MolEval: An Evaluation Toolkit for Molecular Embeddings via LLMs

Anonymous Authors[1]

Abstract

Inspired by SentEval and MTEB for sentence embeddings and DeepChem for molecular machine learning, we introduce MolEval [1]. MolEval tackles the issue of evaluating large language models (LLMs) embeddings, which are traditionally expensive to execute on standard computing hardware. It achieves this by offering a repository of pre-computed molecule embeddings alongside a versatile platform that facilitates the evaluation of any embeddings derived from molecular structures. This approach not only streamlines the assessment process but also makes it more accessible to researchers and practitioners in the field.

1. Introduction

Recent developments in large language models highlight the need for a platform to evaluate their effectiveness in molecular embedding tasks. General-purpose language models may not be optimized for molecular embedding, but benchmarking their performance against models pre-trained on molecular data can offer a comprehensive overview of the top molecule embedding models across various tasks. This approach also helps identify language models with the greatest potential for improvement in this area.

Libraries like DeepChem (Ramsundar et al., 2019) offer the capability to load MoleculeNet (Wu et al., 2018) benchmark data through integrated library functions and include implementations of various chemical featurization methods, as well as machine learning techniques. However, DeepChem does not provide the embedding methods required for the models examined in our study. Hence, inspired from SentEval (con, 2018) and MTEB (Massive Text Embedding Benchmark) (Muennighoff et al., 2023) that benchmark sentence embedding methods, we create MolEval a toolkit to benchmark molecule embeddings.

With the advancements in large language models (LLMs), a pressing question arises: Can LLMs comprehend molecular structures and derive meaningful insights from molecular data? Specifically, can they generate high-quality semantic representations? Recent study (Guo et al., 2023) has conducted initial investigations by assessing LLMs' performance in addressing SMILES-related inquiries. Building upon this groundwork, our study delves deeper into examining the capacity of these models to efficiently encode SMILES strings.

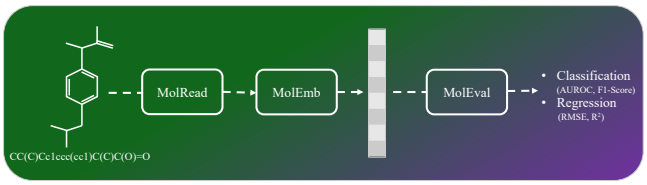


Figure 1. The **MolEval** Framework consists of three primary components: *MolRead*, *MolEmb*, and *MolEval*.

In our investigation, we compare LLMs against pre-trained models, including those trained on vast unlabeled SMILES datasets like Mol2Vec (Jaeger et al., 2018), as well as transformer-based models fine-tuned for specific tasks such as classification tasks, exemplified by MolFormer (Ross et al., 2022), ChemBERTa (Chithrananda et al., 2020), and Roberta-ZINC (Heyer, 2023). These models, having undergone pre-training on millions of molecular structures, require substantial computational resources. For instance, MolFormer necessitates up to 16 V100 graphics processing units (GPUs) (Ross et al., 2022). Consequently, employing pre-trained LLMs like BERT (Devlin et al., 2019), GPT (Radford et al., 2019), LLaMA (Touvron et al., 2023a;b) for generating embeddings becomes computationally more viable. These LLMs have already been trained on extensive datasets, rendering them readily deployable for processing SMILES strings and deriving molecular embeddings without the need for extensive hardware.

MolEval aims to shed light on model performance across various embedding tasks, positioning itself as a key resource for discovering universal molecular embeddings that are applicable to multiple tasks. The platform features nine datasets spanning 12 languages and covers three embedding tasks: classification, multi-task classification, and regression. Available as an open-source package, MolEval allows for the evaluation of any embedding model with fewer than ten lines of code. We hope our work simplifies the selection of the right embedding model and facilitates future research

[1] Here is the link to the anonymous Github for MolEval

in embeddings.

2. Framework Overview

As illustrated in Figure 1, the MolEval framework comprises three main steps: *MolRead*, *MolEmb*, and *MolEval*.

2.1. MolRead

We utilize datasets sourced from MoleculeNet (Wu et al., 2018), a comprehensive collection comprising diverse datasets spanning various tasks, including the identification of properties such as toxicity, bioactivity, and the determination of whether a molecule serves as an inhibitor. MoleculeNet stands as a widely recognized benchmark dataset within the field of computational chemistry and drug discovery. It serves the critical purpose of assessing and contrasting the efficacy of diverse machine learning models and algorithms across tasks concerning molecular property prediction, compound screening, and other cheminformatics endeavors (Chithrananda et al., 2020; Li & Jiang, 2021; Zhang et al., 2021; Ross et al., 2022; Liu et al., 2023; Zang et al., 2023; Guo et al., 2023).

The input for MolRead consists of dataset names summarized in Table 1, while the output includes SMILES strings and their corresponding labels—both for classification and regression—separated for subsequent processing steps.

2.2. MolEmb

In our MolEmb Step, we incorporated 13 representative models. The input for this step consists of the SMILES strings from each dataset and the name of the embedding method. The output is the vector representation of these inputs.

To extract embeddings for transformer-based models, we initiated the process by downloading and loading the model weights through the Transformers library. Subsequently, we generated the embeddings. For LLaMA weights, we acquired the provided weights from Meta for LLaMAs and then converted them into PyTorch format. The embeddings were extracted from the last layer of the LLMs, aligning with established practice (Reimers & Gurevych, 2019). Additionally, for GPT embeddings, we opted for the latest model from OpenAI, named *text-small-3-embeddings*.

The generation of LLaMA and LLaMA2 embeddings necessitated the utilization of four NVIDIA A2 GPUs to handle the 7 billion parameter version of LLaMAs. Operating under this configuration, the average speed for generating embeddings amounted to one molecule per second. In total, we generated embeddings for over 65,000 molecules in our experiments.

In this research, we evaluate several representation models, categorizing them into two main sections:

- **Special-purpose models:** These models are developed specifically for molecule representations, either pre-trained on SMILES using transformers (Heyer, 2023; Chithrananda et al., 2020; Ross et al., 2022), or without transformers but learnt from SMILES strings using Skip gram negative sampling (Jaeger et al., 2018), or without machine learning (Rogers & Hahn, 2010).
- **General-purpose models:** These models are pre-trained on general purpose text data (Devlin et al., 2019; Liu et al., 2020). Since the data is large, they may contain SMILES data. LLaMA (Touvron et al., 2023a;b) and GPT (OpenAI, 2023) mentioned specifically that SMILES data are collected and fed the trainer. Some are further fine-tuned on NLI data (SBert (Reimers & Gurevych, 2019), SimCSE (Gao et al., 2021), AnglEBERT (Reimers & Gurevych, 2019))

2.3. MolEval

In our MolEval step, we obtain the embeddings for each dataset from MolEmb, along with the labels from MolRead. MolEval then performs evaluation tasks for both classification and regression.

In adherence to the methodology outlined in MoleculeNet (Wu et al., 2018), for classification tasks, we adopted a stratified partitioning approach, dividing the datasets into 5 stratified folds to ensure robust benchmarking. This strategy guarantees that each fold maintains consistent proportions of observations for each target class as observed in the complete dataset. We employed a logistic regression model from scikit-learn, configured with default parameters including L2 regularization, 'lbfgs' for optimization, and a maximum of 100 iterations allowed for the solvers to converge. Performance metrics reported include the mean and standard deviation of F1-score and AUROC, calculated across the five folds.

For regression tasks, we used a 5-fold cross-validation to evaluate model performance. Utilizing a Ridge regression model from scikit-learn, with default parameters were employed including a tolerance of 0.001 for optimization and an auto solver to select the most suitable solver method based on the data type. Reported metrics encompass the mean and standard deviation of RMSE and R^2, computed across the five folds.

3. Experiments

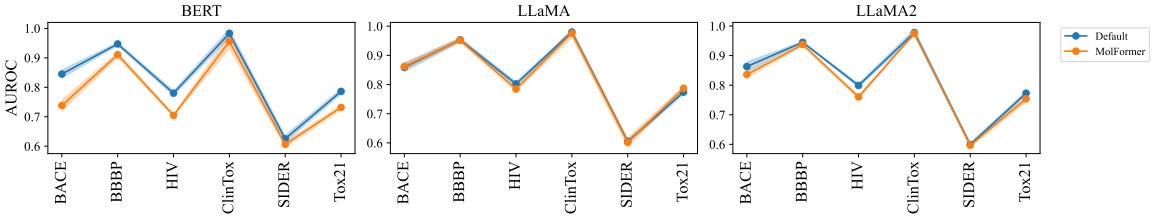
We experimented with 13 representative models, each evaluated by using 9 datasets as described in Table 1 and Table 3. Following are the observations we could make based on the results:

Table 1. The comparison of representative models on classification tasks. The Reported Performance Metrics Are the Mean and Standard Deviation of their Metrics, Calculated Across the 5-folds. The Best Performance is Highlighted With Green. The Line Between the Models Separate the General Purpose Models from Models Specialized on Molecules. The Best Performance in Each Section is Highlighted in **Bold**.

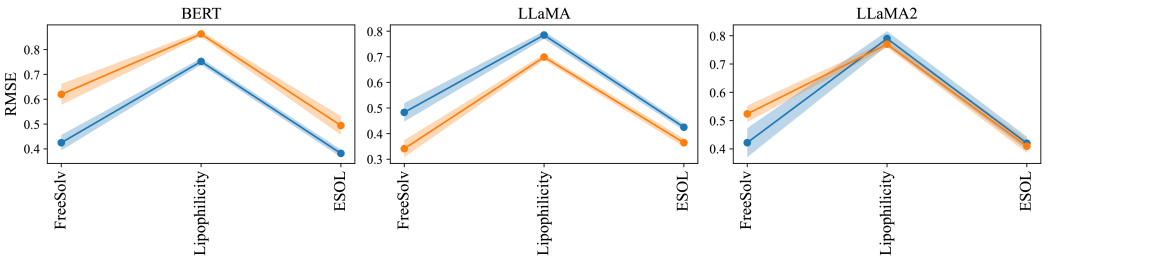
Dataset		BBBP	BACE	HIV	ClinTox	SIDER	Tox21
# Compound		2039	1513	41127	1478	1427	7831
# Tasks		1	1	1	2	27	12
Models	Dim. Size	AUROC					
MorganFP (radius=2)	1024	0.896 ± 0.014	**0.880 ± 0.020**	0.797 ± 0.019	0.799 ± 0.063	**0.629 ± 0.01**	0.761 ± 0.010
Mol2Vec	300	0.863 ± 0.020	0.858 ± 0.014	0.776 ± 0.021	0.842 ± 0.036	0.625 ± 0.006	0.768 ± 0.011
ChemBERTa	384	0.944 ± 0.012	0.862 ± 0.011	0.767 ± 0.019	0.965 ± 0.010	0.628 ± 0.012	0.781 ± 0.008
Roberta-ZINC	768	**0.944 ± 0.010**	0.871 ± 0.018	0.792 ± 0.013	0.980 ± 0.011	0.615 ± 0.011	**0.786 ± 0.006**
MolFormer	768	0.934 ± 0.007	0.860 ± 0.010	**0.804 ± 0.010**	**0.982 ± 0.013**	0.605 ± 0.009	0.775 ± 0.012
BERT	768	0.947 ± 0.007	0.845 ± 0.016	0.780 ± 0.011	0.983 ± 0.017	**0.625 ± 0.014**	**0.786 ± 0.011**
RoBERTa	768	0.939 ± 0.008	0.837 ± 0.015	0.769 ± 0.016	0.837 ± 0.015	0.621 ± 0.005	0.773 ± 0.009
SBERT	384	0.912 ± 0.009	0.726 ± 0.033	0.716 ± 0.008	0.958 ± 0.016	0.606 ± 0.012	0.739 ± 0.014
SimSCE	768	0.937 ± 0.006	0.845 ± 0.020	0.770 ± 0.007	0.975 ± 0.015	0.618 ± 0.007	0.776 ± 0.011
AngleBERT	768	0.938 ± 0.008	0.845 ± 0.020	0.773 ± 0.015	0.973 ± 0.019	0.622 ± 0.004	0.777 ± 0.011
GPT	1536	0.921 ± 0.015	0.743 ± 0.030	0.746 ± 0.009	0.963 ± 0.019	0.612 ± 0.013	0.757 ± 0.015
LLaMA	4096	**0.953 ± 0.009**	0.859 ± 0.017	**0.802 ± 0.010**	**0.980 ± 0.008**	0.605 ± 0.008	0.774 ± 0.010
LLaMA2	4096	0.945 ± 0.004	**0.863 ± 0.018**	0.799 ± 0.008	0.978 ± 0.014	0.599 ± 0.009	0.773 ± 0.009

Table 2. Comparison of Tokenizers for Molecular SMILES String. MolFormer Perform the Best Since it Tokenizes SMILES Strings Atom-wise.

Model	Tokenization Strategy	Example Tokenization of 'CCS(=O)(=O)CCBr'
ChemBERTa Tokenizer	Byte-Pair Encoding-based	['C', 'C', 'S', '(', '=', 'O', ')', '(', '=', 'O', ')', 'C', 'C', 'B', 'r']
MolFormer Tokenizer	SMILE Regex	**['C', 'C', 'S', '(', '=', 'O', ')', '(', '=', 'O', ')', 'C', 'C', 'Br']**
Roberta-ZINC Tokenizer	Byte-Pair Encoding-based	['CCS', '(=', 'O', ')(=', 'O', ')', 'CCBr']
BERT Tokenizer	Subword-based tokenization	['CC', '##S', '(', '=', 'O', ')', '(', '=', 'O', ')', 'CC', '##B', '##r']
SimCSE	Subword-based tokenization	['cc', '##s', '(', '=', 'o', ')', '(', '=', 'o', ')', 'cc', '##br']
AngleBERT	Subword-based tokenization	['cc', '##s', '(', '=', 'o', ')', '(', '=', 'o', ')', 'cc', '##br']
SBERT	Subword-based tokenization	['cc', '##s', '(', '=', 'o', ')', '(', '=', 'o', ')', 'cc', '##br']
GPT Tokenizer	cl100k-base	['CC', 'S', '(', '=', 'O', ')(', '=', 'O', ')', 'CC', 'Br']
LLaMA Tokenizer	SentencePiece Byte-Pair Encoding-based	['_C', , 'CS', '(', '=', 'O', ')(', '=', 'O', ')', 'CC', 'Br']
LLaMA2 Tokenizer	SentencePiece Byte-Pair Encoding-based	['_C', , 'CS', '(', '=', 'O', ')(', '=', 'O', ')', 'CC', 'Br']



a) Classification Tasks



b) Regression Tasks

Figure 2. Impact of Tokenizer on Classification and Regression Tasks

1. Transformer-based models typically outperform traditional models. As shown in Table 1, traditional models (Morgan FP and Mol2Vec) rank at the top in only 3 out of 9 cases. Meanwhile, transformer-based models outperform these traditional models in the remaining instances.

2. Performance varies depending on the task and, at times, the specific dataset involved. This variety indicate

Table 3. The comparison of representative models on regression tasks. The The Reported Performance Metrics Are the Mean and Standard Deviation of their Metrics, Calculated Across the 5-folds. The Best Performance is Highlighted With Green. The Line Between the Models Separate the General Purpose Models from Models Specialized on Molecules. The Best Performance in Each Section is Highlighted in **Bold**. Since the Metric is RMSE, a Lower Value is Better.

Dataset		FreeSolv	ESOL	Lipophilicity
# Compound		642	1128	4200
Models	Dim. Size	RMSE		
MorganFP (radius=2)	1024	0.534 ± 0.101	0.703 ± 0.020	0.817 ± 0.025
Mol2Vec	300	0.537 ± 0.274	0.495 ± 0.042	**0.678 ± 0.024**
ChemBERTa	384	**0.331 ± 0.034**	**0.365 ± 0.007**	0.716 ± 0.022
Roberta-ZINC	768	0.447 ± 0.046	0.514 ± 0.047	0.761 ± 0.008
MolFormer	768	0.545 ± 0.047	0.493 ± 0.027	0.740 ± 0.012
BERT	768	0.425 ± 0.031	**0.382 ± 0.015**	**0.752 ± 0.013**
RoBERTa	768	0.472 ± 0.039	0.405 ± 0.018	0.788 ± 0.013
SBERT	384	0.537 ± 0.070	0.517 ± 0.032	0.872 ± 0.014
SimCSE	768	**0.401 ± 0.042**	0.394 ± 0.022	0.773 ± 0.007
AngleBERT	768	0.407 ± 0.044	0.409 ± 0.013	0.774 ± 0.005
GPT	1536	0.567 ± 0.087	0.562 ± 0.030	0.852 ± 0.010
LLaMA	4096	0.483 ± 0.036	0.425 ± 0.013	0.785 ± 0.015
LLaMA2	4096	0.422 ± 0.051	0.420 ± 0.023	0.790 ± 0.026

that no single model consistently outperforms others across all settings. Different models may excel in certain contexts due to variations in dataset complexity and data characteristics. Therefore, evaluating a range of models is essential to determine the most suitable one for each unique scenario.

3. Fine-tuned LLM models on sentences often perform worse than vanilla models for SMILES string embeddings.

As illustrated in Table 1, models such as AngleBERT, SimCSE, and SBERT, which are fine-tuned on the Natural Language Inference (NLI) task, do not perform as well as vanilla models such as BERT and LLaMA in SMILES embedding tasks. This suggests that the underperformance of GPT may also be attributable to its fine-tuning on the NLI task.

4. When special-purpose models outperform general-purpose ones in classification tasks, the performance difference is often negligible.

Table 1 clearly demonstrates that the performance difference between the best models in the general-purpose and specialized categories is often very small, indicating that both model types are highly competitive across various tasks. For instance, when tested on the HIV dataset, the Molformer—a special-purpose model—scores 0.802, while the LLaMA—a general-purpose model—marginally outperforms it with a score of 0.804. This narrow gap in performance implies that the choice between special-purpose and general-purpose models may rely more on the specific requirements of use cases and dataset attributes, rather than on a definitive superiority of one type over the other. Additionally, although pre-training a model like MolFormer requires up to 16 V100 GPUs, using an already pre-trained model like LLaMA can be computationally cheaper and more efficient.

5. Change in the Tokenization method does not effect positively on the performance of LLMs. As shown in Table 2, each method employs a distinct tokenizer. To explore the significance of tokenizer variation, we conducted an evaluation assessing the impact on LLaMA and BERT models. This analysis aimed to understand how tokenizer changes affect the performance of LLMs. Specifically, we selected the MolFormer tokenizer that is based on study by Schwaller et al. (Schwaller et al., 2018), which tokenized SMILES strings atom-wise using the following regular expression:

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

Our study examines the effects of using the MolFormer tokenizer with BERT and LLaMA models. The results in Figure 2.a. show that while MolFormer accurately tokenizes SMILES strings atom-wise, it often decreases performance in classification tasks. However, as Figure 2.b. demonstrates, LLaMA models benefit in regression tasks with the MolFormer tokenizer. Overall, our findings suggest that choosing the models' default tokenizers yields better performance.

4. Conclusions

Molecular representations based on string representations are a critical focus in computational chemistry research. Our *MolEval* framework facilitates comprehensive benchmarking of various models on molecular property prediction, simplifying evaluation processes. It includes both generic text encoder models and specialized molecular embedding models, revealing that while generic models are competitive in classification tasks, they underperform in regression tasks. This platform enables researchers to select the most appropriate model for further enhancement. Our work establishes a foundation for future advancements in using LLMs for molecular embeddings, with forthcoming efforts aimed at exploring the quality of these embeddings by leveraging techniques from natural language sentence embedding. Additionally, we plan to expand *MolEval* with more tasks as consensus on optimal molecule embedding evaluations evolves, hoping that our toolkit will help standardize research outputs.

References

senteval: an evaluation toolkit for universal sentence representations. In Calzolari, N., Choukri, K., Cieri, C., and Declerck, T. (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Asso-

ciation (ELRA), May 2018.

Chithrananda, S., Grand, G., and Ramsundar, B. chemberta: large-scale self-supervised pretraining for molecular property prediction. *Machine Learning for Molecules Workshop at NeurIPS 2020.*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. bert: pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Gao, T., Yao, X., and Chen, D. simcse: simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910. Association for Computational Linguistics, November 2021. doi: 10.18653/v1/2021.emnlp-main.552.

Guo, T., Nan, B., Liang, Z., Guo, Z., Chawla, N., Wiest, O., Zhang, X., et al. what can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36: 59662–59688, 2023.

Heyer, K. Roberta-zinc-480m. https://huggingface.co/entropy/roberta_zinc_480m, 2023.

Jaeger, S., Fulle, S., and Turk, S. mol2vec: unsupervised machine learning approach with chemical intuition. *Journal of chemical information and modeling*, 58(1):27–35, 2018.

Li, J. and Jiang, X. mol-bert: an effective molecular representation with bert for molecular property prediction. *Wireless Communications and Mobile Computing*, 2021, 2021.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. roberta: a robustly optimized bert pretraining approach, 2020.

Liu, Y., Zhang, R., Li, T., Jiang, J., Ma, J., and Wang, P. molrope-bert: an enhanced molecular representation with rotary position embedding for molecular property prediction. *Journal of Molecular Graphics and Modelling*, 118: 108344, 2023.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. mteb: massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, May 2023. URL https://aclanthology.org/2023.eacl-main.148.

OpenAI. Chatgpt [large language model], 2023. URL https://platform.openai.com/docs.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Ramsundar, B., Eastman, P., Walters, P., Pande, V., Leswing, K., and Wu, Z. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019.

Reimers, N. and Gurevych, I. sentence-bert: sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

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

Ross, J., Belgodere, B., Chenthamarakshan, V., Padhi, I., Mroueh, Y., and Das, P. large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022.

Schwaller, P., Gaudin, T., Lanyi, D., Bekas, C., and Laino, T. "found in translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chemical science*, 9(28):6091–6098, 2018.

Touvron, H., Lavril, T., and Izacard. llama: open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. llama 2: open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Zang, X., Zhao, X., and Tang, B. hierarchical molecular graph self-supervised learning for property prediction. *Communications Chemistry*, 6(1):34, 2023.

Zhang, X.-C., Wu, C.-K., Yang, Z.-J., Wu, Z.-X., Yi, J.-C., Hsieh, C.-Y., Hou, T.-J., and Cao, D.-S. mg-bert: leveraging unsupervised atomic representation learning for molecular property prediction. *Briefings in bioinformatics*, 22(6):bbab152, 2021.