# Predicting Drug-Drug Interactions using Deep Generative Models on Graphs

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Latent representations of drugs and their targets produced by contemporary graph autoencoder-based models have proved useful in predicting many types of node-pair interactions on large networks, including drug-drug, drug-target, and target-target interactions. However, most existing approaches model the node's latent spaces in which node distributions are rigid and disjoint; these limitations hinder the methods from generating new links among pairs of nodes. In this paper, we present the effectiveness of variational graph autoencoders (VGAE) in modeling latent node representations on multimodal networks. Our approach can produce flexible latent spaces for each node type of the multimodal graph; the embeddings are used later for predicting links among node pairs under different edge types. To further enhance the models' performance, we suggest a new method that concatenates Morgan fingerprints, which capture the molecular structures of each drug, with their latent embeddings before preceding them to the decoding stage for link prediction. Our proposed model shows competitive results on two multimodal networks: (1) a multi-graph consisting of drug and protein nodes, and (2) a multi-graph consisting of drug and cell line nodes.

## 1 Introduction

Investigations on biomedical multimodal graphs consisting of drugs, proteins, or cells and their interrelations are essential in drug repurposing as they can help discover innovative node-pair interactions, which may be helpful in medical treatments, on these networks. Most approaches to tackling drug repurposing problems estimate drug side effects and drug responses to cell lines to provide important information about the use of those drugs. In particular, polypharmacy is the practice of treating complex diseases or co-existing health conditions by using combinations of multiple medications [1]. Drug combinations consisting of many drugs affecting distinct targeted proteins can effectively modulate the process of severe diseases by facilitating the diverse biological process of different proteins [2]. Albeit commonly applied, polypharmacy is one of the major underlying issues that cause adverse medical outcomes, also known as side effects caused by drug combinations [3]. Understanding drug-drug interactions (DDIs) is pivotal in predicting the potential side effects of multiple co-administered medications. Moreover, accurate estimation of drug responses to targeted cell lines is another major factor in providing precision medicines to patients. However, it is unfeasible to render clinical testing of all drug combinations due to the tremendous number of relations between drug pairs or drugs and their targets (e.g., proteins).

Submitted to NeurIPS 2022 AI for Science Workshop.

Machine learning models have been widely applied to provide tractable solutions in predicting potential drug interactions. Data-driven techniques (e.g., machine learning or deep learning) have proved their capabilities for dealing with complex patterns in data. Such methods have produced remarkable results in various fields, such as computer vision, natural language processing, and audio processing. Therefore, it is reasonable to employ machine learning approaches to investigate the interactions among complex combinations of medications. Previous work on DDIs [4–6] used various hand-crafted drug features (e.g., chemical properties or Morgan fingerprints) to compute the similarity of each drug pair, and drug-drug interactions can be predicted based on the drug-pair proximity scores. Nevertheless, using fixed drug representations can result in sub-optimal outcomes since they do not sufficiently capture the complex interrelations of drugs.

Recent approaches rely on graph neural networks (GNNs) [7] to directly learn drug and protein representations from structural data (i.e. graphs) augmented with additional information for node features. Zitnik et al. [3] proposed a graph autoencoder to predict polypharmacy side effects on a multimodal graph consisting of drug and protein nodes with various edge types. Similarly, Yin et al. [8] introduced DeepDrug combining graph convolutional neural networks (GCN) [9] and convolutional neural networks (CNN)[10] to learn the structural and sequential representations of drugs and proteins. In addition, Wang et al. [11] suggested using several graph attention (GAT) [12] layers to learn on hand-crafted features of drugs to predict drug-pair interactions. Nyamabo et al. [13] use multiple GAT layers to compute the drug representations and aggregate them using co-attention to produce final predictions. These mentioned GNN-based models effectively learn the latent representations of drugs and proteins, which can be used in several downstream tasks.

Predicting drug-drug or drug-target interactions can be regarded as a link prediction task on graphs. We argue that latent spaces produced by the aforementioned approaches are disjoint and not interpretable; hence, they are not capable of generating new links on some graph benchmarks. In this paper, we suggest using deep graph generative models to make the latent spaces continuous and more interpretable; as a result, they are superior to traditional graph autoencoders (GAE) in predicting new links on biomedical multimodal graphs. Instead of yielding deterministic latent representations, variational graph autoencoders (VGAE) [14] use a probabilistic approach to compute the latent variables. In this work, we aim to use VGAE to learn the representations of drugs and proteins on a multimodal graph; then, we predict several node-pair interactions (e.g., drug-pair polypharmacy side effects, drug-cell line response, etc.) using the learned representations.

Our contributions are two-fold as follows:

- We demonstrate that using VGAE can attain superior performance in predicting drug-drug interactions on multimodal networks, outperforming GAE-based architectures and hand-crafted feature based methods.
- We leverage drug molecular representations (i.e. Morgan fingerprints) concatenated with drug latent representations before the decoding stage to further enhance the model's performance in predicting drug-pair polypharmacy side effects.

## 2 Related work

Link prediction is one of key problems in graph learning. In link-level tasks, the network consists of nodes and an incomplete set of edges between them, and partial information on the graph are used to infer the missing edges. The task has a wide range of variants and applications such as knowledge graph completion [15], link prediction on citation networks [16], and protein-protein interaction prediction [17].

Graph neural networks (GNNs) are deep learning models that learn to generate representations on graph-structured data. Most GNN-based approaches use a message passing paradigm proposed in [18] [19] wherein each node aggregates vectorized information from its neighbors and updates the messages to produce new node representations. Kipf and Welling [9] introduced a scalable method that approximates first-order spectral graph convolutions, which are equivalent to one-hop

neighborhood message passing neural networks. Besides, Veličković et al. [12] proposed graph attention networks (GATs) that use soft attention mechanism to weight the messages propagated by the neighbors. Hamilton et al. [20] introduced GraphSAGE that efficiently uses node attribute information to generate representations on large graphs in inductive settings.

Deep generative models aim to generate realistic samples, which should satisfy properties in nature. Different from traditional methods that rely on hand-crafted features, data-driven approaches can be categorized into variational autoencoders (VAEs) [21], generative adversarial networks (GANs) [22], and autoregressive models. In the field of deep generative models on graphs, several works ([23, 24] aim at adding stochasticity among latents to make the models more robust to complicated data distributions. Simonovsky and Komodakis [25] proposed a regularized graph VAE model wherein the decoder outputs probabilistic graphs. In addition, Bojchevski et al. [26] introduced a stochastic neural networks that are trained by Wassterstain GAN object to learn the distribution of random walks on graphs. For autoregressive methods, Liao et al. [27] proposed a attention-based GNN network to generate graph nodes and associated edges after many decision steps in a generation process.

The objective of our work is to apply a probabilistic framework in predicting missing links on biomedical multimodal networks. We extend varational graph autonecoders (VGAE) [14] to multimodal networks such that separate posterior latent distributions are computed for each node type after several message passing and aggregation layers among different node and edge types. Moreover, we also incorporate molecular structures into the latent variables of drug nodes to capture more inter-intra information across the medications, resulting in superior performance in drug-pair prediction.

# 3 Approach

## 3.1 Problem Setup

Given a multimodal graph, our task is to predict whether two nodes of the same or different types are connected or not under a specific edge type. This can be regarded as a binary classification problem on the graph's edges. In our work, the graph $G$ consists of two node types (e.g., drug and protein), and each side effect is represented as an edge type. In addition to polypharmacy interactions, $G$ also involves edges representing drug-protein and protein-protein interactions. Formally, let denote $G = (V, E, X)$, where $V = V_d \cup V_p$ is a union of two node sets of different types (i.e. $V_d$ is the set of drug nodes and $V_p$ is the set of protein nodes), $E$ is a set of edges, and $X = X_d \oplus X_p$ is a concatenated matrix denoting the node features of different node types. Each edge in $E$ is a triplet $(v_i, e, v_j)$ in which node $v_i$ interacts with node $v_j$ under a specific edge type $e$. As our task is in a binary classification setting, we perform negative sampling to generate non-edge labels to make the model robust to negative examples. Thus, the objective is to learn a function $f : E \times E \rightarrow \{0, 1\}$, where $f$ predicts the label of a particular triplet $(v_i, e, v_j)$.

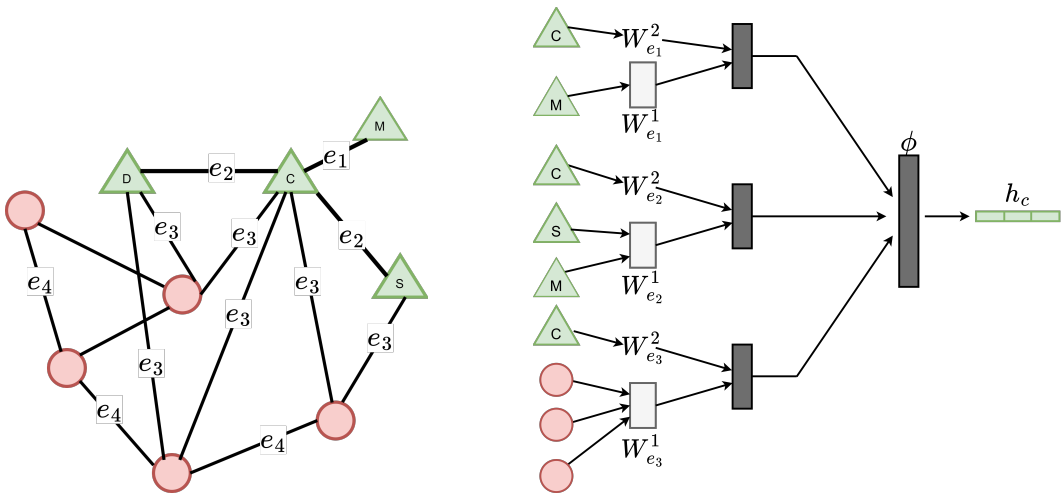
## 3.2 Model Architecture

Our proposed model has three main components: an encoder, a latent encoder, and a decoder. They are described as follows:

**Encoder** a two-layer graph convolutional neural network operating message passing scheme [19, 20] on $G$ and producing node embeddings for each node type (e.g., drugs and proteins). Each layer in the network has the following form:

$$h_i = \phi\Big(\sum_{e} \sum_{j \in \mathcal{N}_e^i} W_e^1 x_j + W_e^2 x_i\Big)$$

where $\mathcal{N}_e^i$ denotes the neighbor set of node $x_i$ under the edge type $e$. $W_e^1 \in \mathbb{R}^{d_k \times d_1}$, $W_e^2 \in \mathbb{R}^{d_k \times d_2}$ are two edge-type specific transformation matrices that map $x_i \in \mathbb{R}^{d_1}$ and its neighbors $x_j \in \mathbb{R}^{d_2}$ into $d_k$-dimensional vector spaces, resulting in $h_i \in \mathbb{R}^{d_k}$. It is worth noting that $d_1$ and $d_2$ are not necessarily the same because $x_i$ and $x_j$ can be nodes of two different node types. Also, $\phi$ denotes a nonlinear




(a) (b)

Figure 1: Overview of a biomedical multimodal graph and one graph convolution layer in our framework. (a) A graph consists of two node types (e.g., red nodes are protein and the green are drug nodes) and five edge types $\{e_i\}$. (b) A layer of the encoder in which the representation of the source node $c$ is computed by aggregating its neighbors' information under different edge types. Black rectangles denote aggregation , whereas white rectangles indicate neural networks that share parameters across the nodes. $\{(W^1_{e_i}, W^2_{e_i})\}$ denote trainable weight matrices of different edge types $\{e_i\}$. $h_c$ can be preceded to either a successive convolution layer or the latent encoder.

activation function; we use rectified linear unit (ReLU) [28] in our experiments. Figure 1 illustrates an overview of the encoder in our framework.

**Latent Encoder** For each node type $v$, there are two multilayer perceptrons (MLPs) receiving the node embeddings from the encoder and computing the predicted mean $\mu$ and logarithm of the standard deviation $\log \sigma$ of the posterior latent distribution:

$$q_v(Z_v|X,E)=\prod_{i=1}^{|V_v|} q_v(z_v^i|X,E)$$

$q_v(z_v^i|X,E)=\mathcal{N}(z_v^i|\mu_v^i, \text{diag}((\sigma_v^i)^2)$ denotes the posterior distribution of a node of a specific node type. Here, $\mu_v$ and $\log \sigma_v$ are computed as follows:

$$\mu_v=W^2_{\mu_v} \tanh(W^1_{\mu_v} h_v)$$

$$\log \sigma_v=W^2_{\sigma_v} \tanh(W^1_{\sigma_v} h_v)$$

where $W^i_{\mu_v} \in \mathbb{R}^{d_k \times d}$, $W^i_{\sigma_v} \in \mathbb{R}^{d_k \times d}$ are the weight matrices, $\mu_v$ and $\log \sigma_v$ are the matrices of mean vector $\mu_v^i$ and logarithm of standard deviation vector $\log \sigma_v^i$, respectively. Figure 2 demonstrates how we integrate molecular structures of drug nodes with their latent representation to improve the performance in side effect link prediction.

**Decoder** a tensor factorization using latent embeddings to predict the node interactions on $G$. We follow the approach proposed in [3] to design the decoder:

$$g(v_i, e, v_j) \begin{cases} z_i^T D_e R D_e z_j & \text{if } v_i \text{ and } v_j \text{ are drugs} \\ z_i^T M_e z_j & \text{if } v_i \text{ and } v_j \text{ are a protein and a drug, or vice versa.} \end{cases}$$

where $D_e, R, M_e \in \mathbb{R}^{d \times d}$ are learnable parameters. $R$ denotes the global matrix representing all drug-drug interactions among all polypharmacy side effects; $M_e$ is a edge-type-specific matrix modeling

drug-protein and protein-protein relations. Also, $D_e$ is a diagonal matrix, and its on-diagonal entries model the significance factors of $z_i$ and $z_j$ in multiple dimensions under the side effect type $e$. Finally, the probability of edge $(v_i, e, v_j)$ is calculated via a sigmoid function $\sigma$:

$$p_e(v_i, v_j) = \sigma(g(v_i, e, v_j))$$

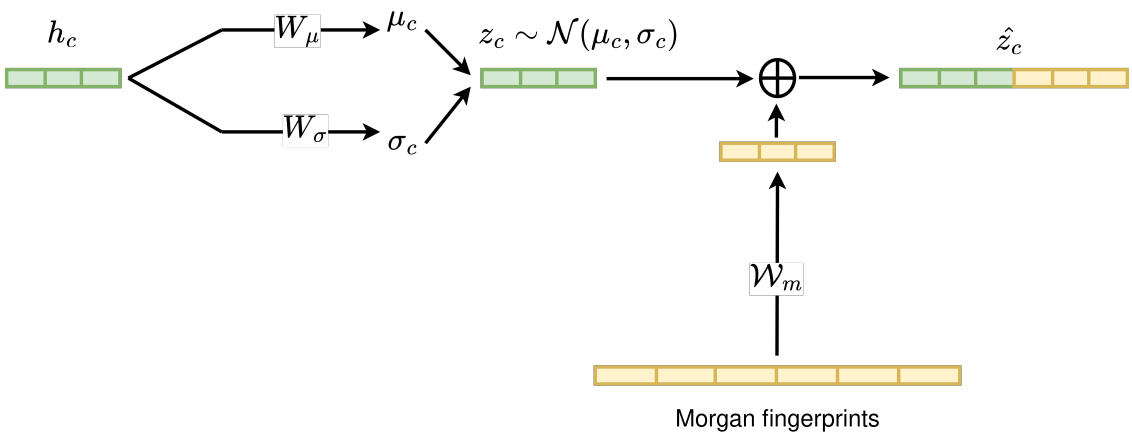



Figure 2: Overview of the latent encoder mentioned in 3.2. Morgan fingerprints are concatenated with latent vector $z_c$ to improve the performance of polypharmacy side effects prediction. $\{W_\mu, W_\sigma, W_m\}$ are trainable weight matrices.$\oplus$ denotes concatenation.

# 4 Experiments

Our implementation is done with the PyTorch Geometric library [29, 30]. The code base is developed based on the implementation of VGAE published in the library's repository.

## 4.1 Polypharmacy side effects

We conduct experiments on the dataset introduced in [3]. It is a multimodal graph consisting of 19,085 protein and 645 drug nodes. There are three types of interaction including drug-drug, drug-protein, and protein-protein. In particular, the dataset contains 964 commonly occurring polypharmacy side effects, resulting in 964 edge types in drug-drug interaction. Protein-protein and drug-protein interactions are regarded as two other edge types; therefore, the multimodal graph has 966 edge types in total. Additionally, total numbers of edges of each interaction are displayed in Sec.A of the Appendix.

We randomly split the edges into training, validation, and testing sets with a ratio of $8{:}1{:}1$. Then, the edges in the training edge set are randomly divided into $80\%$ edges for message passing and $20\%$ edges for supervision. We use Adam optimizer [31] to minimize the loss function $L$ showed in Equation 1 in 300 epochs with a learning rate of $0.001$. We run the experiments with six different random seeds.

$$L = \sum_{(v_i, e, v_j) \in E} -\log p_e(v_i, v_j) - \mathbb{E}_{v_n \sim P_e(v_i)}[1 - \log p_e(v_i, v_n)] - \sum_v \lambda_v \mathbb{D}_{KL}(q_v(Z_v|X, E)|p_v(Z_v)) \tag{1}$$

where $p_v(Z_v)$ denotes the latent space's prior distribution of the node type $v$. The first term of $L$ denotes the cross-entropy loss of the probabilities of positive and negative edges which is sampled by choosing a random node $v_n$ for each node $v_i$ under a specific edge type $e$, while the second term is the weighted sum of KL-divergence of each node type $v$; $\{\lambda_v\}$ are hyper-parameters. In our experiments,

$\lambda_d$ and $\lambda_p$ equal to $0.9$ and $0.9$ for drug and protein nodes, respectively. Finally, we use one-hot representations indicating node indices as features for both drug and protein nodes.

### 4.2 Anticancer drug response

In addition to drug-pair polypharmacy side effects, we evaluate our approach for the anticancer drug response prediction problem; the task is to predict the interactions between drug and cell line on a multimodal network. Integrated information between drugs and cell lines are an effective approach to calculate anticancer drug responses using computational methods; various type of information can be exploited such as drug chemical structures, gene expression profiles, etc., to provide more accurate predictions. In this work, we evaluate the performance of VGAE on the CCLE dataset proposed in [32] which contains comprehensive information of drugs and cell lines. Three types of pairwise similarities are provided for each modality, resulting in nine types of combination used in computation among drug and cell line pairs; in addition, pairwise logarithms of the half-maximal inhibitory concentration (IC50) scores between drugs and cell lines are given as values to be predicted. The combinations are detailed in Table 2.

To apply VGAE in predicting the IC50 scores among drug-cell line pairs, we construct an undirected multimodal graph consisting of drug and cell line nodes with the similarity scores indicating the edge weights under three types of relation, including drug-drug, cell line-cell line, and drug-cell line. The problem is regarded as a link prediction task in which an amount of edges connecting drug and cell line nodes are masked, and the objective is to compute real values $w_e$ denoting the weights of these missing links. The implementation of VGAE is almost the same as detailed in 3.2 with modifications of the decoding stage in which representations of two nodes are concatenated before being preceded to a simple multilayer perceptron with two hidden layers of size 16, ReLU nonlinearity, and a linear layer on top to predict a final score denoting the edge weights among them.

We randomly split the edges with a ratio 7:1:2 for training, validation, and testing purposes, respectively. In this task, the models are trained to reconstruct a weighted adjacency matrix and regularized by KL-divergence of the node latent distributions with hyperparameters $\lambda_d$ and $\lambda_c$ denoting the coefficients for drug and cell nodes, respectively; $\lambda_d$=$\lambda_c$=$0.001$ in our experiments. We train the models in 500 epochs with a learning rate of 0.01 to minimize the loss function $L$ as follows:

$$L = \sum_{(v_i,e,v_j)\in E} (\widehat{s_e}(v_i,v_j) - s_e(v_i,v_j))^2 - \sum_v \lambda_v \mathbb{D}_{KL}(q_v(Z_v|X,E)|p_v(Z_v)) \tag{2}$$

where $\widehat{s_e}(v_i,v_j)$ indicates the predicted edge weights between node $v_i$ and $v_j$, whereas $s_e(v_i,v_j)$ are their ground truths. Finally, experiments are run with 20 different random seeds.

## 5 Results

### 5.1 Polypharmacy side effects prediction

Table 1: Average performance on polypharmacy link prediction across all side effects

| Method | AUROC | AUPRC | AP@50 |
|---|---|---|---|
| RESCAL tensor factorization | 0.693 | 0.613 | 0.476 |
| DEDICOM tensor factorization | 0.705 | 0.637 | 567 |
| DeepWalk neural embeddings | 0.761 | 0.737 | 0.658 |
| Concatenated drug features | 0.793 | 0.764 | 0.712 |
| Decagon | 0.872 | 0.832 | 0.803 |
| GAE | $0.843 \pm 0.006$ | $0.817 \pm 0.006$ | $0.783 \pm 0.012$ |
| VGAE (ours) | $0.855 \pm 0.006$ | $0.834 \pm 0.005$ | $0.835 \pm 0.011$ |
| VGAE + Morgan fingerprints (ours) | $\mathbf{0.891 \pm 0.007}$ | $\mathbf{0.872 \pm 0.006}$ | $\mathbf{0.887 \pm 0.004}$ |

We compare the performance of VGAE to alternative approaches. In addition to VGAE, we also implement a graph autoencoder and train the model in the same setting detailed in Sec. 4. The baseline results are taken from [3] including: RESCAL Tensor Factorization [33], DEDICOM Tensor Factorization [34], DeepWalk Neural Embeddings [35, 36], Concatenated Drug Features, and Decagon [3]. It is worth noting that Decagon is also a GAE-based model with two GCN layers, yet Zitnik et al. [3] use side information (e.g., side effects of individual drugs) as additional features for drug nodes. By constrast, our VGAE and GAE are trained on one-hot representations for drug and protein nodes. To examine the effects of Morgan fingerprints in drug-drug interaction decoding, we augment the Morgan fingerprint information into the latent vectors of drug nodes before preceding them to tensor-based decoders to produce final predictions.

Table 1 shows the results of all approaches in predicting the polypharmacy side effects. The models are evaluated based on three metrics, including area under ROC curve (AUROC), area under precision-recall curve (AUPRC), and average precision at 50 (AP@50). The scores reveal that VGAE without augmenting Morgan fingerprints outperforms traditional tensor-based approaches by a large margin, resulting in up to $23.4\%$ (AUROC), $36.1\%$ (AUPRC), and $75.4\%$ (AP@50). We also compare VGAE with two machine-learning-based methods. The model achieves a $26.7\%$ gain over DeepWalk neural embeddings and $17.2\%$ over Concatenated drug features in AP@50 scores. Furthermore, albeit trained on one-hot feature vectors, vanilla VGAE can achieve competitive performance with Decagon, outperforming the baseline by $4.0\%$ (AP@50). This indicates the effectiveness of VGAE in recommending potential side effects in a featureless drug-protein multi-modal network. Finally, VGAE with additional molecular information at the decoding stage provides the best performance across the approaches. The method achieves the highest scores among all three metrics and outperforms the others by a large margin. The results reveal that using molecular fingerprints is a simple yet effective solution in predicting drug-drug interactions as suggested in [37].

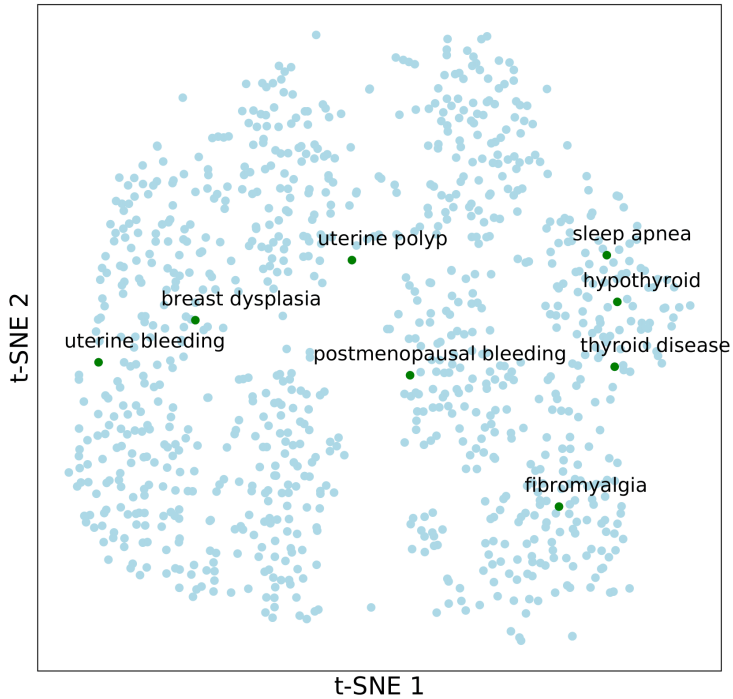



Figure 3: Visualization of polypharmacy side effects modeled by VGAE

**Visualization of side effect embeddings** To further demonstrate the capability of VGAE in learning to model drug-pair polypharmacy side effects, we plot their representations in 2D dimension using

the t-SNE [38] method. The embedding of each side effect is derived by taking on-diagonal entries of the tensor $D_e$ to create a $d$-dimensional vector which is accordingly projected into 2D space. Figure 3 illustrates a vector space in which side effects' representations establish a clustering structure.

### 5.2 Anticancer drug response prediction

In this task, the models are evaluated using four criteria including root mean square error (RMSE), coefficient of determination ($R^2$), Pearson correlation coefficient (PCC), and fitness which is computed as:

$$fitness = R^2 + PCC - RMSE$$

Table 2 shows the experimental results of drug response prediction on nine combinations of pairwise similarities of drugs and cell lines. VGAE trained on copy number variation with KEGG-pathways as similarity scores for cell lines and drugs achieve the best performance, yielding the lowest RMSE at 0.46 $\pm$ 0.02 and highest scores in $R^2$ = 0.67 $\pm$ 0.03, PCC = 0.85 $\pm$ 0.01, and fitness = 1.05 $\pm$ 0.06. Moreover, Table 3 demonstrates the comparisons between VGAE and other baselines which are taken from [32] including CDRscan [39], CDCN [40], SRMF [41], CaDRRes [42], ADRML [32], and k-nearest neighbors (KNN). The results reveal that compared with the baselines, VGAE has competitive performance, especially in the fitness score determining the effectiveness of the approaches in the drug response prediction task.

Table 2: Performance of VGAE on various types of similarities of drugs and cell lines

| Cell line similarity | Drug similarity | RMSE ↓ | $R^2$ ↑ | PCC ↑ | fitness ↑ |
|---|---|---|---|---|---|
| Mutation | Chemical | 0.48 $\pm$ 0.02 | 0.64 $\pm$ 0.04 | 0.84 $\pm$ 0.01 | 1.00 $\pm$ 0.06 |
| Gene expression | Chemical | 0.47 $\pm$ 0.01 | 0.66 $\pm$ 0.03 | 0.85 $\pm$ 0.00 | 1.03 $\pm$ 0.05 |
| Copy number variation | Chemical | 0.47 $\pm$ 0.02 | 0.65 $\pm$ 0.03 | 0.84 $\pm$ 0.01 | 1.02 $\pm$ 0.05 |
| Mutation | Target protein | 0.48 $\pm$ 0.02 | 0.66 $\pm$ 0.03 | 0.84 $\pm$ 0.01 | 1.01 $\pm$ 0.05 |
| Gene expression | Target protein | 0.47 $\pm$ 0.02 | 0.66 $\pm$ 0.03 | 0.84 $\pm$ 0.01 | 1.03 $\pm$ 0.05 |
| Copy number variation | Target protein | 0.48 $\pm$ 0.01 | 0.65 $\pm$ 0.03 | 0.84 $\pm$ 0.01 | 1.02 $\pm$ 0.05 |
| Mutation | KEGG | 0.47 $\pm$ 0.02 | 0.66 $\pm$ 0.03 | 0.85 $\pm$ 0.01 | 1.03 $\pm$ 0.06 |
| Gene expression | KEGG | 0.47 $\pm$ 0.02 | 0.65 $\pm$ 0.04 | 0.84 $\pm$ 0.01 | 1.03 $\pm$ 0.08 |
| Copy number variation | KEGG | **0.46 $\pm$ 0.02** | **0.67 $\pm$ 0.03** | **0.85 $\pm$ 0.01** | **1.05 $\pm$ 0.06** |

Table 3: Methods' performance in predicting anticancer drug response

| Method | RMSE ↓ | $R^2$ ↑ | PCC ↑ | fitness ↑ |
|---|---|---|---|---|
| ADRML | 0.49 | **0.68** | **0.85** | 1.04 |
| CDRscan | 0.76 | 0.67 | 0.83 | 0.74 |
| CDCN | 0.48 | 0.67 | 0.83 | 1.02 |
| SRMF | **0.25** | 0.40 | 0.80 | 0.95 |
| CaDRRes | 0.53 | 0.31 | 0.52 | 0.3 |
| KNN | 0.56 | 0.57 | 0.78 | 0.79 |
| VGAE (ours) | 0.46 $\pm$ 0.02 | 0.67 $\pm$ 0.03 | **0.85 $\pm$ 0.01** | **1.05 $\pm$ 0.06** |

## 6 Conclusion

In this work, we evaluate the effectiveness of variational graph autoencoders in predicting potential polypharmacy side effects on multimodal networks. The results reveal that VGAE trained on one-hot feature vectors outperforms other approaches. Moreover, augmenting Morgan fingerprints before the decoding stage helps boost the performance of VGAE. This suggests further examination of the use of molecular fingerprints in drug-drug interaction problems.

# References


[1] Mukesh Bansal, Jichen Yang, Charles Karan, Michael P Menden, James C Costello, Hao Tang, Guanghua Xiao, Yajuan Li, Jeffrey Allen, Rui Zhong, et al. A community computational challenge to predict the activity of pairs of compounds. *Nature biotechnology*, 32(12):1213–1222, 2014.

[2] Yi Sun, Zhen Sheng, Chao Ma, Kailin Tang, Ruixin Zhu, Zhuanbin Wu, Ruling Shen, Jun Feng, Dingfeng Wu, Danyi Huang, et al. Combining genomic and network characteristics for extended capability in predicting synergistic drugs for cancer. *Nature communications*, 6(1): 1–10, 2015.

[3] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 06 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty294. URL `https://doi.org/10.1093/bioinformatics/bty294`.

[4] Assaf Gottlieb, Gideon Y Stein, Yoram Oron, Eytan Ruppin, and Roded Sharan. Indi: a computational framework for inferring drug interactions and their associated recommendations. *Molecular systems biology*, 8(1):592, 2012.

[5] Santiago Vilar, Rave Harpaz, Eugenio Uriarte, Lourdes Santana, Raul Rabadan, and Carol Friedman. Drug—drug interaction through molecular structure similarity analysis. *Journal of the American Medical Informatics Association*, 19(6):1066–1074, 2012.

[6] Feixiong Cheng and Zhongming Zhao. Machine learning-based prediction of drug–drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties. *Journal of the American Medical Informatics Association*, 21(e2):e278–e286, 2014.

[7] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[8] Qijin Yin, Xusheng Cao, Rui Fan, Qiao Liu, Rui Jiang, and Wanwen Zeng. Deepdrug: A general graph-based deep learning framework for drug-drug interactions and drug-target interactions prediction. *biorxiv*, pages 2020–11, 2022.

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

[10] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[11] Jinxian Wang, Xuejun Liu, Siyuan Shen, Lei Deng, and Hui Liu. Deepdds: deep graph neural network with attention mechanism to predict synergistic drug combinations. *Briefings in Bioinformatics*, 23(1):bbab390, 2022.

[12] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJXMpikCZ`. accepted as poster.

[13] Arnold K Nyamabo, Hui Yu, and Jian-Yu Shi. Ssi–ddi: substructure–substructure interactions for drug–drug interaction prediction. *Briefings in Bioinformatics*, 22(6):bbab133, 2021.

[14] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[15] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press, 2015. ISBN 0262511290.

[16] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1ZdKJ-0W`.

[17] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):190–198, 2017.

[18] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

[19] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/gilmer17a.html`.

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf`.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL `https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf`.

[23] John Ingraham and Debora Marks. Variational inference for sparse and undirected models. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1607–1616. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/ingraham17a.html`.

[24] Hao Wu, Jonas Köhler, and Frank Noe. Stochastic normalizing flows. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5933–5944. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/41d80bfc327ef980528426fc810a6d7a-Paper.pdf`.

[25] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders, 2018. URL `https://arxiv.org/abs/1802.03480`.

[26] Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. NetGAN: Generating graphs via random walks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 610–619. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/bojchevski18a.html`.

[27] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/d0921d442ee91b896ad95059d13df618-Paper.pdf`.

[28] George E. Dahl, Tara N. Sainath, and Geoffrey E. Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8609–8613, 2013. doi: 10.1109/ICASSP.2013.6639346.

[29] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf`.

[31] Sebastian Bock, Josef Goppold, and Martin Weiß. An improvement of the convergence proof of the adam-optimizer. *arXiv preprint arXiv:1804.10587*, 2018.

[32] Fatemeh Ahmadi Moughari and Changiz Eslahchi. Adrml: anticancer drug response prediction using manifold learning. *Scientific reports*, 10(1):1–18, 2020.

[33] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 809–816, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[34] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Trans. Intell. Syst. Technol.*, 8(2), oct 2016. ISSN 2157-6904. doi: 10.1145/2915921. URL `https://doi.org/10.1145/2915921`.

[35] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623732. URL `https://doi.org/10.1145/2623330.2623732`.

[36] Nansu Zong, Hyeoneui Kim, Victoria Ngo, and Olivier Harismendy. Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations. *Bioinformatics*, 33(15):2337–2344, 2017.

[37] Yanan Long, Horace Pan, Chao Zhang, Hy Truong Son, Risi Kondor, and Andrey Rzhetsky. Molecular fingerprints are a simple yet effective solution to the drug–drug interaction problem. *The 2022 ICML Workshop on Computational Biology*, 2022. URL `https://icml-compbio.github.io/2022/papers/WCBICML2022_paper_72.pdf`.

[38] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

[39] Yoosup Chang, Hyejin Park, Hyun-Jin Yang, Seungju Lee, Kwee-Yum Lee, Tae Soon Kim, Jongsun Jung, and Jae-Min Shin. Cancer drug response profile scan (cdrscan): a deep learning model that predicts drug effectiveness from cancer genomic signature. *Scientific reports*, 8(1): 1–11, 2018.

[40] Dong Wei, Chuanying Liu, Xiaoqi Zheng, and Yushuang Li. Comprehensive anticancer drug response prediction based on a simple cell line-drug complex network model. *BMC bioinformatics*, 20(1):1–15, 2019.

[41] Chayaporn Suphavilai, Denis Bertrand, and Niranjan Nagarajan. Predicting cancer drug response using a recommender system. *Bioinformatics*, 34(22):3907–3914, 2018.

[42] Lin Wang, Xiaozhong Li, Louxin Zhang, and Qiang Gao. Improved anticancer drug response prediction in cell lines using matrix factorization with similarity regularization. *BMC cancer*, 17(1):1–12, 2017.

# A Additional Table

Table 4: Statistics for each type of interaction

| Interaction | # Edges |
|---|---|
| Drug-Drug | 4,651,131 |
| Drug-Protein | 18,596 |
| Protein-Protein | 715,612 |

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]