A Appendix

A.1 General Information

A.1.1 Links

The FlexMol code is available at our public repository: https://github.com/Steven51516/FlexMol. The code to reproduce the experiments described in this paper can be found in the experiments directory of the repository. The FlexMol experiment split used in our experiment are adapted from MolTrans: https://github.com/kexinhuang12345/MolTrans.

A.1.2 Licenses

FlexMol is under the BSD 3-Clause License. We, the authors, bear all responsibility in case of violation of rights.

A.2 Drug Encoders Implemented in FlexMol

A.2.1 Sequence and Fingerprint-based Encoders

Morgan: Generates a 1024-length bit vector encoding circular radius-2 substructures, processed with an MLP. [28]

Daylight: Produces a 2048-length vector encoding path-based substructures, processed with an MLP. [28]

ErG: Creates a 315-dimensional 2D pharmacophore description for scaffold hopping, processed with an MLP. [32]

PubChem: Generates an 881-length bit vector where each bit corresponds to a significant substructure, processed using an MLP. [16]

ChemBERTa: Generates embeddings from SMILES strings using the pretrained ChemBERTa model, processed with a linear layer or MLP. [23]

ESPF: Produces a 2586-length sub-structure partition vector, processed with an MLP. [14]

CNN: One-hot encodes SMILES strings and processes them through a multi-layer 1D convolutional neural network, followed by a global max pooling layer. [14]

Transformer: Generates sub-structure partition fingerprints from SMILES strings and encodes them using a self-attention-based transformer model. [14]

A.2.2 2D Graph-based Encoders

Preprocessing involves creating 2D molecular graphs from SMILES strings using RDKit. These graphs are then encoded using various graph neural network models implemented with DGL:

GCN, GAT, GIN: Standard graph neural networks to capture relational and topological features in 2D drug graphs. [17, 34, 37]

AttentiveFP: Utilizes attention mechanisms to prioritize significant molecular substructures. [36]

NeuralFP: Employs neural fingerprinting methods to capture detailed molecular features. [8]

MPNN: Uses message-passing neural networks to transmit information among atoms and bonds in the graph.[10]

A.2.3 3D Graph-based Encoders

Preprocessing involves creating 3D molecular graphs from SMILES strings using RDKit, considering spatial conformation. These graphs are then encoded using:

SchNet: SchNet models to capture 3D spatial relationships. [29]

MGCN: Multi-level graph convolutional networks to learn spatial features. [20]

A.3 Protein Encoders Implemented in FlexMol

A.3.1 Sequence-based Encoders

CNN: One-hot encodes the amino acid sequences and processes them through a multi-layer 1D convolutional neural network, followed by a global max pooling layer. [14]

Transformer: Generates sub-structure partition fingerprints from amino acid sequences and encodes them using a self-attention-based transformer model. [14]

AAC: Generates an 8,420-length vector representing amino acid k-mers, processed with an MLP. [26]

ESPF: Produces a 4,114-length sub-structure partition vector, processed with an MLP. [14]

PseudoAAC: Generates a 30-length vector considering protein hydrophobicity and hydrophilicity patterns, processed with an MLP. [4]

Quasi-seq: Generates a 100-length quasi-sequence order descriptor using sequence-order-coupling numbers, processed with an MLP. [3]

Conjoint triad: Produces a 343-length vector based on the frequency distribution of three continuous amino acids, processed with an MLP. [30]

Auto correlation: Generates a 720-length vector based on the autocorrelation of physicochemical properties along the sequence, processed with an MLP. [12]

CTD: Produces a 147-length vector by calculating composition, transition, and distribution descriptors, processed with an MLP. [7]

ESM: Directly generates embeddings using a pretrained ESM model, processed with a linear layer or MLP. [27]

ProtTrans-t5, ProtTrans-bert, ProtTrans-albert: Directly generates embeddings using pretrained models (T5, BERT, ALBERT respectively), processed with a linear layer or MLP. [9]

A.3.2 3D Graph-based Encoders

Preprocessing involves creating 3D graphs from protein PDB structures. These graphs are then encoded using various graph neural network models:

GCN, GAT, GIN: Standard graph neural networks to capture spatial features in 3D protein structures. [17, 34, 37]

GCN_ESM, GAT_ESM, GIN_ESM: Combines standard GNNs with additional ESM features for enhanced node representations. [35]

PocketDC: Identifies and constructs graphs from binding pockets using DeepChem, encoded with GCN. [39]

GVP: Utilizes Geometric Vector Perceptrons (GVP) to capture geometric and vectorial features. [15]

GearNet: Employs pretrained GearNet layers with relational message passing to capture geometric properties and spatial features. [40]

A.4 Interaction Layers Implemented in FlexMol

Bilinear Attention: The Bilinear Attention Network (BAN) layer captures interactions between 2D feature sets by computing bilinear transformations, followed by attention pooling and batch normalization. [1]

Bilinear Fusion: Combines 1D features from two sources using a bilinear transformation and ReLU activation, capturing multiplicative interactions for enhanced feature representation. [19]

Bidirectional Cross Attention: Combines 2D embeddings from two sources using bidirectional attention and max pooling, creating a unified representation. [25]

Highway: Combines 1D features using multiple highway layers with gated mechanisms to regulate information flow. [41]

Gated Fusion: Combines 1D features from two sources using gated mechanisms and transformations, producing a fused representation. [22]

Multi-Head Attention: Applies attention mechanisms to 2D features using multiple heads, with optional residual connections and layer normalization. [33]

Concatenation: Concatenation is the simplest form of combining multiple feature embeddings by joining them end-to-end.

A.5 Example Usage of FlexMol

This section provides a simple example to using FlexMol for drug-target interaction prediction. A more detailed set of tutorials can be found in the `tutorials` directory of our repository.

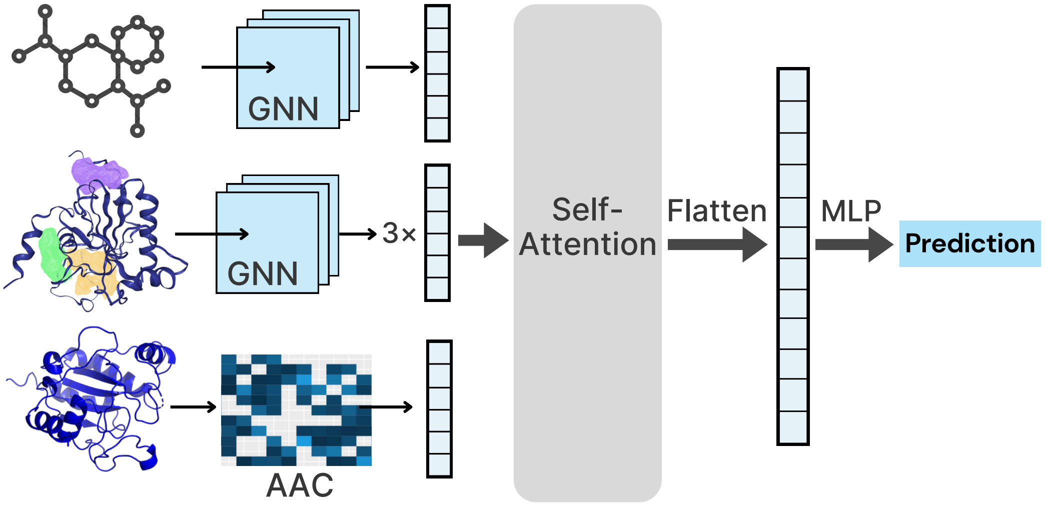

Figure 1: The example model constructed using FlexMol involves several stages. First, drug and protein sequences/structures are encoded using selected encoders: GCN for drug encoding, PocketDC for protein structure encoding, and AAC for protein sequence encoding. These encoded features are then stacked and passed through a self-attention interaction layer to capture complex relationships. The output of the interaction layer is flattened and processed through a Multi-Layer Perceptron (MLP) with specified hidden layers.

A.5.1 Loading the Dataset

First, we import the necessary modules from FlexMol and load the DAVIS dataset.

```
# Import necessary modules from FlexMol
from FlexMol.dataset.loader import load_DTI
from FlexMol.encoder import FlexMol
from FlexMol.task import BinaryTrainer

# Load the DAVIS dataset from the specified directory
dir = "data/DAVIS/"
train_df = load_DTI(dir + "train.txt")
val_df = load_DTI(dir + "val.txt")
```

```python
test_df = load_DTI(dir + "test.txt")
```

The load_DTI function is a general-purpose utility that loads drug-target interaction data from the specified directory into DataFrames for training, validation, and testing. The loaded DataFrame contains four columns: drug sequence, protein sequence, protein PDB ID, and interaction label.

A.5.2 Initializing Encoders

Next, we initialize the drug and protein encoders.

```python
# Initialize drug and protein encoders
# Graph Convolutional Network for drug encoding
# PocketDC encoder for protein structures
# Amino Acid Composition encoder for protein sequences
FM = FlexMol()
de = FM.init_drug_encoder("GCN")
pe1 = FM.init_prot_encoder("PocketDC", pdb=True, pocket_num=3)
pe2 = FM.init_prot_encoder("AAC")
```

Here, we use a Graph Convolutional Network (GCN) for drug encoding, PocketDC for protein structure encoding, and Amino Acid Composition (AAC) for protein sequence encoding.

A.5.3 Stacking Features and Setting Interaction Layers

We then stack the features from the different encoders and set the interaction layer to self-attention.

```python
# Stack the features from different encoders
# Set interaction layer to self-attention
# Flatten the attention outputs for MLP input
# Apply a Multi-Layer Perceptron (MLP) with specified hidden layers
features_stack = FM.stack([de, pe1, pe2])
attention = FM.set_interaction(features_stack, "self-attention")
features_flatten = FM.flatten(attention)
output = FM.apply_mlp(features_flatten, hidden_layers=[512, 512, 256], head
    =1)
```

The stack method combines the outputs of the encoders, and the set_interaction method applies a self-attention mechanism to these combined features. The flatten method prepares the attention layer outputs for the MLP, and the apply_mlp method sets up the MLP with specified hidden layers.

A.5.4 Building and Training the Model

Finally, we initialize the BinaryTrainer and train/test the model.

```python
# Build the FlexMol model
# Initialize the BinaryTrainer for training the model
FM.build_model()
trainer = BinaryTrainer(
    FM, early_stopping="roc-auc", test_metrics=["roc-auc", "pr-auc"],
    device="cuda:0", epochs=50, patience=10, lr=0.0001, batch_size=64,
    metrics_dir = "metrics/example_run"
)
```

The BinaryTrainer is configured for training binary classification tasks with early stopping based on the ROC-AUC metric and evaluates the model using both ROC-AUC and PR-AUC metrics. After testing, the metrics are saved to the user-specified directory.

The code provided in this section serves as a practical example of how FlexMol can be utilized for molecular relational learning tasks, showcasing its flexibility and ease of use.

A.6 Additional Experiments for DDI

A.6.1 Experiment Setup

Dataset: We used DrugBank, downloaded using the TDC Python library [13], for the evaluation of FlexMol-baselines. DrugBank contains 191,808 DDI tuples with 1,706 drugs. Each drug is represented in SMILES format, from which molecular graphical representations are generated using the Python library RDKit. There are 86 interaction types describing how one drug affects the metabolism of another. Each DDI pair is considered a positive sample, from which a negative sample was generated using the method described in the GMPNN-CS framework [24].

Evaluation Method: We performed a stratified split of the dataset to maintain the same interaction type proportions in the training (60%), validation (20%), and test (20%) sets. This was repeated three times, resulting in three stratified randomized folds.

We constructed six FlexMol baselines as detailed in Table 1. The models were trained in mini-batches of 512 with a learning rate of 0.0001. Five state-of-the-art(SOTA) methods were selected for comparison: MHCADDI [6], GMPNN-CS [24], GAT-DDI [24], GMPNN-U [24], and MR-GNN [38].

Table 1: FlexMol Experimental Settings on the DrugBank dataset

Experiment No.	**Drug Encoder 1**	**Drug Encoder 2**	**Interaction**	**Input Feature**
1	CNN	CNN	-	d_s
2	CNN	GCN	-	$d_s + d_g$
3	CNN+PubChem	GCN+PubChem	-	$d_s + d_g$
4	PubChem	PubChem	-	d_s
5	Transformer	Transformer	-	d_s
6	Transformer	Transformer	Cross Attention	d_s

Note: d_s = drug sequence, d_g = drug graph, ’-’ denotes concatenation for combining embeddings.

A.6.2 DDI Experiment Results

Table 2: Comparison of Model Performance on the DrugBank dataset.

Experiment No. / Method	**Accuracy**	**ROC-AUC**	**PR-AUC**	**Precision**	**Recall**
MR-GNN	96.04 ± 0.05	98.87 ± 0.04	**98.57 ± 0.06**	94.48 ± 0.08	**97.78 ± 0.03**
MHCADDI	83.80 ± 0.27	91.16 ± 0.31	89.26 ± 0.37	78.90 ± 0.06	92.26 ± 0.63
SSI-DDI	**96.33 ± 0.09**	**98.95 ± 0.08**	98.57 ± 0.14	**95.09 ± 0.08**	97.70 ± 0.14
GAT-DDI	89.81 ± 1.00	95.21 ± 0.70	93.56 ± 0.90	87.04 ± 1.11	93.56 ± 0.52
GMPNN-CS	95.30 ± 0.05	98.46 ± 0.01	97.94 ± 0.02	93.60 ± 0.07	97.22 ± 0.1
1	83.39 ± 0.08	90.01 ± 0.10	86.35 ± 0.13	81.40 ± 0.10	86.20 ± 0.06
2	80.42 ± 0.20	87.70 ± 0.60	83.92 ± 1.52	78.22 ± 2.20	84.41 ± 0.94
3	83.40 ± 0.26	89.68 ± 0.10	85.73 ± 0.20	80.83 ± 0.06	85.60 ± 0.47
4	82.12 ± 0.15	89.20 ± 0.30	85.54 ± 0.44	80.67 ± 0.30	84.57 ± 0.40
5	87.03 ± 0.19	92.33 ± 0.12	89.13 ± 0.16	83.42 ± 0.10	92.34 ± 0.35
6	87.27 ± 0.24	92.49 ± 0.09	89.37 ± 0.12	83.71 ± 0.20	92.92 ± 0.83

Table 2 shows that while simple combinations using FlexMol generally perform slightly lower than SOTA methods, they remain closely competitive. Experiment #2, in particular, demonstrates that using different encoders for the two drugs tends to decrease performance. Additionally, the interaction layers added in experiment #6 improve model performance compared to experiment #5, highlighting the importance of interaction layers in enhancing the predictive capabilities of the model.

A.7 Additional Experiments for PPI

A.7.1 Experiment Setup

Dataset: We used the Guo yeast dataset [11], which includes 11,188 PPI pairs, with 5,594 positive and 5,594 negative interactions. The data was collected from the Saccharomyces cerevisiae core subset of the Database of Interacting Proteins (DIP), version DIP_20070219. The dataset is available at `https://github.com/aidantee/xCAPT5/tree/master/data/Golden-standard-datasets/Guo-2008`

Evaluation Method: We tested using 5-fold cross-validation with a random split following the xCAPT5 framework [5].

We constructed six FlexMol baselines as detailed in Table 3. The models were trained in mini-batches of 128 with a learning rate of 0.001. Five state-of-the-art(SOTA) methods were selected for comparison: PIPR [2], FSNN-LGBM [21], MARPPI [18], TAGPPI[31], and xCAPT5[5].

Table 3: FlexMol Experimental Settings on the Guo dataset

Experiment No.	Protein Encoder 1	Protein Encoder 2	Interaction	Input Feature
1	CNN	CNN	-	p_s
2	CNN	GCN	-	$p_s + p_g$
3	AAC	AAC	-	p_s
4	AAC+CNN	AAC+GCN	-	$p_s + p_g$
5	Transformer	Transformer	-	p_s
6	Transformer	Transformer	Cross Attention	p_s

Note: p_s = protein sequence, p_g = protein graph, '-' denotes concatenation for combining embeddings.

A.7.2 PPI Experiment Results

Table 4: Comparison of Model Performance on the Guo dataset.

Experiment No. / Method	Accuracy	Precision	Recall	F1-Score
PIPR	96.47 ± 0.21	96.31 ± 0.23	96.67 ± 0.22	96.48 ± 0.20
FSNN-LGBM	98.46 ± 0.20	98.73 ± 0.25	98.18 ± 0.18	98.45 ± 0.20
MARPPI	96.03 ± 0.76	98.12 ± 0.98	93.51 ± 1.22	NA
TAGPPI	97.81	98.10	98.26	97.80
HNSPPI	98.57 ± 0.11	98.30 ± 0.22	98.85 ± 0.13	98.57 ± 0.11
xCAPT5	**99.76 ± 0.05**	**99.76 ± 0.04**	**99.75 ± 0.07**	**99.37 ± 0.27**
1	77.07 ± 0.87	85.52 ± 0.69	66.81 ± 2.43	71.81 ± 1.59
2	88.12 ± 1.93	89.61 ± 0.64	86.32 ± 4.47	87.91 ± 2.53
3	89.71 ± 2.04	89.31 ± 1.32	86.34 ± 0.34	89.72 ± 2.25
4	90.02 ± 1.52	91.96 ± 1.80	87.92 ± 4.83	89.73 ± 2.22
5	90.26 ± 0.23	90.24 ± 2.25	90.42 ± 2.32	90.31 ± 1.03
6	89.18 ± 1.09	90.22 ± 2.28	89.10 ± 0.93	89.51 ± 1.02

Note: NA indicates that data is not available in the reference literature. The TAGPPI method does not include standard deviation values in the literature, and thus no standard deviations are reported here.

Form table 4, we observed that all results from FlexMol baselines are significantly lower than SOTA methods. This suggests that the PPI task is more challenging and requires more specialized modeling methods rather than simple encoder combinations. However, some trends were noted. For instance, the combination of encoders in Experiment #4 improves performance compared to Experiments

#2 and #3. Interestingly, we found that the cross-attention in Experiment #6 does not improve performance, indicating that the influence of interaction layers can vary depending on the specific dataset and task.

References

[1] Peizhen Bai, Filip Miljković, Bino John, and Haiping Lu. Interpretable bilinear attention network with domain adaptation improves drug–target prediction. *Nature Machine Intelligence*, 5(2):126–136, Feb 2023.

[2] Minghao Chen, Chien Ju-Ting Ju, Guangyu Zhou, Xinkai Chen, Ting Zhang, Kai-Wei Chang, Carlo Zaniolo, and Wei Wang. Multifaceted protein-protein interaction prediction based on siamese residual rcnn. *Bioinformatics*, 35(14):305–314, 2019.

[3] Kuo-Chen Chou. Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. *Biochemical and Biophysical Research Communications*, 278(2):477–483, Nov 2000.

[4] Kuo-Chen Chou. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, 43(3):246–255, Mar 2001.

[5] Thanh Hai Dang and Tien Anh Vu. Xcapt5: Protein–protein interaction prediction using deep and wide multi-kernel pooling convolutional neural networks with protein language model. *BMC Bioinformatics*, 25(1), Mar 2024.

[6] Andreea Deac, Yu-Hsiang Huang, Petar Veličković, et al. Drug-drug adverse effect prediction with graph co-attention. In *ICML Workshop on Computational Biology*. ICLR, May 2019.

[7] I Dubchak, I Muchnik, S R Holbrook, and S H Kim. Prediction of protein folding class using global description of amino acid sequence. *Proceedings of the National Academy of Sciences*, 92(19):8700–8704, Sep 1995.

[8] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292*, 2015.

[9] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Wang Yu, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[10] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[11] Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Research*, 36(9):3025–3030, Apr 2008.

[12] David S. Horne. Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities. *Biopolymers*, 27(3):451–477, Mar 1988.

[13] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Advances in Neural Information Processing Systems*, 2 2021.

[14] Kexin Huang, Tianfan Fu, Lucas M Glass, Marinka Zitnik, Cao Xiao, and Jimeng Sun. Deeppurpose: A deep learning library for drug–target interaction prediction. *Bioinformatics*, 36(22–23):5545–5547, Dec 2020.

[15] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021.

[16] Sunghwan Kim, Jian Chen, Tiezheng Cheng, Asta Gindulyte, Jane He, Sherry He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bin Yu, Leonid Zaslavsky, Jie Zhang, and Evan E Bolton. Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109, 2019.

[17] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[18] Xiaoyan Li, Peng Han, Weiliang Chen, Changxin Gao, Shuaipeng Wang, Ting Song, Meiyu Niu, and Alejandro Rodriguez-Patón. Marppi: boosting prediction of protein–protein interactions with multi-scale architecture residual network. *Briefings in Bioinformatics*, 24(1):524, 2023.

[19] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015.

[20] Chengqiang Lu, Qi Liu, Chao Wang, Zhenya Huang, Peize Lin, and Lixin He. Molecular property prediction: A multilevel quantum interactions modeling perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1052–1060, Jul 2019.

[21] Saurav Mahapatra and Satyabrata S Sahu. Improved prediction of protein–protein interaction using a hybrid of functional-link siamese neural network and gradient boosting machines. *Briefings in Bioinformatics*, 22(6):255, 2021.

[22] Qi Mao, Hanxiao Wang, Roger W. H. Lau, Peter K. K. Loh, and Xiaorong Li. Gated multimodal units for information fusion. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 2314–2320, 2017.

[23] Sylwia Nowakowska. *Chemberta-2: Fine-tuning for molecule's HIV replication inhibition prediction*, Sep 2023.

[24] Arnold K Nyamabo, Hui Yu, Zun Liu, and Jian-Yu Shi. Drug–drug interaction prediction with learnable size-adaptive molecular substructures. *Briefings in Bioinformatics*, 23(1), Oct 2021.

[25] Ying Qian, Xinyi Li, Jian Wu, and Qian Zhang. Mcl-dti: Using drug multimodal information and bi-directional cross-attention learning method for predicting drug–target interaction. *BMC Bioinformatics*, 24(1), Aug 2023.

[26] M. Reczko, D. Karras, and H. Bohr. An update of the def database of protein fold class predictions. *Nucleic Acids Research*, 25(1):235–235, Jan 1997.

[27] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019.

[28] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

[29] Kristof T Schütt, Pieter-Jan Kindermans, Huziel E Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in Neural Information Processing Systems*, 30:992–1002, 2017.

[30] Juwen Shen, Jian Zhang, Xiaomin Luo, Weiliang Zhu, Kunqian Yu, Kaixian Chen, Yixue Li, and Huanliang Jiang. Predicting protein-protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences*, 104(11):4337–4341, 2007.

[31] Bo Song, Xiaoyong Luo, Xin Luo, Yong Liu, Zhibo Niu, and Xiangxiang Zeng. Learning spatial structures of proteins improves protein–protein interaction prediction. *Briefings in Bioinformatics*, 23(2):558, 2022.

[32] Nikolaus Stiefl, Ian A. Watson, Knut Baumann, and Andrea Zaliani. Erg: 2d pharmacophore descriptions for scaffold hopping. *ChemInform*, 37(17), Apr 2006.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[35] Fang Wu, Lirong Wu, Dragomir Radev, Jinbo Xu, and Stan Z. Li. Integration of pre-trained protein language models into geometric deep learning networks. *Communications Biology*, 6(1), Aug 2023.

[36] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, Aug 2019.

[37] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[38] Nuo Xu, Pinghui Wang, Long Chen, et al. Mr-gnn: Multi-resolution and dual graph neural network for predicting structured entity interactions. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2019-August, pages 3968–3974. International Joint Conferences on Artificial Intelligence Organization, 2019.

[39] Mehdi Yazdani-Jahromi, Niloofar Yousefi, Aida Tayebi, Elayaraja Kolanthai, Craig J Neal, Sudipta Seal, and Ozlem Ozmen Garibay. Attentionsitedti: An interpretable graph-based model for drug-target interaction prediction using nlp sentence-level relation classification. *Briefings in Bioinformatics*, 23(4), Jul 2022.

[40] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations*, 2023.

[41] Yan Zhu, Lingling Zhao, Naifeng Wen, Junjie Wang, and Chunyu Wang. Datadta: A multi-feature and dual-interaction aggregation framework for drug–target binding affinity prediction. *Bioinformatics*, 39(9), Sep 2023.