.

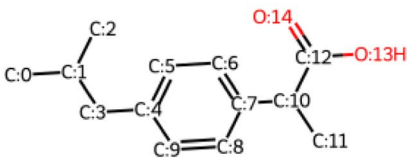
The training dataset of LLaMA [28, 33] predominantly comprises webpages, accounting for over 80% of its content. This is supplemented by various sources, including 6.5% code-centric data from GitHub and StackExchange, 4.5% literary content from books, and 2.5% scientific material primarily sourced from arXiv.

In contrast, GPT [33, 34] was developed using a comprehensive and mixed dataset. This dataset includes diverse sources like CommonCrawl, WebText2, two different book collections (Books1 and Books2), and Wikipedia.

SMILES is utilized as a "chemical language" that encodes the structural elements of a chemical graph-including atoms, bonds, and rings-into a brief textual format. This is achieved through a systematic, depth-first tree traversal of the chemical structure. The method uses alphanumeric characters to represent atoms (such as C, S, Br) and symbols such as '-', '=', and '#' to indicate different types of chemical bonds. For instance, the SMILES notation for Ibuprofen is *CC(C)Cc1ccc(cc1)C(C)C(O)=O* (Fig. 1).



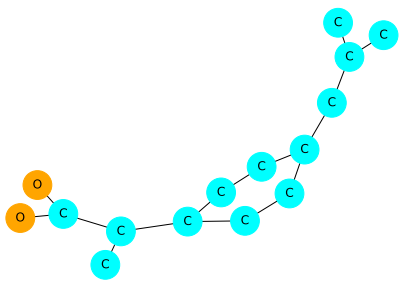
(a) Kekule diagram



(b) Kekule diagram with properties

CC(C)Cc1ccc(cc1)C(C)C(O)=O

(c) SMILES string



(d) Molecular graph

Fig. 1 Drug chemical representations

Table 1 Comparison of tokenizers for molecular SMILES string

Model	Tokenization strategy	Example tokenization of 'CCS(=O)(=O)CCBr'
BERT tokenizer	Subword-based tokenization	['CC', '##S', '(', '=', 'O', ')', '(', '=', 'O', ')', 'CC', '##B', '##r']
GPT tokenizer	cl100k-base	['CC', 'S', '(', '=', 'O', ')(', '=', 'O', ')', 'CC', 'Br']
LLaMA2 tokenizer	SentencePiece byte-pair encoding-based	['_C', , 'CS', '(', '=', 'O', ')(', '=', 'O', ')', 'CC', 'Br']
ChemBERTa tokenizer	Byte-pair encoding-based	['C', 'C', 'S', '(', '=', 'O', ')', '(', '=', 'O', ')', 'C', 'C', 'B', 'r']
MolFormer-XL tokenizer	SMILE regex	['C', 'C', 'S', '(', '=', 'O', ')', '(', '=', 'O', ')', 'C', 'C', 'Br']

Table 2 Comparison of embedding models used in this study

Model	Dim. Size	# Layers	# Parameters	Speed* (s)
Morgan FP (Radius=2)	1024	Not applicable	Not applicable	0.0015
BERT	768	12	110 M	2.9777
ChamBERTa	384	3	3 M	4.8544
MolFormer	768	12	44 M	20.9644
GPT	1536	96	175 B	0.2597
LLaMA	4095	32	7 B	50.8919
LLaMA2	4095	32	7 B	51.6308

*Speed of generating embedding. Speed is dependent on the machine

Table 1 compares how each model tokenizes SMILES strings. ChemBERTa, explicitly designed for molecular embeddings, tokenizes SMILES using the Byte-Pair Encoder (BPE) strategy. Meanwhile, MolFormer-XL employs a SMILES-specific regular expression method, as described by Schwaller et al. [35], using an atom-wise tokenization strategy with the regular expression pattern that is formatted as follows and is able to differentiate between atom characters and symbols for chemical bonds:

```
(\[[^\]]+]|Br?|Cl?|N|O|S|P|F|I|b|c|n|o|s|p|\(|\)|\.|=|#|\+|
\\\\\/|:|~|@|\?|>|\*|\$|\%[0-9]{2}|[0-9])
```

However, LLaMA, as a general-purpose model, employs a different tokenization approach. Its tokenizer is based on SentencePiece Byte-Pair Encoding (BPE). This tokenizer processes the input string character by character, searching for the largest known subword units it can match based on its training. Consequently, as it can be seen in Table 1, it treats 'CS' from the 'CCS(=O)(=O)CCBr' string as a single token, possibly interpreting it as an abbreviation in natural language. However, 'C' and 'S' should be considered as separate tokens, since each represents a distinct atom.

Table 2 compares molecular embedding in terms of the number of layers, parameters and their speed in generating a SMILES embedding. Compared with Morgan FP, language models are extremely slow. However, GPT performs the fastest among the language models, while LLaMA models are the slowest. There is also a relation between the number of layers and the speed of embedding generation. Although GPT remains an exception.

Experiments

Our study aims to generate molecular representation via LLMs and then evaluate the representation on various downstream tasks. To demonstrate the effectiveness of LLMs' molecular representations, we benchmarked their performance on numerous challenging classification and regression tasks from MoleculeNet [36] as well as link prediction from BioSnap [37] and DrugBank [38]. The objective of link prediction in this research is to map the drugs as nodes and their interactions as edges and identify whether there is a missing edge between two drug nodes.

Experimental setup

We experimented with seven models, each evaluated by six classifications, three regression and two link prediction tasks. To generate embeddings from LLaMAs, BERT, ChemBERTa, and MolFormer models, we first download and load the model weights using the Transformers library and then generate the embeddings. For LLaMA weights, we download the weights provided by Meta for LLaMAs and then convert them into PyTorch format. We extract embeddings from the last layer of the LLMs, following the practice in [39]. Pooling strategies can impact performance, and we explored a variety of combinations. The overall result remains the same. Hence, for the sake of simplicity, we use only the last layer. For GPT embeddings, we choose the recent model, *text-small-3-embeddings*.

To generate LLaMA and LLaMA2 embeddings, we employed four NVIDIA A2 GPUs to load the 7 billion parameter version of LLaMAs. In this configuration, the average speed of generating embeddings is one molecule per second. In our experiments, we generated embeddings for over 65,000 molecules.

Following MoleculeNet [36], for classification tasks, we partition the datasets into 5-stratified folds to ensure robust benchmarking. This approach ensures that each fold maintains the same proportion of observations for each target class as in the complete dataset. We employ a logistic regression model from scikit-learn, equipped with the following default parameters: L2 regularization, 'lbfgs' for optimization, and maximum 100 iterations allowed for the solvers to converge. The reported performance metrics are the mean and standard deviation of the F1-score and AUROC, calculated across the five folds.

For regression tasks, we implement five-fold cross-validation to assess model performance. We employ a Ridge regression model which is a linear regression model with l2 regularization. From scikit-learn with the following default parameters: tolerance of 0.001 for the optimization and a auto solver to automatically chooses the most appropriate solver method based on the data type. The metrics reported are the mean and standard deviation of the RMSE and the R^2, calculated across the five folds.

Following MIRACLE [40], a state-of-the-art method in DDI, for link prediction, we split all interaction samples from the DrugBank and BioSnap datasets into training and test sets using a 4:1 ratio. We further select 1/4 of the training dataset as a validation set. The reported results are the mean and standard deviation of AUROC and AUPR across 10 different runs of the GCN model. We set each parameter learning rate using an exponentially decaying schedule with an initial learning rate of 0.0002 and a multiplicative factor of 0.96. For the proposed model's hyperparameters, we set the dimension of

the hidden state of drugs as 256 and 3 layers for the GCN encoder. To further regularise the model, dropout with $p = 0.3$ is applied to every intermediate layer's output. We use Pytorch-geometric [41] for GCN. GCN Model is trained using Adam optimizer.

Benchmarking data sets

For classification and regression tasks, we use datasets from MoleculeNet [36], which is a collection of diverse datasets that cover a range of tasks, such as identifying properties like toxicity, bioactivity, and whether a molecule is an inhibitor. MoleculeNet is a widely used benchmark dataset in the field of computational chemistry and drug discovery and it is designed to evaluate and compare the performance of various machine learning models and algorithms on tasks related to molecular property prediction, compound screening, and other cheminformatics tasks [3–6, 18, 23, 42].

For the link prediction task, however, we utilize two DDI networks: BioSnap [37] and DrugBank [38]. These datasets represent interactions among FDA-approved drugs as a biological network, with drugs as nodes and interactions as edges.

We extracted the SMILES strings of drugs in the DrugBank database. It should be noted that we conduct data removal because of some improper drug SMILES strings in Drugbank, which can not be converted into molecular graphs, as determined by the RDKit library. The errors include so-old storage format of SMILES strings, wrong characters, etc. Through these curation efforts, we have fortified the quality and coherence of our DDI network, ensuring its suitability for comprehensive analysis and interpretation.

For the BioSnap dataset, 1320 drugs have SMILES strings, while the DrugBank dataset has 1690 drugs with SMILES strings. Hence, the number of edges for BioSnap and DrugBank reduced to 41,577 and 190,609, respectively.

Performance analysis

Results on classification tasks

Figure 2a, Table 3, and 4 present our experiments on classification tasks. Surprisingly, LLaMA embeddings achieve comparable performance to established pre-trained models such as MolFormer-XL [4] and ChemBERTa [6] across all datasets. Conversely, GPT embeddings underperform in every case. Intriguingly, Morgan FP representations nearly match the performance of other pre-trained methods but are more computationally efficient; generating Morgan FP for a large dataset takes less than a minute without the need for a GPU, whereas LLaMA requires GPUs and processes only 117 molecules per minute (Table 2). We also tested other classifiers, including SVM and Random Forest, with similar results. The small standard deviation in the evaluation scores indicates that these performance differences are statistically significant. Despite ChemBERTa and MolFormer-XL being pre-trained on millions of compounds from PubChem and ZINC, they perform comparably or, in some instances, less effectively than the BERT model. This showcases the importance of fine-tuning the results of pre-trained models.

Results on regression tasks

Figure 2a and Table 5 present the evaluation results for the regression tasks. Similar to the classification results, GPT underperforms relative to other models, and in some instances, it even falls short of Morgan Fingerprint's performance. ChemBERTa

Table 3 Results on classification tasks

Dataset	**BBBP**		**BACE**		**HIV**	
# Compounds	2039		1513		41127	
Negative:Positive	≈1:3		≈1:1		≈28:1	
Models	**F1-Score**	**AUROC**	**F1-Score**	**AUROC**	**F1-Score**	**AUROC**
Morgan FP	0.921 ± 0.003	0.896 ± 0.014	**0.778 ± 0.027**	**0.880 ± 0.020**	0.373 ± 0.028	0.797 ± 0.019
BERT	**0.935 ± 0.005**	0.947 ± 0.007	0.744 ± 0.023	0.845 ± 0.016	0.182 ± 0.032	0.780 ± 0.011
ChemBERTa	0.926 ± 0.011	0.944 ± 0.012	0.767 ± 0.020	0.862 ± 0.011	0.294 ± 0.033	0.767 ± 0.019
MolFormer-XL	0.927 ± 0.006	0.934 ± 0.007	0.762 ± 0.012	0.860 ± 0.010	0.317 ± 0.032	**0.804 ± 0.010**
GPT	0.908 ± 0.007	0.921 ± 0.015	0.648 ± 0.025	0.743 ± 0.030	0.039 ± 0.010	0.746 ± 0.009
LLaMA	0.933 ± 0.006	**0.953 ± 0.009**	0.766 ± 0.024	0.859 ± 0.017	**0.391 ± 0.013**	0.802 ± 0.010
LLaMA2	0.930 ± 0.006	0.945 ± 0.004	0.772 ± 0.023	0.863 ± 0.018	0.378 ± 0.017	0.799 ± 0.008

The reported performance metrics are the mean and standard deviation of the F1-score and AUROC, calculated across the five-folds. The Best Performance is Highlighted in Bold

Table 4 Results on multi-task classification tasks

Dataset	**ClinTox**		**SIDER**		**Tox21**	
# Compounds	1478		1427		7831	
# Tasks	2		27		12	
Models	**F1-Score**	**AUROC**	**F1-Score**	**AUROC**	**F1-Score**	**AUROC**
Morgan FP	0.647 ± 0.065	0.799 ± 0.063	**0.634 ± 0.008**	**0.629 ± 0.01**	0.314 ± 0.019	0.761 ± 0.010
BERT	0.919 ± 0.035	0.983 ± 0.017	0.617 ± 0.008	0.625 ± 0.014	0.192 ± 0.019	**0.786 ± 0.011**
ChemBERTa	0.896 ± 0.019	0.965 ± 0.01	0.628 ± 0.014	0.628 ± 0.012	0.236 ± 0.013	0.781 ± 0.008
MolFormer-XL	**0.929 ± 0.038**	**0.982 ± 0.013**	0.624 ± 0.012	0.605 ± 0.009	0.315 ± 0.008	0.775 ± 0.012
GPT	0.520 ± 0.035	0.963 ± 0.019	0.601 ± 0.005	0.612 ± 0.013	0.032 ± 0.008	0.757 ± 0.015
LLaMA	0.881 ± 0.053	0.980 ± 0.008	0.627 ± 0.007	0.605 ± 0.008	**0.339 ± 0.015**	0.774 ± 0.010
LLaMA2	0.905 ± 0.036	0.978 ± 0.014	0.627 ± 0.004	0.599 ± 0.009	0.332 ± 0.012	0.773 ± 0.009

The reported performance metrics are the mean and standard deviation of the F1-score and AUROC, calculated across the five-folds. The Best Performance is Highlighted in Bold

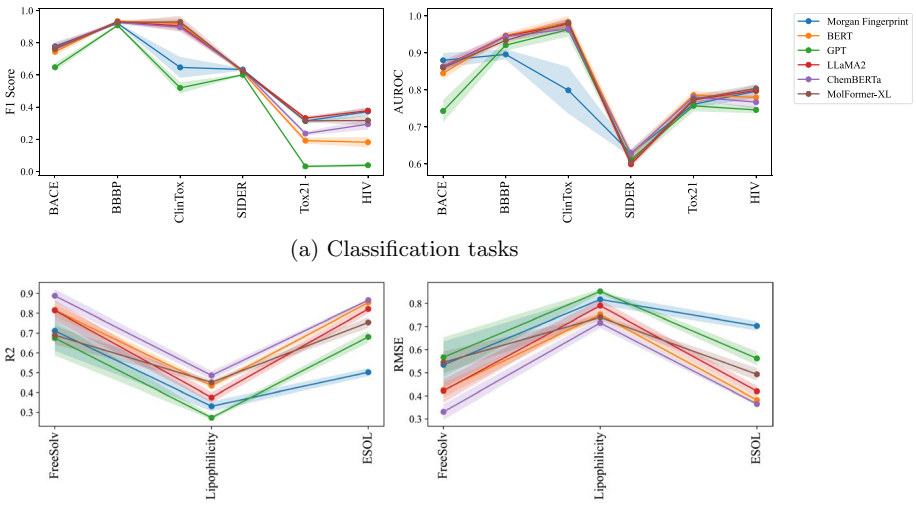


Fig. 2 Results on classification and regression tasks. Each line represent the mean value of five-Fold cross validation while the shaded area shows their standard deviation

consistently emerges as the top-performing model for regression across all tested datasets. BERT and LLaMA exhibit performances that are closely comparable to ChemBERTa in the regression tasks. Additionally, we observed a general decline in the performance of all methods when applied to larger datasets, such as Lipophilicity.

Results on link prediction tasks

Table 6 presents the results for the link prediction tasks on DDI networks. LLaMA consistently outperforms all other models across both datasets by a significant margin. Notably, Morgan FP surpasses the performance of embeddings from pre-trained models. It appears that the size of the embeddings impacts model performance, as larger embeddings generally yield better results. Nevertheless, despite having the same size, there are still noticeable performance differences between the LLaMA and LLaMA2 models.

Ablation study

LLaMA Vs LLaMA2 Figure 3 compares the LLaMA and LLaMA2 models. The performance of these two models is similar, mainly across various tasks. However, there are notable differences in specific instances. For example, in the link prediction tasks (Table 6), LLaMA2 outperforms LLaMA. This trend is also observed in classification and regression tasks, where LLaMA2 generally matches or exceeds the performance of LLaMA. Both models share similar architecture and training presets. Nevertheless, LLaMA2 has been trained on 40% more data and supports twice the context length of its predecessor, enhancing its capability to understand more complex language structures [28, 29].

Dimension reduction We investigated the impact of dimension reduction on LLMs with substantial embedding sizes, as illustrated in Fig. 4. Using Principal Component Analysis (PCA) for dimension reduction, we experimented with various reduction sizes. Our findings indicate that the impact of dimension reduction on the

Table 5 Results on regression tasks

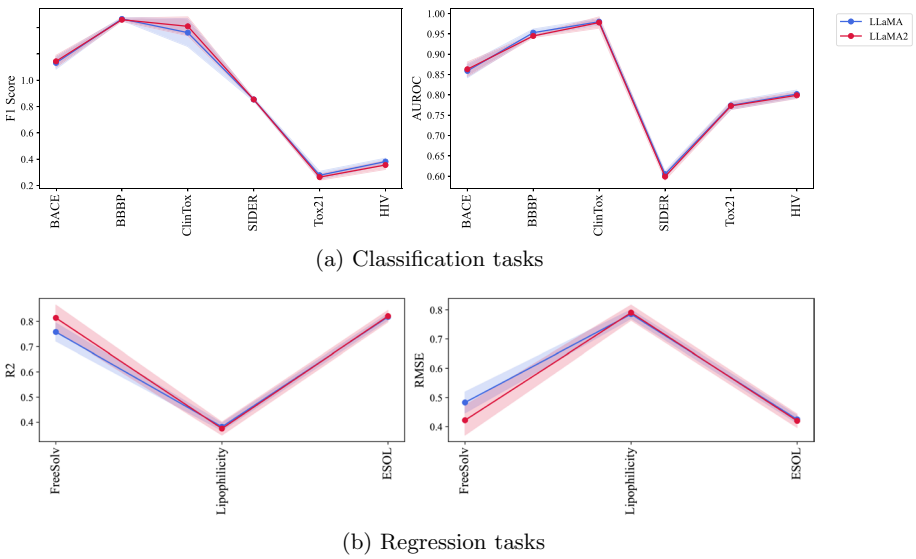
Dataset	**FreeSolv**		**Lipophilicity**		**ESOL**	
# Compounds	642		4200		1128	
Models	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**
Morgan FP	0.534 ± 0.101	0.712 ± 0.101	0.817 ± 0.025	0.331 ± 0.025	0.703 ± 0.020	0.502 ± 0.020
BERT	0.425 ± 0.031	0.816 ± 0.031	0.752 ± 0.013	0.434 ± 0.013	0.382 ± 0.015	0.854 ± 0.015
ChemBERTa	**0.331 ± 0.034**	**0.888 ± 0.034**	**0.716 ± 0.022**	**0.486 ± 0.022**	**0.365 ± 0.007**	**0.866 ± 0.007**
MolFormer-XL	0.545 ± 0.047	0.690 ± 0.047	0.740 ± 0.012	0.451 ± 0.012	0.493 ± 0.027	0.754 ± 0.027
GPT	0.567 ± 0.087	0.675 ± 0.087	0.852 ± 0.010	0.273 ± 0.010	0.562 ± 0.030	0.681 ± 0.030
LLaMA	0.483 ± 0.036	0.758 ± 0.036	0.785 ± 0.015	0.382 ± 0.015	0.425 ± 0.013	0.818 ± 0.013
LLaMA2	0.422 ± 0.051	0.814 ± 0.051	0.790 ± 0.026	0.375 ± 0.026	0.420 ± 0.023	0.821 ± 0.023

The reported performance metrics are the mean and standard deviation of the RMSE and R^2, calculated across the five-folds. The Best Performance is Highlighted in Bold

Table 6 Results on link prediction tasks

Dataset	**BioSnap**		**DrugBank**	
# Nodes	1320		1690	
# Edges	41577		190609	
Average node degree	64.087		224.38	
Models	**AUROC**	**AUPR**	**AUROC**	**AUPR**
Morgan FP	0.871 ± 0.00	0.847 ± 0.00	0.876 ± 0.00	0.855 ± 0.00
BERT	0.621 ± 0.02	0.563 ± 0.08	0.660 ± 0.02	0.639 ± 0.01
ChemBERTa	0.527 ± 0.02	0.547 ± 0.08	0.519 ± 0.02	0.457 ± 0.01
MolFormer-XL	0.550 ± 0.02	0.701 ± 0.08	0.611 ± 0.02	0.644 ± 0.01
GPT	0.856 ± 0.06	0.812 ± 0.08	0.836 ± 0.05	0.748 ± 0.09
LLaMA	0.921 ± 0.00	0.884 ± 0.02	0.927 ± 0.00	0.872 ± 0.01
LLaMA2	**0.941 ± 0.00**	**0.902 ± 0.02**	**0.961 ± 0.00**	**0.933 ± 0.01**

The reported performance metrics are the mean and standard deviation of the AUROC and AUPR, calculated across the 10 runs. The Best Performance is Highlighted in Bold



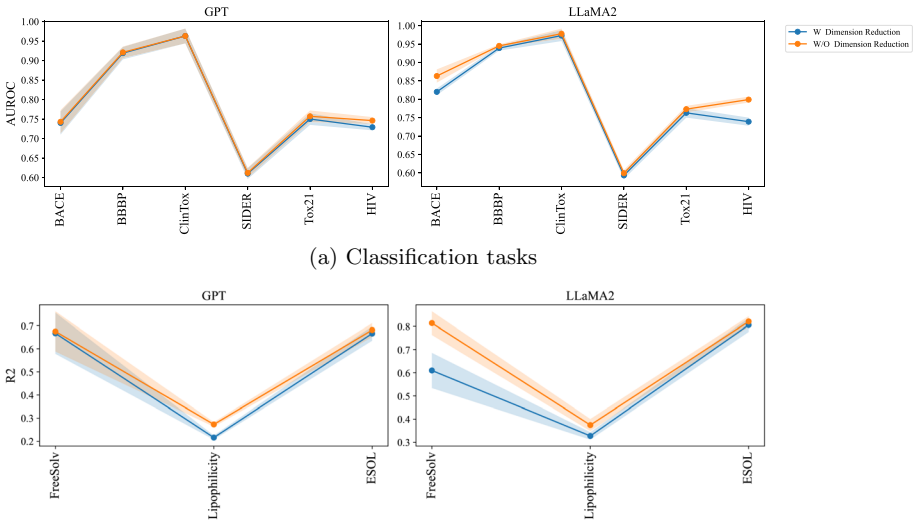
(a) Classification tasks

(b) Regression tasks

Fig. 3 Comparison of LLaMA and LLaMA2 performance

classification performance of GPT and LLaMA models is minimal, although there is a noticeable decrease in performance post-reduction. In contrast, for regression tasks, dimension reduction significantly lowers the performance of the models. This suggests a correlation between the size of the embeddings in LLMs and their effectiveness in handling regression tasks.

LLM and anistropy It is well-documented that LLM embeddings suffer from the isotropy problem, meaning they are not uniformly distributed in terms of direction [43–45]. Instead, these embeddings occupy a narrow cone in the vector space, making them anisotropic. The anisotropy problem in LLM model embeddings is evident from



(a) Classification tasks

(b) Regression tasks

Fig. 4 Effect of dimension reduction on the performance of LLMs

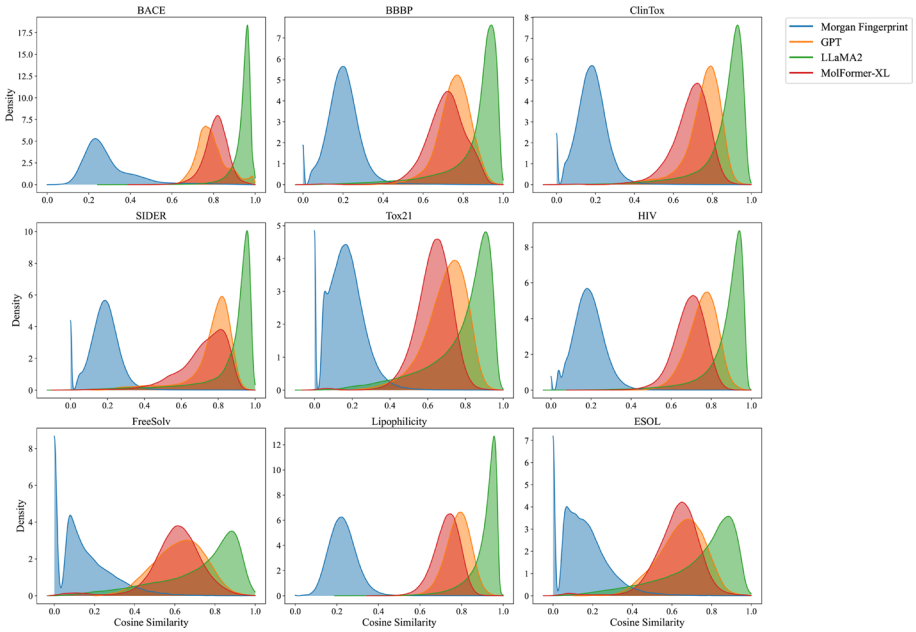


Fig. 5 Cosine similarity distribution between SMILES embeddings

the narrow shape of the cosine similarity distribution and the higher average cosine similarity values.

Our comparative analysis also reveals that LLMs embeddings demonstrate a higher degree of anisotropy than pre-trained embeddings and Morgan FP (Fig. 5). This is evident since the distribution of cosine similarity of embeddings is more closely grouped together in their representation (Fig. 5). However, our experiments indicate that better isotropy does not imply a performance gain in machine-learning tasks. As can be

seen, the cosine similarity distribution of LLaMA2 embeddings is a lot narrower than GPT and Morgan FP; however, LLaMA2 outperforms both models in most cases.

As illustrated in Fig. 6, we also noticed that the PCA representation of GPT's embeddings is predominantly concentrated within a range smaller than 1. This observation also suggests a high likelihood that the GPT embeddings have been pre-normalized.

GPT Vs LLaMA Figure 7 demonstrates that LLaMA consistently outperforms GPT across all datasets by a significant margin. This raises the question of whether these differences are due to the architectural design or the specific training of the models. As outlined in the GPT-4 technical report, GPT models are capable of interpreting SMILES strings. Notably, approximately 2.5% of the LLaMA training dataset, as reported in [28, 33], consists of scientific material primarily sourced from arXiv, including bioinformatics papers.

Both LLaMA and GPT models utilize a transformer-based architecture with a heavy reliance on self-attention mechanisms and a decoder-only configuration. However, the opaque nature of GPT as a "black box" model complicates direct comparisons with LLaMA regarding whether their efficiency stems solely from architecture or pre-training specifics. Nonetheless, considering their training on SMILES strings, the data from Fig. 7 and Table 6 suggest that the LLaMA architecture is particularly adept at handling complex language structures like SMILES strings. Furthermore, Table 1 reveals that while the LLaMA2 tokenizer may not perform as well as the MolFormer tokenizer, it tokenizes SMILES strings more effectively than BERT. Unfortunately, we cannot compare the GPT tokenizer directly with other models due to limitations in OpenAI's API access.

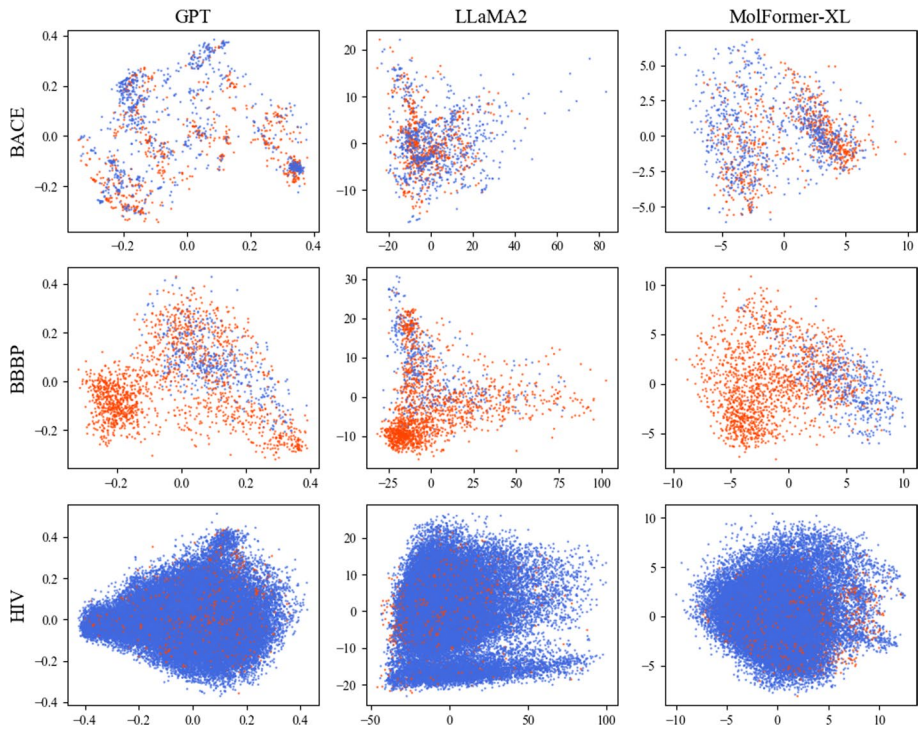


Fig. 6 PCA representation embedding for classification task. Red represent positive samples while blue represent negative samples

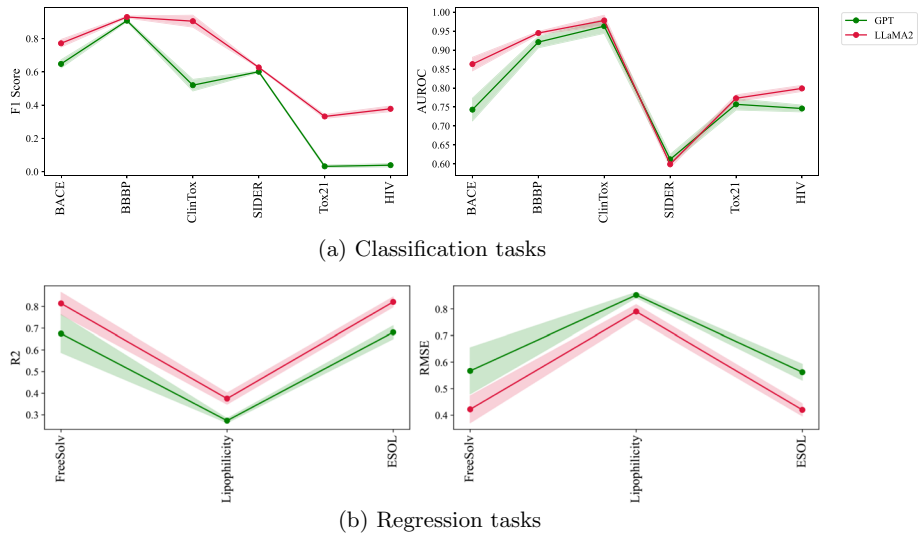


Fig. 7 Comparison of LLaMA2 and GPT

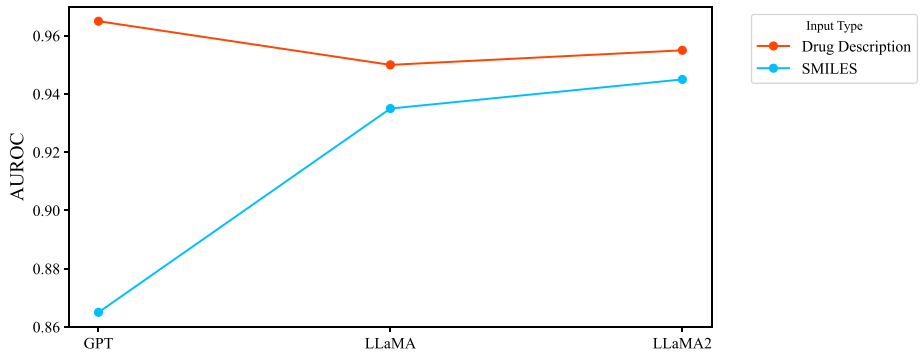


Fig. 8 Impact of drug description for DDI prediction on BioSnap dataset

Link prediction with SMILES VS drug description We also extracted the text-format drug description information of drugs from the DrugBank database. Drug description embedding in DDI prediction significantly outperforms using SMILES strings when leveraging LLMs. This enhancement is consistent with applying LLMs pre-trained on general text data, as depicted in Fig. 8. When applied to drug descriptions closer to natural language, GPT outperforms the LLaMA models on both datasets and uses both AUROC and AUPRC metrics.

Conclusions

In summary, this research underscores the potential of LLMs like GPT and LLaMA for molecular embedding. We specifically recommend LLaMA models over GPT due to their superior performance in generating molecular embeddings from SMILES strings, which is notable in our studies. These findings suggest that LLaMA could be particularly effective in predicting molecular properties and drug interactions. Although models like LLaMA and GPT are not explicitly designed for SMILES string embedding-unlike specialized models such as ChemBERTa and MolFormer-XL-they still demonstrate

competitive performance. Our work lays the groundwork for future improvements in utilizing LLMs for molecular embedding. Future efforts will focus on enhancing the quality of molecular embeddings derived from LLMs inspired by natural language sentence embedding techniques, such as fine-tuning and modifications to LLaMA tokenization.

Acknowledgements
The authors thank the anonymous reviewers for their valuable suggestions.

Author contributions
SS, JL and AN conceived the presented idea. SS obtained the embeddings, perform evaluation and wrote the main manuscript text and prepared figures. AB helped with the evaluation code. AF obtained the LLaMA embeddings. JL and AN supervised the work and helped with the main manuscript. All authors discussed the results and contributed to the final manuscript.

Funding
This research is supported by the National Science and Engineering Research Council of Canada (NSERC) (NSERC RGPIN-2016-05017 and NSERC RGPIN-2019-05350).

Availability of data and materials
Datasets for the validation of our work were obtained from the original studies and processed into a format suitable for analysis. Processed data is available for download from our GitHub repository.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable

Competing interests
The authors declare that they have no competing interests.

Received: 31 December 2023 Accepted: 18 June 2024
Published online: 26 June 2024

References

1. Li P, Wang J, Qiao Y, Chen H, Yu Y, Yao X, et al. An effective self-supervised framework for learning expressive molecular global representations to drug discovery. Br Bioinform. 2021;22(6):bbab109.
2. Lv Q, Chen G, Zhao L, Zhong W, Yu-Chian CC. Mol2Context-vec: learning molecular representation from context awareness for drug discovery. Br Bioinform. 2021;22(6):bbab317.
3. Liu Y, Zhang R, Li T, Jiang J, Ma J, Wang P. MolRoPE-BERT: An enhanced molecular representation with Rotary Position Embedding for molecular property prediction. J Mol Graph Model. 2023;118: 108344.
4. Ross J, Belgodere B, Chenthamarakshan V, Padhi I, Mroueh Y, Das P. Large-scale chemical language representations capture molecular structure and properties. Nat Mach Intell. 2022;4(12):1256–64.
5. Zhang XC, Wu CK, Yang ZJ, Wu ZX, Yi JC, Hsieh CY, et al. MG-BERT: leveraging unsupervised atomic representation learning for molecular property prediction. Br Bioinform. 2021;22(6):bbab152.
6. Chithrananda S, Grand G, Ramsundar B. ChemBERTa: large-scale self-supervised pretraining for molecular property prediction. Preprint at arXiv:2010.09885. 2020; p. 1–7.
7. Zhou D, Xu Z, Li W, Xie X, Peng S. MultiDTI: drug-target interaction prediction based on multi-modal representation learning to bridge the gap between new chemical entities and known heterogeneous network. Bioinformatics. 2021;37(23):4485–92.
8. Thafar MA, Alshahrani M, Albaradei S, Gojobori T, Essack M, Gao X. Affinity2Vec: drug-target binding affinity prediction through representation learning, graph mining, and machine learning. Sci Rep. 2022;12(1):1–18.
9. Jin Y, Lu J, Shi R, Yang Y. EmbedDTI: enhancing the molecular representations via sequence embedding and graph convolutional network for the prediction of drug-target interaction. Biomolecules. 2021;11(12):1783.
10. Purkayastha S, Mondal I, Sarkar S, Goyal P, Pillai JK. Drug-drug interactions prediction based on drug embedding and graph auto-encoder. In: 2019 IEEE 19th international conference on bioinformatics and bioengineering (BIBE). IEEE; 2019. pp. 547–552.
11. Han X, Xie R, Li X, Li J. SmileGNN: drug-drug interaction prediction based on the smiles and graph neural network. Life. 2022;12(2):319.
12. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: North American Chapter of the association for computational linguistics; 2019. pp. 4171–4186. Available from https://api.semanticscholar.org/CorpusID:52967399.
13. Vaswani A, Shazeer N, Parmar N. Attention is all you need. Adv Neural Inf Process Syst. 2017;30.
14. Jaeger S, Fulle S, Turk S. Mol2vec: unsupervised machine learning approach with chemical intuition. J Chem Inf Model. 2018;58(1):27–35.

15. 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; 2019. pp. 429–436.
16. Fabian B, Edlich T, Gaspar H, Segler M, Meyers J, Fiscato M, et al. Molecular representation learning with language models and domain-relevant auxiliary tasks. Mach Learn Mol Workshop NeurIPS 2020;2020.
17. Koge D, Ono N, Huang M, Altaf-Ul-Amin M, Kanaya S. Embedding of molecular structure using molecular hypergraph variational autoencoder with metric learning. Mol Inf. 2021;40(2):2000203.
18. Guo T, Nan B, Liang Z, Guo Z, Chawla N, Wiest O, et al. What can large language models do in chemistry? A comprehensive benchmark on eight tasks. Adv Neural Inf Process Syst. 2023;36:59662–88.
19. Goh GB, Hodas NO, Siegel C, Vishnu A. SMILES2Vec: an interpretable general-purpose deep neural network for predicting chemical properties. Preprint at arXiv: 1712.02034.
20. Morgan HL. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. J Chem Doc. 1965;5(2):107–13.
21. Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, et al. Convolutional networks on graphs for learning molecular fingerprints. Adv Neural Inf Process Syst. 2015;28.
22. Wang Y, Wang J, Cao Z, Barati FA. Molecular contrastive learning of representations via graph neural networks. Nat Mach Intell. 2022;4(3):279–87.
23. Zang X, Zhao X, Tang B. Hierarchical molecular graph self-supervised learning for property prediction. Commun Chem. 2023;6(1):34.
24. Xu Z, Wang S, Zhu F, Huang J. Seq2seq fingerprint: an unsupervised deep molecular embedding for drug discovery. In: Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics; 2017. pp. 285–294.
25. Zhang YF, Wang X, Kaushik AC, Chu Y, Shan X, Zhao MZ, et al. SPVec: a Word2vec-inspired feature representation method for drug-target interaction prediction. Front Chem. 2020;7:895.
26. Su J, Lu Y, Pan S, Murtadha A, Wen B, Liu Y. Roformer: enhanced transformer with rotary position embedding. Neurocomputing. 2024;568: 127063. https://doi.org/10.1016/j.neucom.2023.127063.
27. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. OpenAI Blog. 2019;1–24.
28. Touvron H, Lavril T, Izacard. LLaMA: open and efficient foundation language models. Preprint at arXiv:2302.13971; 2023.
29. Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, et al. Llama 2: open foundation and fine-tuned chat models. Preprint at arXiv:2307.09288; 2023.
30. Hassani H, Silva ES. The role of ChatGPT in data science: how ai-assisted conversational interfaces are revolutionizing the field. Big Data Cogn Comput. 2023;7(2):62.
31. OpenAI. OpenAI, editor.: ChatGPT (Large language model). OpenAI. https://platform.openai.com/docs.
32. Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, et al. Palm: scaling language modeling with pathways. J Mach Learn Res. 2023;24(240):1–113.
33. Zhao WX, Zhou K, Li J, Tang T, Wang X, Hou Y, et al. A survey of large language models. Preprint at arXiv:2303.18223; 2023.
34. Brown T, Mann B, Ryder N, Subbiah. Language models are few-shot learners. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. Advances in neural information processing systems. Curran Associates, Inc.; 2020. p. 1877–901.
35. Schwaller P, Laino T, Gaudin T, Bolgar P, Hunter CA, Bekas C, et al. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. ACS Cent Sci. 2019;5(9):1572–83.
36. Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, et al. MoleculeNet: a benchmark for molecular machine learning. Chem Sci. 2018;9(2):513–30.
37. Zitnik M, Sosivc R, Maheshwari S, Leskovec J. University S, editor.: BioSNAP datasets: stanford biomedical network dataset collection. ACM. http://snap.stanford.edu/biodata.
38. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, et al. DrugBank 5.0: a major update to the DrugBank database for 2018. Nucleic Acids Res. 2018;46(D1):D1074–82.
39. Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Conference on empirical methods in natural language processing; 2019. pp. 3982–3992. Available from: https://api.semanticscholar.org/CorpusID:201646309.
40. Wang Y, Min Y, Chen X, Wu J. Multi-view graph contrastive representation learning for drug-drug interaction prediction. In: Proceedings of the web conference. vol. 2021, 2021. pp. 2921–33.
41. Fey M, Lenssen JE. Fast graph representation learning with PyTorch geometric. Representation learning on graphs and manifolds at ICLR 2019 Workshop. 2019.
42. Li J, Jiang X. Mol-BERT: an effective molecular representation with BERT for molecular property prediction. Wirel Commun Mob Comput. 2021;2021.
43. Timkey W, van Schijndel M. All bark and no bite: rogue dimensions in transformer language models obscure representational quality. In: Proceedings of the 2021 conference on empirical methods in natural language processing. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics; 2021. p. 4527–4546. Available from: https://aclanthology.org/2021.emnlp-main.372.
44. Kovaleva O, Kulshreshtha S, Rogers A, Rumshisky A. BERT busters: outlier dimensions that disrupt transformers. In: Findings; 2021. pp. 3392–3405. Available from: https://api.semanticscholar.org/CorpusID:235313996.
45. Rudman W, Gillman N, Rayne T, Eickhoff C. IsoScore: measuring the uniformity of embedding space utilization. In: Findings of the association for computational linguistics: ACL 2022. Dublin: Association for Computational Linguistics; 2022. pp. 3325–3339. Available from https://aclanthology.org/2022.findings-acl.262.

Publisher's Note