Abstract—The computational drug repositioning aims to discover new uses for marketed drugs, which can accelerate the drug development process and play an important role in the existing drug discovery system. However, the number of validated drug-disease associations is scarce compared to the number of drugs and diseases in the real world. Too few labeled samples will make the classification model unable to learn effective latent factors of drugs, resulting in poor generalization performance. In this work, we propose a multi-task self-supervised learning framework for computational drug repositioning. The framework tackles label sparsity by learning a better drug representation. Specifically, we take the drug-disease association prediction problem as the main task, and the auxiliary task is to use data augmentation strategies and contrast learning to mine the internal relationships of the original drug features, so as to automatically learn a better drug representation without supervised labels. And through joint training, it is ensured that the auxiliary task can improve the prediction accuracy of the main task. More precisely, the auxiliary task improves drug representation and serving as additional regularization to improve generalization. Furthermore, we design a multi-input decoding network to improve the reconstruction ability of the autoencoder network. The experimental results demonstrate the effectiveness of the multi-task self-supervised learning framework, and its predictive ability is superior to the state-of-the-art model.

Index Terms—Computational drug repositioning, drug discovery, label sparsity, self-supervised learning.

I. INTRODUCTION

Traditional drug discovery has a long R&D cycle, high investment cost, high risk and low success rate [1], [2]. It takes roughly 10–15 years and $0.8–$1 billion to bring a new drug to market from development [3], [4]. The difficulty of new drug development is that about 90% of new drugs do not pass Phase I clinical trials because their new chemical structures lead to unpredictable side effects during actual use. Therefore, there is a need for a new technology that can accelerate the drug development process and ensure that drug discovery has a high success rate and low risk [5], [6].

Manuscript received 9 August 2022; revised 30 January 2023; accepted 5 March 2023. Date of publication 8 March 2023; date of current version 9 October 2023. This work was supported by the China National R&D Key Research Program under Grant 2020YFB1711204 and 2019YFB1705702. (Corresponding Author: Genke Yang.)

Xinxing Yang, Genke Yang, and Jian Chu are with the Ningbo Artificial Intelligence Institute, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Department of Automation, Shanghai Jiaotong University, Shanghai 200240, China (e-mail: yangxinxing@sjtu.edu.cn; gkyang@sjtu.edu.cn; chujian@sjtu.edu.cn).

Digital Object Identifier 10.1109/TCBB.2023.3254163

Computational drug repositioning aims to uncover new uses for marketed drugs based on known drug-disease associations [7], [8]. The logic behind it is that the small molecule drugs currently on the market have multi-target properties, which means that they can inhibit or activate unknown targets, thereby producing therapeutic effects on unknown diseases. Computational drug repositioning uncovers potential therapeutic patterns of drugs and diseases through computational models and the large number of validated drug-disease associations. Based on these patterns, new therapeutic uses of the target drug can be inferred [9], [10].

The popular computational drug repositioning models can be divided into two categories, namely graph-based model [11], [12], [13], [14] and matrix factorization-based model [15], [16]. The first step of the graph-based model is to construct a heterogeneous network based on multi-source information of drugs and diseases, and then use algorithms such as random walks to mine potential drug-disease associations on the above heterogeneous network. Wang et al. [17] integrated information from multiple sources, including omics information of diseases, drugs and targets, to construct a heterogeneous graph. A random walk algorithm is then performed on the entire heterogeneous graph to compute potential drug-disease associations. Luo et al. [18] constructed a heterogeneous graph based on drug similarity information, disease similarity information and drug-disease associations. The Bi-Random walk algorithm was then used to walk through the heterogeneous graph to predict new drug-disease associations. Zeng et al. [19] proposed a network-based deep learning method, deepDR. The model integrates multiple drug and disease-related networks for mining new uses of drugs. These networks are the drug-disease association network, the drug side effect network, the drug-target network, and the multiple drug interaction network. The deepDR model is used to predict the probability of a therapeutic relationship between drugs and diseases by learning higher-order features in these networks through multiple autoencoder models. Previous graph-based models assume that neighbors are independent of each other in heterogeneous graphs, which leads to the loss of local structural information. Therefore, Meng et al. [20] performed graph convolution operations on drug-disease association networks, drug-drug similarity networks and disease-disease similarity networks to learn a unified representation of drugs and diseases. Then the drug and disease representations are input into a multi-layer fully-connected network with network regularization elements to obtain drug-disease association probabilities.
The idea of the matrix factorization-based model is to represent the latent factor of drugs and diseases using vectors in embedding space. The probability of a therapeutic association between a drug and a disease is subsequently calculated by the similarity function. Zhang et al. [21] proposed a computational model based on Bayesian inductive matrix for mining new uses of marketed drugs. Unlike previous work that used data from a single source to calculate drug and disease similarity, DRIMC used multiple data sources. In addition, the features of the drugs (diseases) are considered with information about their respective neighbors. These features were then combined with the induction matrix to obtain the predicted values. Yan et al. [22] proposed a multi-view matrix factorization model for predicting novel drug-disease associations. The model mines the combined drug/disease similarity matrix from multiple sources of drug/disease structural information. And this similarity matrix is combined with the drug-disease association matrix to derive the treatment probability between the target drug and the target disease through regularization techniques and multi-view learning. Yang et al. [23] demonstrated that this approach does not represent the relationship between drugs and diseases correctly. Therefore, they proposed neural metric factorization for computational drug repositioning (NMFDR). The NMFDR model used a modified euclidean distance and point space to represent the drug-disease relationship and demonstrated that this approach is superior to the inner product operation and vector space on several real datasets. Yang et al. [24] believe that the fusion of multiple similarity information of drugs or diseases cannot maximize the ability to express drug or disease characteristics. Therefore, they concatenate multiple similarity matrices of drugs and multiple similarity information of diseases through validated drug-disease associations to form a comprehensive similarity matrix. Then, matrix completion work is performed on the matrix to mine potential drug-disease associations. Yang et al. [25] argue that previous heterogeneous graph inference models do not take into account the sparsity of heterogeneous association networks, thereby degrading the performance of computational drug repositioning models. Therefore, they first use bounded matrix completion to complete the missing data, so as to increase the density of the heterogeneous association network. Then they used various similarity information to connect drugs and disease nodes in the heterogeneous network. Finally, they utilized a heterogeneous graph inference algorithm to mine potential drug-disease associations from this heterogeneous network. Li et al. [26] believe that the existing matrix factorization model cannot effectively capture the characteristics of drugs or diseases, thus reducing the mining ability of the model. So they used a more powerful feature extraction model, the graph neural network, to capture the effective features of drugs or diseases. And in order to overcome the cold-start problem in computational drug repositioning, they embedded auxiliary information into the graph neural network, and verified the effectiveness of the model on multiple real-world datasets.

The focus of this paper is on the two-tower model (matrix factorization-based model) in computational drug repositioning, where the main idea is to use two different sets of neural networks to learn the latent factor of the drug and disease. The degree of compatibility of the two latent factors is subsequently calculated using a similarity function (inner product) as the probability that the drug can treat the disease. This class of models is described in detail in Section II-A.

The key to the usefulness of this model is to find the appropriate latent factor vector to represent the drug and disease. However, as the number of layers of the neural network increases, the number of parameters to be learned also increases exponentially. It is noteworthy that the loss function used to train these models is usually formulated as a supervised learning problem. These supervised signals are sourced from validated drug-disease associations. However, the validated drug-disease associations are very scarce compared to the larger number of drugs and diseases in the real world. This also means that it is difficult to train the large number of parameters in the two-tower model with these few labeled samples.

Inspired by the successful application of self-supervised learning in computer vision [27] and recommendation system [28], we believe that self-supervised learning can provide a different perspective to enhance the representation of drugs via unvalidated data. Therefore, in this work, we propose a multi-task self-supervised learning framework (SSLDR) to tackle the label sparsity problem in computational drug repositioning. Specifically, we take the drug-disease association prediction problem as the main task, and the auxiliary task (self-supervised learning) is to use data augmentation strategies and contrast learning to mine the internal relationships of the original drug features, so as to automatically learn a better drug representation without supervised labels. The auxiliary task of the SSLDR framework can be divided into three steps. The first step of the auxiliary task is to use the similarity information to select the negative neighbors of the target drug. Where \( k = \text{arg min}_{k} \text{DrugSim}_i(x^k_i) \), \( \text{DrugSim}_i \in \mathbb{R}^{n \times n} \) is the similarity matrix between drugs, which is calculated using the Simplified Molecular Input Line Entry System (SMILES) string of the drug. \( n \) represents the number of drugs. The second step is to use two different data augmentation strategies to perform representation learning on the target drug \( i \) and its negative neighbor \( k \) to obtain two different representations for each of them. In practice, we use two different chemical structure representations of drugs as data augmentation strategies. They are the SMILES string and International Chemical Identifier (InChI) of the drug, respectively. The third step is to map the above four drug representations (two belonging to the target drug \( i \), and two belonging to the negative neighbor \( k \)) to the embedding space to obtain the respective latent factors. And the purpose is to shorten the distance of the latent factor belonging to the target drug \( i \) in the embedding space, and to make the distance between the latent factor of the target drug \( i \) and the latent factor of the negative neighbor \( k \) longer in the embedding space. Through the continuous comparison of the latent factor of the target drug \( i \) and the negative neighbor \( k \), a better latent factor can be automatically learned from the unlabeled drug data.

If we just initialize the corresponding latent factor in the two-tower model with the drug representation obtained after the above self-supervised learning, this is essentially a pre-training
approach that cannot effectively improve the accuracy of the main task (drug-disease association prediction). Therefore, to ensure that the auxiliary task can improve the accuracy of the main task, we leverage a multi-task training strategy where the main task (supervised) and the auxiliary task (self-supervised) are jointly optimized. To be precise, the embedding layer employed in the auxiliary task shares parameters with the drug embedding layer of the two-tower model in the main task. In Section II-C, we explore why the joint training strategy can contribute to the prediction performance of the main task. It essentially adds a regularization term to the loss function of the main task, which enhances the generalization performance of the main task.

In addition, some works use autoencoders as embedding layers for mining the latent factor of drugs and diseases. With the deepening of the network layers in the autoencoder, the problem of information loss occurs, and the decoder cannot restore the original input. However, the idea of the autoencoder is that a good latent factor must be able to restore the original input. Therefore, in order to solve the problem that the latent factor cannot be restored to the original input due to information loss, we design a multi-input decoder. Unlike the previous decoder layer which only accepts the output of the previous network layer, we input the latent factor into each decoder layer to enhance the reconstructing ability of the decoder. Subsequent experiments verify that the latent factor has better prediction performance after the above operation.

Our main contributions in this work are as follows.

1) We present a multi-task self-supervised learning framework for tackling the label sparsity problem in computational drug repositioning. The framework jointly optimizes the main task ([drug-disease association prediction]) and auxiliary task (self-supervised learning) through parameter sharing to improve the representation of drugs, which can be used to improve the prediction performance of potential drug-disease associations.

2) We improve the decoder in the autoencoder so that each decoder can take into account latent factor information. Thus, a good latent factor can be obtained without loss of information.

3) On three real-world drug-disease association datasets, we demonstrate that self-supervised learning as an auxiliary task improves the performance of the main model. In addition, it is proved that autoencoders with multi-input decoder have better prediction performance.

The rest of this paper is organized as follows. Section II introduces the dataset used in this work and the proposed multi-task self-supervised learning framework and multi-input decoder. Section III is the experimental part, including multiple ablation experiments, comparison experiments with state-of-the-art and case study. Section IV summarizes the work of this paper and discusses future directions of work.

II. MATERIALS AND METHODS

In this work, we design a multi-task self-supervised learning framework for solving the problem of label sparsity in computational drug repositioning by learning better drug representations.

We first introduce the dataset used in this work in Section II-A. Second, we introduce the working principle of the two-tower model in computational drug repositioning in Section II-B. Subsequently, we describe how the multi-task self-supervised learning framework improves the accuracy of drug-disease prediction in Section II-C. Finally, in Section II-D, we describe the workflow of the multi-input decoder.

A. Datasets

In this work, we use three popular real-world datasets [18]. The drugs in these three datasets are marketed drugs from DrugBank database [29], diseases from Online Mendelian Inheritance in Man database [30] and the sparsity of each dataset is different. Table I lists the statistics of the above dataset, in which the Gottlieb dataset contains 593 drugs, 313 diseases and 1933 treatment relationships. The Cdataset contains 663 drugs, 409 diseases and 2532 treatment relationships. The DNdataset contains 1490 drugs, 4516 diseases and 1008 treatment relationships. We removed uninformative drugs and diseases from the DNdataset, i.e., these drugs did not have any treatment for diseases, and these diseases did not have any known treatment drugs. 550 drugs and 360 diseases were retained in the final DNdataset.

The above dataset also contains similarity matrix between drugs, DrugSim, which is calculated from the SMILES chemical structures of the drugs [31], [32]. The similarity matrix between diseases is calculated from the medical description information between them. In addition, the repoDB dataset, in which, negative and positive samples were clearly defined and large enough for classification tasks. The repoDB dataset contains 1571 drugs and 2051 diseases, 6677 have been approved drug-disease associations and 4123 failed drug-disease associations. For ease of calculation, we only consider drugs with a treatment number greater than 5, and diseases associated with at least 7 drugs.

B. The Two-Tower Model in Computational Drug Repositioning

Computational drug repositioning can be defined as a binary classification problem, given a target drug \( i \) and a target disease \( j \), we input their respective features into the model \( \mathcal{M} \), resulting in a prediction value of 0 or 1, where 0 means the drug \( i \) is not able to treat the disease \( j \) and 1 means the drug \( i \) is able to treat the disease \( j \).

The current popular model of computational drug repositioning is the two-tower model. This framework is using two different sets of neural networks to learn the latent factor of drugs and diseases. Its architecture is shown in Fig. 1. In this

| Datasets   | Drugs | Diseases | Validated Associations | Sparsity   |
|------------|-------|----------|------------------------|------------|
| Gottlieb   | 593   | 313      | 1933                   | 98.95%     |
| Cdataset   | 663   | 409      | 2532                   | 99.06%     |
| DNdataset  | 1490  | 4516     | 1008                   | 99.98%     |

First, we introduce the dataset used in this work in Section II-A. Second, we introduce the working principle of the two-tower model in computational drug repositioning in Section II-B. Subsequently, we describe how the multi-task self-supervised learning framework improves the accuracy of drug-disease prediction in Section II-C. Finally, in Section II-D, we describe the workflow of the multi-input decoder.
architecture, the features of the drug and the disease are separately input into an embedding layer containing a series of neural networks for extracting the respective latent factor. The latent factor of the drug and the disease are subsequently computed by a similarity algorithm (inner product) to derive the prediction value $R_{ij}$, which represents the probability that the drug can treat the disease.

C. The Multi-Task Self-Supervised Learning Framework

We present a multi-task self-supervised learning framework for computing drug repositioning. The framework is used to tackle label sparsity by learning a better drug representation. Specifically, we treat the prediction of drug-disease associations as the main task (supervised learning). And automatically mining the internal relationships of drug features is the auxiliary task (self-supervised learning), which aims to learn a good drug representation in the presence of unlabeled data. In this subsection, we first introduce the auxiliary task, and then describe how the auxiliary task improves the accuracy of the main task through a joint training strategy.

The auxiliary task of the SSLDR framework can be divided into three steps. The first step is to use the similarity information to select the neighbors of the target drug. The second step is representation learning for the target drug and negative neighbors using two different data augmentation strategies. The third step is to map the above four representations into the embedding space through the embedding layer so that the latent factors belonging to the same drug are close to each other in the embedding space, and those not belonging to the same drug are far away. In this way, a better latent factor can be automatically learned from the unlabeled data. In particular, we use the SMILES string and InChI to encode the chemical structure information of a drug with a small number of characters. And to enable the strings to be input by the deep learning model, we use the Word2Vec algorithm to convert the strings into numeric vectors that can be received by the system. In addition, we also use the drug behavior records $R_{ix}$ and similarity information $DrugSim$ instead of the transfer functions $h$ and $g$.

As shown in (3), we then input $x_i$ and $x'_i$ into the embedding functions $\mathcal{H}$ and $\mathcal{G}$, respectively, to obtain $z_i$ and $z'_i$ as the two latent factors of the target drug $i$. The same operation is performed for $x_k$ and $x'_k$ to obtain two latent factors of negative drug $k$, $z_k$ and $z'_k$. In practice, we use the autoencoder model as the embedding function

$$
\begin{align*}
  z_i &= \mathcal{G}(x_i), \quad z'_i = \mathcal{H}(x'_i) \\
  z_k &= \mathcal{G}(x_k), \quad z'_k = \mathcal{H}(x'_k)
\end{align*}
$$

(3)

After obtaining the latent factor of the target drug $i$ and the negative drug $k$, we want to make the distance between $z_i$ and $z'_i$ belonging to the same drug in the embedding space as close as possible. The distance between $z_i$ and $z_k$, $z'_i$, and $z'_k$ that do not belong to the same drug becomes as far as possible in the embedding space. Therefore, as shown in (4), we define the following loss function to let these latent factors contrast themselves

$$
\mathcal{L}_{auxiliary} = D(z_i, z'_i) - D(z_i, z_k) - D(z'_i, z'_k)
$$

(4)

Where $D$ is the distance metric function. In this work, we use the euclidean distance as the distance metric function. For example, $D(z_i, z'_i) = \|z_i - z'_i\|^2$.

The main task in the SSLDR framework is based on the two-tower model with the flow chart shown in Fig. 3, which contains information can be obtained from the similarity matrix of drugs $DrugSim$, which can be downloaded from public websites.

$$
k \leftarrow S(i)
$$

(1)

Subsequently, as shown in (2), we do data augmentation for the target drug $i$ and negative drug $k$ using two different transfer functions $h$ and $g$. In the framework of self-supervised learning, the more complex the auxiliary task is, the more it can improve the predictive performance of the main task. Therefore, we adopt two different transfer functions as data augmentation strategies to increase the complexity of the auxiliary tasks, thereby improving the performance of the main task. For the target drug $i$, we want to learn two different representations $x_i$ and $x'_i$ after different data augmentation to ensure that the model still recognizes that $x_i$ and $x'_i$ represent the same drug $i$. Same for the negative drug $k$, two different representations $x_k$ and $x'_k$ are also learned by different data augmentation

$$
\begin{align*}
  x_i &\leftarrow g(drug_i), \quad x'_i \leftarrow h(drug_i) \\
  x_k &\leftarrow g(drug_k), \quad x'_k \leftarrow h(drug_k)
\end{align*}
$$

(2)

In practice, we use two different chemical structure representations of drugs as data augmentation strategies. The transfer functions $h$ and $g$ are replaced by the Simplified Molecular Input Line Entry System (SMILES) string of the drug and the International Chemical Identifier (InChI), respectively. Both SMILES strings and InChI represent the chemical structure information of a drug with a small number of characters. And to enable the strings to be input by the deep learning model, we use the Word2Vec algorithm to convert the strings into numeric vectors that can be received by the system. In addition, we also use the drug behavior records $R_{ix}$ and similarity information $DrugSim_{ix}$ instead of the transfer functions $h$ and $g$.

As shown in (3), we then input $x_i$ and $x'_i$ into the embedding functions $\mathcal{H}$ and $\mathcal{G}$, respectively, to obtain $z_i$ and $z'_i$ as the two latent factors of the target drug $i$. The same operation is performed for $x_k$ and $x'_k$ to obtain two latent factors of negative drug $k$, $z_k$ and $z'_k$. In practice, we use the autoencoder model as the embedding function

$$
\begin{align*}
  z_i &= \mathcal{G}(x_i), \quad z'_i = \mathcal{H}(x'_i) \\
  z_k &= \mathcal{G}(x_k), \quad z'_k = \mathcal{H}(x'_k)
\end{align*}
$$

(3)
Fig. 2. The architecture of the auxiliary task.

(a) The general architecture of the auxiliary task.

(b) The implementation detail of auxiliary task in this work.

Fig. 3. The workflow of main task in the SSLDR framework.
three modules, namely the input layer, embedding layer and prediction layer.

**Input Layer.** We take the treatment information of the target drug \( i \) for all diseases as its input feature, which is the \( i \)th row of \( R \), \( R_i \). \( R \) is the drug-disease association matrix. The treated information of the target disease \( j \) for all drugs is used as its input feature, which is the \( j \)th column of \( R \), \( R_j \). The benefit of this feature is the ability to keep a record of behavioral preferences for drugs or diseases.

**Embedding Layer.** We use an autoencoder model with 2 encoders and 3 decoders as the component to extract the latent factor of a drug (disease). Taking the drug as an example, the operation formula is shown below

\[
d_i = f(W_2^T f(W_1^T R_i + b_1) + b_2) \quad (5)
\]

\[
\hat{R}_i = f(V_3^T f(V_2^T d_i + b_3) + b_4) + b_5 \quad (6)
\]

\[
L_{drug} = \| \hat{R}_i - R_i \|_2^2 \quad (7)
\]

Where (5) is the encoding part, (6) is the decoding part, and (7) is the loss function of the autoencoder. The \( d_i \) is obtained by minimizing \( L_{drug} \). Similarly, the latent factor of the disease \( j \), \( s_j \), can be obtained by minimizing \( L_{disease} \)

\[
L_{disease} = \| \hat{R}_j - R_j \|_2^2 \quad (8)
\]

**Prediction Layer.** The predicted probability \( \hat{R}_{ij} \) that the target drug \( i \) can treat the target disease \( j \) is obtained by performing the inner product operation on \( d_i \) and \( s_j \)

\[
\hat{R}_{ij} = d_i^T s_j \quad (9)
\]

However, if the latent factor of the drugs learned by the auxiliary task is used as the initial value in the corresponding embedding layer of the main task, this is essentially a pre-training approach that cannot effectively improve the prediction accuracy of the main task. Therefore, to ensure that the drug representation learned by the auxiliary task can improve the prediction accuracy of the main task (drug-disease association prediction), we leverage a multi-task training strategy where the drug-disease association prediction task (supervised) and the auxiliary task (self-supervised) are jointly optimized (Fig. 4).

**D. The Autoencoder With Multiple-Input Decoder**

The two-tower model mentioned above uses the autoencoder to extract latent factors of drugs or diseases. The deeper the number of layers in the autoencoder model, the more efficient latent factor it can capture. However, according to the logic of the autoencoder model, a good latent factor depends on whether it can restore the original features. After the original features undergo multi-layer encoding and decoding operations, the problem of information loss occurs due to the disappearance of original information. To solve this problem, we introduce a multi-task training strategy that jointly optimizes the main task and auxiliary task, where the latent factor learned from the auxiliary task is used as the initial value in the corresponding embedding layer of the main task. This strategy not only improves the prediction accuracy of the main task but also enhances the generalization ability of the model.
of gradients and other reasons. As a result, the latent factor cannot restore the original features.

We believe that the biggest reason why the latent factor cannot restore the original input features is that the decoder only receives a single input from the previous layer, which may become sparse as the number of layers deepens. Therefore, based on the decoding architecture of the main task, we additionally add the latent factor to the input of each decoder (Fig. 5), so that each layer of the decoder can take into account the information from the latent factor. Its decoding operation is shown in the following (12)

\[ \hat{R}_{is} = f(V_i^T (f(V_i^T d_i + b_3) + \beta d_i) + b_4) + \beta d_i) + b_5) \]

(12)

Observing the above equation, it can be concluded that when the adjustment parameter \( \beta \) is 0, it is the same as the original decoder. However, when the value of the adjustment parameter \( \beta \) is greater than 0, the decoder can take into account the information of latent factor and overcome the problem of information loss. Algorithm 1 is the pseudo-code of the SSLDR framework.

### III. EXPERIMENTS AND DISCUSSION

We provide empirical results on three real-world drug-disease association datasets to demonstrate the effectiveness of our proposed multi-task self-supervised learning framework and multi-input decoder. The experiments designed in this section were used to answer the following research questions.

- **RQ1** Can our proposed strategy of jointly optimizing the auxiliary task and the main task improve the prediction accuracy of the latter?
- **RQ2** Can our proposed autoencoder with multi-input decoder have an advantage in prediction performance compared to traditional autoencoder?
- **RQ3** Can our proposed multi-task self-supervised learning framework outperform the state-of-the-art model?
- **RQ4** How can our proposed multi-task self-supervised learning framework help in practical applications?

### A. Evaluation Metrics

The experiments in this section use 5-fold cross-validation to assess the generalization ability of the model. We first treat the known drug-disease associations as positive samples and divide them into 5 parts equally. We take turns to use four parts of them sequentially as the training set and the remaining one part as the test set. In addition, we added all unknown drug-disease associations as negative samples to the testset. The parameters in the model are subsequently trained by the training set and the generalization performance of the model is evaluated by the testset. Finally, the average of the computed results of the 5 rounds is calculated, and the value represents the result of the 5-fold cross-validation of the model.
Computational drug repositioning is a binary classification problem. In order to fairly compare the generalization performance of the models, we use two popular evaluation metrics for evaluating the performance of the models. They are AUPR (Area Under Precision-Recall Curve) and F1-Score, which can more objectively evaluate the performance of the model in the case of positive and negative sample disproportion.

### B. Parameter Setting

The variation interval of the latent factor vector dimension of drugs and diseases is [8, 16, 32, 64, 128, 256]. The variation interval of the parameters of the loss function of the autoencoder is [0.1, 0.3, 0.5, 0.7, 0.9]. The learning rate of the model optimizer varies in the interval [0.1, 0.05, 0.01, 0.005, 0.001]. In the experiments in this section, the default values for the above parameters are 64, 0.5 and 0.001.

### C. Effectiveness of Joint Optimization of Auxiliary Task and Main Task (RQ1)

To answer RQ1, we evaluate the impact of the auxiliary tasks on the main task under the joint training strategy. We compare SSLDR with the following baseline. The baseline is SSLDR-M, a variant of the SSLDR model with auxiliary tasks removed. By comparing the experimental results of SSLDR and SSLDR-M, we can intuitively compare whether the auxiliary task can improve the prediction accuracy of the main task.

Table II shows the experimental results of the SSLDR model and the SSLDR-M model on the three real-world datasets. First, it can be intuitively found from the Table II that the SSLDR model outperforms the SSLDR-M model on all metrics and datasets. On the three datasets, the average improvement under the AUPR and F1-Score metrics are 3.4% and 8.9%. The above results illustrate that the loss function of the auxiliary task is used as the regularization term of the loss function of the main task by a joint training strategy, which optimizes the search space of the parameters in the main task. This enables the model of the main task to have better generalization performance.

Furthermore, we find that the performance gap between the SSLDR model and the SSLDR-M model is directly proportional to the sparsity between the datasets, i.e., the greater sparsity of the datasets, the larger performance gap between the SSLDR model and the SSLDR-M model. The sparsity of Gottlieb dataset, Cdataset and DNdataset increases in turn, and compared with the SSLDR-M model, the average improvement of SSLDR on these three datasets are 3.4%, 6.8% and 8.3%, respectively. This is because the parameters in the SSLDR-M model rely on labeled data for training. The small amount of labeled data in the sparse dataset prevents SSLDR-M model learning effective latent factor of drug and disease. And the SSLDR model additionally uses self-supervision and joint optimization to ensure that the main task learns a better latent factor of the drug. Therefore, the main task can have a better prediction effect. The discussion of the above experimental results proves that the auxiliary task can solve the sparsity of data, thereby improving the prediction accuracy of the main task. Through the above discussion, we believe that the framework can overcome data sparsity, thereby improving the predictive performance of the model.

### D. Effectiveness of Autoencoder With Multi-Input Decoder (RQ2)

To answer RQ2, we evaluate the prediction performance of the autoencoder with multi-input decoder. We compare SSLDR with the following baseline. The baseline is SSLDR-A, a variant of SSLDR, which uses the original autoencoder to replace the autoencoder with multiple-input decoder in the SSLDR model. Specifically, the decoder of the original autoencoder only receives input from the previous network layer. In addition, the hyperparameters γ and δ of the SSLDR-A model are set to 0. The purpose of this is to simulate the impact of a more severe loss of information on the model. Through the direct comparison between SSLDR and SSLDR-A, we can verify whether the multi-input decoder can overcome the problem of information loss, thereby improving the prediction ability of the latent factor.

Table III shows the experimental results of SSLDR model and SSLDR-A model on three datasets. It can be significantly found that the SSLDR model with a multi-input decoder outperforms the SSLDR-A model with the single-input decoder on all metrics and datasets. On the three datasets, the average improvements under the AUPR and F1-Score metrics are 11.1% and 11%. The above experimental results show that adding the latent factor to the input of each decoder allows it to take into account the information from the latent factor, so that the model can learn a better latent factor, thereby improving its expressiveness and the predictive power for drug-disease associations.

### E. Comparison of Experimental Results (RQ3)

To answer RQ3, we compare the experimental results of the SSLDR model with the following mainstream computational drug repositioning models.

- **MF [33]:** The matrix factorization model uses the inner product and the latent factor to infer the probability of a therapeutic relationship between the drug and the disease.
- **SVM [34]:** The support vector machine (SVM) are currently popular binary classification models.


| Dataset | Gottlieb | AUPR | F1-Score | Cdataset | AUPR | F1-Score | DNdataset | AUPR | F1-Score |
|---------|----------|------|----------|----------|------|----------|-----------|------|----------|
| SSLDR-A | 0.16     | 0.249| 0.215    | 0.297    | 0.053 | 0.117    |           |      |          |
| SSLDR   | 0.176    | 0.266| 0.245    | 0.322    | 0.038 | 0.138    |           |      |          |

**TABLE IV**

| Dataset | Gottlieb | AUPR | F1-Score | Cdataset | AUPR | F1-Score | DNdataset | AUPR | F1-Score |
|---------|----------|------|----------|----------|------|----------|-----------|------|----------|
| MF      | 0.033    | 0.01 | 0.017    | 0.01     | 0.005 | 0.003    |           |      |          |
| GMF     | 0.015    | 0.048| 0.018    | 0.042    | 0.007 | 0.026    |           |      |          |
| SVM     | 0.009    | 0.031| 0.008    | 0.026    | 0.005 | 0.022    |           |      |          |
| MLP     | 0.021    | 0.05 | 0.027    | 0.049    | 0.007 | 0.038    |           |      |          |
| MetricF | 0.039    | 0.074| 0.063    | 0.081    | 0.012 | 0.058    |           |      |          |
| SSLDR-A | 0.16     | 0.249| 0.215    | 0.297    | 0.053 | 0.117    |           |      |          |
| SSLDR-M | 0.171    | 0.256| 0.236    | 0.293    | 0.056 | 0.122    |           |      |          |
| SSLDR   | 0.176    | 0.266| 0.245    | 0.322    | 0.058 | 0.138    |           |      |          |

**Fig. 6.** The confusion matrix of SSLDR model when threshold = 0.1.

**Fig. 7.** The confusion matrix of SSLDR model when threshold = 0.3.

- **GMF [35]:** The Neural Collaborative Filtering uses neural networks and Hadamard products to uncover potential new uses for drugs.
- **MLP [36]:** The multi-layer perceptron (MLP) consists of multiple neural networks and sigmoid activation functions for binary classification problems.
- **MetricF [37]:** Metric factorization (MetricF) uses points and distances to represent drug-disease associations with greater expressiveness.

Table IV presents the experimental results of the SSLDR model with all the comparison algorithms. Figs. 6 and 7 show the confusion matrix of the SSLDR model at different thresholds.
The following conclusions can be drawn from observing the table.

The MF and GMF, as primary latent factor-based models, achieved poor prediction results because they were not able to incorporate more auxiliary information about drugs or diseases. The prediction performance of SVM and MLP is basically similar to that of GMF model (deep learning model). From the experimental results, it can be found that the performance of the SSLDR model is better than the MetricF model. The comparison with MetricF model can explain the effectiveness of the SSLDR model to a certain extent.

It is not difficult to find that through the contrastive learning between the two enhanced representations of the drug, the model can learn a better drug representation, thereby improving the generalization ability of the model. In addition, the additional input of the decoding layer enhances the reconstruction ability of the model. This also enhances the representation of drugs and diseases to a certain extent, thereby improving the predictive power of the model.

F. Case Study (RQ4)

We selected 3 drugs from the Gottlieb dataset to validate the usefulness of SSLDR in practical applications. These three drugs are doxorubicin, gemcitabine and vincristine, all of which are used to treat oncology diseases. Oncological diseases are currently the focus of research and development by pharmaceutical companies, so it is of great value to find potential therapeutic drugs for these diseases.

Table V lists the diseases recommended by the SSLDR model for these three drugs. The bolded diseases in the table indicate that they have been verified in the CTD dataset to have a therapeutic relationship with the corresponding drugs. For the drugs doxorubicin and gemcitabine, two new diseases were correctly predicted, and both were hits in the first and fifth spots. The last drug, vincristine, has 3 diseases that are correctly recommended in the list of recommended diseases.

The results of the above case studies show that compared with the previous computational drug repositioning models, the disease list recommended by SSLDR model has a higher hit rate, and most diseases are successfully predicted under the condition of higher ranking. Therefore, this can significantly accelerate the process of drug screening and research and development, and has great economic and practical value for practical application scenarios.

G. Experimental Results on the repoDB Dataset

We validate the SSLDR model using repoDB, a better dataset for computational drug repositioning. The repoDB dataset is provided by Brown et al. [38], in which, negative and positive samples were clearly defined and large enough for classification tasks. The repoDB dataset contains 1571 drugs and 2051 diseases, 6677 have been approved drug-disease associations and 4123 failed drug-disease associations. Five-fold cross-validation is also used on the repoDB dataset. The experimental results of the SSLDR model and the comparison model on the repoDB dataset are shown in Table VI.

Observing Table VI, it can be found that the AUPR and F1-Score of the SSLDR model proposed in this work are 0.267 and 0.346, respectively, which are better than all the comparison models. And it is noteworthy that the performance of the SSLDR model is 12.6% and 8.2% higher than that of the SSLDR-M and SSLDR-A models, respectively. This indicate that the SSLDR model proposed in this work is also applicable to the repoDB dataset.
H. Effectiveness of the SSLDR Framework on Other Models

In order to verify whether the self-supervised learning framework proposed in this work can be applied to other computational drug repositioning models as the data augmentation strategy, we set up the following experiments. We chose two models, one is GMF, a benchmark model for computational drug repositioning, and the other is one of the current state-of-the-art models, NMFDR [23]. Taking the GMF model as an example, we replace the model of the main task in SSLDR with the GMF model, and then jointly optimize it with the auxiliary tasks. In this way, the SSLDR framework can be used as the data augmentation strategy to improve the latent factor of drugs in the GMF model. The same operation is also adopted for the NMFDR model.

The experimental results are shown in Table VII. We found that the experimental performance of GMF and NMFDR models improved under all evaluation metrics after using SSLDR as the data augmentation strategy. The performance improvements of GMF model on the four datasets are 25.2%, 22.2%, 21.9% and 4.1%, respectively. The performance improvements of NMFDR model on the four datasets are 5%, 5.3%, 11.9% and 2%, respectively.

In addition, we found that the above experimental results have a small improvement on the repoDB dataset. This is because the training data in the repoDB dataset is sufficient, and the latent factor of the drug has been well trained. The great improvement in the first three sparse datasets shows that the SSLDR framework can improve the generalization ability of the GMF model and the NMFDR model. And it is worthy to note that the experimental results of the NMFDR model on the four datasets are better than SSLDR. However, after using SSLDR as the data augmentation strategy, the prediction performance of the NMFDR model is further improved. This shows that the SSLDR framework proposed in this work has certain value as the data augmentation strategy, which can improve the generalization ability of other models.

| Dataset       | Gottlieb | Cdataset | repoDB |
|---------------|----------|----------|--------|
|               | AUPR     | F1-Score | AUPR   |
|               |          |          | F1-Score | AUPR | F1-Score | AUPR | F1-Score |
| GMF           | 0.015    | 0.048    | 0.018  | 0.042 | 0.007 | 0.026 | 0.085 | 0.164 |
| GMF+SSLDR     | 0.021    | 0.033    | 0.023  | 0.049 | 0.009 | 0.03  | 0.088 | 0.165 |
| NMFDR         | 0.202    | 0.266    | 0.279  | 0.322 | 0.057 | 0.123 | 0.36  | 0.396 |
| NMFDR+SSLDR   | 0.214    | 0.277    | 0.294  | 0.339 | 0.066 | 0.133 | 0.363 | 0.401 |

IV. CONCLUSION

In this work, we propose a multi-task self-supervised learning framework SSLDR for the problem of label sparseness in computational drug repositioning. Under the strategy of joint training, the framework uses auxiliary tasks to improve the latent factor of drugs to enhance the generalization performance of the main task "drug-disease association prediction". And we propose a multi-input decoder to improve the ability of the autoencoder to mine latent factors of drugs or diseases.

Experimental results on multiple real-world datasets validate the superiority of the multi-task self-supervised learning framework and multi-input decoder.

For future work, we plan to explore how to improve the latent factor of disease so that it can be better applied to computational drug repositioning scenarios. In addition, the framework proposed in this work is based on the matrix factorization model, and how to apply this framework to graph-based model is also the direction of our future work.

REFERENCES

[1] M. Dickson and J. P. Gagnon, “Key factors in the rising cost of new drug discovery and development,” Nature Rev. Drug Discov., vol. 3, no. 5, pp. 417–429, 2004.
[2] N. A. Tamimi and P. Ellis, “Drug development: From concept to marketing!,” Nephron Clin. Pract., vol. 113, no. 3, pp. c125–c131, 2009.
[3] T. T. Ashburn and K. B. Thor, “Drug repositioning: Identifying and developing new uses for existing drugs,” Nature Rev. Drug Discov., vol. 3, no. 8, pp. 673–683, 2004.
[4] N. Nosengo, “Can you teach old drugs new tricks?,” Nature News, vol. 534, no. 7607, 2016, Art. no. 314.
[5] J. L. E. Pritchard, T. A. O. Mara, and D. M. Glubb, “Enhancing the promise of drug repositioning through genetics,” Front. Pharmacol., vol. 8, 2017, Art. no. 896.
[6] M. Lotfi Shahrerea, N. Ghafti, S. R. Mousavib, J. Varshosaza, and J. R. Green, “A review of network-based approaches to drug repositioning,” Brief. Bioinf., vol. 19, no. 5, pp. 878–892, 2018.
[7] B. Padhy et al., “Drug repositioning: Re-investigating existing drugs for new therapeutic indications,” J. Postgraduate Med., vol. 57, no. 2, 2011, Art. no. 153.
[8] S. S. Sadeghi and M. R. Keyvanpour, “An analytical review of computational drug repurposing,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 18, no. 2, pp. 472–488, Mar./Apr. 2019.
[9] J. K. Yella, S. Yaddanapudi, Y. Wang, and A. G. Jegga, “Changing trends in computational drug repositioning,” Pharmaceuticals, vol. 11, no. 2, 2018, Art. no. 57.
[10] H. Luo, M. Li, M. Yang, F.-X. Wu, Y. Li, and J. Wang, “Biomedical data and computational models for drug repositioning: A comprehensive review,” Brief. Bioinf., vol. 22, no. 2, pp. 1604–1619, 2021.
[11] A. Gottlieb, G. Y. Stein, E. Ruppin, and R. Sharan, “Predict: A method for inferring novel drug indications with application to personalized medicine,” Mol. Syst. Biol., vol. 7, no. 1, 2011, Art. no. 496.
[12] H. Chen, F. Cheng, and J. Li, “iDrug: Integration of drug repositioning and drug-target prediction via cross-network embedding,” PLoS Comput. Biol., vol. 16, no. 7, 2020, Art. no. e1008040.
[13] J. Yang et al., “Computational drug repositioning based on the relationships between substructure–indication,” Brief. Bioinf., vol. 22, no. 4, 2021, Art. no. bbaa348.
[14] H. Luo, M. Li, S. Wang, Q. Liu, Y. Li, and J. Wang, “Computational drug repositioning using low-rank matrix approximation and randomized algorithms,” Bioinformatics, vol. 34, no. 11, pp. 1904–1912, 2018.
[15] X. Yang et al., “Additional neural matrix factorization model for computational drug repositioning,” BMC Bioinf., vol. 20, no. 1, pp. 1–11, 2019.
[16] J. He et al., “Hybrid attentional memory network for computational drug repositioning,” BMC Bioinf., vol. 21, no. 1, pp. 1–17, 2020.
[17] W. Wang, S. Yang, X. Zhang, and J. Li, “Drug repositioning by integrating target information through a heterogeneous network model,” Bioinformatics, vol. 30, no. 20, pp. 2923–2930, 2014.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
[18] H. Luo et al., “Drug repositioning based on comprehensive similarity measures and bi-random walk algorithm,” Bioinformatics, vol. 32, no. 17, pp. 2664–2671, 2016.

[19] X. Zeng, S. Zhu, X. Liu, Y. Zhou, R. Nussinov, and F. Cheng, “deepDR: A network-based deep learning approach to in silico drug repositioning,” Bioinformatics, vol. 35, no. 24, pp. 5191–5198, 2019.

[20] Y. Meng, C. Lu, M. Jin, J. Xu, X. Zeng, and J. Yang, “A weighted bilinear neural collaborative filtering approach for drug repositioning,” Brief. Bioinf., vol. 23, no. 2, 2022, Art. no. bbab581.

[21] W. Zhang, H. Xu, X. Li, Q. Gao, and L. Wang, “DRIMC: An improved drug repositioning approach using Bayesian inductive matrix completion,” Bioinformatics, vol. 36, no. 9, pp. 2839–2847, 2020.

[22] Y. Yan, M. Yang, H. Zhao, G. Duan, X. Peng, and J. Wang, “Drug repositioning based on multi-view learning with matrix completion,” Brief. Bioinf., vol. 23, no. 3, 2022, Art. no. bbbc054.

[23] X. Yang, G. Yang, and J. Chu, “The neural metric factorization for computational drug repositioning,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 20, no. 1, pp. 731–741, Jan./Feb. 2022.

[24] M. Yang, G. Wu, Q. Zhao, Y. Li, and J. Wang, “Computational drug repositioning based on multi-similarities bilinear matrix factorization,” Brief. Bioinf., vol. 22, no. 4, 2021, Art. no. bbaa267.

[25] M. Yang, L. Huang, Y. Xu, C. Lu, and J. Wang, “Heterogeneous graph inference with matrix completion for computational drug repositioning,” Bioinformatics, vol. 36, no. 22/23, pp. 5456–5464, 2020.

[26] S. Li and X. Pan, “A computational drug repositioning model based on hybrid similarity side information powered graph neural network,” Future Gener. Comput. Syst., vol. 125, pp. 24–31, 2021.

[27] G. Larsson, M. Maire, and G. Shakhnarovich, “Learning representations for automatic colorization,” in Proc. Eur. Conf. Comput. Vis., Springer, 2016, pp. 577–593.

[28] T. Yao et al., “Self-supervised learning for large-scale item recommendations,” in Proc. 30th ACM Int. Conf. Inf. Knowl. Manage., 2021, pp. 4321–4330.

[29] C. Knox et al., “Drugbank 3.0: A comprehensive resource for ‘OMICS’ research on drugs,” Nucleic Acids Res., vol. 39, no. suppl_1, pp. D1035–D1041, 2010.

[30] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, “Online Mendelian inheritance in man (OMIM), a knowledgebase of human genes and genetic disorders,” Nucleic Acids Res., vol. 33, no. suppl_1, pp. D514–D517, 2005.

[31] D. Weininger, “Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules,” J. Chem. Inf. Comput. Sci., vol. 28, no. 1, pp. 31–36, 1988.

[32] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, and E. Wilhagen, “The chemistry development kit (CDK): An open-source Java library for chemo-and bioinformatics,” J. Chem. Inf. Comput. Sci., vol. 43, no. 2, pp. 493–500, 2003.

[33] Y. Koren, “Factorization meets the neighborhood: A multifaceted collaborative filtering model,” in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2008, pp. 426–434.

[34] W. S. Noble, “What is a support vector machine?,” Nature Biotechnol., vol. 24, no. 12, pp. 1565–1567, 2006.

[35] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural collaborative filtering,” in Proc. 26th Int. Conf. World Wide Web, 2017, pp. 173–182.

[36] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[37] S. Zhang, L. Yao, B. Wu, X. Xu, X. Zhang, and L. Zhu, “Unraveling metric vector spaces with factorization for recommendation,” IEEE Trans. Ind. Informat., vol. 16, no. 2, pp. 732–742, Feb. 2019.

[38] A. S. Brown and C. J. Patel, “A standard database for drug repositioning,” Sci. Data, vol. 4, no. 1, pp. 1–7, 2017.

Xinxing Yang received the BS and MS degrees, in 2013 and 2017 respectively. He is currently working toward the PhD degree with the Department of Automation, Shanghai Jiao Tong University. His current research interests include data mining, recommender systems, machine learning and its applications in drug discovery.

Genke Yang received the BS degree in mathematics from Shanxi University, in 1984, the MS degree in mathematics from Xi’nan Normal University, in 1987, and the PhD degree in systems engineering from Xi’an Jiaotong University, in 1998. He has been a full-time professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. He is currently a member of the Collaborative Innovation Center for Advanced Ship and Deep-Sea Exploration, Shanghai. His research interests include supply chain management, logistics, production planning and scheduling, discrete event dynamics systems, and computer integrated manufacturing.

Jian Chu received the BS, MS, and PhD degrees from Zhejiang University, Hangzhou, China, in 1982, 1984, and 1989, respectively, and the PhD degree in joint education program from Zhejiang University and Kyoto University, Kyoto, Japan. He was a post-doctoral researcher with the Institute of Advanced Process Control, Zhejiang University, where he was a full professor in 1993, and a doctoral advisor in 1994. He is now the chief researcher of Shanghai Jiao Tong University. His current research interests include control theory and applications, research and development of computer control systems, and advanced process control software.