Geometry-enhanced molecular representations using X-attention

Final Report

Anshuman Sinha[1*] **Bhavay Aggarwal**[1*]

[1] Georgia Institute of Technology anshs@gatech.edu, baggarwal9@gatech.edu

Abstract

Drug and small molecule discovery holds an important status in the ongoing human evolution. Molecular property prediction is a critical task in drug discovery and other areas of molecular science. Accurate molecular representation learning is essential for achieving high performance in this task and other downstream tasks. Current molecular representation learning methods, including graph convolutional networks (GCNs), have limitations in capturing important molecular features. While capturing the topological properties and structures these graphs tend to miss on the important geometrical features of the molecules. The paper ”Geometry-enhanced molecular representation learning for property prediction” proposes a new method for molecular representation learning that incorporates geometry information using geometry-based graph neural network architecture and dedicated self-supervised learning pertaining tasks to learn geometry knowledge. In this research, we tried to extend this method using additional pertaining tasks and by further training the self-supervised atom and bond embedding using X attention, a technique that has shown promise in improving representation learning. Our self supervised learning algorithm is shown to learn distinguishable features as shown by the latent space of the embedding. The final multi-modal embedding shows improved results in supervised learning downstream tasks.

Keywords

Drug discovery, Molecular embedding, Graphical models, Self-Supervised Learning (SSL), X-attention.

Introduction

Molecular representation learning is the process of encoding molecular structures into numerical vectors that can be used as input to machine learning models for molecular property prediction. [2] The goal of molecular representation learning is to capture important features of the molecule that are relevant to predicting its properties or behaviour. The aim of this project is to use various deep learning techniques to achieve molecular embedding which can better perform downstream supervised learning tasks. [9, 7] While predicting molecular properties is an important task in drug discovery and materials science, but it requires large amounts of data to train machine learning models. However, in many cases, collecting sufficient data is not always feasible due to the cost and time required for laboratory experiments. In such scenarios, leveraging self-supervised graph embedding can help overcome the data limitation and enable accurate predictions. With similar motivation we propose further use of self-supervised learning methods using additional pretraining tasks to achieve better initial embedding from the molecular graphs in the first place, before implementing our multi-modal x-attention.

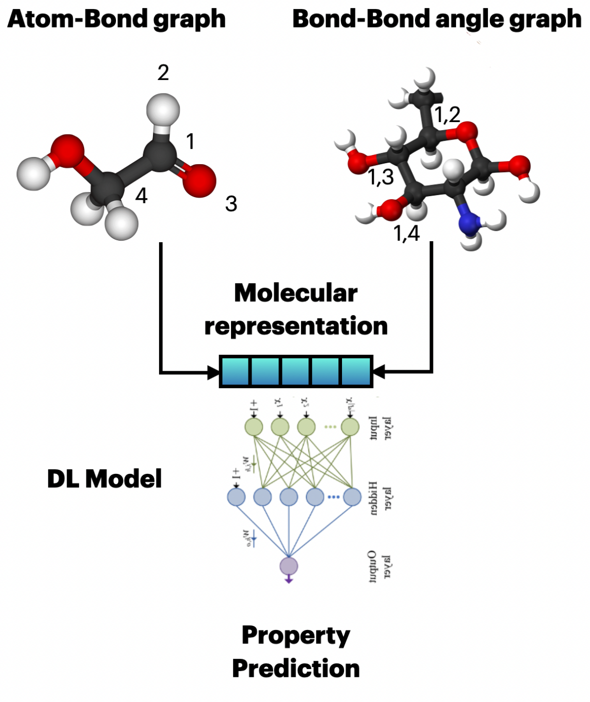


Figure 1: Model pipeline of the current work which includes Embedding generation of a multi-modal molecular graph for downstream property prediciton.

To obtain effective molecular representation, previous works have used graph-based encoding methods like GNNs and GCNs. [13, 11] The Graph Neural Networks (GNNs)

[*] These authors contributed equally.

have shown great promise in generating molecular embeddings by learning from molecular graph structures. However, GNNs require a significant amount of labeled data to achieve good performance, which is often a bottleneck in molecular embedding generation. In this project, we propose to use self-supervised pre-training of graphs as a way to enhance the performance of GNNs in molecular embedding generation tasks. As introduced in the previous paragraph we plan to use self-supervised techniques. Graph-based methods, such as GCNs, are a popular approach to molecular representation learning. However, these methods treat the molecules as topological graphs without fully utilizing the molecular geometry information and hence have limitations in their ability to capture complex molecular features, such as stereochemistry and conformational flexibility. Fang et al. [2] proposes a new method for molecular representation learning that addresses some of these limitations. The method incorporates geometry information, such as bond lengths and angles, into the representation learning process. It also uses the uses several dedicated geometry-level self-supervised learning strategies to learn the molecular geometry knowledge.

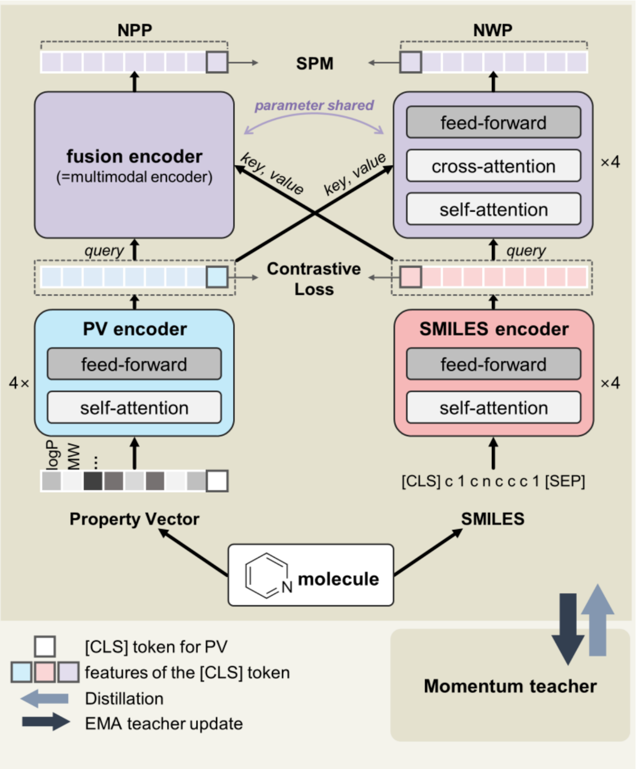


Figure 2: The overview of the X attention model architecture.

Self supersived pre-training: Self-supervised pre-training of graphs is a technique for training Graph Neural Networks (GNNs) using unsupervised learning methods. [3, 10, 12] In self-supervised pre-training, the GNN model is trained to learn meaningful representations of the graph data without relying on explicit supervision from labeled data. The key idea behind self-supervised pre-training is to design pretext tasks that require the GNN model to learn a useful representation of the graph structure. For example, in the masked node prediction task, a subset of the nodes in the graph are randomly masked, and the GNN model is trained to predict the missing node features based on the information from the neighboring nodes. Similarly, in the graph context prediction task, the GNN model is trained to predict the context of a subgraph given a randomly sampled central node. By pre-training a GNN model using self-supervised learning, the model can learn to capture the underlying patterns and structure in the graph data, which can then be fine-tuned on downstream tasks with labeled data. Self-supervised pre-training has been shown to be an effective strategy for improving the performance of GNNs, particularly in scenarios where labeled data is scarce or expensive to obtain and they have also been implemented in various cases of molecular graphs. [8, 6]

Objectives

The main objective of this project is to investigate the effectiveness of self-supervised pre-training of multi-modal graph representation using X-attention for molecular embedding generation. Specifically, we aim to achieve the following goals:

1. Design self-supervised pretext tasks that can capture the structural properties of molecular graphs.
2. Pre-train a GNN model using the self-supervised pretext tasks to learn meaningful representations of molecular graphs.
3. Perform additional training using the X Attention model on both the pretrained GEM embedding and the improved GEM embeddings.
4. Fine-tune the pre-trained GNN model on downstream molecular embedding generation tasks using labeled data.
5. Compare the performance of the pre-trained GNN model with state-of-the-art molecular embedding generation methods.

Methodology

We implemented the following methodology in the stepwise manner as follows. We will further explain all these steps in this section.

1. Generate A-B and BA graph data structures. Graph generator
2. Pre-train the model with Zinc dataset (BA and BAB). Graph SSL
3. With SSL Embeddings Train X-attention model. X-attention
4. For downstream application generate AB and BA embeddings. Downstream task
5. Generate multi-modal embedding using X-attention. X-attention

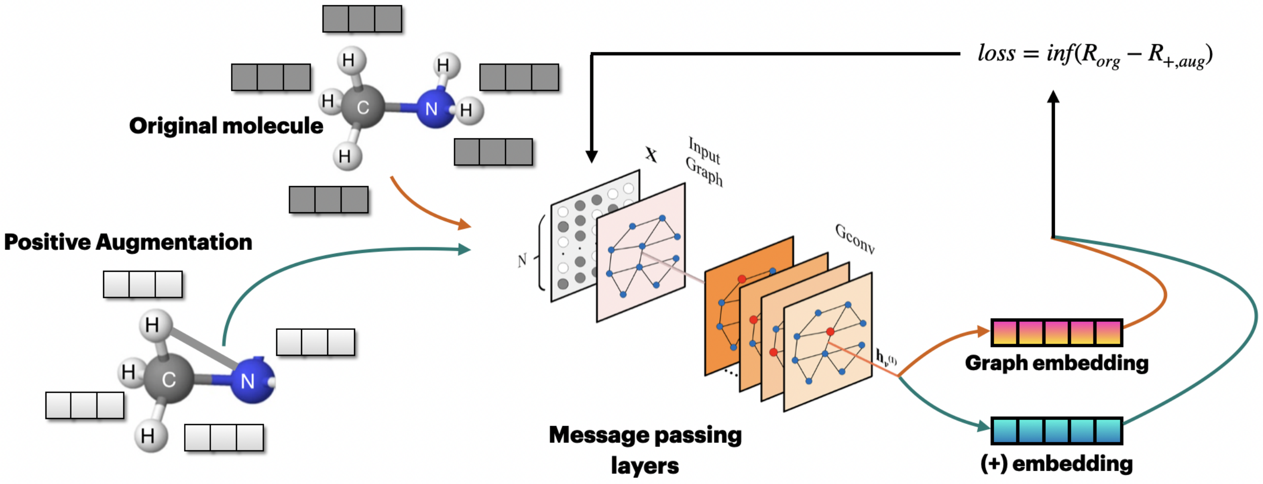


Figure 3: Model pipeline of the current work which includes Embedding generation of a multi-modal molecular graph for downstream property prediciton.

6. Make final prediction from the material properties. Downstream tasks

Following the steps as shown above, we firstly developed our own model for generating the Atom-Bond graph and Bond-Bond Angle graph of the molecules. The Atom-Bond graph comprises of Atoms as nodes of the graph and Bonds between atoms as edges of the graph. The Node features for this graph are: Atomic number, . The edge features include: Bond length, One hot encoded aromaticity, Cyclic nature of the bond (Boolean). While the Bond-Bond Angle graph comprises of bonds as nodes of the graph and bond angle between bonds as edges of the graph. The Node features for this graph are related to the bonds, such as: Bond length, nature of bond etc. The edge features includes: Bond angle values (in radian).

After generating both the graphs we use them separately to train our SSL model. The SSL model will be trained on the same graph without using its labels, such that our model can learn the inherent structure of the molecules. We try and create the positive augementation of each molecule, such that we have a similar structure of the molecule in our dataset. While all the other structure in the batch will be the dissimilar molecule. The process of creating such a dataset is shown as follows in the following Table 1. The loss function $L_{InfoNCE}$ used for training the SSL model comes from information theory applied to contrastive learning. It measures the loss of information in going from 1 structure to another.

$$L_{InfoNCE} = -log\left(\frac{e^{d(v,v^+)}}{e^{d(v,v^+)+\sum_{u\in\{v^-\}}} e^{d(v,u)}}\right)$$

The next step is to train the X-atention framework. For training the X-attention network, we used publicly available molecular datasets such as the ZINC database [5] from the set of our generated embeddings from SSL. We randomly sampled 5k samples from the zinc database for our X-attention model. We extract the atom-bond and bond-angle embeddings for our sampled dataset. For comparing the results, we selected two datasets, ClinTox for classification and FreeSolv for regression. With the generated embeddings of SSL.

We wish to find a common embedding that uses both these critical features to get a vector representing both topology and structure. Fang et al.[2] implemented two different approaches for this - Mean Pooling and Graph Pooling. The authors, however, do not mention the difference in performance observed when using these two different pooling methods. We wanted to experiment with further training the atom-bond and bond-angle graph embeddings and try to capture the details of these embeddings better and to create more generalizable molecular representations. One promising approach we found was demonstrated by Chang et al.[1] in the SPMM that utilizes self-supervised learning along with the molecules SMILES representation and chemical properties to learn molecular representations. The novel architectural component of the model was X-attention, a form of alternating cross-attention to learn joining embeddings. X attention was introduced by Jaeyoung et al.[4] where they used it to embed learned image and its corresponding caption representations into the same space. The idea behind it is that both the image and its caption should represent the same context; hence, their representations should be comparable and in the same

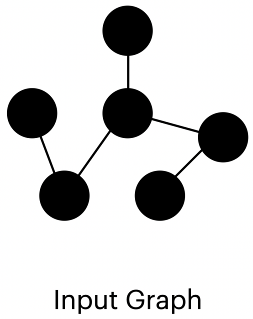
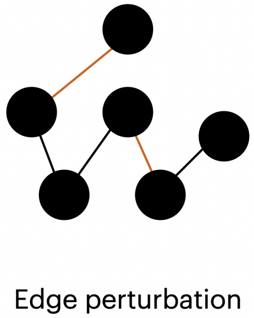
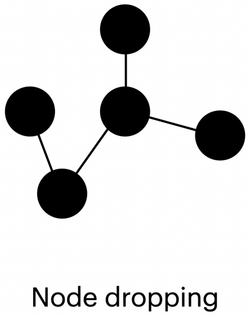
Augumentation	Structure
Original Molecule	
Edge purturbation	
Node purturbation	

Table 1: The features used when training our X-attention model for multi-model training tasks.

space. To make the representations similar, contrastive learning is used, which pushes the image and caption representations together while pushing away the negative pairs. This enables modality-agnostic learning, which gives the model a better understanding of the relationship between the modalities through more efficient data utilization by alternately using the given image-text pair as key/value and query. The intermodal similarity is defined as

$$s_{AB2BA} = \frac{exp(sim(AB,BA_n)/\tau}{\Sigma_{n=1}^{N} exp(sim(AB,BA_n)/\tau)}$$
$$s_{BA2AB} = \frac{exp(sim(BA,AB_m)/\tau}{\Sigma_{m=1}^{M} exp(sim(BA,AB_m)/\tau)}$$

where AB is the atom-bond graph embedding, and BA is the bond-angle graph embedding. The intramodal similarity is calculated similarly,

$$s_{AB2BA} = \frac{exp(sim(AB,AB_n)/\tau}{\Sigma_{n=1}^{N} exp(sim(AB,AB_n)/\tau)}$$
$$s_{BA2AB} = \frac{exp(sim(BA,BA_m)/\tau}{\Sigma_{m=1}^{M} exp(sim(BA,BA_m)/\tau)}$$

Although Chang et al.[1], we do not perform hard-negative mining to select our positive and negative pairs. Instead, each sample's pair of embedding acts as the positive pair while the other samples act as a negative pair. The overall contrastive loss is defined using the cross-entropy loss H and one-hot similarity

$$L_{contrastive} = 0.5 * (H(y_{AB2BA}, s_{AB2BA}) + H(y_{BA2AB}, s_{BA2AB}) + H(y_{AB2AB}, s_{AB2AB}) + H(y_{BA2BA}, s_{BA2BA}))$$

For the downstream task of comparing the results, we selected two datasets, ClinTox for classification and FreeSolv for regression. With the generated embeddings from the trained SSL, we apply those embeddings to the trained X-attention model and generate a final combined multi-modal embedding as shown from Figure 4.

After contrastive learning, the representations are fed into the Fusion Encoder, which alternates attention between 2 different tasks resembling an X. This model can be visualized in Figure 2. Considering that the SPMM model does not utilize the structure of molecules yet performs comparable to others and is even SOTA for the BBBP dataset, we believed that X attention could be further help improve the atom and bond embeddings created by GEM. The atom and bond embeddings can be thought of as two different modalities but represent the same context, and instead of aggregating them into a single graph embedding, we can further use contrastive learning and the fusion encoder to learn a better representation. However, since we are using the pre-trained embeddings from GEM, we cannot perform self-supervised training and cannot fully utilize the SPMM architecture. We instead focus only on the X-attention module and modify it to our needs. The output of the GEM model is two embeddings of different

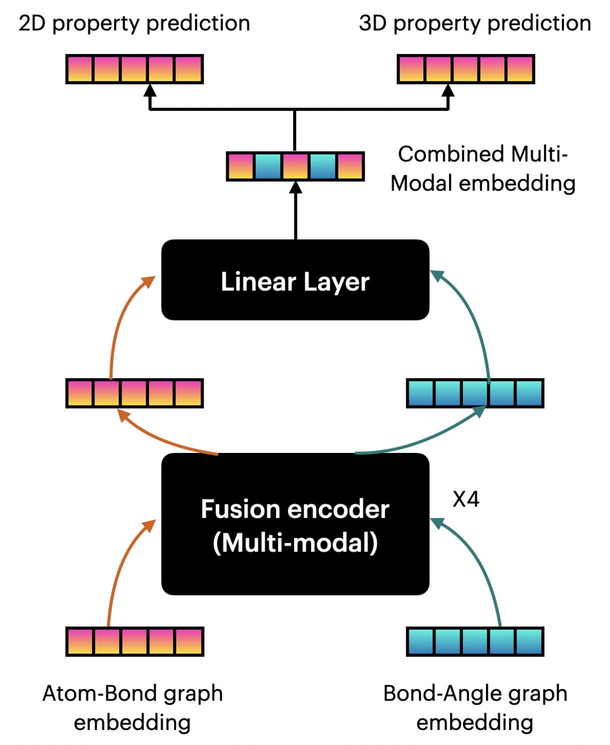

2D property prediction
3D property prediction
Combined Multi-Modal embedding
Linear Layer
Fusion encoder (Multi-modal)
X4
Atom-Bond graph embedding
Bond-Angle graph embedding

Figure 4: The overview of our modified X attention model architecture.

modalities. We use a contrastive learning loss to push them closer into a joint embedding space. We use a modified fusion encoder which consists of 4 stacks of cross-attention layers. These layers help us to capture the inter-modality relationships and infuse them in our new embeddings. We then concatenate the embeddings and pass them through to a series of linear layers to jointly learn them while reducing their dimension. The output of the linear layers are the final molecular representations that combine both the atom-bond and bond-angle representations. To train the model, we extract features of the molecule using RDKit. To learn the chemical and topological features of the molecule, which is similar to learning of the atom-bond graph, we use 2D descriptors. To capture the geometry and the spatial information, which is similar to learning the bond-angle graph, we use 3D descriptors. The architecture of our model is visualized in 4. The two prediction tasks calculate the Mean Absolute Error(MAE) between the actual and the predicted descriptors. Our overall loss function looks like -

$$Loss = L_{contrastive} + L_{2D} + L_{3D}$$

After generating our new embeddings, we use a two-layer linear network.

For the downstream prediction task; we tried to predict the material property which was the initial task of the project. For this task we tried two prediction task one classification and the other regression task. The data-set and the results are discussed in the next section.

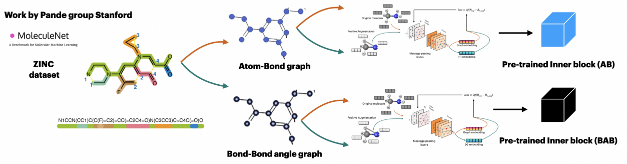

Work by Pande group Stanford
MoleculeNet
A Benchmark for Molecular Machine Learning
ZINC dataset
N1CCN(CC1)C(C(F)=C2)=CC(=C2C4=O)N(C3CC3)C=C4C(=O)O
Atom-Bond graph
Bond-Bond angle graph
Original molecule
Positive Augmentation
Message passing layers
Graph embedding
(+) embedding
Pre-trained Inner block (AB)
Pre-trained Inner block (BAB)

Figure 5: SSL training procedure

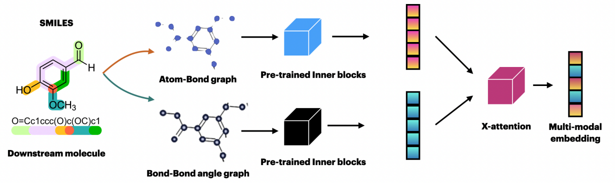


Figure 6: X-attention training procedure

Results

The overall results of the project are discussed in the following section. In the previous section we have discussed the various models which we have built in the project. The results section are divided as follows.

(a) Results for SSL training.
(b) Results for X-attention model.
(c) Results for the downstream model.

(a) Results for SSL training.

We train the SSL framework on A-B graph and B-AB graph separately, in order to obtain the pre-trained GNN layers associated with each of them. The message passing layers of the GNN tries to learn the features of the molecule such that it can generate embeddings which match the (+)ve augmentation. Training Dataset: Zinc dataset ; samples in train subset: 174,619 , samples in val subset: 49,891

We trained the SSL model with 3 GCN layers, and hidden layers with dimension 64 for 20 epochs with the learning rate of 0.001. The results for the L_{infNCE} loss in Training and Validation as shown in Figure 7 seems to saturate within our set 20 epochs.

We also visualised the generated latent space of the molecules for the A-B graph and the B-AB graph. The latent space shown in Figure 8 does seem provide some separation, but doesn't look quite good. (A classification task would've been better as shown in previous works)

(b) Results for X-attention training.

From the generated embeddings of the trained SSL model. We select 5k molecule embedding of each atom-bond and bond-angle-bond graph to train this model. The limited availability of resources restricted us to train on this limited data-set. The loss plots for this training is shown in Figure 9

The training and validation loss

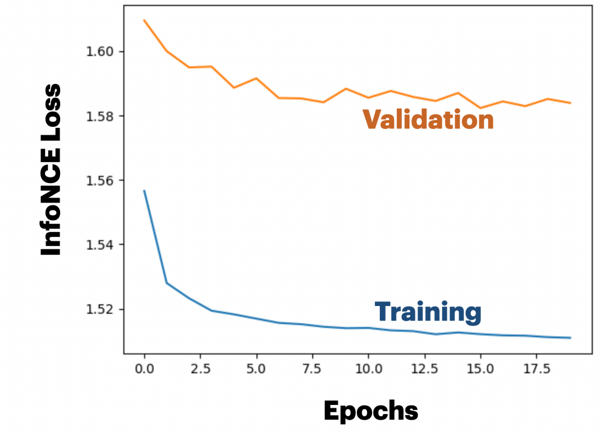


Figure 7: The loss plots generated during the training the SSL.

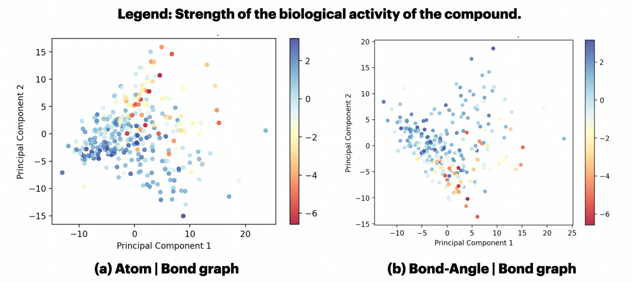


Figure 8: The latent space generated by the trained SSL model.

(c) Results for the downstream model.

After training our sampled zinc database on our X-attention model, we extract molecular representations for each molecule in our database. For classification, we used the ClinTox dataset which contains 1,478 molecules and two different tasks from which we chose to predict the molecule's toxicity which is a binary classification problem. For regression, we used the FreeSolv dataset which contains 642 molecules and their experimental and calculated hydration free energies. We focused on predicting the experimental hydration free energy. We used a two-layer linear model for both of these tasks. We re-evaluated the GEM embeddings using the pooled graph embeddings on both datasets. Table 2 contains the results on both tasks.

Dataset	GEM	Our results
FreeSolv (RMSE)	2.83	3.514
ClinTox (Accuracy)	89.8%	90.92%

Table 2: The features used when training our X-attention model for multi-model training tasks.

Our model performs better on the ClinTox classification task but worse on the FreeSolv regression task. We believe that if trained on the original dataset, without any sampling, we can achieve better results and train a much more generalizable and robust model.

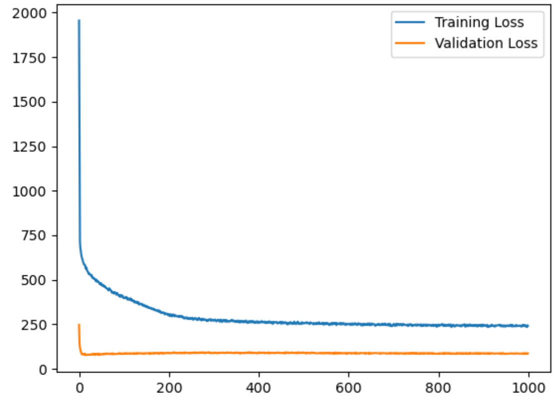


Figure 9: Training and Validation loss per epoch while training our attention model on the zinc dataset.

Conclusion

In this project, we proposed additional pre-taining tasks to capture molecular representations better and use X attention to jointly learn from the multi-modal atom and bond representations and use a contrastive learning approach to learn the difference from the different modalities and generate a better representation combining both of them. We evaluated our experiments on downstream tasks as defined in the original GEM framework. The results of this project had some improvements over our evaluations of GEM which shows future potential. SSL result seems to distinguish the embedding space but further improvement could've lead to better downstream results. The X-attention model was able to generate better embedding, but the model generalisability can be improved further. Since our X-attention model was trained on a limited set of molecules and hence it does not generalise well across different downstream datasets containing a diverse set of molecules. For further improvements in results, instead of implementing a 2-step approach, we can combine the SSL and X-attention method for generating the molecular embeddings.

References

[1] Chang, J.; and Ye, J. C. 2023. Bidirectional Generation of Structure and Properties Through a Single Molecular Foundation Model.

[2] Fang, X.; Liu, L.; Lei, J.; He, D.; Zhang, S.; Zhou, J.; Wang, F.; Wu, H.; and Wang, H. 2022. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2): 127–134.

[3] Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Representation Learning on Graphs: Methods and Applications. *IEEE Data Engineering Bulletin*.

[4] Huh, J.; Park, S.; Lee, J. E.; and Ye, J. C. 2023. Improving Medical Speech-to-Text Accuracy with Vision-Language Pre-training Model. *arXiv preprint arXiv:2303.00091*.

[5] Irwin, J. J.; and Shoichet, B. K. 2005. ZINC- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1): 177–182.

[6] Krenn, M.; Hase, F.; Nigam, A.; Friederich, P.; and Aspuru-Guzik, A. 2020. SELFIES: Self-Referencing Embedded Strings for Molecular Fingerprints. In *Advances in Neural Information Processing Systems*.

[7] Lavecchia, A. 2015. Machine-learning approaches in drug discovery: methods and applications. *Drug discovery today*, 20(3): 318–331.

[8] Lewis, P.; Kovalevskyi, O.; Liu, Y.; Petroski Such, F.; Wenzel, A.; Zeller, M.; and Stoyanov, V. 2020. MolBERT: Pre-trained general purpose chemical text embeddings. In *Advances in Neural Information Processing Systems*.

[9] Searls, D. B. 2005. Data integration: challenges for drug discovery. *Nature reviews Drug discovery*, 4(1): 45–58.

[10] Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2019. Deep Graph Infomax. In *International Conference on Learning Representations*.

[11] Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. PMLR.

[12] Xu, Z.; Zhang, H.; Chen, Y.; Long, G.; and Liu, X. 2020. Graph Contrastive Learning with Augmentations. In *International Conference on Machine Learning*, 10616–10625.

[13] Zhang, S.; Tong, H.; Xu, J.; and Maciejewski, R. 2019. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1): 1–23.

Supplementary information

Feature	3D Descriptors
2D Descriptors	BalabanJ, BertzCT, Chi0, Chi0n, Chi0v, Chi1, Chi1n, Chi1v, Chi2n, Chi2v, Chi3n, Chi3v, Chi4n, Chi4v, ExactMolWt, FpDensityMorgan1, FpDensityMorgan2, FpDensityMorgan3, FractionCSP3, HallKierAlpha, HeavyAtomCount, HeavyAtomMolWt, Kappa1, Kappa2, Kappa3, LabuteASA, MaxAbsEStateIndex, MaxEStateIndex, MinAbsEStateIndex, MinEStateIndex, MolLogP, MolMR, MolWt, NHOHCount, NOCount, NumAliphaticCarbocycles, NumAliphaticHeterocycles, NumAliphaticRings, NumAromaticCarbocycles, NumAromaticHeterocycles, NumAromaticRings, NumHAcceptors, NumHDonors, NumHeteroatoms, NumRadicalElectrons, NumRotatableBonds, NumSaturatedCarbocycles, NumSaturatedHeterocycles, NumSaturatedRings, NumValenceElectrons, RingCount, TPSA, QED
3D Descriptors	Asphericity, Eccentricity, InertialShapeFactor, NPR1, NPR2, PMI1, PMI2, PMI3, RadiusOfGyration, RadiusOfGyration

Table 3: The features used when training our X-attention model for multi-model training tasks.