Predicting Molecular Toxicity under LUPI Approach

Xinyi Zhao, Haoyuan Cai

Abstract

In this project, we use Learning Under Privileged Information method to help improve the performance of molecules' toxicity prediction. Firstly we use molecules' gene expressions(L1000) as privileged information to improve a one-layer model based on ECFP. Then we introduce TOX21 data and implement graph convolution models GCN and GAT. Finally we apply the LUPI method to the two graph convolution models and examine the improvement of their performance and convergence rate. The results show that LUPI improves the performance of all these three models, and the improvement is more obvious when models become more complex. LUPI also decreases the number of epochs needed to converge, and reduces the chance of overfitting in the two graph-based models. Our results demonstrate that L1000 data has a close relationship to TOX21 data, and LUPI plays an important role in assisting molecules' toxicity prediction, and it helps even when only a small portion of privileged data is available.

Keywords: Learning Under Privileged Information; Toxicity Prediction; Graph Convolution; Machine Learning

1 Introduction

Predicting toxicity is an important task in many fields, especially in drug discovery. Since more and more datasets become available, the abundance of data makes machine learning a useful way to predict molecules' toxicity. Given the SMILES or the structures of the molecules, we can use different learning models to predict whether the molecules have side effects on some targets.

Previous works(conventional and graph-based models, in section 2) about toxicity prediction only use single type of data, i.e. molecule structure or gene changes. Intuitively, we can use one type of data to help another type of data train. However, the intersection between the data is sparse. And it brings many limitations to the methods that we just combine these two data.

In our work, we introduce a method LUPI (learning under privileged information) [1] raised by Vapnik. We treat the gene expression data as our privileged information. Instead of predicting the toxicity from the molecules structure, we use the gene expressions of these molecules to help increase the efficiency and accuracy of learning. We have applied LUPI into three molecule-predicting models, and the accuracy and efficiency both get improved.

The related works are listed in the next section. In the third section of the paper, we will introduce the basic methods and how we collect data. In the fourth section, we will explicitly introduce our experiments and analyze our results. And in the final section, we will make conclusions and introduce some future plans for our work.

2 Related Works

Deepchem [2] has provided a baseline for most of the toxicity prediction problems. It has covered conventional models like regression, random forests, etc, and graph-based models like weave, MPNN, GCN, GAT, and so on.

Besides, there are some works about predicting the property of molecules by using chemical-induced gene expressions in Cultured Human cells [3]. It used dataset from LINCS [4], which includes the changes of 1000 genes after applying one drug.

Meanwhile, there also exist some works using other properties rather than L1000 gene expressions as privileged information [5], but the effects of them are still not satisfactory. We are going to improve the prediction results further, by treating L1000 data as privileged information and applying graph-based models.

3 Data and Methods

3.1 Learning under privileged information

The growing availability of data provides chances to obtain knowledge by using machine learning methods, but also requires us to design efficient algorithms to transfer knowledge from various data sources, even when they are unavailable during validation. Such data is called Privileged Information. [5]

Traditional machine learning method is: given i.i.d. data pairs (x_i, y_i), $x_i \in X$, $y_i \in \{0, 1\}$, we train a function f that approximates these points and minimizes a loss function L. Moreover, the feature space for training and validation are the same.

However, under the Learning Under Privileged Information method, the training examples are in this form:

$$(x_i, x_i^*, y_i), x_i \in X, x_i^* \in X^*, y_i \in \{0, 1\}$$

X^* is privileged information, which is only accessible during training. In validation, the feature space contains only X. Therefore, the core of LUPI method is to train two models for (X, y) and (X^*, y) together, and design a feasible loss function to transfer knowledge from X^* to X efficiently.

To meet this requirement, the loss function should be a weighted sum of these two parts:

1. original loss function suitable for data points in (X, y)
2. mutual information between these two models' prediction results

3.2 Methods for toxicity prediction

In our work, we use three methods to predict the toxicity. The methods include some naive ways(just one fully connected layer), and some advanced ways(GCN and GAT). The main difference between the easy method and graph-based methods is that graph-based methods also train the function of getting features of the graph while easy methods only use fixed representation of one single graph(dynamic hashing VS fixed hashing). In this section, we will explain them explicitly. And also the realizations of the graph-based methods come from DGL(deep graph library) [6].

3.2.1 Fully connected layer with ECFP

The state of the art in molecular fingerprints are extended-connectivity circular fingerprints(ECFP) [7]. It give one molecule a string consisting of 2048 0s and 1s. It is designed to capture molecular features relevant to molecular activity. While not designed for substructure searching, they are well suited to tasks related to predicting and gaining insight into drug activity. Some works have used this string to predict the toxicity of the small molecules. In the deeptox paper [8], the author use ECFP information to train the deep learning network.

Figure 1 shows how we construct an ECFP.[1] In our work, we use the 2048-length string as our input, and use one layer of fully connected layers to do the training. In this section, we want to show how much the LUPI approach influences on easy model.

[1]Figure 1 is from https://depth-first.com/articles/2019/01/11/extended-connectivity-fingerprints/

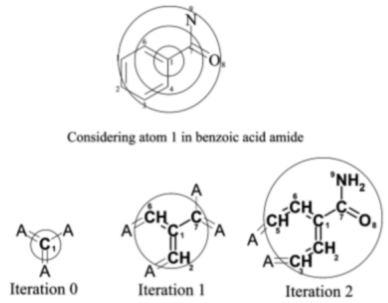

Figure 1: An illustration for the construction of ECFP

3.2.2 Graph Convolutional Network

Graph Convolution is a method to exploit the "local geometry" of a system to extract information from each node's neighbors in a graph. A general graph convolutional network(GCN) is to approximate a function $f(X, A)$, X is the nodes' features, A is the weighted adjacency matrix of the graph. We consider a multi-layer GCN with this layer propagation rule [9]:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{1}$$

Here $\tilde{A} = A + I$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, $H^{(l)} \in \mathbb{R}^{N\times D}$ is input from l^{th} layer, W^l is trainable weight matrix. The structure of GCN can be illustrated in Figure 2[2].

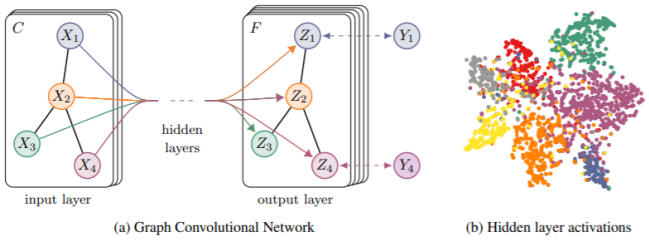

Figure 2: An illustration for GCN propagation rule and visualization for its hidden layer activations

In intuition, $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ changes each node's feature into a weighted sum of itself and its neighbors' feature. After several layers, the model has enough capability to figure out the local geometry from the original node features. Essentially, this form of propagation rule can be regarded as a first-order approximation of localized spectral filters on graphs[3].

3.2.3 Graph Attention Network

Graph Attention Network(GAT) is another method to obtain knowledge from a graph using different layer propagation rule [10]. The main difference of GCN and GAT is that GAT introduces "attention function" to represent the importance of each node's neighbors, which can be realized by one layer network in experiment. With the "attention function" a, GAT can characterize nonlinear relationship between each node and its neighbors when we merge their features, which is more powerful than multi layers of weighted sum and ReLU in GCN model.

Here we define $\mathbf{h} = \left\{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\right\}, \vec{h}_i \in \mathbb{R}^F$, a set of node features as a layer's input.

$\mathbf{h}' = \left\{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\right\}, \vec{h}'_i \in \mathbb{R}^F$ is the layer's output.

[2]This picture is from [9], Figure 1
[3]A rigorous proof is in [9].

Firstly we perform a trainable linear transform $\mathbf{W} \in \mathbb{R}^{F' \times F}$ to every node. Then we use a shared "attention function" $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \to \mathbb{R}$ to compute "attention coefficients":

$$e_{ij} = a\left(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j\right)$$

In intuition, e_{ij} indicates the importance of node j's feature to node i. Now we need to utilize the graph structure. We dropout all the e_{ij} that node pair (i,j) has no edge. Then we normalize e_{ij} using softmax function:

$$\alpha_{ij} = \text{softmax}_j\left(e_{ij}\right) = \frac{\exp\left(e_{ij}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(e_{ik}\right)}$$

Here $\mathcal{N}_i$ is neighborhood of node i in the graph. Then we output α_{ij} as this layer's features. The structure of GAT is shown in Figure 3.[4]

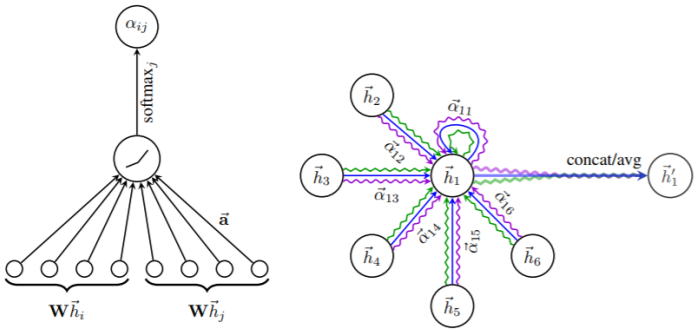

Figure 3: An illustration for Graph Attention Network and "Attention Function"

In our experiment, we use the molecular structure as the input graph, and apply these two graph convolutional methods to gain features from the graph structure.

3.3 Data collection

In our experiment, we mainly use the toxcity information from the dataset TOX21 [11]. This dataset is a challenge competition and consists of more than 8,000 molecules toxicity information on 12 targets. And our experiments are based on this dataset.

To collect L1000 gene expressions information from LINCS. We use the method introduced in the paper [12] and get the data from [13]. Every compound in the LINCS dataset has a lot of treatment records, including different reacting time and different dozes. But in our experiment, we only need one representation for one compound. We first pick up the higher-quality gold data. And then for each treatment record for one molecule, we calculate the correlation between this record and other records by comparing the similarity between them (replicates, doses, treatment durations, etc). And then use the correlation to calculate the weighted average of all the treatments, which can be written as

$$z \sim \frac{\sum_{i=1}^{k} w_i z_i}{\sqrt{\sum_{i=1}^{k} w_i^2}}$$

And after merging the data, we find that 887 molecules are both in TOX21 dataset and L1000 dataset. And we use the L1000 data of these molecules to serve as teachers and improve the performance of tox21 dataset prediction.

[4]This picture is from [10], Figure 1

4 Experiments

4.1 How to apply LUPI approach

We performed three experiments to verify the advantages of LUPI approach. Firstly we combined the naive ECFP model with gene expression data, and checked its performance at 12 targets(shown in section 4.2.1). Then we implemented two graph based models, GCN and GAT. We appended LUPI model to GCN and GAT respectively, and examined LUPI's improvement in terms of *auc_roc* score and acceleration in terms of epochs needed to reach convergence.

In the original ECFP model, we use BCE loss function to approximate the data points. We define y^i, p^i_{GC} are the probability distributions of our target result, the prediction output in graph convolution model, respectively. All of them are Bernoulli distributions. Then the original loss function in the left ECFP model is as follows:

$$L_0(y, p_{GC}) = \frac{1}{N} \sum_i H(y^i; p^i_{GC}) \tag{2}$$

To apply LUPI, we introduce molecules' gene expression(contained in L1000 data) and construct another model. When we introduced privileged knowledge, we need to modify the loss function to train the two models together and realize knowledge transfering. As is illustrated in section 3.1, we design the loss function as a weighted sum of two parts:

$$L(y, p_{GC}, p_{LUPI}) = \frac{1}{N} \sum_i H(y^i; p^i_{GC}) + \beta \exp(-\alpha H(p^i_{LUPI}, p^i_{GC})) * H(y^i; p^i_{LUPI}) \tag{3}$$

Here p^i_{LUPI} is the probability distribution of the prediction output in LUPI model. α, β are two hyper-parameters we need to set manually. $H(y^i; p^i_{GC})$ is BCE loss function in naive ECFP model. $H(y^i; p^i_{LUPI})$ is the performance of privileged knowledge, and $\exp(-\alpha H(p^i_{LUPI}; p^i_{GC}))$ is proportional to mutual information of these two models. We use the following picture to illustrate how we applyed LUPI to naive ECFP model.

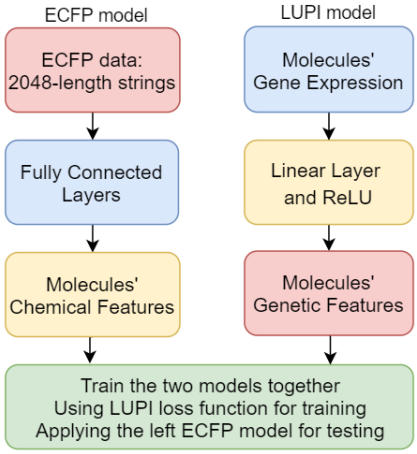

Figure 4: Applying LUPI to ECFP model

Then we implemented the two graph convolutional models, GCN and GAT, and use LUPI to seek a leap on its performance. The structure of GCN model is shown in Figure 5.

The structure of GAT model is similar to it, except that we replace the "Linear Layer and ReLU" by "Attention Function Layer".

In original model without privileged information, we do graph convolution from molecules' structural formula(contained in TOX21 data) to obtain molecules' chemical features and prediction results. In GCN

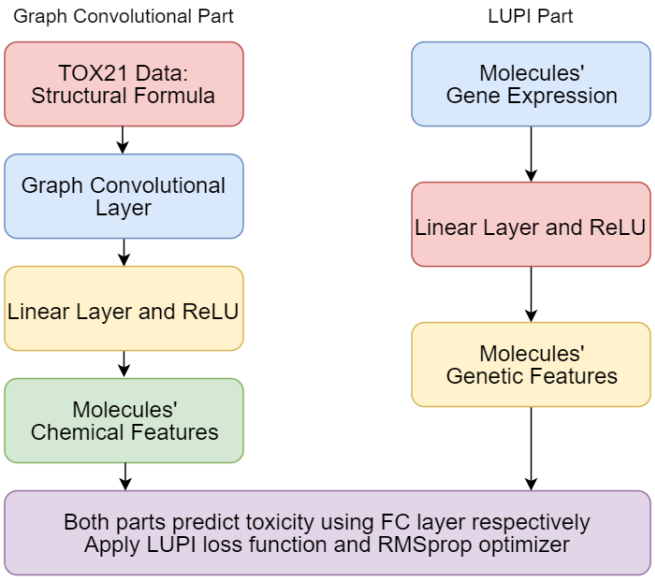

Figure 5: Applying LUPI to GCN model

graph convolution method, we use multi-layers of merging neighborhood's features to utilize "local geometry". In GAT graph convolution method, we focus on the "attention function": realized by the network layer illustrated in 3.2.3. Then we combine these two models together and we use above loss function Eq.(3) to help transfer knowledge. Also we can use multitask approach [14] to improve our performance.

4.2 Results

4.2.1 Easy Model Approach

When experimenting on ECFP information, we do two experiments. In the first experiment, we just use 887 molecules to train the model without privileged information and with the privileged information. Here is the result.

Table 1: Results using 800+ data

	accuracy without lupi	accuracy for lupi	roc_auc score without lupi	roc_auc score for lupi
NR-AR	0.7881445	0.773714	0.682375281	0.656012879
NR-AR-LBD	0.789476	0.786165	0.70441266	0.695990826
NR-AhR	0.6136275	0.6053225	0.67042735	0.65458922
NR-Aromatase	0.714269	0.706568	0.646566833	0.633857806
NR-ER	0.5914175	0.63158	0.571724429	0.586357664
NR-ER-LBD	0.70064	0.704744	0.623582517	0.639769161
NR-PPAR-gamma	0.698341	0.667024	0.592958682	0.648889828
SR-ARE	0.62366	0.622286	0.608373372	0.609506828
SR-ATAD5	0.668177	0.695747	0.53184529	0.550238357
SR-HSE	0.629801	0.655408	0.577077404	0.59068771
SR-MMP	0.549968	0.559684	0.653420184	0.641747884
SR-p53	0.625554	0.654967	0.599148466	0.618946298
Average	0.666089625	0.671934125	0.621826039	0.627216205

Then we try to use 4000+ data to train. And at this time, only 800+ data have the privileged information (teacher information).

Table 2: Results using 4000+ data

	accuracy without lupi	accuracy for lupi	roc_auc score without lupi	roc_auc score for lupi
NR-AR	0.852292891	0.837181131	0.700487334	0.698638901
NR-AR-LBD	0.841351725	0.836669259	0.747018309	0.788864648
NR-AhR	0.733029468	0.733029468	0.720085266	0.719975034
NR-Aromatase	0.829940273	0.844777824	0.701918643	0.720223979
NR-ER	0.706423339	0.701897614	0.600208563	0.626236045
NR-ER-LBD	0.818187089	0.796566684	0.680338364	0.686765924
NR-PPAR-gamma	0.914703443	0.910830295	0.693173751	0.694614817
SR-ARE	0.674545625	0.670533082	0.642220937	0.648785425
SR-ATAD5	0.862574545	0.867507532	0.678755879	0.67888636
SR-HSE	0.835439413	0.839016071	0.604098488	0.602556361
SR-MMP	0.675168242	0.676393195	0.766656663	0.765516206
SR-p53	0.826280795	0.826280795	0.678940518	0.674599538
Average	0.797494737	0.795056913	0.684491893	0.692138603

We can see that with the increasing of the data, the accuracy and roc-auc scores both improves without privileged information and with privileged information. But we can see that the effects of privileged information on both experiments are nearly same. That is to say, only part of the useful privileged data can influence the learning effects.

4.2.2 Graph-based Model

When working on graph-based Models(GCN and GAT), instead of training 12 targets one by one, we train the 12 targets together and calculate the overall training effects (The reason is that it can save time and different targets can share the information). In this process, we introduce a function called early-stopping to judge when we should stop training. If the validation roc-auc doesn't improve in 10 epochs, we will drop out of the train and finally train on the test data. And here is the results about GCN and GAT.

	GCN(no LUPI)	GCN(LUPI)	GAT(no LUPI)	GAT(LUPI)
Number of Epochs	161	149	55	38
Validation roc_auc	0.8082	0.8077	0.8211	0.8183
Test roc_auc	0.8098	0.8169	0.8227	0.8347

And we can see that privileged information not only increases the roc-auc score, but also reduces the number of epochs.

4.3 Analysis

We can see that privileged information improves the training whatever the portion of the overall data. As is shown in the three tables above, privileged information can improve the performance of our model in terms of test *roc_auc* score (about 0.01 on average in section 4.2.1 and 4.2.2). On some targets in section 4.2.1 (like NR-AR-LBD in Table 2), we can even improve up to 0.04. It reflects that gene expressions (privileged information) is closely related to molecules' toxicity on some of the targets.

And it can reduce the number of training epochs by providing more useful information among the training processes. Using early-stopping approach when executing our graph-based model, we can check how many epochs we need to make the test *roc_auc* score converge, which exactly means the knowledge in our privileged information is fully distilled. In section 4.2.2, we can see our LUPI can reduce the number of epochs by 17 in graph-based model, a significant acceleration for the training process. Note that privileged information in fact expands the feature space in our input, so if these gene expression data has little link with

toxicity results, the number of epochs needed to converge would increase a lot. Therefore the acceleration in LUPI method is a witness for the superiority of introducing a good "teacher" to help our machine learning.

Besides, LUPI approach can avoid overfitting problem. In section 4.2.2, we can see the result in validation *roc_auc* score even decreases by 0.002 in graph-based model. However, the test *roc_auc* score increases over 0.01 in those two models, a big improvement in test score compared with the falling back in validation. Therefore, although it doesn't work that well on validation dataset, it really does a good job on test dataset. We can demonstrate this phenomenon in intuition. As the privileged information expands the feature space, the difficulty for the graph convolutional model to "memorize" the data increases a lot due to the limited complexity of its parameters, thus rather than overfit training data, our model will try to figure out the connections between toxicity data and gene expression data, resulting in a little falling back in validation data but getting rid of overfitting.

Moreover, comparing the three models above (naive ECFP model in section 4.2.1, graph-based GCN, GAT in section 4.2.2), we find that both the performace and the increment of GAT are higher than GCN, and the two graph-based models outperform the naive ECFP model. It's because GAT includes an attention layer, which can learn the "importance" of each node's neighborhood. Also, we can guess that the privileged information may result in much more improvement on more complicated models.

5 Conclusion

Molecules' toxicity features are not only determined by its chemical features, but also related to their gene expression. LUPI plays an important role in exploiting the relationship between these two dataset, thus it brings about a notable improvement in all those three models' prediction results. LUPI also expands the feature space when training, and reduces the chance for the graph-based models to overfit. Also, the original TOX21 data is not complete and has several blank positions, in which case LUPI can serve as a supplement to help predict the lacked data. Moreover, LUPI can accelerate training speed when combined with early-stopping method, and reduces the number of epochs needed to converge. And we find that the more complex original model is, the better improvement LUPI may have.

When using privileged information, we improve the roc-auc score a little bit. But comparing with LUPI used in other fields, the effects of LUPI in toxicity prediction is not that notable. There exist some works using other properties rather than L1000 gene expressions as privileged information [5]. And the effects of that work are also not significant. The reasons behind this may come from the uncertainty and inaccuracy of the chemical data. For example, our genetic data can not differentiate information from distinct organs, which results from the high expense for genetic toxicity measures. Also, the intersection between TOX21 data and L1000 gene expression data is still too small, thus the potential of our LUPI method is still limited by the lacked data. Moreover, we can no longer ensure the quality of our collected gene expression data, because measuring molecules' gene expressions is expensive and error-prone, thus the noise of privileged data may be too intense for us to achieve a qualitative improvement.

To sum up, our work verifies L1000 data can serve as an assistance for molecules' toxicity prediction, and even only a small portion of privileged data can still make improvements on the training of the whole dataset, which may be affected by the size and the quality of privileged data. And if we can obtain more accurate data or can find more privileged data, the improvement may be much more prominent.

The code can be found on `https://github.com/xinyi-zhao/ADMET-under-LUPI-approach`

Acknowledgement: Thank FangPing Wan, Shuya Li and Jianyang Zeng for helping us do this work.

Abbreviation

AUC, area under the receiver operating characteristic curve;
ECFP, Extended Connectivity Fingerprints;
GAT, Graph Attention Network;
GCN, Graph Convolutional Network;
LINCS, Library of Integrated Network-based Cellular Sig- natures;
LUPI, Learning Under Privileged Information;
REACH, Registration, Evaluation, Authorization, and Restriction of Chemical Substances;
SMILES, Simplified molecular-input line-entry system.

References

[1] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: similarity control and knowledge transfer. *Journal of machine learning research*, 16(2023-2049):2, 2015.

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

[3] Ruifeng Liu, Xueping Yu, and Anders Wallqvist. Using chemical-induced gene expression in cultured human cells to predict chemical toxicity. *Chemical research in toxicology*, 29(11):1883–1893, 2016.

[4] `http://www.lincsproject.org/`.

[5] Niharika Gauraha, Lars Carlsson, and Ola Spjuth. Conformal prediction in learning under privileged information paradigm with applications in drug discovery. *arXiv preprint arXiv:1803.11136*, 2018.

[6] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J Smola, and Zheng Zhang. Deep graph library: Towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[7] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. PMID: 20426451.

[8] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. Deeptox: toxicity prediction using deep learning. *Frontiers in Environmental Science*, 3:80, 2016.

[9] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[10] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[11] `http://tripod.nih.gov/tox21/challenge`.

[12] Ruifeng Liu, Mohamed Diwan M AbdulHameed, and Anders Wallqvist. Molecular structure-based large-scale prediction of chemical-induced gene expression changes. *Journal of chemical information and modeling*, 57(9):2194–2202, 2017.

[13] `https://figshare.com/articles/L1000_Drug_level_Consensus_Expression_Profiles/1476293/2`.

[14] Fengyi Tang, Cao Xiao, Fei Wang, Jiayu Zhou, and H Lehman Li-wei. Retaining privileged information for multi-task learning. 2019.