GraSeq: Graph and Sequence Fusion Learning for Molecular Property Prediction

Zhichun Guo
University of Notre Dame
Notre Dame, IN 46556
zguo5@nd.edu

Wenhao Yu
University of Notre Dame
Notre Dame, IN 46556
wyu1@nd.edu

Chuxu Zhang
University of Notre Dame
Notre Dame, IN 46556
chuxuzhang@gmail.com

Meng Jiang
University of Notre Dame
Notre Dame, IN 46556
mjiang2@nd.edu

Nitesh Chawla
University of Notre Dame
Notre Dame, IN 46556
nchawla@nd.edu

ABSTRACT

With the recent advancement of deep learning, molecular representation learning - automating the discovery of feature representation of molecular structure, has attracted significant attention from both engineering and science researchers. It can facilitate a variety of downstream applications, including bio-property prediction, chemical reactivity prediction, etc. Despite the fact that current *SMILES* string or molecular graph molecular representation learning algorithms (via sequence modeling and graph neural networks, respectively) have achieved promising results, there is no work to integrate the capabilities of both approaches in preserving molecular characteristics (e.g, atomic cluster, chemical bond) for further improvement. In this paper, we propose **GraSeq**, a joint graph and sequence representation learning model for molecular property prediction. Specifically, **GraSeq** makes a complementary combination of graph neural networks and recurrent neural networks for modeling two types of molecular inputs, respectively. In addition, it is trained by the multitask loss of unsupervised reconstruction and various downstream tasks, using limited size of labeled datasets. In a battery of chemical property prediction tests, we demonstrate that our GraSeq model achieves better performance than state-of-the-art approaches.

KEYWORDS

Molecular Representation Learning, Sequence Model, Graph Neural Network, Fusion Learning

ACM Reference Format:
Zhichun Guo, Wenhao Yu, Chuxu Zhang, Meng Jiang, and Nitesh Chawla. 2020. GraSeq: Graph and Sequence Fusion Learning for Molecular Property Prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3340531.3411981

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411981

1 INTRODUCTION

Virtual screening of large compound libraries plays an essential role in the drug development pipeline [38]. Structure-based virtual screening results in a set of leads; molecular compounds which may have a particular bio-activity of interest. In the pioneering days of virtual screening it was hoped that the increase in lead candidates brought about by efficient computational algorithms would result in a large influx of new drugs reaching approval from regulatory agencies, however, that has not been the case despite great advances over several decades [16, 33].

The most significant costs in getting a new drug to market are time and resources wasted on research and clinical trials of drug candidates which ultimately never receive approval.[7] Improvements to virtual screening methods can play a vital role in reducing these costs by screening out these failed candidates earlier, thereby resulting in a smaller set of leads to be investigated further. As with many fields, recent advances in deep learning has given rise to renewed interest in developing data-driven virtual screening methods in the hopes to see further improvements to the accuracy of these methods.[48]

Molecular property prediction is an important part of virtual screening. For prediction of learning targets which are properties of the molecular structure, and not a particular conformation of that structure, existing neural network based approaches for molecular representation can be categorized into two main groups: graph-based [21, 37] and sequence-based [15, 47, 51, 52] representations as shown in Figure 1.

Graph-based representations build on the recent successes of graph neural networks[13, 44] to incorporate significant amounts of domain knowledge about the structure of the molecule into the design of the model. By viewing atoms as nodes and chemical bonds as edges, graph-convolutions enable models to focus on the most important connections in the molecular structure, those directly bonded to an atom, when building an atomic embedding from a graph neural network. Furthermore, the preservation of bonding information (e.g. single, double, triple, aromatic) can be incorporated into the model in a way that cannot be done for sequence-based representations.

Sequence-based representation, on the other hand, use textual encodings of molecular structure, such as simplified molecular-input line-entry system (*SMILES*) [43], to formulate the problem

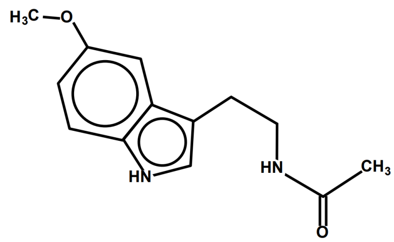
Molecule Graph	
SMILES	**CC(=O)NCCC1=CNc2c1cc(OC)cc2**

Figure 1: A molecule (e.g., melatonin) could be represented as two different forms; a molecular graph and simplified molecular-input line-entry system (*SMILES*) sequence.

in a way similar to natural language processing applications. By viewing molecular structure as sequence data, many natural language processing techniques (e.g, Word2Vec [27], BERT [6]) can be adopted, which have demonstrated powerful capabilities in predicting different kinds of molecular properties[15, 47, 51, 52]. While sequence-based representations forego domain knowledge about the connectivity of the molecule in model design, there are many non-local interactions important to many properties of a molecule. Graph-based models are limited in how distant atoms can interact based on the number of graph-convolution iterations. Sequence-based representations are also able to incorporate special molecular configurations (e.g. chirality) which have so far not been implemented into any graph-based models.

Pharmaceutical companies face a race for effective and efficient drug discovery. Drug development via wet-lab experimentation is a difficult, expensive, and time-consuming process [38]. The new potential drug candidate will be selected only if all chemical drug properties (e.g., affinity, selectivity, metabolic stability) are biologically tested in the lab environment and deemed to reach the drug requirement tests. If any of these tests fail, new potential medicines with new chemical compounds need to be tested with the aforementioned complex experimental processes went over again. This leads to significantly added costs and time. Therefore, computational approaches are becoming popular and now play a key role in drug discovery [1, 30, 38]. The traditional approaches are not data-driven and heavily depend on human expertise knowledge [34, 35]. The recent advent of deep learning provides drug discovery with new possibilities through neural network based molecular representation learning [8, 12, 36, 47, 51]. For example, Sabrina et al. [15] automated the process of molecular representation for various downstream applications (e.g, bio-property prediction, molecular design, and chemistry reactivity prediction).

Existing neural network based molecular representation learning approaches could be categorized into two different groups according to the input data type of molecules: simplified molecular-input line-entry (*SMILES*) [43] and molecular graph (as shown in Figure 1). On the one hand, *SMILES* is a sequence notation for describing the structure of chemical species, which is unique for each structure generated from the canonicalization algorithm. *SMILES* sequence can mainly bring the following two benefits:

- Ionic groups and atomic groups are represented in the canonical way, which avoids confusion with their surrounding atomic groups. For instance, the ammonium is denoted as $[NH4+]$ rather than $HHNHH$.
- Some specially defined symbols are used to preserve chemical properties such as chemical valence, isotopes, etc.

By viewing molecule structure as sequence data, many natural language processing technologies (e.g, Word2Vec [27], BERT [6]) could be adopted to model molecular sequence, which have demonstrated powerful capabilities in predicting different kinds of molecular properties [15, 47, 51, 52]. However, merely taking molecules with sophisticated internal connectivity as simple sequential data lacks sufficient interpretable and expressive capabilities. On the other hand, a chemical molecule can be naturally seen as a graph where nodes represent atoms and edges represent interatomic connectivities. Molecular graph brings two unique benefits comparing with *SMILES* sequence:

- Molecular graph can capture the spatial connectivity of different atoms, especially for star structure and ring structure (e.g., alkyl and benzene ring).
- Chemical molecular bonds are well preserved, which might have influence on the molecular properties. For instance, carbon dioxide has divalent bonds between carbon and oxygen.

By viewing molecule data as graph data, recent deep learning methods for graphs, such as graph neural networks [13, 44], can build effective components to learn molecular properties and have demonstrated considerable performance on different benchmark datasets [21, 37]. However, similar to sequence modeling in *SMILES*, simply using molecular graphs to model molecules cannot effectively learn molecular representations. Therefore, one research question is *can we integrate the capabilities of both molecular graph and SMILES sequence to further enhance molecule representation expressive power and improve performance for different downstream tasks?* In addition, the limited size of labeled molecule properties is another issue which restrains the prediction performance. Thus, another research question is *how can we put forward to obtain higher prediction results with structures of the chemical molecules themselves?*

In this work, we focus on addressing the above two research questions by leveraging both the graph and sequence information to learn effective molecular representations for different downstream tasks for drug discovery. We propose a graph and sequence fusion learning model, called **GraSeq**, to capture significant information from both *SMILES* sequence and molecular graph. In addition, our model employs a neural variational inference to reconstruct *SMILES* sequence as an additional downstream task, which helps learn better molecular representation and improve model performance on out-of-distribution generalization, besides using limited labeled data to train the model.

Contributions. The contributions and features of this work are summarized as follows:

- Proposed a graph and sequence fusion learning model, called **GraSeq**, to capture significant information from both *SMILES*

sequence and molecular graph, which could maximize the utilization of limited labels on known chemical properties.
- Integrated multitask loss functions in two parts for model training: unsupervised loss (for lacking labeled data) and supervised loss (for training different downstream tasks to maximize the utilization of limited labels).
- Demonstrated extensive comparisons with various baselines including different graph-based models and sequence-based models. Our **GraSeq** can outperform baselines with significant improvement (average +2.5% on AUC) in predicting a wide range of chemical properties on six benchmark datasets.

2 RELATED WORK

In this section, we review existing work related to our study, which mainly includes molecular graph representation learning and molecular sequence representation learning. Researchers have started to investigate molecular representation learning methods for a long time. The early works mainly focused on developing hand-crafted features for molecular representations that can reflect structural similarities and biological activities of molecules including Extended-Connectivity Fingerprints [31], Coulomb Matrix [32], Symmetry Function [2], and Bag-of-Bonds [10]. These methods heavily depend on hand-crafted features, which lack generalizability and scalability.

Molecular Graph Representation Learning. With the rapid development of deep learning algorithms, graph neural networks have gained a lot of attention for learning molecular representations since they can learn appropriate molecular representations that are invariant to graph isomorphism in an end-to-end fashion [5, 8, 12, 24, 36, 37]. Shindo et al. [37] proposed gated graph recursive neural networks (GGRNet) by considering a molecule as a complete directed graph where each atom has three-dimensional coordinates, and update hidden vectors of atoms depending on the distances between them. Hu et al. [12] proposed four different self-supervised tasks to pretrain graph neural networks at both levels of nodes and entire graphs, which significantly improved performance on out-of-distribution generalization.

Molecular Sequence Representation Learning. Scholars in the field of natural language processing tried to leverage sequential models (e.g., RNN) to learn molecular representation [15, 42, 47, 51, 52]. Sabrina et al. [15] proposed Mol2Vec, which imitated Word2Vec with an unsupervised method initially trained with unlabeled molecules, subsequently used machine learning models for property predictions as supervised tasks. Inspired by neural machine translation models, Xu et al. [47] and Zhang et al. [51] proposed sequence to sequence fingerprint model by mapping the *SMILES* string to a fixed-sized vector and then translates it back to the original *SMILES* string. SMILES-BERT [42] was motivated by the recent natural language model BERT [6]. It first trained with unsupervised learning mechanism Masked *SMILES* Recovery on large scale unlabeled data, then fine-tuned with specific downstream tasks. Zheng et al. [52] proposed a novel approach with self-attentive Bi-LSTM to complete *SMILES* strings syntax analysis, which could achieve superior performance on multiple benchmark datasets.

All of the above works were based on a certain model of graph or sequence while none has fused them together. There have been some studies about graph and sequence fusion learning for different applications [4, 28, 49, 50]. For example, Chen et al. [4] proposed a novel reinforcement learning based Graph2Seq model for natural language generation which encodes the graph representation of a text passage and decodes a text sequence. Besides, some complicated network structures may also contain sequential information such as evolutionary graphs. Parejia et al. [28] proposed EvoGCN to model evolutionary graphs by leveraging GCN as a feature extractor of each stage and RNN for sequence learning from the extracted features (e.g., node embeddings). However, no existing work integrates both molecular graph and sequence representations learning approaches for molecular applications. In this paper, we propose a fusion model of molecular graph and sequence. This is the first work to apply the idea of graph and sequence fusion learning for molecular representation learning, and explores the internal complementarity of graph and sequence.

3 GraSeq FRAMEWORK

In this section, we first formally define the problem of molecular property prediction and introduce the *GraSeq* model. Then, we present model details including graph layer, sequence layer, fusion layer, and molecule reconstruction layer.

3.1 Problem Definition

Let $G = (\mathcal{V}, \mathcal{E})$ denote a molecular graph with node attributes X_v for $v \in \mathcal{V}$ and edge attributes e_{uv} for $(u, v) \in \mathcal{E}$. Concretely speaking, a node in a molecular graph represents a chemical atom, and an edge represents a chemical bond between two atoms. At the same time, a molecule could also be represented as a sequence S, in which each $s \in S$ is an atom associated with a specific node in graph G. Each item s can be mapped from a node v through a predefined function $\psi(\cdot)$, denoted as $s = \psi(v)$. Formally, the problem of molecular property prediction is defined as follow:

Problem 1. Molecular Property Prediction. Given a set of molecules $\mathcal{M} = \{M_i\}_{i=1}^{|\mathcal{M}|}$, where each molecule $M \in \mathcal{M}$ is an union of its molecular graph and molecular sequence, and their labels $\mathcal{Y} = \{y_i\}_{i=1}^{|\mathcal{M}|}$, the problem is to learn a molecular representation vector $\mathbf{h}_M$ for predicting label (property) of each M in specific downstream tasks, that is find a function $f : \mathcal{M} \to \mathcal{Y}$.

To solve this problem, we propose a GraSeq model consisting of four parts: first, it encodes molecular graph with graph neural networks; second, it passes embeddings obtained from graph layer into a sequence encoder in order to learn contextual information of molecules represented by sequence; third, it uses a fusion layer to combine the output from graph layer and sequence layer; finally, the joint model is trained by multiple tasks augmented with molecule reconstruction as a self-supervised task. In this section, we present each part in detail. The whole framework is shown in Figure 2.

3.2 Graph Encoder

Graph neural networks (GNNs) [44] leverage both graph connectivity and node/edge features to learn a representation vector $\mathbf{h}_v$ for each node $v \in \mathcal{V}$. In molecular graph, each node represents

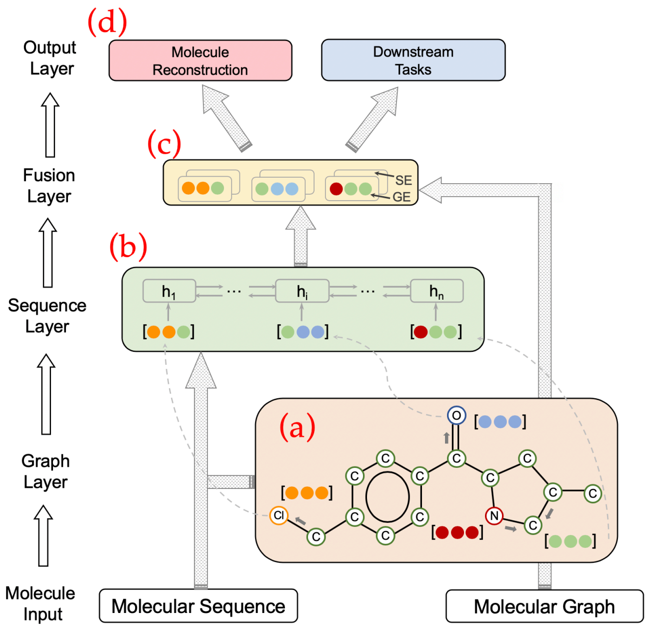


Figure 2: The GraSeq model consists of four parts: first, it encodes molecular graph with graph neural networks; second, it applies language model to encode molecule sequence; third, it uses a fusion layer to combine outputs graph embedding (denoted as GE) and sequence embedding (denoted as SE); finally, it takes multiple downstream tasks as supervision, also associates with the molecule reconstruction task.

an atom and each edge represents a chemical bond between two atoms. The edge type could be single, double, or triple associated with different weights $\mathbf{w}_{uv}$ according to the types of corresponding chemical bonds. As shown in Figure 2.(a), graph neural networks typically use neighborhood aggregation to iteratively update the representation of a node by aggregating representations of its neighboring nodes and edges. After k iterations, a node representation $\mathbf{h}_v^{(k)}$ is able to capture the structural information within its k-hop neighborhoods. Formally, the k-th layer of a node representation $\mathbf{h}_v^{(k)}$ obtained from a graph neural network is represented as:

$$\mathbf{h}_{\mathcal{N}(v)}^{(k)} = \text{AGGREGATE}_k(\mathbf{w}_{uv} \cdot \mathbf{h}_u^{(k-1)}, \forall u \in \mathcal{N}(v)), \tag{1}$$

$$\mathbf{h}_v^{(k)} = \sigma(\mathbf{W}^{(k)} \cdot \text{CONCAT}(\mathbf{h}_v^{(k-1)}, \mathbf{h}_{\mathcal{N}(v)}^{(k)})). \tag{2}$$

Note that we initialize $\mathbf{h}_v^{(0)} = \mathbf{X}_v$, and $\mathcal{N}(v)$ is a set neighbors of node v, and $\sigma(\cdot)$ is a non-linear activation function (e.g., LeakyReLU). Therefore, we can learn the representation of each node in molecular graph through graph encoder, i.e., $\mathbf{h}_v = \mathbf{h}_v^{(k)}/||\mathbf{h}_v^{(k)}||_2$.

3.3 Sequence Encoder

By leveraging the correspondence between *SMILES* strings and molecular graphs, we could represent each molecule as a sequence S. Each item in the sequence $s \in S$ is associated with a specific node in the molecular graph. Then we map the embedding obtained from graph encoder to the atom in the sequence through function $\psi(\cdot)$. The bi-directional long short term memory (Bi-LSTM) [11] is utilized to encode the representations of each node and learn contextual information of a molecule (shown in Figure 2.(b)):

$$\mathbf{h}_s = [\overleftarrow{\text{LSTM}}(\psi(\mathbf{h}_v)) \oplus \overrightarrow{\text{LSTM}}(\psi(\mathbf{h}_v))], \tag{3}$$

where $\mathbf{h}_s$ denotes the contextual representation of s, and $\oplus$ denotes vector concatenation operation.

3.4 Fusion Layer

In order to obtain the final output $\mathbf{h}_M$ of each molecule, we combine the output $\mathbf{h}_s$ from the sequence layer with the hidden state $\mathbf{h}_v$ from graph layer through a dimensional-wise fusion gate F. F is accomplished by the *sigmoid* activation function to encode two parts of representation (shown in Figure 2.(c)):

$$F = \text{sigmoid}(\mathbf{W}_1 \cdot \mathbf{h}_G + \mathbf{W}_2 \cdot \mathbf{h}_S) \tag{4}$$

where $\mathbf{h}_G$ and $\mathbf{h}_S$ are vector matrices of the whole graph and sequence by concatenating all $\mathbf{h}_v$ and $\mathbf{h}_s$, $\mathbf{W}_1$, $\mathbf{W}_2$ and b are trainable parameters of the fusion gate. Then the final vector representation output of a specific molecule $\mathbf{h}_m$ is generated though F:

$$\mathbf{h}_M = F \odot \mathbf{h}_G + (1 - F) \odot \mathbf{h}_S \tag{5}$$

3.5 Reconstruction Layer

The Seq2Seq model [41] is the most common strategy for obtaining robust sentence representations in natural language processing tasks as they can effectively leverage information from unlabeled data. Therefore, motivated by such idea, we further take molecular representation (obtained through graph encoder and sequence encoder) as the input of the decoder to enhance the molecular representation learning process through molecular reconstruction, which is shown in Figure 2(d). Specifically, in the reconstruction setup, the output of the decoder is reconstructed molecular sequence, denoted as $\hat{S} = \{\hat{s}_1, \hat{s}_2, \cdots \hat{s}_n\}$,

$$p(\hat{S}|S) = p(\hat{s}_{1:n}|s_{1:n}) = p(\hat{s}_1|\mathbf{h}_m) \prod_{t=2}^{T} p(\hat{w}_t|\mathbf{h}_m, \hat{w}_{1:t-1}). \tag{6}$$

Traditional Seq2Seq framework may lead to poor performance especially facing out-of-distribution generalization [14]. Therefore, we introduce Neural variational inference (NVI) [26] framework for sequence modeling. NVI can infer a stochastic latent variable $z \sim p(z|S)$, and construct an inference network to approximate the true posterior distribution $p(z|S)$. This strategy endows latent variable z with better ability for sequence reconstruction. Conditioning on the latent code z, a decoder network $q(\hat{S}|z)$ maps z back to reconstruct the original molecular sequence S. Since this process is intractable in most cases, a variational lower bound is typically employed as the objective to be maximized [18]:

$$\begin{aligned}\mathcal{L}_{vae} &= E_{q_\phi(z|S)}[\log p_\theta(\hat{S}|z)] - \text{D}_{\text{KL}}(q_\phi(z|S)|p(z)) \\ &= E_{q_\phi(z|S)}[\log p_\theta(\hat{S}|z) + \log p(z) - \log q_\phi(z|S)] \\ &\leq \log \int p_\theta(\hat{S}|z)p(z)dz = \log p_\theta(S),\end{aligned} \tag{7}$$

where θ and ϕ denote decoder and encoder parameters, respectively. The lower bound $\mathcal{L}_{vae}(\theta, \phi; S)$ is maximized w.r.t. both encoder and decoder parameters. In this work, we adopt another LSTM as decoder. At decoding time step t, the LSTM decoder reads the previous embedding $\mathbf{s}_{t-1}$ and context vector $\mathbf{c}_{t-1}$ to compute the new hidden state $\mathbf{s}_t$. We utilize the last encoder hidden state in $\mathbf{h}_m$ to initialize the decoder LSTM hidden state. The context vector $\mathbf{c}_t$ for time t is computed through attention mechanism [22]. Attention mechanism matches each $\mathbf{s}_t$ in decoder with each hidden state $\mathbf{h}_i$ in encoder to get an importance score. Formally,

$$\mathbf{s}_t = \overrightarrow{\mathrm{LSTM}}(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{c}_{t-1}), \tag{8}$$

$$e_{ti} = \mathbf{v}_a^\top \tanh(\mathbf{W}_s \cdot \mathbf{s}_{t-1} + \mathbf{W}_h \cdot \mathbf{h}_i), \tag{9}$$

$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{ti}\mathbf{h}_i, \text{ where } \alpha_{t,i} = \frac{\exp(e_{ti})}{\sum_{i=1}^{n}\exp(e_{ti})} \tag{10}$$

The hidden state passes through a *maxout* hidden layer to predict the next word (atom in this work) with a *softmax* layer over the decoder vocabulary (all atoms in this work):

$$p(\hat{s}_t|\hat{s}_1, \cdots, \hat{s}_{t-1}) = \mathrm{softmax}(\mathbf{W}_{out} \cdot \mathbf{s}_t + \mathbf{b}) \tag{11}$$

where $\mathbf{W}_s$ is trainable parameter and $\mathbf{s}_t$ is hidden state at step t.

3.6 Output Layer

At the last part, we formulate the optimization function (i.e., training loss) as two parts: label loss and reconstruction loss. Label loss is defined as negative likelihood of predicting correct labels of multiple downstream tasks $T \in \mathcal{T}$:

$$\mathcal{L}_{label} = -\sum_{M \in \mathcal{M}} \sum_{T \in \mathcal{T}} \mathrm{softmax}(\mathbf{W}_T \cdot \mathbf{h}_M + \mathbf{b}), \tag{12}$$

where $\mathbf{W}_T$ and $\mathbf{b}$ are learnable parameters, $\mathbf{h}_M$ is obtained from Eq.(5). Therefore, combining with reconstruction loss discussed in in Eq.(7), the overall loss could be written as:

$$\mathcal{L} = \mathcal{L}_{label} + \alpha\mathcal{L}_{vae}. \tag{13}$$

where α is a trade-off factor to control the importance of each task. The hyperparameter sensitivity analysis is shown in Section 5.6.

4 EXPERIMENTS

In this section, we first introduce datasets and experimental settings. Then we report extensive experimental results to demonstrate the effectiveness of our proposed model.

4.1 Datasets

We obtain datasets from two different resources, ZINC [39] and MoleculeNet [45]. ZINC is a public access database collecting molecular compounds for the virtual screening, which contains over twenty million available molecules. MoleculeNet is another large scale benchmark for molecular machine learning.

Classification Datasets. We obtain 806,991 molecules that are reactive from ZINC, where we extract two downstream datasets: LopP and FDA benchmarks. We also utilize four binary classification datasets contained in MoleculeNet. The statistic summary of datasets is in Table 1. The properties of these six datasets are as follows:

- **LogP**: Solubility of molecules. Molecules whose LogP >= 1.88 are positive, otherwise negative.
- **FDA**: Approved drug compounds by FDA.
- **BBBP** [25]: Blood-brain barrier permeability.
- **BACE** [40]: Quantitative (IC50) and qualitative (binary label) binding results for a set of inhibitors of human β-secretase 1(BACE-1).
- **Tox21**[1]: Toxicity on 12 biological targets, including nuclear receptors and stress response pathways.
- **ToxCast** [29]: Toxicity experiments on over 600 tasks for compounds based on in vitro high-throughput screening.

Dataset Split. All molecule datasets are split into training, validation, and testing collections by following 8/1/1 ratio. As mentioned in MoleculeNet [45], we should apply different splitting methods to effectively evaluate machine learning methods depending on datasets' contents. For several datasets, scaffold splitting is a better choice than random splitting, which splits the datasets according to molecules' two-dimensional structural frameworks. It tries to separate structurally different molecules into different collections, which offers challenges for learning methods. We split the datasets using these two different split methods followed by the recommendations of MoleculeNet [45], as shown in Table 1.

Dataset Processing. The initial molecular representations of our datasets are SMILES strings. We utilize Rdkit.Chem [20] to transfer these SMILES strings to molecular graphs. We can now identify multiple types of bonds, such as single, double, and triple. In our experiments, we add two edges between two nodes if the bond connecting them is double. Add three edges if it is triple.

4.2 Baselines

We compare our method with multiple baseline methods including various graph learning and sequence learning methods for molecular property prediction:

- **GraphSAGE [9].** It generates the nodes' embedding by sampling and aggregating their neighbors' embedding, which can effectively capture the graph information. In our model, we select this model as graph encoder.
- **GCN [19].** It is a widely used graph-based model, which contains an effective convolutional neural network component. GCN outperforms various models by learning both local graph structure and features of nodes.
- **Seq2SeqFP [47].** Seq2Seq fingerprint is base on Seq2Seq model, which is one typical NLP model. It uses unsupervised methods to learn molecular representations.
- **Mol2vec [15].** Mol2vec learns vector representations of molecular structures by Word2Vec [27]. Similar molecular structures have similar vector representations. The vector representation of the compound can be obtained by combining the vectors of its molecular substructures.
- **Seq3SeqFP [51].** It is based on Seq2Seq model. It defines a loss function which contains both self-recovery loss and inference task loss. This model is similar with sequence encoder part in our method.

[1]https://tripod.nih.gov/tox21/ challenge/

Dataset	LogP	FDA	BACE	BBBP	Tox21	ToxCast
Split Method	Random	Random	Scaffold	Scaffold	Random	Random
# Ins.	10000	3230	1513	2037	7831	8575
# Tasks	1	1	1	1	12	617
# Train	8000	2584	1210	1639	6264	6877
# Dev.	1000	323	151	205	783	860
# Test	1000	323	152	193	784	860

Table 1: The statistical details of six datasets used in our paper. LogP and FDA are obtained from ZINC[39]. BACE, BBBP, Tox21 and ToxCast are obtained from MoleculeNet[45]. We split the datasets into train, development, and test collections following an 8/1/1 ratio using random or scaffold splitting methods.

- **GIN [46]** It is the Graph Isomorphism Network, which has been proved with high representational performance.
- **SMILES-BERT [42].** SMILES-BERT puts forward a semi-supervised model which contains attention mechanism. It utilizes the transformer layer and obtains state-of-art performance on several ZINC datasets.
- **PreGNN [12].** It develops self-supervised method to pre-train GNN, which makes GNN learn both useful local and global representations. It achieves state-of-art performance on molecule property prediction.

4.3 Competitive Methods

Besides comparing with baseline methods, we also implement the model variants (ablation studies) to show the effectiveness our proposed model.

- **GNN.** It is GraphSage trained by downstream tasks loss.
- **Seq2seq.** It is Seq2Seq model trained by downstream tasks loss.
- **SeqRec.** It is Seq2Seq model trained by both downstream tasks loss and reconstruction loss of variational auto-encoder model.
- **GraSeq.** It is our proposed model, which consists of three parts: graph encoder, sequence encoder and output layer.
- **GraSeq-R.** It is based on GraSeq, and adds molecule reconstruction as a self-supervised task.
- **GraSeq-F.** It is based on GraSeq, and adds fusion layer combining the output of graph encoder and sequence encoder.
- **GraSeq-RF.** It is based on GraSeq, and adds both molecule reconstruction layer and fusion layer.

For fair comparisons with different methods, we use a supervised output layer for each experiment. The experiments with reconstruction layers will be trained by both reconstruction loss and downstream task loss.

4.4 Experimental Settings

We initialize each atom with a 64-dim random embedding vector. The same type of atom has the same initial embedding. The output layer dimension of GraphSage is 64. The numbers of hidden units in GRU encoder are all set as 64. All decoders are multi-layer perceptions (MLP) with one 64 units hidden layer. The latent embedding size is 64. The model is trained for 100 epochs by Adam optimizer [17]. For the KL-divergence, we use a KL cost annealing scheme [3], which serves the purpose of letting the VAE learn useful representations before they are smoothed out. We increase the weight β of the KL-divergence by a rate of $2/epochs$ per epoch until it reaches 1. We set the learning rate as 0.0005, and implement the model using Pytorch.

Dataset	**BACE**	**BBBP**	**Tox21**	**ToxCast**
Sequence-based Models				
Seq3seqFP (2018) [51]	77.25	90.73	79.76	71.07
Mol2Vec (2018) [15]	81.37	85.05	74.97	66.78
Graph-based Models				
GraphSage (2017) [9]	68.32	86.62	72.60	62.26
GCN (2017) [19]	70.03	81.23	73.20	68.39
GIN (2019) [46]	74.20	80.36	73.04	64.27
PreGNN (2020) [12]	**84.50**	68.70	78.10	65.70
GraSeq-Best	83.82	**94.26**	**81.95**	**73.30**

Table 2: Comparison of prediction results (AUC) between GraSeq and baseline models on 4 MoleculeNet [45] datasets.

4.5 Evaluation Metrics

First, as four datasets (BBBP, BACE, Tox21, ToxCast) obtained from MoleculeNet [45] are recommended to be evaluated by ROC-AUC. So we calculate the *Area under the Receiver Operating Characteristic Curve (AUC)* for evaluation in almost all the experiments. Moreover, we plot the *Receiver operating characteristic (ROC)* curves. Second, several previous work also use *Accuracy* to evaluate performances on two ZINC datasets. Therefore, we also take *Accuracy* as an evaluation metric for our model. Third, since it is a standard multi-class classification task and we also report *Micro F1* in the experiments.

4.6 Experimental Results

Comparison with Baselines. As shown in Table 2, we can observe that the GraSeq model with different variants offers a significant improvement over baseline models on the four datasets from MoleculeNet [45]. Comparing with best baseline methods, our model could improve AUC scores by **+3.53%**, **+2.19%** and **+2.23%** on BBBP, Tox21 and ToxCast datasets, respectively. The only exception is BACE dataset, on which PreGNN performs the best.

Methods	Accuracy
Seq2seqFP (2017) [47]	76.82
Seq3seqFP (2018) [51]	89.72
SMILES-BERT (2019) [42]	91.54
GraSeq-Best (Ours)	**94.70**

Table 3: Comparison of Accuracy between GraSeq and state-of-art sequence models on LogP dataset. We only report Accuracy mentioned in their paper because they did not have released code or runnable code.

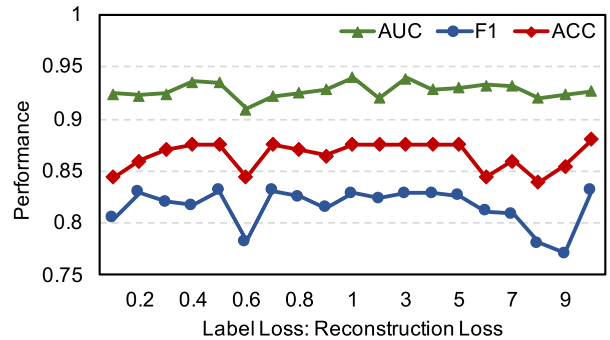


Figure 3: Our proposed GraSeq-R method is insensitive to the ratio between label loss and reconstruction loss in terms of AUC, F1, and Accuracy on BBBP dataset.

PreGNN takes multi pre-training strategies to incorporate both local (node-level) and global (graph-level) knowledge. It usually outperforms other graph-based models. But this performance gain requires costly pre-training. Additionally, as PreGNN fails to utilize sequence information as our model, its performance on all the other datasets are lower than ours.

We further compare GraSeq model with state-of-the-art sequence based molecular representation learning models, which is shown in Table 3. Since they have not released codes or their released codes are not runnable, we directly reported performances in their published papers. We can observe that our model could outperform all baseline models on LogP dataset, and improve **+3.16%** on Accuracy comparing with the state-of-the-art method SMILES-BERT.

Overall, we find that the sequence-based models perform better than the graph-based ones. This may due to the descriptive power of input representations (sequences vs. graphs). The graph-based models (i.e., GraphSage, GCN, GIN, and PreGNN) take molecular graphs as input, while the sequence-based models (i.e., Mol2Vec, and Seq3SeqFP) take sequences as input. Our experiments show that the sequences may be better for learning in general, but neither of these two kinds of models can effectively use information from both the graph and sequence representations like ours. We will further discuss the impact of input representation in Section 4.7.

Analyzing the effectiveness of fusion learning. Here we compare the effectiveness by choosing different variants in our fusion strategies. The AUC results are shown in Table 4. We can observe that our proposed GraSeq models demonstrate better performance comparing with graph-based model, sequence-based model, and sequence reconstruction model. Our best results of GraSeq-series models can outperform best single-input model by improving **+5.19%** on BACE, **+2.85%** on BBBP, **+1.46%** on Tox21, **+2.23%** on Toxcast, +0.58% on LogP, and **+1.66%** on FDA, respectively. Thus, fusing graph-based model and sequence-based model can effectively utilize two types of molecular representations and capture significant information from them. By using GraphSage, we update each atom by sampling atoms neighbors and aggregating their information iteratively. Furthermore, we use the sequence to sequence model to learn contextual information of a molecular sequence, which is better for learning chemical properties from atomic groups and electronics offsets. As reported in the Table 4, the molecular features extracted from molecular graphs and molecular sequences are complementary combined by our proposed method.

Comparing four variants of our proposed methods, GraSeq-F performs best on four datasets (BACE, BBBP, Toxcast and FDA). GraSeq and GraSeq-RF perform best on Tox21 and LogP, separately. Comparing with GraSeq, GraSeq-F could improve AUC by **+4.83%**, **+1.79%** and +0.80% on BACE, BBBP and Toxcast dataset. This is because GraSeq-F adds a fusion layer on the top of sequence encoder as shown in Figure 2(c), which can be seen as a type of boosting or residual learning that allows the sequence encoder to compensate on what the graph encoder fails to learn (e.g. isotopes). As we know, reconstruction loss of variational autoencoder is able to improve the stability by minimizing a variational lower bound. However, with limited training datas, molecular reconstruction is hard to be learnt effectively, leading to the model paying less attention on downstream task learning. Therefore, we can observe that GraSeq-R and GraSeq-RF does not improve model performance comparing with GraSeq-F except on Tox21 and LogP datasets.

Performance on different datasets. As mentioned in Section 1, one challenge of molecular properties prediction is the limited size of labeled data. To solve this problem, we design GraSeq to capture information both from molecular graphs and molecular sequences to maximize the utilization of existing datasets. We also define a multitask loss function containing both reconstruction loss and downstream task loss. It is obvious that downstream task loss is more important for chemical molecular prediction tasks since supervised loss is directly to the point than unsupervised loss (through reconstruction). So the lower missing label rate always leads to the higher prediction performance. For instance, all of molecules in BACE, BBBP, LogP and FDA datasets have labels but ToxCast has a 71% label missing rate and Tox21 has a 17.05% label missing rate. The prediction performance on Tox21 and Toxcast are lower than other four datasets as shown in Table 4, although the size of Tox21 and ToxCast is larger than FDA, BACE, and BBBP. Dataset size usually comes into effect when the missing label rate is similar. The prediction performance using the four data sets without any missing label (i.e., BACE, BBBP, LogP, and FDA) is positively correlated with their size. The amount of data probably contributes to reduced reconstruction loss.

	Model Methods Selection				MoleculeNet				ZINC	
	GE	SE	FL	RL	BACE	BBBP	Tox21	Toxcast	LogP	FDA
GraphSage	✔				68.32	86.62	72.60	62.26	97.83	97.03
Seq2seq		✔			77.25	90.73	79.76	71.07	98.21	97.44
SeqRec		✔		✔	78.63	91.41	80.49	71.04	98.16	97.24
GraSeq	✔	✔			78.99	92.47	81.10	72.50	98.79	98.58
GraSeq-R	✔	✔		✔	82.47	92.35	**81.95**	72.28	98.78	98.32
GraSeq-F	✔	✔	✔		**83.82**	**94.26**	81.37	**73.30**	98.57	**99.10**
GraSeq-RF	✔	✔	✔	✔	81.21	91.36	81.53	72.45	**98.83**	98.38

Table 4: We compare AUC of different fusion selections (GraSeq, GraSeq-R, GraSeq-F, GraSeq-RF) and single representation based models (GraphSage, Seq2seq, SeqRec) on 6 different downstream task datasets. Our GraSeq series models outperform Graph-based model and Sequence-based model by a significant margin. Graph-F achieves best performance at most time.

Parameter insensitivity analysis. Figure 3 shows the accuracy, F1 score, and AUC changing trend of GraSeq-R on BBBP datasets, where the ratio between label loss and reconstruction loss varies from 0.1 to 10. We observe that our proposed GraSeq-R method is insensitive. It always performs better than any of the baseline methods (AUC is always higher than 0.9).

4.7 Case Study

t-SNE visualization of embedding. To qualitatively validate the effectiveness of our approach, we visualize the embedding of our GraSeq-F with that of graph model and sequence model using t-SNE [23]. Figure 4 shows the embedding visualizations, where the blue dots denote true positive labels, the green cross symbols represent false negative labels, the orange dots are true negative labels and red cross symbols are false positive labels in BBBP dataset. Note that we flip the visualization coordinates (which does not change the embedding space) for better comparison.

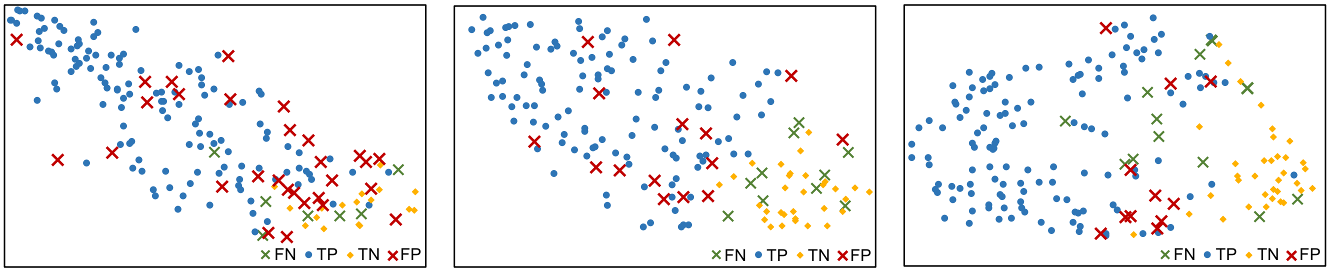
We can find that our GraSeq-F best separates the BBBP data points from the non-BBBP ones. First, GrapSeq-F is less likely to misclassify BBBP data points. In the visualization of GraSeq-F, we do not find any non-BBBP data points on the left side; while in the visualization of GNN and Seq2seq, several non-BBBP data points reside on the left side, mixing with the BBBP ones. The non-BBBP data points on the left are all misclassified as BBBP by GNN and Seq2seq, as shown by the red cross symbols in Figure 4(a) and (b). It indicates higher false positive rates of these models. Second, it is easy to know that our GraSeq-F is more likely to identify the non-BBBP data points correctly, as there are fewer BBBP points located at the bottom-right corner where most non-BBBP points reside. In contrast, for GNN model, many BBBP points appear in the bottom-right corner mixing with the non-BBBP ones. Figure 4(a) shows that only a few non-BBBP points are identified correctly (as illustrated by a few orange dots but various red cross symbols). This indicates a low true negative rate but a high false positive rate of GNN. Seq2Seq model has similar performance to our GraSeq-F in this sense, as similar numbers of orange dots (indicating true negative samples) are found in Figure 4(b) and (c). Finally, we find that the incorrect predictions (red and green crosses) by our GraSeq-F mostly gather on the border of the two classes while the incorrect predictions of GNN and Seq2Seq spread over the entire space. This also indicates that our GraSeq-F produces more desired embedding than the other methods. With a better separation hyperplane, the performance is further improved.

	G-S	S-G	G+S
Accuracy	88.9%	85.7%	44.4%

Table 5: The accuracy of our GraSeq for incorrect predicted molecules by GNN and Seq2seq. G-S, S-G and G+S represent incorrect predictions by GraphSage only, Seq2seq only, both GraphSage and Seq2seq.

Examination of GraSeq's effectiveness. We examine the effectiveness of our proposed model in complementing other models by providing the missing information, as shown in Table 5. By following set operations, we denote"G-S", "S-G", and "G+S" as incorrect predictions by GraphSage only, Seq2seq only, and both GraphSage and Seq2Seq, respectively. For the molecules that are predicted incorrectly by only one model, our model performs well (88.9% for "G-S" and 85% for "S-G"). It indicates that when information is missing in one type of representation (i.e., strings or graphs), our model can leverage information from the other type to achieve correct predictions. However, for the molecules predicted incorrectly by both the graph and sequence-based models ("G+S"), our model also does not perform well (with an accuracy of 44.4%). This indicates that when the critical information is captured by neither the graph-based nor the sequence-based models, our model is also insufficient to recover the missing information.

Analysis of relationships between model performance and data characteristics. We further examine the characteristics of molecules, which are more likely to be predicted incorrectly by graph-based models and sequence-based models respectively. We denote the molecules predicted incorrectly by GraphSage as "G" and those by Seq2seq as "S". By examining these incorrectly predicted samples, we find that graph-based models and sequence-based models are likely to miss certain types of information, due to the nature of the models. First, we can see that GraphSage performs poorly on



(a) Graph Model (b) Sequence Model (c) Our GraSeq Model

Figure 4: Embedding visualization of Graph model, Sequence model, and our Graseq model. We visualize the embedding space using t-SNE [23], where the scale of X-Y axis is relative without practical significance. The blue dots denote true positive (TP) labels in BBBP dataset, and the orange dots represent true negative (TN) labels in BBBP dataset. We highlights false positive (FP) predictions in red and false negative (FN) predictions in green. We observe that our GraSeq Method learns better molecular representations than Graph Model and Sequence Model.

molecules with long SMILES strings. The average SMILES string length in "G" (77.3) is longer than that in "S" (48.8), as shown in the left histogram in Figure 5. Then, we take 77.3 as a threshold for long SMILES string and calculate the prediction accuracy of GraphSage and Seq2Seq. We find the accuracy of GNN (39.2%) is much lower than that of Seq2seq (82.1%) for these molecules, as shown in the middle histogram of Figure 5. These observations indicate that long SMILES stings are unfriendly with graph-based models on property prediction tasks. Graph-based models do not capture the long-term dependencies without the well re-memorable unit. In contrast, Seq2Seq is based on LSTM, which can leverage the previously appeared information even when the SMILES string is long. Second, we find that graph-based models may miss information provided by the chemical symbols, which may be relate to specific properties. For example, we examine the molecules with the character "@" in their SMILES strings. The character "@" indicates atoms' chirality. We find that the percentage of molecules with "@" in "G" (72.2%) is much higher than that in "S" (28.5%), as shown in the right histogram. This indicates that the character "@" indeed has considerable influence on molecules' properties while graph-based models fail to capture this important information. Third, we find that the sequence-based models often miss information in two kinds of molecules. The first kind of molecules exhibits simple structures with short SMILES strings, such as "*O=C1C=CNN=C1*". It is difficult for Seq2Seq to learn useful function groups from this kind of short strings. The other kind of molecules are complex with many substructures, such as "*C3=C(N2CCN(CC(COC1=CC(=C(OC)C(=C1)OC)OC)O)CC2)C(=CC=C3)OC*". There are many parentheses in this SMILES string, indicating molecular branches. This kind of structures is difficult to be learned by Seq2seq from the strings, but relatively easy to be learned by GraphSage from the graph representations.

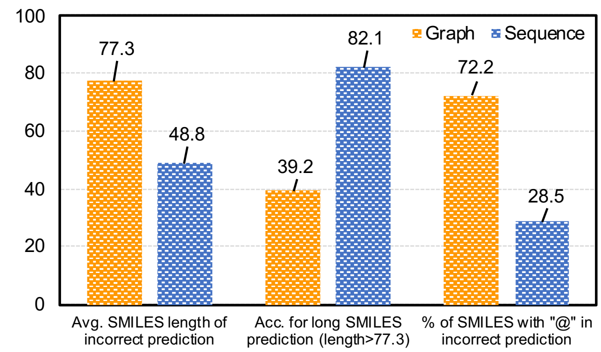


Figure 5: Statistical analysis of prediction results: 1) The length of incorrect predicted SMILES by Graph-based model is longer than sequence-based model; 2) Sequence model could make better predictions on relatively large (> 77 atoms) molecules; 3) Missing specific characters(such as "@") influences graph model prediction result.

5 CONCLUSIONS

In this work, we focused on leveraging both graphs and sequences to learn effective representations of molecules for different downstream molecular property prediction tasks. We propose a fusion model of graph and sequence, called **GraSeq**, to capture significant information from both *SMILES* string and molecule graph. Experimental results over 6 different downstream tasks datasets show our proposed GraSeq with different fusion selections could outperform various state-of-art methods by a significant margin. According to the case study, we can observe there are limitations when only taking single molecular representation (molecular graphs or SMILES strings) as input. GraSeq enables the information extracted from two different representations complementary rather than the opposite, which has been proved both in case study and ablation study.

ACKNOWLEDGEMENTS

We thank Dr. Olaf Wiest and John Herr for their constructive comments and suggestions. We also thank the anonymous referees for their valuable comments and helpful suggestions. This work was supported in part by NSF Grant xxx.

REFERENCES

[1] Mark Ashton, John Barnard, Florence Casset, Michael Charlton, Geoffrey Downs, Dominique Gorse, John Holliday, Roger Lahana, and Peter Willett. 2002. Identification of diverse database subsets using property-based and fragment-based molecular descriptions. *Quantitative Structure-Activity Relationships* 21, 6 (2002), 598–604.

[2] Jörg Behler and Michele Parrinello. 2007. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters* 98, 14 (2007), 146401.

[3] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*.

[4] Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2020. Reinforcement learning based graph-to-sequence model for natural question generation. *ICLR* (2020).

[5] Hyeoncheol Cho and Insung S Choi. 2018. Three-dimensionally embedded graph convolutional network (3dgcn) for molecule interpretation. *arXiv preprint arXiv:1811.09794* (2018).

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[7] Joseph A DiMasi, Henry G Grabowski, and Ronald W Hansen. 2016. Innovation in the pharmaceutical industry: new estimates of R&D costs. *Journal of health economics* 47 (2016), 20–33.

[8] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1263–1272.

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[10] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O Anatole Von Lilienfeld, Klaus-Robert Muller, and Alexandre Tkatchenko. 2015. Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space. *The journal of physical chemistry letters* 6, 12 (2015), 2326–2331.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[12] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2020. Pre-training graph neural networks. *International Conference on Learning Representation* (2020).

[13] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2020. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*.

[14] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1587–1596.

[15] Sabrina Jaeger, Simone Fulle, and Samo Turk. 2018. Mol2vec: unsupervised machine learning approach with chemical intuition. *Journal of chemical information and modeling* 58, 1 (2018), 27–35.

[16] William L Jorgensen. 2009. Efficient drug lead discovery and optimization. *Accounts of chemical research* 42, 6 (2009), 724–733.

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of International Conference for Learning Representation*.

[18] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. *International Conference on Learning Representation* (2014).

[19] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *International Conference for Learning Representation* (2017).

[20] Greg Landrum. 2013. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling.

[21] Ke Liu, Xiangyan Sun, Lei Jia, Jun Ma, Haoming Xing, Junqiu Wu, Hua Gao, Yax Sun, Florian Boulnois, and Jie Fan. 2019. Chemi-Net: a molecular graph convolutional network for accurate drug property prediction. *International journal of molecular sciences* 20, 14 (2019), 3389.

[22] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.

[23] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[24] Elman Mansimov, Omar Mahmood, Seokho Kang, and Kyunghyun Cho. 2019. Molecular geometry prediction using a deep generative graph neural network. *Scientific Reports* 9, 1 (2019), 1–13.

[25] Ines Filipa Martins, Ana L Teixeira, Luis Pinheiro, and Andre O Falcao. 2012. A Bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling* 52, 6 (2012), 1686–1697.

[26] Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*. 1727–1736.

[27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[28] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E Leisersen. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)* (2020).

[29] Ann M Richard, Richard S Judson, Keith A Houck, Christopher M Grulke, Patra Volarath, Inthirany Thillainadarajah, Chihae Yang, James Rathman, Matthew T Martin, John F Wambaugh, et al. 2016. ToxCast chemical landscape: paving the road to 21st century toxicology. *Chemical research in toxicology* 29, 8 (2016), 1225–1251.

[30] Sereina Riniker and Gregory A Landrum. 2013. Similarity maps-a visualization strategy for molecular fingerprints and machine-learning methods. *Journal of cheminformatics* 5, 1 (2013), 43.

[31] David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of chemical information and modeling* 50, 5 (2010), 742–754.

[32] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. 2012. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters* 108, 5 (2012), 058301.

[33] Gisbert Schneider. 2010. Virtual screening: an endless staircase? *Nature Reviews Drug Discovery* 9, 4 (2010), 273–276.

[34] Gisbert Schneider, Odile Clément-Chomienne, Laurence Hilfiger, Petra Schneider, Stefan Kirsch, Hans-Joachim Böhm, and Werner Neidhart. 2000. Virtual screening for bioactive molecules by evolutionary de novo design. *Angewandte Chemie International Edition* 39, 22 (2000), 4130–4133.

[35] Gisbert Schneider, Man-Ling Lee, Martin Stahl, and Petra Schneider. 2000. De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *Journal of computer-aided molecular design* 14, 5 (2000), 487–494.

[36] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*. 991–1001.

[37] Hiroyuki Shindo and Yuji Matsumoto. 2019. Gated Graph Recursive Neural Networks for Molecular Property Prediction. *arXiv preprint arXiv:1909.00259* (2019).

[38] Gregory Sliwoski, Sandeepkumar Kothiwale, Jens Meiler, and Edward W Lowe. 2014. Computational methods in drug discovery. *Pharmacological reviews* 66, 1 (2014), 334–395.

[39] Teague Sterling and John J Irwin. 2015. ZINC 15–ligand discovery for everyone. *Journal of chemical information and modeling* 55, 11 (2015), 2324–2337.

[40] Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. 2016. Computational modeling of β-secretase 1 (BACE-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling* 56, 10 (2016), 1936–1949.

[41] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[42] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. 2019. SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 429–436.

[43] David Weininger, Arthur Weininger, and Joseph L Weininger. 1989. SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of chemical information and computer sciences* 29, 2 (1989), 97–101.

[44] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[45] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* 9, 2 (2018), 513–530.

[46] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? *International Conference for Learning Representation* (2019).

[47] Zheng Xu, Sheng Wang, Feiyun Zhu, and Junzhou Huang. 2017. Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery. In *Proceedings of ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. 285–294.

[48] Xin Yang, Yifei Wang, Ryan Byrne, Gisbert Schneider, and Shengyong Yang. 2019. Concepts of artificial intelligence for computer-assisted drug discovery. *Chemical reviews* 119, 18 (2019), 10520–10594.

[49] Wenhao Yu, Mengxia Yu, Tong Zhao, and Meng Jiang. 2020. Identifying referential intention with heterogeneous contexts. In *Proceedings of The Web Conference 2020*. 962–972.

[50] Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2031–2043.

[51] Xiaoyu Zhang, Sheng Wang, Feiyun Zhu, Zheng Xu, Yuhong Wang, and Junzhou Huang. 2018. Seq3seq fingerprint: towards end-to-end semi-supervised deep drug discovery. In *Proceedings of ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. 404–413.

[52] Shuangjia Zheng, Xin Yan, Yuedong Yang, and Jun Xu. 2019. Identifying Structure–Property Relationships through SMILES Syntax Analysis with Self-Attention Mechanism. *Journal of chemical information and modeling* 59, 2 (2019), 914–923.