# Deep Graph Networks for Drug Repurposing with Multi-Protein Targets

Davide Bacciu, *Senior Member, IEEE*, Federico Errica, Alessio Gravina*, Lorenzo Madeddu, Marco Podda, Giovanni Stilo

***Abstract*—In the early phases of the COVID-19 pandemic, repurposing of drugs approved for use in other diseases helped counteract the aggressiveness of the virus. Therefore, the availability of effective and flexible methodologies to speed up and prioritize the repurposing process is fundamental to tackle present and future challenges to worldwide health. This work addresses the problem of drug repurposing through the lens of deep learning for graphs, by designing an architecture that exploits both structural and biological information to propose a reduced set of drugs that may be effective against an unknown disease. Our main contribution is a method to repurpose a drug against multiple proteins, rather than the most common single-drug/single-protein setting. The method leverages graph embeddings to encode the relevant proteins' and drugs' information based on gene ontology data and structural similarities. Finally, we publicly release a comprehensive and unified data repository for graph-based analysis to foster further studies on COVID-19 and drug repurposing. We empirically validate the proposed approach in a general drug repurposing setting, showing that it generalizes better than single protein repurposing schemes. We conclude the manuscript with an exemplified application of our method to the COVID-19 use case. All source code is publicly available.**

***Index Terms*—deep graph networks, graph neural networks, drug repurposing, COVID-19.**

## I. Introduction

The COVID-19 pandemic has undoubtedly revolutionized our lives, calling for a response that has required a coordinated effort worldwide and across multiple disciplines. Often, such interdisciplinary collaborations comprise Artificial Intelligence (AI) expertise needed for data exploitation and properly manage the crisis. The study presented in this paper was developed within the framework of one of such collective endeavour. Namely, we report the outcomes of a volunteering initiative developed within CLAIRE, the Confederation of Laboratories for AI Research in Europe (CLAIRE), to help tackle the COVID-19 pandemic [1]. In particular, this work describes some significant outcomes developed by the Bioinformatics research group as concerns deep graph networks for drug repurposing.

* Corresponding Author

D. Bacciu, F. Errica, A. Gravina, and M. Podda are with the Department of Computer Science, University of Pisa, Italy. (e-mail: {bacciu, marco.podda}@di.unipi.it and {federico.errica, alessio.gravina}@phd.unipi.it)

L. Madeddu is with the Department of Translational and Precision Medicine, Sapienza University of Rome, Italy. (e-mail: lorenzo.madeddu@uniroma1.it)

G. Stilo is with the Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Italy. (e-mail: giovanni.stilo@univaq.it)

While part of the medical research community was (and still is) focused on studying the virus and producing new vaccines, an equal effort has been put into devising therapeutic treatments for those afflicted by COVID-related diseases. In this respect, drug repurposing, i.e., the use of available drugs to treat different diseases than the ones for which they were developed, offers an appealing alternative to traditional drug discovery. The idea is to provide effective therapies in a timely manner by using what is available in the drug market, rather than investing massive amounts of time and resources to develop novel drugs. The very first stage of drug repurposing involves *in silico* screening of available drugs amenable to repurposing. In this phase, machine learning methods are often used to select promising candidates and improve the throughput of the subsequent *in vitro* or *in vivo* studies.

Clearly, the screening for drugs to repurpose is tailored on a specific biological target, typically represented as a set of specific proteins. Such proteins, however, should not be considered as independent and isolated entities. Rather, considering them as compounds of functionally related entities often improves the final outcome, allowing to select stronger candidates. This has motivated a whole body of research leveraging the functional relationships between proteins represented as a network of protein-protein interactions. By complementing such representation with known drug-protein associations, the problem of drug repurposing can be cast as a predictive task on the resulting network of protein-protein-drug associations. The task is generally framed as a link prediction problem, where the link between known drugs and proteins of interest must be discovered [2], [3]. The main differences between these methods lie in how proteins, drugs, and the protein-drug interaction network are represented and processed altogether. With the recent re-discovery of deep learning for graphs [4], we approach the problem from a structure-aware perspective, trying to jointly exploit the relationships among proteins and their cross-interactions with the molecular structure of drugs.

Our contribution is three-fold and can be summarized as follows. First, we introduce a method to repurpose a drug given a set of multiple target proteins considered jointly, rather than assessing single-protein/single-drug associations (as widespread in the literature). By this, we claim to better capture the richness of biological targets, which cannot be fully represented by the latter. Secondly, we discuss different embedding methods for proteins (with Node2Vec) and drugs (with Deep Graph Networks) able to encode their functional and chemical information, respectively, into vectorial representations. Lastly, in a joint effort with the CLAIRE consortium, we have systematically aggregated information gathered from

different sources to release them publicly as a networked resource aimed at fostering research on COVID-19 related topics[1]. With the same spirit, we also release openly the code implementing our methodology and reproducing our empirical analysis[2].

We provide an experimental validation comparing single-protein and multi-protein repurposing scenarios on generic drugs. We achieve AUROC values greater than 0.82 for the former and 0.92 for the latter scenario, which hints at the benefit and potential advantage of considering a richer representation of the biological target through protein ensembles for the drug repurposing task. We also analyze the robustness of the proposed approach with respect to changes in the size of the multi-protein target set. The experiments show that the proposed approach can yield high-precision repurposing results even when only a fraction of the related protein targets are used, which is especially helpful in those applications where the disease-related knowledge is not yet complete (i.e., when a novel disease is discovered). Lastly, we apply our multi-protein method to repurpose existing drugs against a repertoire of COVID-19 proteins and compare it to the single-protein approach. While assessing the therapeutic potential of the repurposed drugs predicted by our model goes beyond the scope of this work, the results further highlight the differences between our strategy presented in this work in comparison to the standard approach.

The rest of the paper is structured as follows: in Section II we review graph representation learning and machine learning for drug repurposing, which constitute the backbone of our approach. In Section III we describe our methodology in detail. Section IV describes the biological networks used in the study. In Section V, we present the experiments and their setup, in conjunction with the discussion of the obtained results. Finally, in Section VI, we highlight the advancements that we produced in the field.

## II. Background

We first provide a general introduction to the field of machine learning for graphs, followed up by a discussion of to the related drug repurposing literature.

### A. Graph Representation Learning

Graph representation learning seeks data-driven methodologies to extract the relevant information from topologically complex structures [4]–[7], in contrast with feature-engineering approaches that leverage prior knowledge to handcraft a graph into a flat vector. Broadly speaking, Deep Graph Networks (DGNs) currently dominate the research landscape of learning from graph-structured data, due to their efficiency and ability to implement an adaptive message-passing scheme between the nodes. This is realized thanks to a local and iterative processing of information: locality means that each node has information about a restricted neighborhood, while iteration allows an efficient exchange of information among nodes which produces representations that are possibly informed by the global structure of the graph. DGNs are commonly used to produce a representation/embedding, i.e., a vector, for each node of the graph. These node embeddings are often summed/averaged to obtain a *graph embedding*, which is fed into a standard machine learning predictor to solve graph regression or classification tasks. Alternatively, node embeddings can be used as-is to address node-related problems. Pioneering works are the Neural Network for Graphs [8] and the Graph Neural Network [9]. While the latter implements the recurrent paradigm of computation with contractive constraints to ensure convergence, the former is designed as a feed-forward network implementing the "spatial convolution" mechanism that is common in today's literature, such as in the Graph Convolutional Network (GCN) [10].

In this work, we use unsupervised approaches to build the graph representations described above from the available raw data. We focus on random walk-based methods that compute node embeddings based on the statistics extracted from such walks. Node2Vec [11] is an unsupervised approach that maps nodes to an embedding space by maximizing the likelihood of preserving the nodes' neighborhoods. This is achieved by performing random walks of fixed length starting from the node to embed, and maximizing the probability that the nodes encountered in the walks co-occur with the target node using a Skip-Gram objective function [12]. Additional hyper-parameters control the trade-off between depth and width of the random walk. In this work, we use Node2Vec to compute representations of proteins starting from three different Gene Ontology (GO) graphs (Section III-D). Note that other techniques such as DeepWalk [13] (of which Node2Vec is a generalized version), LINE [14] and HARP [15], can be employed instead of the chosen one.

Another approach for unsupervised graph representation learning is the Graph Auto-Encoder (GAE) [16], which relies on a Deep Graph Network to reconstruct the graph structure. Depending on the underlying model, GAE takes into account both node, edge and adjacency information to create node embeddings that are similar if they are adjacent and dissimilar otherwise.

### B. Drug Repurposing

In the era of COVID-19 and its variants, effective and fast drug development has become urgent. However, the discovery of new drugs is a high-risk and expensive process that can last years. Recent studies [17], [18] estimated that developing a new drug costs 2–3 billion dollars and takes roughly 12 years. To overcome these limitations, researchers are resorting to drug repurposing, which was shown to be more efficient and safer than traditional drug development. Drug repurposing, or repositioning, identifies new clinical indications or therapeutic effects for approved drugs. Yet, most of the novel drug-disease associations are mainly due to serendipity or come from intuitions based on biological knowledge [19]. That explains why, in the last decade, several computational approaches have been proposed to filter (*in silico*) the most promising drug-disease relationships for the (*in vitro*) clinical and biological experiments.

[1]https://github.com/CLAIRE-COVID-T4/covid-data

[2]https://github.com/gravins/covid19-drug-repurposing-with-DGNs

Computational approaches based on "omics" data processing have proved to be effective for drug repurposing thanks to their capability to extract complex molecular relationships hidden in the mechanisms of our organism [2], [20]–[22]. Most of these methods rely on feature or network-based strategies. Feature-based methods apply machine learning techniques to raw biological data to predict novel drug-disease associations. For instance, Gottlieb et al. [20] compute drug-drug and disease-disease similarities based on omics features. Then, they feed the computed similarities into a logistic regression classifier to predict drug-disease associations. Instead, Ozturk et al. [21] employ a convolutional neural network (CNN) on protein sequences to predict the binding affinity between drugs and genes. Network-based methods discover drug-disease associations by applying network analysis or graph-based machine learning techniques to biological data. Guney et al. [2] apply a network-based proximity measure to the human interactome to predict novel drug-disease associations. They observed that a drug close to a disease in the interactome is more likely to be effective than a distant drug. Zeng et al. [22] developed deepDR, a multi-modal deep auto-encoder that leverages information from 10 heterogeneous biological networks for drug repurposing.

Without a doubt, the COVID-19 pandemic has greatly boosted research at the intersection of biology and network science [23]–[27]. Zhou et al. [23] combined network-proximity measures on the human interactome and gene sequences analysis to repurposing drugs for COVID-19. Gysi et al. [24] rank drug candidates for COVID-19 by using three network-based drug repurposing strategies: network proximity measures, diffusion-based methods, and graph deep learning-based techniques. Zeng et al. [25] construct a Knowledge-Base (KB) network of biological entities such as drugs and diseases and augment it with known relationships of several Coronaviruses (e.g., SARS-CoV-1 and MERS-CoV). Then, they use RotatE [28], a network-based deep-learning model to learn low-dimensional representations of entities and relationship types of the KB and predict drug candidates for COVID-19. Ioannidis et al. [26], similarly to [25], frame the drug repurposing problem as a link prediction task in a biological KB network augmented by gene relationships with SARS-CoV-1, MERS-CoV and COVID-19. For link prediction, they apply a novel GCN-based inductive model to the KB network. Ray et al. [27] predict drug candidates for COVID-19 by applying a variational Graph Auto-Encoder (VGAE) [16] to the Human interactome augmented with drug-target and COVID-19 host-protein interactions.

We remark that these methods are typically hard to compare, as they rely on feature engineering steps on different underlying data sources. In this sense, the novel aspects of these works mainly lie in the feature engineering strategy, combined with a peculiar architecture designed for specific kinds of omics data that leads to significantly different evaluation analyses. For instance, DeepPurpose [29] learns protein and drugs embedding respectively using protein sequences and SMILES sequences. Liu et al. [30] introduce NRLMF, a method which models the probability that a drug would interact with a target by logistic matrix factorization. Differently, [31] discusses a scheme involving pre-training of the model to learn protein and drug embeddings from aminoacid sequences and molecular structures, followed by fine tuning on a graph reconstruction task. The latter targets reconstructing the PPI and the drug-drug interaction network initialized with the pretrained embeddings.

The work of [32] learns embedding representations of the GO-terms associated with a protein; similarly, we also incorporate structural information using the GO directed acyclic graph, but we additionally leverage drug representations that preserve either chemical or structural properties of the drug itself, rather than just encoding the drug into some latent space using a DGN. On a similar note, the KB method of [33] heavily relies on a feature engineering preprocessing step to construct triples of the form (drug, relationship, disease/protein), which are fed into popular transductive KB learning models.

In contrast, we prefer to let the model leverage the relevant interactions from the raw drug-PPI network in an inductive fashion. Thanks to inductive learning, our model easily incorporates new information in the graph as it becomes available without re-training the whole architecture. Finally, the work of Hsieh et al. [34] aggregates multiple sources into a large heterogeneous KB (similarly to what we propose) for training a standard variational graph auto-encoder. Due to our novel architecture and broad choice of the curated data sources, our model adaptively maps a drug to a latent representation. However, it is still hard to fairly compare with other competitors. In addition, to the best of our knowledge, no other studies tested a drug-target interaction prediction model in a multi-protein repurposing scenario.

To sum up, the method we propose is closely related to the link prediction paradigm presented above, in that we also try to predict an interaction between a drug and proteins. However, with respect to the mentioned approaches which only operate by predicting the association between a single drug and a single protein, we will argue that our method can also predict interactions between a drug and a set of proteins. Moreover, we specifically represent proteins according to their function by constructing protein embeddings based on *structural* gene ontology information.

## III. Drug repurposing by deep graph networks with multi-protein targets

This section describes the proposed drug repurposing framework in a top-down fashion. We begin with useful definitions and a high-level description of the architecture before discussing the specific methodologies that encode drugs and proteins into meaningful embeddings.

### A. Mathematical Notation

For the purpose of this work, we define a graph as a tuple $g = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{E})$. The set $\mathcal{V}$ contains the vertices (also called nodes) that represent interacting entities, whereas the set $\mathcal{E}$ explicitly defines the connections between nodes. In this work we deal with undirected graphs, where the connection between two nodes $u$ and $v$ is described by the unordered pair $\{u, v\}$.

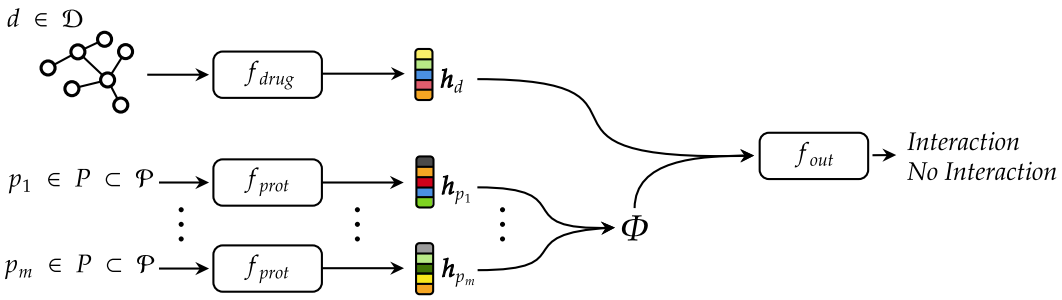


Fig. 1: A high-level overview of the proposed model for drug repurposing. Given a drug $d \in \mathcal{D}$ and a set of proteins $P \subset \mathcal{P}$, the model computes a drug representation $\mathbf{h}_d$ using a drug embedding module $f_{drug}$, and a protein representation $\mathbf{h}_{p_i}$ for each protein $p_i \in P$ using a protein embedding module $f_{prot}$. The protein representations are aggregated into a single vector by the aggregator $\Phi$. Finally, the vectors representing the drug and the set of proteins are passed to an output module $f_{out}$, which computes the desired prediction.

However, the ML models we consider work with directed edges: therefore, each undirected edge is implicitly replaced by two oppositely oriented arcs. We represent node features as a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times |F|}$, where $|F|$ is the number of available features. The $v$-th row of $\mathbf{X}$ is denoted as $\mathbf{x}_v$ and represents a single node's features. Similarly, we represent edge features as a matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}| \times |E|}$, where $|E|$ is the number of edge features, and we indicate edge features' vectors as $\mathbf{e}_{uv}$. Finally, we denote the neighborhood (or adjacency set) of a node $u \in \mathcal{V}$ as the set $\mathcal{N}_u = \{v \in \mathcal{V} \mid \{u, v\} \in \mathcal{E}\}$.

### B. High-Level Overview

In this work, we are given a set of drugs $\mathcal{D}$, and a set of human proteins $\mathcal{P}$. The interactions among the proteins come from the interactome described in Section IV. Given a set of proteins $P \subseteq \mathcal{P}$, we say that a drug $d$ interacts with the set $P$ if it has an effect on the biological process modulated by the interactions among the proteins in $P$. Note that, in some cases, drugs are known to interact only with a single protein, i.e., $P$ could contain only a single protein.

We formulate the drug repurposing task as that of discovering previously unknown interactions among drugs and protein sets of interest. To do so, we organize drugs, proteins, and their known interactions in a dataset $\mathbb{D} = \{\langle d_i, P_i, y_i \rangle\}_{i=1}^{n}$ of triplets, where $d_i \in \mathcal{D}$, $P_i \subseteq \mathcal{P}$, and $y_i \in \{0, 1\}$ is a binary target value telling whether the drug $d_i$ interacts with the proteins in set $P_i$. Thus the objective is, given an unseen combination of a drug and a set of (possibly interacting) proteins, to correctly predict whether the drug interacts with the set or not.

To learn this task, we propose a Deep Learning model comprising three different components:

- a drug embedding module $f_{drug}$;
- a protein embedding module $f_{prot}$;
- an output module $f_{out}$.

In the following, we provide a description of how a triplet $\langle d, P, y \rangle$ is processed by the model at a high level. The process is summarized visually in Figure 1. First, the drug embedding module $f_{drug}$ is used to "featurize" $d$, that is, to compute a representation vector from the molecular graph:

$$\mathbf{h}_d = f_{drug}(d).$$

The details of this module are discussed in Section III-C. Similarly, the proteins $p \in P$ are processed by the protein embedding module $f_{prot}$, which computes a vectorial representation for each protein in the set, focusing on their functional aspects:

$$\mathbf{h}_p = f_{prot}(p).$$

The mechanism by which a protein representation is computed is discussed in Section III-D. Once all the protein representations have been computed, they are aggregated via a permutation invariant operator $\Phi$ to obtain a vector representation of $P$ as follows:

$$\mathbf{h}_P = \Phi(\{\mathbf{h}_p \mid p \in P\}).$$

Finally, the two representations are passed to the output function module $f_{out}$. The result is a prediction:

$$o = f_{out}(\mathbf{h}_d, \mathbf{h}_P),$$

which corresponds to the likelihood of an interaction between $d$ and the set of proteins $P$. The details of the output module are presented in Section III-E. The model is trained to minimize the following loss:

$$\mathcal{L}(\mathbb{D}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{pred}(d_i, P_i, y_i),$$

which corresponds to the Binary Cross-Entropy (BCE) loss accumulated across all the triplets in the dataset:

$$\mathcal{L}_{pred}(d_i, P_i, y_i) = -\mathrm{BCE}(o_i, y_i),$$

where $\mathrm{BCE}(o_i, y_i) = y_i \log(o_i) + (1 - y_i) \log(1 - o_i)$ as usual.

### C. Drug Embedding Module

The purpose of the drug embedding module is to extract a vectorial representation from a drug. We represent drugs as molecular graphs where nodes are atoms, and edges are

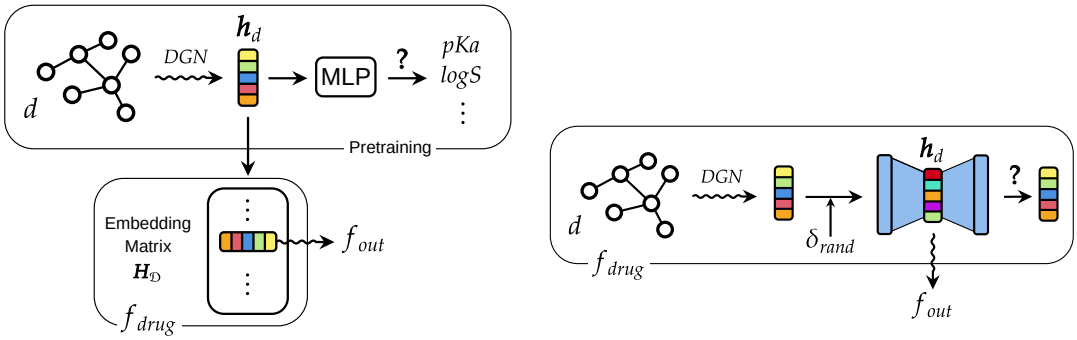


Fig. 2: The two implementations of the drug embedding module $f_{drug}$ considered in this study. In (a), we pretrain a DGN on a property prediction task, and stack the trained DGN representations in a matrix $\mathbf{H}_{\mathcal{D}}$, which is looked up to get the drug embeddings during the training of the drug repurposing model. In (b), the DGN representations are augmented with noise $\delta_{rand}$ and passed to a denoising autoencoder (in pale blue). The hidden state of the autoencoder is passed to the downstream $f_{out}$ module during training in an end-to-end fashion, and the overall loss optimized is augmented with the autoencoder loss.

chemical bonds between them. Bonds can be of four types: single, double, triple or aromatic.

At a high level, the drug embedding module is a DGN which inputs a molecular graph and outputs a representation for each of the graph nodes. Node representations are computed by iteratively applying a series of Graph Convolutional Layers (GCLs), i.e., parameterized transformations that combine the previous representation of a node (initially set to a vector of node features) with the representations of the nodes in its neighborhood. Given the molecular graph $g = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ associated to a drug $d$, a general formulation of the transformation operated by a GCL is the following:

$$\mathbf{x}_v^\ell = \mathrm{GCL}(\mathbf{x}_v^{\ell-1}, \{\mathbf{x}_u^{\ell-1} \mid u \in \mathcal{N}_v\}), \forall v \in \mathcal{V}.$$

In the previous formula, $\ell = 1, \ldots, L$ indexes the current layer, and the initial node representation $\mathbf{x}_v^0$ is a vector of node features. After $L$ GCLs are applied, the node representations are combined to obtain a unique vector representing the whole graph. In this work, this is achieved by first aggregating the node representations at a specific layer into a single vector, and then concatenating each layer-wise representation together. More formally, a drug embedding $\mathbf{h}_d$ is computed as follows:

$$\mathbf{h}_d = \Big\|_{\ell=1}^{L} \gamma(\{\mathbf{x}_v^\ell \mid v \in \mathcal{V}\}),$$

where $\gamma$ is a *global pooling* function that aggregates the node representations into a layer-wise graph representation, and $\|_{l=1}^{L}$ represents concatenation of the graph representations at each layer. In this work, we opted for implementing $\gamma$ as the sum function.

In the drug repurposing experiments, we leverage and test two alternative strategies to obtain the drug embeddings, shown visually in Figure 2. Below, we describe each of them separately.

*a) Pretrained DGN:* the first variant is a DGN pretrained on a property prediction task. The rationale behind the choice of this variant is to provide the downstream output module with drug representations organized by *chemical similarity*: that is, two drugs are considered similar if their chemical properties are similar. To enforce this prior, we pretrain the weights of the drug embedding module on a property prediction task. The purpose of the pretraining is to obtain an embedding matrix $\mathbf{H}_{\mathcal{D}} \in \mathbb{R}^{|\mathcal{D}| \times h}$, where $h$ is the drug embedding dimension, which can be queried to get drug embeddings during the training of the drug repurposing model. The architecture of this variant is composed of a stack of GCLs commonly used in quantum chemistry predictions tasks [35], plus a downstream MLP, hereby termed $\mathrm{MLP}_{prop}$, for property prediction. Precisely, the GCL is defined as:

$$\mathbf{x}_v^\ell = \mathbf{x}_v^{\ell-1} + \sum_{u \in \mathcal{N}_v} \mathrm{MLP}(\mathbf{e}_{uv})\, \mathbf{x}_u^{\ell-1}, \forall v \in \mathcal{V}$$

where $\mathbf{e}_{u,v}$ is the vector of features associated with the edge that connects nodes $u$ and $v$ in the graph. Pretraining is realized through regression tasks for the prediction of the following (continuous) chemical properties: boiling and melting points, solubility, pKa, logS, Octanol-Water partition constant, and Caco-2 permeability, as collected from the DrugBank database [36]. For a given drug $d$, the model is pretrained to minimize the Mean Squared Error (MSE) loss between the properties as predicted by the DGN and a corresponding vector of target properties $\mathbf{y}_d$:

$$\mathrm{MSE}(\mathrm{MLP}_{prop}(\mathbf{h}_d), \mathbf{y}_d).$$

After pretraining, and during the training procedure of the drug repurposing model, the embedding of a drug $d$ is obtained as $\mathbf{h}_d = \mathbf{H}_{\mathcal{D}}(d)$, where $\mathbf{H}_{\mathcal{D}}(d)$ denotes a lookup operation for the drug embedding of $d$ on the pretrained drug embedding matrix.

*b) End-to-end DGN with denoising Autoencoder:* the second variant consists of a stack of GCLs combined with a downstream Denoising Autoencoder (dAE) [37]. The rationale behind this architectural choice is to provide the output module with drug representations organized by *structural similarity*: that is, two drugs are considered similar if their structure is similar. To enforce this prior, we use a GCL architecture

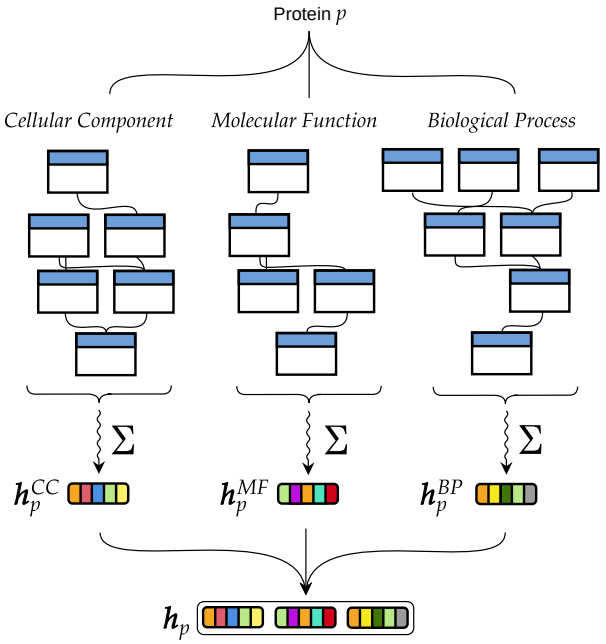


Fig. 3: The schematics of the protein embedding module $f_{prot}$. Given a protein $p$, we fetch the Node2Vec embeddings of the associated GO terms in the three GO DAGs (shown in the figure as connected white boxes with pale blue headers). The embeddings are summed together to form a GO-wise representation (e.g., the vector $\mathbf{h}_p^{CC}$ for the Cellular Component ontology). The three embeddings are finally concatenated together to form the protein representation $\mathbf{h}_p$.

suitable for unsupervised learning, and use the dAE to regularize the training objective of the drug repurposing model. Specifically, we adopt the GCL of Hu et al. [38], where the node representations are computed as follows:

$$\mathbf{x}_v^\ell = \text{MLP}(\mathbf{x}_v^{\ell-1} + \sum_{u \in \mathcal{N}_v} \text{ReLU}(\mathbf{x}_u^{\ell-1} + \mathbf{e}_{uv})), \forall v \in \mathcal{V}.$$

Once the drug representation $\mathbf{h}_d$ is computed by the DGN, it is corrupted with Gaussian noise $\delta_{rand}$ and passed as input to a standard 1-layer AE which is trained to reconstruct the original representation. More precisely, the regularizing objective by which the dAE is trained is the following:

$$\mathcal{L}_{AE}(d_i) = \text{MSE}(\text{AE}(\mathbf{h}_d + \delta_{rand}), \mathbf{h}_d).$$

### D. Protein Embedding Module

The purpose of the Protein Embedding module is to extract the information about proteins in vectorial form. Differently from the drug embedding module, we do so considering their *function*, and not their *structure*. Specifically, we leverage Gene Ontologies (GOs) [39], which are hierarchies of predefined semantic labels and their relations, characterizing the function of genes and their products. Essentially, proteins are annotated with one or more GO terms, belonging to three different domains. Each of these domains has its own GO, represented as a Directed Acyclic Graph (DAG) where nodes are terms and relations among them are of the kind "is_a". The three domains are:

- Cellular Component (CC), i.e., terms related to the cellular structure. The corresponding DAG comprises 4183 nodes and 4727 edges;
- Molecular Function (MF), i.e., terms related to the activity. The corresponding DAG comprises 11125 nodes and 13575 edges;
- Biological Process (BP), i.e., terms related to he biological function performed. The corresponding DAG comprises 29211 nodes and 56398 edges.

In GOs, the similarity among proteins is conveyed by the associated GO terms: two proteins are considered similar if they share many GO terms, or, more loosely, if they share many parent GO terms. To effectively encode these relationships, we apply the graph representation learning algorithm Node2Vec [11] to the *unweighted* DAG representing each ontology. We recall that Node2Vec works by collecting a set of random walks for each node of a target graph (hereby, the DAG representing the GO); crucially, GO terms (i.e., nodes of our DAG) with similar sets of random walks will have similar representations. The depth vs. width trade-off of the walks is controlled by two parameters, $\alpha$ and $\beta$. Each walk in a GO DAG is composed of a sequence of terms, represented as one-hot vectors, and two nodes are considered similar if their corresponding random walk sequences share many co-occurrences. To learn this similarity, the terms contained in the sequences are embedded using a skip-gram model [12]. The application of Node2Vec to the nodes of each DAG yields three embedding matrices, one for each GO domain: $\mathbf{E}^{CC} \in \mathbb{R}^{4183 \times z}$, $\mathbf{E}^{MF} \in \mathbb{R}^{11125 \times z}$, and $\mathbf{E}^{BP} \in \mathbb{R}^{29211 \times z}$. In each embedding matrix, the $k$-th row identifies the $z$-dimensional embedding of the $k$-th GO term. To embed the proteins with these three matrices, we proceed as follows. We first collect, from the AmiGO database [40], the GO terms by which the proteins are annotated. Let $T_p^O = \{t_1^O, t_2^O, \ldots\}$ be the generic set of GO terms associated to a protein $p \in \mathcal{P}$, with $O \in \mathcal{O} = \{CC, MF, BP\}$. For a given GO domain, the representation associated to $p$ is the vector:

$$\mathbf{h}_p^O = \sum_{t \in T_p^O} \mathbf{E}^O(t),$$

where the notation $\mathbf{E}^O(t)$ indicates the vector obtained selecting the row of $\mathbf{E}^O$ corresponding to the GO term $t$. Finally, the representation of the protein is obtained by concatenating these three intermediate representations together:

$$\mathbf{h}_p = \Big\|_{O \in \mathcal{O}} \mathbf{h}_p^O.$$

The overall architecture of the protein embedding module is shown in Figure 3. Notice that the protein embedding module is pretrained: during the training of the drug repurposing model, the protein embeddings are provided as a matrix, and the process of computing a protein embedding simply amounts to an embedding lookup.

### E. Output Module

The output module receives as input a drug and a proteins set embedding, and outputs a prediction which corresponds to the likelihood of an interaction between the drug and the proteins set. To compute a vector representation of the set of proteins, we aggregate all protein representations using a permutation invariant function $\Phi$; such a function allows to generalize to graphs of arbitrary sizes and requires no ordering of the nodes in the graph. Because different drugs are associated with protein sets of different cardinalities, the choice of the aggregation function may have an impact on generalization performances. Thus, we evaluate the performance of *mean*, *max*, and *sum* aggregations. The protein set representation is then concatenated with the drug representation provided by the drug embedding module, and the result is fed to an MLP that computes the desired interaction likelihood. Depending on the used drug embedding module, learning happens only at the MLP level or in an end-to-end fashion (i.e., the weights of the drug embedding module are updated together with those of the output MLP). In the latter, the overall objective function minimized by the drug repurposing model becomes:

$$\mathcal{L}(\mathbb{D}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{pred}(d_i, P_i, y_i) + \mathcal{L}_{AE}(d_i).$$

Importantly, during the training of the drug repurposing model, we do not pass the drug embeddings $\mathbf{h}_d$ as computed by the DGN, but we pass the hidden state of the AE instead (even though we slightly abuse the notation $\mathbf{h}_d$ to still indicate a drug embedding). Therefore, the dAE loss can be seen as a regularizer which imposes a smoothness constraint on the drug representations. Please refer to Figure 1 for a visual representation of the overall process.

## IV. A curated network of protein-drug-gene data for drug-repurposing

As outlined in Section III-B, we test the proposed method in a link prediction fashion using an augmented human interactome network for drug repurposing. To construct the augmented human interactome network, as well as the drug and protein embeddings (see Sections III-C and III-D), we use the data available in our public repository (See Section I), which integrates curated omics' information from biomedical literature. The purpose of this repository is to actively support the COVID-19 research community by collecting in one place all the clinical evidence on COVID-19 and the human genomic and proteomic information. The repository contains data characterizing molecular aspects of human diseases, drugs and protein-protein interactions between the human organism and the COVID-19.

The chosen biological network is a human interactome - a network where nodes are human's proteins and links are their interactions - augmented by drug-protein relationships. The human proteins (nodes) interacting with the COVID-19 are labelled as disease proteins. Hereafter, we briefly discuss the biological characteristics of the data used to build the reference network:

- Protein-Protein Interaction: Protein-Protein Interactions (PPIs) are physical interactions between two or more proteins. A relevant finding of the interactome is that proteins involved in the same processes can cluster together in the network. Protein-protein interactions are important because they allow us to understand a protein's function and its behaviour. The repository contains $217.161$ interactions among $15.970$ human proteins. The PPIs were previously collected by [41] from 15 popular databases (e.g., BioGRID, HPRD, MINT, IntAct, etc.) based on several kinds of high-quality experimental evidences (e.g., Yeast 2-Hybrid, mass spectrometry, etc.).
- Drug-protein interaction: A drug is designed to produce a specific desirable therapeutic effect on the target organism. The relation between a drug and the target molecules of the organism, usually a protein, is named drug-target association or interaction. The repository contains $46.235$ drug-host interactions yielded by $6.605$ drugs. The drug-target interactions were previously collected by [41], [42].
- COVID-19 host proteins: Human protein interacting with the COVID-19 virus and involved its pathogenic mechanisms. The repository includes the $332$ human proteins associated with COVID-19 discovered by [43].

## V. Experiments

In the following, we describe the experiments conducted to measure the effectiveness and the robustness of our approach.

### A. Setup

We consider two main experimental scenarios. The first, here referred to as *Single-protein task*, considers the common setting of predicting the interaction between a single protein and a drug. For this task, given a protein $d$, we constructed one sample $\langle d, \{p\}, y = 1\rangle$ for every protein $p$ that is associated with $d$ in the interactome. In short, the task consists of inferring the relationship between the protein and the drug. The second task, here referred to as *Multi-protein task*, constitutes one of the novelties of this work. In this scenario, a sample has the form $\langle d, P, y = 1\rangle$, where $P \subset \mathcal{P}$ is a set of proteins associated with a drug $d$. The task consists of inferring the association between sets of protein of variable size and a target drug. In this work, we experiment with two different $\Phi$ variants: either $\Phi_{sum} = sum(R) \,||\, max(R)$, or $\Phi_{mean} = mean(R) \,||\, max(R)$, where $sum$, $mean$, and $max$ are performed element-wise, and $||$ denotes concatenation.

### B. Negative Samples Generation

In drug repurposing tasks, negative samples are not generally available; in fact, literature tells us if a protein (set) interacts with a drug while the opposite is typically unknown. Most of the existing drugs interact with specific protein families, such as enzymes and receptors [44]. Proteins which bind easily with a drug are called druggable targets, while harder binding proteins are called undruggable. Drug databases such as DrugBank do not typically associate drugs to information about not interacting or undruggable proteins. For this reason,

we constructed a balanced dataset with negative samples generated via a task-specific data augmentation strategy. For the *Single-protein task*, we created a negative sample by fixing a drug $d$ and randomly selecting one protein from the subset $P^- = \{p \in \mathcal{P} \mid p \notin P\}$, which contains the proteins not associated with $d$. Instead, in the *Multi-protein task*, we created $\theta$ negative triplets for each positive sample. The negative triplets are constructed by fixing the drug $d$ and uniformly choosing a set $\bar{P}$ of proteins from $P^-$. The $\theta$ sampled $\bar{P}$ has a cardinality that follows the distribution of the cardinality of all the drug related positive sets $P$. In this case, we have uniformly chosen to generate $\theta = 5$ negative sets which follow these ranges of cardinality: $[1], [2], [3], [4-9], [10-300]$. This process ended up producing $\theta \cdot |P|$ negative triplets, or in other words, $|P|$ negative triplets for each of the $\theta$ cardinality ranges.

### C. Experimental protocol and implementation details

To perform risk assessment, we split the data according to a hold-out strategy into training (80%) and test (20%) sets, making sure that a link does not appear in both sets. Internally to the training set, we used a 5-fold cross validation schema for model selection. Specifically for the *Multi-protein task*, data was stratified according to two strategies: either $i$) the target $y$; or $ii$) both the target $y$ and the cardinality of the set of proteins. Throughout the experiments, we optimized the Area Under the ROC curve (AUROC), which is a good estimate of the classification performances since the dataset is balanced.

We recall that each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. Moreover, in the experiments we measured Recall (i.e., the ratio between true positive and the actual true samples), Precision (i.e., the ratio between true positive and the samples predicted as true), and the F1 score (i.e., the harmonic mean of precision and recall). Lastly, we performed a DeLong's test [45], to assess if the AUROCs of a pair of models are statistically significantly different.

We considered as drug features the information related to atoms and bonds. In particular, for the atoms we leveraged the one-hot-encoding of symbol, type of hybridization, number of hydrogens, and degree. For the bonds we considered the one-hot-encoding of the stereo configuration, type (i.e., single, double, triple, or aromatic), and if the bond is in a ring.

As described in Section III-C, we evaluated two different model architectures, which differ in the drug embedding module they use internally. The former is the one where the drug embedding module is pretrained on the chemical property prediction task, and it is a 3-layer DGN. We refer to this architecture as *Chemical-Similarity-based prediction Network (CSN)*. On the other hand, the latter exploits structural similarity, is based on a dAE, and it is trained end-to-end. Similarly to the previous configuration, we used a 3-layer DGN. We refer to this architecture as *Structural-Similarity-based prediction Network (SSN)*. Both drug-embedding modules generate embeddings of dimension 96. For the protein embedding module, we applied Node2Vec with $\alpha = \beta = 1$, an embedding dimension of 128, a context window of size 7, and a total of 5 training epochs.

We compared SSN and CSN with four baselines in the Single-protein task. The first two are DeepDTA [46] and GraphDTA [47], which are state of the art approaches in the domain of drug-protein affinity prediction. The last two baselines leverage protein embeddings generated using Node2Vec over the PPI network and drug embeddings computed with Extended-Connectivity Fingerprint (ECFP) method [48]. The first baseline transforms drugs and proteins representations into a hidden embedding space. Then, it computes the dot product to predict the interaction between a drug-protein pair. The second approach feeds into an MLP the concatenation of drug and protein embeddings. In this case, Node2Vec's hyper-parameters are set as before except for the embedding dimension which is scaled to 384 in order to have the same length as our methodology. We employed the ECFP method with fingerprint length of 1024 and radius equal to 3.

We performed hyper-parameter tuning via grid-search. The grid contains:

- *AdamW* optimizer [49] with three learning rates, i.e., $2e^{-3}$, $2e^{-4}$, $2e^{-5}$;
- six $f_{out}$'s architecture configurations: 2 hidden layers with dimension of $[512, 64]$, $[256, 32]$, $[128, 16]$; or 3 hidden layers with structure $[512, 128, 32]$, $[256, 64, 16]$, $[128, 64, 32]$;
- nine hidden embedding space dimensions for dot product baseline: 4096, 3000, 2048, 1024, 512, 256, 128, 64, and 32;
- use of batch normalization between $f_{out}$'s hidden layers;
- batch size equal to 512, and 10000 epochs.

We used the same grid for both CSN and SSN models.

### D. Results

We start by analyzing the results of the *Single-protein task*. Table I shows that both CSN and SSN perform better than baseline approaches. Indeed, they achieve a score which is more than 2 points of AUROC higher with respect to the baselines. Specifically, the CSN architecture achieves a test score of approximately 0.83 of AUROC. Notably, the SSN model improves this result by more than 2 points.

| Model config. | Model Selection | | Risk Assessment | | | | |
|---|---|---|---|---|---|---|---|
| | Train | Valid | Train | Test | | | |
| | AUROC | AUROC | AUROC | AUROC | F1 | Recall | Precision |
| DeepDTA | $0.5000_{\pm 0.0000}$ | $0.5000_{\pm 0.0000}$ | 0.5000 | 0.5000 | 0.6667 | 1.0 | 0.5000 |
| GraphDTA | $0.5000_{\pm 0.0000}$ | $0.5000_{\pm 0.0000}$ | 0.5000 | 0.5000 | 0.6667 | 1.0 | 0.5000 |
| DotProd | $0.9862_{\pm 0.0237}$ | $0.7893_{\pm 0.0031}$ | 0.9985 | 0.8006 | 0.9169 | 0.9110 | 0.9260 |
| MLP | $0.9882_{\pm 0.0046}$ | $0.7823_{\pm 0.0050}$ | 0.9953 | 0.7972 | 0.9321 | 0.9395 | 0.9248 |
| CSN | $0.9987_{\pm 0.0004}$ | $0.8197_{\pm 0.0046}$ | 0.9974 | 0.8285 | 0.8309 | 0.8426 | 0.8195 |
| SSN | $0.9995_{\pm 0.0001}$ | $0.8441_{\pm 0.0026}$ | 0.9994 | 0.8549 | 0.8560 | 0.8499 | 0.8621 |

TABLE I: AUROC results for the Single-protein task.

This result is consistent with the end-to-end approach which allows building drug representations that are more suitable for the downstream task. We observe that both DeepDTA and GraphDTA are unable to distinguish between negative and positive samples, always predicting the negative class. We provide further evidence of the strong performance of

our model by comparing the test AUROCs pairwise using DeLong's statistical test [45] at a significance level $\alpha = 0.05$. Recall that the null hypothesis for DeLong's test is that the two AUROCs do not differ statistically. In Fig. 4, we observe that all the AUROCs are statistically different to one each other, with the only exception of GraphDTA compared to DeepDTA.

In the *Multi-protein task*, both models obtain much higher validation and test AUROC scores, in the range of 0.92-0.94; these results suggest that the choice of exploiting sets of proteins may be indeed effective. In Table II, we assess the contribution of the aggregation functions to the performances. It appears that the mean aggregation ($\Phi_{mean}$) is more beneficial for the CSN variant, while the sum aggregation ($\Phi_{sum}$), gives the best results for the SSN variant. More precisely, we measure a difference of 0.55% for the CSN variant, and a difference of 0.73% for the SSN variant. On average, the SSN variant improves with respect to CSN by almost 1%.

| Model config. | Model Selection | | Risk Assessment | | | | |
|---|---|---|---|---|---|---|---|
| | Train | Valid | Train | Test | | | |
| | AUROC | AUROC | AUROC | AUROC | F1 | Recall | Precision |
| CSN($\Phi_{sum}$) | $0.9999_{\pm 0.0002}$ | $0.9168_{\pm 0.0062}$ | 1.0 | 0.9264 | 0.9274 | 0.9412 | 0.9140 |
| CSN($\Phi_{mean}$) | $0.9994_{\pm 0.0006}$ | $0.9212_{\pm 0.0057}$ | 1.0 | 0.9319 | 0.9334 | 0.9557 | 0.9122 |
| SSN($\Phi_{sum}$) | $0.9993_{\pm 0.0011}$ | $0.9300_{\pm 0.0054}$ | 1.0 | 0.9413 | 0.9418 | 0.9514 | 0.9324 |
| SSN($\Phi_{mean}$) | $1.0_{\pm 0.0}$ | $0.9267_{\pm 0.0028}$ | 0.9952 | 0.9340 | 0.9347 | 0.9446 | 0.9249 |

TABLE II: AUROC results for the Multi-protein task with stratification on $y$.

Table III show that the obtained results remain consistent even if the stratification method is changed: specifically also in this case, the SSN architecture performs better than CSN.

| Model config. | Model Selection | | Risk Assessment | | | | |
|---|---|---|---|---|---|---|---|
| | Train | Valid | Train | Test | | | |
| | AUROC | AUROC | AUROC | AUROC | F1 | Recall | Precision |
| CSN($\Phi_{sum}$) | $0.9982_{\pm 0.0016}$ | $0.9182_{\pm 0.0089}$ | 0.9998 | 0.9272 | 0.9278 | 0.9353 | 0.9204 |
| CSN($\Phi_{mean}$) | $0.9996_{\pm 0.0005}$ | $0.9255_{\pm 0.0065}$ | 1.0 | 0.9332 | 0.9342 | 0.9498 | 0.9193 |
| SSN($\Phi_{sum}$) | $0.9992_{\pm 0.0012}$ | $0.9302_{\pm 0.0046}$ | 0.9958 | 0.9361 | 0.9367 | 0.9438 | 0.9296 |
| SSN($\Phi_{mean}$) | $0.9994_{\pm 0.0010}$ | $0.9315_{\pm 0.0023}$ | 1.0 | 0.9429 | 0.9438 | 0.9583 | 0.9298 |

TABLE III: AUROC results for the Multi-protein task with stratification on $y$ and number of associated proteins.

### *E. Assessing robustness to ablated protein sets*

In the last experiment, we assess the robustness of SSN (the best performing one) model in the Multi-protein task where the disease-related knowledge is not yet hypothetically fully discovered.

To do so, we first select the known (positive) interactions in the test set that involve more than 4 proteins. Then, we randomly divide the proteins into 4 distinct groups and generate a set of "ablated" samples as follows: one triplet for each of the 4 groups, another for every combination of 2 and 3 groups out of 4. Hence, each ablated instance contains a protein set as large as 25%, 50%, or 75% of the size of the original proteins set. During the process, we ensured that none of the obtained ablated sets was used by the model during training, to avoiding biased results.

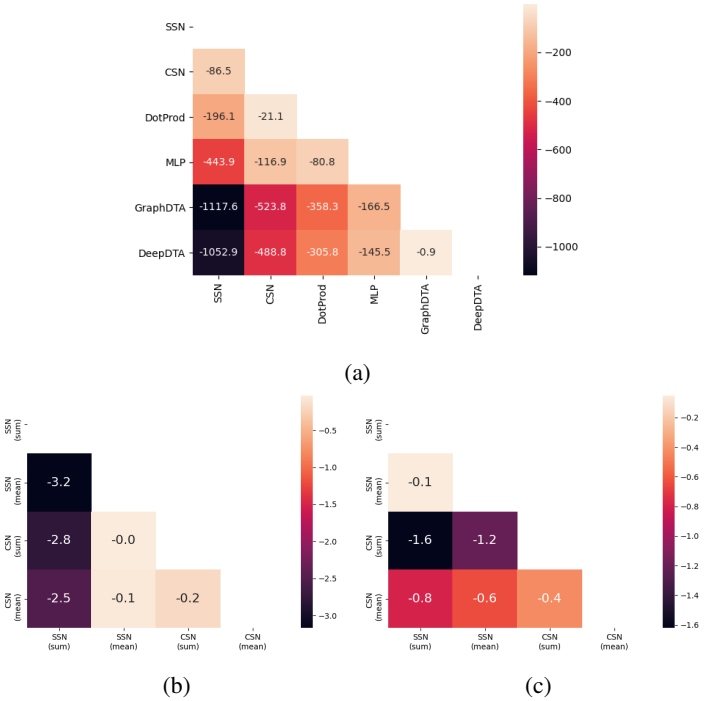


Fig. 4: Log-transformed pairwise $p$-values of DeLong's test (at significance level $\alpha = 0.05$) to compare the AUROCs of the different models in the single-protein scenario. Recall that $\log \alpha = -3$.

Tables IV and V show that, regardless of the aggregation function used, both models maintain a strong generalization capability despite diminishing the number of proteins in the set. Notably, both models maintain a recall score greater than 0.89 when only 25% of the proteins are considered. In particular, it appears that SSN($\Phi_{mean}$) is more robust to using partial knowledge: this is reasonable, as the mean aggregation considers the distribution of the feature values, rather than their absolute value. For both F1 and recall, the model improves the performances by approximately 1 point for each data configuration, i.e., 25-50-75%. Moreover, the model shows 21 less false negatives in the setting with 25% of associated proteins, 34 in the case with 50% of proteins, and 14 with 75%. This corresponds, on average, to a 1.4% reduction of false negatives. Also, the standard deviation computed on the output value is relatively low, especially when the size of the protein subset is greater than 50% of the original. Again, SSN($\Phi_{mean}$) shows lower standard deviations.

| | 25% | 50% | 75% | 100% |
|---|---|---|---|---|
| F1 | 0.9444 | 0.9730 | 0.9852 | 0.9971 |
| Recall | 0.8947 | 0.9473 | 0.9708 | 0.9942 |
| std(prediction drug$_i$) | 0.1533 | 0.0741 | 0.0331 | – |
| True Positive | 1224 | 1944 | 1328 | 340 |
| False Negative | 114 | 108 | 40 | 2 |

TABLE IV: Results of the SSN($\Phi_{sum}$) model when tested with different subsets of proteins.

The True Positive and False Negative values reported in

| | 25% | 50% | 75% | 100% |
|---|---|---|---|---|
| F1 | 0.9529 | 0.9816 | 0.9904 | 0.9971 |
| Recall | 0.9101 | 0.9639 | 0.9810 | 0.9942 |
| std(prediction drug$_i$) | 0.1388 | 0.0440 | 0.0226 | – |
| True Positive | 1245 | 1978 | 1342 | 340 |
| False Negative | 123 | 74 | 26 | 2 |

TABLE V: Results of the SSN($\Phi_{mean}$) model when tested with different subsets of known interactions.

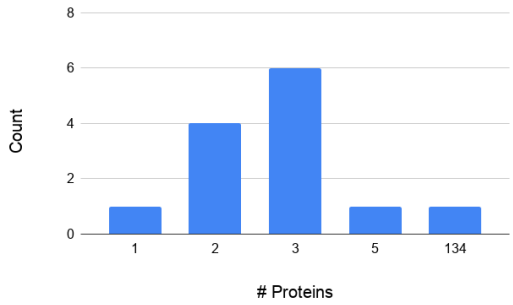


Fig. 5: We show the distribution of the identified COVID-19 protein groups with respect to their dimensions.

Tables IV and V highlight the robustness of the end-to-end architecture. In the worst case, the model wrongly predicts 10% of the true test samples.

### *F. Use case: COVID-19*

Here, we show a concrete example of how the model can be exploited in a real-world drug repurposing scenario, focusing on the COVID-19 disease. Our data consists of 332 COVID-19 related proteins taken from [43], and a set of drug candidates for repurposing taken from the Clinical Trial Summary of DrugBank [50]. To operate in the most neutral setting possible, we proceeded as follows. First, we did not consider drugs for which the SMILES string was not available in our data repository, and filtered out drugs that were not approved, reducing the drug set to 289 candidates. Then, we extracted from the human interactome the connected components to which the 332 proteins belong, removing those with known interactions. From this preprocessing, we obtained 12 groups of COVID-19 related proteins. Figure 5 shows the distribution of the groups sizes, highlighting that most of the groups contain a small number of proteins. We further excluded the biggest component (composed of 134 proteins) from the study, as it is highly implausible that a single drug can interact with a set so large (e.g., in the DrugBank database, a drug interacts with two or three targets on average).

Finally, we predicted the drug-proteins interactions of the 12 groups of proteins and each of the 289 candidate drugs using the best multi-protein model, namely SSN($\Phi_{mean}$). We compared these predictions with those obtained by the same model, when tasked to score the interactions between the same drugs and each of the 332 proteins taken in isolation. The results are reported in Table VI, where we only report for each drug and group the top single-protein interactions for ease of comparison.

| Gene symbols | # Interactions (entire group) | # Interactions (one protein) | Kendall's $\tau$ |
|---|---|---|---|
| {NDUFB9} | 0 | 0 | 0.7822 |
| {ACADM, ETFA} | 182 | 199 | 0.7257 |
| {GOLGA3, GOLGA7} | 0 | 18 | 0.6263 |
| {FASTKD5, NLRX1} | 27 | 17 | 0.5446 |
| {PIGS, GPAA1} | 0 | 1 | 0.5658 |
| {FBN1, FBN2, FBLN5} | 2 | 14 | 0.5586 |
| {TIMM9, TIMM10B, TIMM10} | 0 | 1 | 0.5024 |
| {ERO1B, PLD3, ERP44} | 9 | 205 | 0.0974 |
| {SRP19, SRP54, MDN1} | 0 | 10 | 0.2992 |
| {EIF4H, GLA, MAT2B} | 39 | 105 | 0.5359 |
| {PLEKHF2, RTN4, GFER} | 0 | 174 | 0.4086 |
| {REEP5, GTF2F2, SPART, REEP6, UBXN8} | 0 | 1 | 0.2159 |

TABLE VI: We report, for each analyzed connected component, the numbers of discovered interactions. We also show the Kendall's $\tau$ score for each pair of drug rankings. These results are obtained by SSN($\Phi_{mean}$) in the Multi-protein setting. The gene symbols are associated with the following Entrez Gene IDs: ACADM = 34, EIF4H = 7458, ERO1B = 56605, ERP44 = 23071, ETFA = 2108, FASTKD5 = 60493, FBLN5 = 10516, FBN1 = 2200, FBN2 = 2201, GFER = 2671, GLA = 2717, GOLGA3 = 2802, GOLGA7 = 51125, GPAA1 = 8733, GTF2F2 = 2963, MAT2B = 27430, MDN1 = 23195, NDUFB9 = 4715, NLRX1 = 79671, PIGS = 94005, PLD3 = 23646, PLEKHF2 = 79666, REEP5 = 7905, REEP6 = 92840, RTN4 = 57142, SPART = 23111, SRP19 = 6728, SRP54 = 6729, TIMM10 = 26519, TIMM10B = 26515, TIMM9 = 26520, UBXN8 = 7993.

We compare the two rankings by measuring their concordance with the Kendall's $\tau$ score for rank correlation [51]. We recall that the $\tau$ score lies in the range $[-1, 1]$, where $-1$ means that the two rankings are the reverse of the other, and 1 means that they are identical. In our case, the overall $\tau$ score (obtained by averaging the per-interaction scores reported in the last column of the table) is $0.49 \pm 0.20$, indicating that the two rankings are similar. Notice that the interactions found using the 12 groups are smaller in number if compared with the interactions found by scoring each protein in isolation: more in detail, we found a total of 259 interactions in the former case, against 745 in the latter. This result suggests that using multiple sets of proteins might be more appealing from a practical point of view, since we obtained a smaller set of candidate interactions that are eligible for further *in vitro* analysis.

Lastly, in Table VII we report all the discovered interactions (i.e., those for which the predicted interaction probability is above 0.5) predicted by the model for the 289 drugs and the set of proteins with Entrez ID $\{60493, 79671\}$ (identifying the genes FASTKD5 and NLRX1) further showing the difference of using protein groups against predicting one protein at a time. Notice that, in both cases, the model recognizes drugs tested on multiple COVID-19 clinical trials [50]. Specifically, it identifies 3 drugs that were tested on more than 60 trials (see column CT). In general, it appears that drugs that underwent a larger number of clinical trials are ranked higher by the model

| | *entire group* | | | | *one protein* | | | |
|---|---|---|---|---|---|---|---|---|
| | **Drug** | **DrugBank ID** | **Gene symbols** | **CT** | **Drug** | **DrugBank ID** | **Gene symbol** | **CT** |
| *1* | Mecobalamin | DB03614 | FASTKD5, NLRX1 | 1 | Fondaparinux | DB00569 | NLRX1 | 4 |
| *2* | Roxithromycin | DB00778 | FASTKD5, NLRX1 | 1 | Bivalirudin | DB00006 | FASTKD5 | 1 |
| *3* | Cisatracurium | DB00565 | FASTKD5, NLRX1 | 1 | Icatibant | DB06196 | FASTKD5 | 2 |
| *4* | Fondaparinux | DB00569 | FASTKD5, NLRX1 | 4 | Dotatate gallium Ga-68 | DB13925 | FASTKD5 | 1 |
| *5* | Cyclosporine | DB00091 | FASTKD5, NLRX1 | 10 | Degarelix | DB06699 | FASTKD5 | 1 |
| *6* | Vitamin B12 | DB00115 | FASTKD5, NLRX1 | 3 | Cyclosporine | DB00091 | FASTKD5 | 10 |
| *7* | Erythromycin | DB00199 | FASTKD5, NLRX1 | 1 | Vitamin B12 | DB00115 | FASTKD5 | 3 |
| *8* | Azithromycin | DB00207 | FASTKD5, NLRX1 | 86 | Fondaparinux | DB00569 | FASTKD5 | 4 |
| *9* | Degarelix | DB06699 | FASTKD5, NLRX1 | 1 | Bivalirudin | DB00006 | NLRX1 | 1 |
| *10* | Clarithromycin | DB01211 | FASTKD5, NLRX1 | 4 | Mecobalamin | DB03614 | FASTKD5 | 1 |
| *11* | Ivermectin | DB00602 | FASTKD5, NLRX1 | 64 | Vitamin B12 | DB00115 | NLRX1 | 3 |
| *12* | Ledipasvir | DB09027 | FASTKD5, NLRX1 | 4 | Sirolimus | DB00877 | FASTKD5 | 5 |
| *13* | Tacrolimus | DB00864 | FASTKD5, NLRX1 | 2 | Azithromycin | DB00207 | FASTKD5 | 86 |
| *14* | Sirolimus | DB00877 | FASTKD5, NLRX1 | 5 | Ledipasvir | DB09027 | FASTKD5 | 4 |
| *15* | Icatibant | DB06196 | FASTKD5, NLRX1 | 2 | Velpatasvir | DB11613 | FASTKD5 | 1 |
| *16* | Bivalirudin | DB00006 | FASTKD5, NLRX1 | 1 | Cisatracurium | DB00565 | FASTKD5 | 1 |
| *17* | Dotatate gallium Ga-68 | DB13925 | FASTKD5, NLRX1 | 1 | Ivermectin | DB00602 | FASTKD5 | 64 |
| *18* | Etoposide | DB00773 | FASTKD5, NLRX1 | 1 | Colistin | DB00803 | FASTKD5 | 1 |
| *19* | Velpatasvir | DB11613 | FASTKD5, NLRX1 | 1 | Ivermectin | DB00602 | NLRX1 | 64 |
| *20* | Aliskiren | DB09026 | FASTKD5, NLRX1 | 1 | Clarithromycin | DB01211 | FASTKD5 | 4 |
| *21* | Inosine pranobex | DB13156 | FASTKD5, NLRX1 | 3 | Inosine pranobex | DB13156 | FASTKD5 | 3 |
| *22* | Montelukast | DB00471 | FASTKD5, NLRX1 | 4 | | | | |
| *23* | Vitamin E | DB00163 | FASTKD5, NLRX1 | 3 | | | | |
| *24* | Itraconazole | DB01167 | FASTKD5, NLRX1 | 1 | | | | |
| *25* | Simvastatin | DB00641 | FASTKD5, NLRX1 | 3 | | | | |
| *26* | Ritonavir | DB00503 | FASTKD5, NLRX1 | 95 | | | | |
| *27* | Candesartan cilexetil | DB00796 | FASTKD5, NLRX1 | 1 | | | | |

TABLE VII: We report the ranked list of approved drugs (for the component with gene symbol FASTKD5 and NLRX1, respectively with Entrez Gene IDs $\{60493, 79671\}$) with respect to the number of proteins in input. Each row contains the DrugBank ID of the drug, the Entrez Gene IDs of the considered proteins, and the number of clinical trials (CT) in which the drug is involved. These results are obtained by SSN($\Phi_{mean}$) in the Multi-protein setting.

that leverages the whole set of proteins instead of a single one. As an example, the drug with ID DB00602 (Ivermectin), which has been used in 64 clinical trials, is ranked 11th when using the entire protein set as input, while it is ranked 17th or 19th if we only use one of the two proteins as input to the model.

## VI. Conclusions

Drug repurposing is a time and cost-effective strategy to adopt whenever a fast response to large-scale diseases is needed, such as with the recent COVID-19 outbreak. In such a scenario, the availability of tools and methodologies that can optimize and prioritize the drug repurposing effort assumes paramount importance. At the same time, a pandemic led by a new viral agent poses increased challenges due to the lack of historical information on the disease and its processes, thus reducing the effectiveness of purely data-driven techniques. In a scenario characterized by data scarcity, it is therefore fundamental to leverage consolidated domain knowledge available.

Our work developed since the early onset of the pandemic. Our approach has been heavily influenced by our goal of delivering a flexible and effective model for exploring clinical working hypotheses on candidate drugs, leveraging the few information available on the virus. With this perspective in mind, it was natural to orient our effort towards integrating existing resources concerning biological processes with few virus-human protein interactions in a network-based representation. Simultaneously, we needed to develop informative representations of existing drugs that could effectively scale and generalize from known interactions to working hypotheses expressed under the form of candidate protein targets that clinicians considered crucial in the viral-host interaction.

Concerning these circumstances, we have designed a deep learning-based solution that can discover interactions between a drug and a set of proteins, surpassing the limitations of the literature approaches that can only predict the interaction between a drug and a single protein. More precisely, the proposed model takes as input a drug and a set of (functionally related) proteins to predict an interaction score between the two entities, allowing the identification of undiscovered associations. Internally, it leverages Node2Vec to represent proteins according to their function, and DGNs to represent drugs considering their structural and chemical properties. We have shown experimentally that this novel approach is versatile, and can be used seamlessly to predict interactions between a drug and one or an entire set of proteins with outstanding performances, especially in terms of recalling the correct associations. Empirically, we found that our multi-protein approach improved over the single-protein baseline by 14% AUROC. Moreover, we have found that the model predictions are robust to ablating the set of proteins, which ensures its effectiveness even in cases where the set of proteins

that interact with the drug is only partially known. Lastly, we have presented a use case of the model for COVID-19 drug repurposing, showing the interactions discovered for a set of COVID-related proteins in both multi-protein and single-protein usage.

The approach presented in this paper is very flexible, and can be adapted with very few modifications to a broader set of problems. For example, by replacing the protein embedding module with a corresponding disease embedding module (i.e., a module that encodes a disease as a vector), the approach could be used for drug-disease association prediction [52]. In the same spirit, the drug embedding module could be augmented to encode different drug targets, such as micro-RNA [53], [54], by leveraging sequential features, or to use 3D-aware structural embeddings such as those of AlphaFold [55]. Lastly, the drug embedding module itself could be augmented to model drug-drug interactions for tasks such as synergistic drug combination prediction [56]. These are all interesting directions which we are willing to explore in future works. The presented model also has limitations. In our particular formulation, we were forced to fix several components in advance to reduce the computational cost of model training, including protein embedding size, the number of convolutional layers, and the number of epochs in the protein embedding module. While we acknowledge that this is not ideal, we also note that a more thorough model selection could have resulted in a performance improvement. Similarly, the use of random walk-based approaches such as Node2Vec to encode the proteins intrinsically carries out some limitations, such as the possibility of losing long-term dependencies due to the stochasticity of the method. Future works will try to address both of these limitations by exploring a wider range of hyper-parameters and resorting to different graph embedding approaches to encode proteins through GO terms.

In conclusion, our work provides a new methodology to support clinical experts by making available a versatile and robust tool to perform in-silico exploratory analyses and pre-screening drug collections and associated protein ensembles. On the methodological side, we have provided the machine learning community with insights about the effectiveness of considering high-order drug-protein interactions (in contrast with single-drug-single-protein ones) and of using graph-based data of different nature within a fully end-to-end differentiable deep graph network.

## ACKNOWLEDGMENT

This work has been partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215, Territori Aperti a project funded by Fondo Territori Lavoro e Conoscenza CGIL CISL UIL, and by the Intel COVID-19 Response and Readiness Initiative which has provided the computing facilities for this work. The authors would like to thank Francesco Landolfi, University of Pisa, and the CLAIRE COVID-19 task force members for the insightful discussions throughout the development of this work.

## REFERENCES

[1] G. Bontempi, R. Chavarriaga, H. D. Canck, E. Girardi, H. Hoos, I. Kilbane-Dawe, T. Ball, A. Nowé, J. Sousa, D. Bacciu, M. Aldinucci, M. D. Domenico, A. Saffiotti, and M. Maratea, "The CLAIRE COVID-19 initiative: approach, experiences and recommendations," *Ethics and Information Technology*, 2021.

[2] E. Guney, J. Menche, M. Vidal, and A.-L. Barábasi, "Network-based in silico drug efficacy screening," *Nature communications*, vol. 7, no. 1, pp. 1–13, 2016.

[3] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, "GraphDTA: Predicting drug–target binding affinity with graph neural networks," *Bioinformatics*, 10 2020, btaa921. [Online]. Available: https://doi.org/10.1093/bioinformatics/btaa921

[4] D. Bacciu, F. Errica, A. Micheli, and M. Podda, "A gentle introduction to deep learning for graphs," *Neural Networks*, vol. 129, pp. 203–221, 9 2020.

[5] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 25. 18–42, 2017.

[6] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, and others, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[7] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[8] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009, publisher: IEEE.

[9] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009, publisher: IEEE.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[11] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 2016, pp. 855–864.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.

[13] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[14] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[15] H. Chen, B. Perozzi, Y. Hu, and S. Skiena, "Harp: Hierarchical representation learning for networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[16] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[17] S. Pushpakom, F. Iorio, P. A. Eyers, K. J. Escott, S. Hopper, A. Wells, A. Doig, T. Guilliams, J. Latimer, C. McNamee *et al.*, "Drug repurposing: progress, challenges and recommendations," *Nature reviews Drug discovery*, vol. 18, no. 1, pp. 41–58, 2019.

[18] M. Dickson and J. P. Gagnon, "The cost of new drug discovery and development," *Discovery medicine*, vol. 4, no. 22, pp. 172–179, 2009.

[19] B. Bolgár, A. Arany, G. Temesi, B. Balogh, P. Antal, and P. Matyus, "Drug repositioning for treatment of movement disorders: from serendipity to rational discovery strategies," *Current topics in medicinal chemistry*, vol. 13, no. 18, pp. 2337–2363, 2013.

[20] A. Gottlieb, G. Y. Stein, E. Ruppin, and R. Sharan, "Predict: a method for inferring novel drug indications with application to personalized medicine," *Molecular systems biology*, vol. 7, no. 1, p. 496, 2011.

[21] H. Öztürk, A. Özgür, and E. Ozkirimli, "Deepdta: deep drug–target binding affinity prediction," *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, 2018.

[22] 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.

[23] Y. Zhou, Y. Hou, J. Shen, Y. Huang, W. Martin, and F. Cheng, "Network-based drug repurposing for novel coronavirus 2019-ncov/sars-cov-2," *Cell discovery*, vol. 6, no. 1, pp. 1–18, 2020.
[24] D. M. Gysi, Ítalo Do Valle, M. Zitnik, A. Ameli, X. Gan, O. Varol, S. D. Ghiassian, J. Patten, R. Davey, J. Loscalzo, and A.-L. Barabási, "Network medicine framework for identifying drug repurposing opportunities for covid-19," *ArXiv*, 2020.
[25] X. Zeng, X. Song, T. Ma, X. Pan, Y. Zhou, Y. Hou, Z. Zhang, K. Li, G. Karypis, and F. Cheng, "Repurpose open data to discover therapeutics for covid-19 using deep learning," *Journal of proteome research*, vol. 19, no. 11, pp. 4624–4636, 2020.
[26] V. N. Ioannidis, D. Zheng, and G. Karypis, "Few-shot link prediction via graph neural networks for covid-19 drug-repurposing," *arXiv preprint arXiv:2007.10261*, 2020.
[27] S. Ray, S. Lall, A. Mukhopadhyay, S. Bandyopadhyay, and A. Schönhuth, "Predicting potential drug targets and repurposable drugs for covid-19 via a deep generative model for graphs," *arXiv preprint arXiv:2007.02338*, 2020.
[28] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.
[29] K. Huang, T. Fu, L. M. Glass, M. Zitnik, C. Xiao, and J. Sun, "Deeppurpose: a deep learning library for drug–target interaction prediction," *Bioinformatics*, vol. 36, no. 22-23, pp. 5545–5547, 2020.
[30] Y. Liu, M. Wu, C. Miao, P. Zhao, and X.-L. Li, "Neighborhood regularized logistic matrix factorization for drug-target interaction prediction," *PLOS Computational Biology*, vol. 12, no. 2, pp. 1–26, 02 2016. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1004760
[31] Y. Long, M. Wu, Y. Liu, Y. Fang, C. K. Kwoh, J. Chen, J. Luo, and X. Li, "Pre-training graph neural networks for link prediction in biomedical networks," *Bioinformatics*, Feb. 2022.
[32] K. Y. Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, and P. Zhang, "Interpretable drug target prediction using deep neural representation." in *International Joint Conference on Artificial Intelligence*, vol. 2018, 2018, pp. 3371–3377.
[33] R. Zhang, D. Hristovski, D. Schutte, A. Kastrin, M. Fiszman, and H. Kilicoglu, "Drug repurposing for covid-19 via knowledge graph completion," *Journal of biomedical informatics*, vol. 115, p. 103696, 2021.
[34] K. Hsieh, Y. Wang, L. Chen, Z. Zhao, S. Savitz, X. Jiang, J. Tang, and Y. Kim, "Drug repurposing for covid-19 using graph neural network and harmonizing multiple evidence," *Scientific reports*, vol. 11, no. 1, pp. 1–13, 2021.
[35] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," 2017.
[36] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali, "DrugBank: a knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Research*, vol. 36, no. suppl_1, pp. D901–D906, 11 2007. [Online]. Available: https://www.drugbank.ca/
[37] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, p. 3371–3408, Dec. 2010.
[38] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," 2020.
[39] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.
[40] S. Carbon, A. Ireland, C. J. Mungall, S. Shu, B. Marshall, S. Lewis, the AmiGO Hub, and the Web Presence Working Group, "AmiGO: online access to ontology and annotation data," *Bioinformatics*, vol. 25, no. 2, pp. 288–289, 11 2008.
[41] F. Cheng, I. A. Kovács, and A.-L. Barabási, "Network-based prediction of drug combinations," *Nature communications*, vol. 10, no. 1, pp. 1–11, 2019.
[42] T. U. Consortium, "UniProt: a worldwide hub of protein knowledge," *Nucleic Acids Research*, vol. 47, no. D1, pp. D506–D515, 11 2018. [Online]. Available: https://www.uniprot.org/
[43] D. E. Gordon, G. M. Jang, M. Bouhaddou, J. Xu, K. Obernier, K. M. White, M. J. O'Meara, V. V. Rezelj, J. Z. Guo, D. L. Swaney *et al.*, "A sars-cov-2 protein interaction map reveals targets for drug repurposing," *Nature*, vol. 583, no. 7816, pp. 459–468, 2020.
[44] K. Lundstrom, "An overview on gpcrs and drug discovery: structure-based drug design and structural biology on gpcrs," *G protein-coupled receptors in drug discovery*, pp. 51–66, 2009.
[45] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach," *Biometrics*, vol. 44, no. 3, p. 837, Sep. 1988. [Online]. Available: https://doi.org/10.2307/2531595
[46] H. Öztürk, A. Özgür, and E. Ozkirimli, "Deepdta: deep drug–target binding affinity prediction," *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, 2018.
[47] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, "GraphDTA: predicting drug–target binding affinity with graph neural networks," *Bioinformatics*, vol. 37, no. 8, pp. 1140–1147, 10 2020. [Online]. Available: https://doi.org/10.1093/bioinformatics/btaa921
[48] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, pp. 742–54, 05 2010.
[49] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7
[50] "Covid-19 Clinical Trial Summary," https://go.drugbank.com/covid-19#clinical-trials, Accessed: 2021-02-28.
[51] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.
[52] X. Chen, C. C. Yan, X. Zhang, X. Zhang, F. Dai, J. Yin, and Y. Zhang, "Drug–target interaction prediction: databases, web servers and computational models," *Briefings in Bioinformatics*, vol. 17, no. 4, pp. 696–712, Aug. 2015. [Online]. Available: https://doi.org/10.1093/bib/bbv066
[53] X. Chen, N.-N. Guan, Y.-Z. Sun, J.-Q. Li, and J. Qu, "MicroRNA-small molecule association identification: from experimental results to computational models," *Briefings in Bioinformatics*, Oct. 2018. [Online]. Available: https://doi.org/10.1093/bib/bby098
[54] X. Chen, C. Zhou, C.-C. Wang, and Y. Zhao, "Predicting potential small molecule–miRNA associations based on bounded nuclear norm regularization," *Briefings in Bioinformatics*, vol. 22, no. 6, Aug. 2021. [Online]. Available: https://doi.org/10.1093/bib/bbab328
[55] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, Jul. 2021. [Online]. Available: https://doi.org/10.1038/s41586-021-03819-2
[56] X. Chen, B. Ren, M. Chen, Q. Wang, L. Zhang, and G. Yan, "NLLSS: Predicting synergistic drug combinations based on semi-supervised learning," *PLOS Computational Biology*, vol. 12, no. 7, p. e1004975, Jul. 2016. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1004975

**Davide Bacciu** (S'06–M'09–SM'18) has a Ph.D. in Computer Science and Engineering from IMT Lucca. He is currently Associate Professor at the Computer Science Department, University of Pisa, where he heads the Pervasive AI Lab. His research interests include machine learning for structured data, Bayesian learning, deep learning, reservoir computing, distributed and embedded learning systems. Dr. Bacciu received the 2009 E.R. Caianiello Award for the best Italian Ph.D. thesis on neural networks. He is the Vice President of the Italian Association for Artificial Intelligence, Vice Chair of the IEEE Technical Committee on Neural Networks and the chair of the IEEE CIS Task Force on Learning for Structured Data. He is currently a Senior Editor of the IEEE TNNLS.

**Federico Errica** is a Ph.D. Student in Computer Science from the University of Pisa. He received both his Bachelor and Masters Degrees in Computer Science from the University of Pisa in 2015 and 2018, respectively. He was a visiting researcher at Facebook AI Research, London in 2019 and at University College London in 2021. He is part of the Computational Intelligence and Machine Learning group (CIML). His current research interests revolve around machine learning for graphs, Bayesian networks, deep learning, and hybrid learning models.

**Alessio Gravina** is a Ph.D. Student in Computer Science from the University of Pisa. He received his Bachelor and Master of Science in Computer Science from University of Pisa in 2018 and 2020, respectively. In 2018 he won the Fujistu AI-NLP Challenge, while in 2019 he was a visiting student at University College Dublin (UCD), and a visiting researcher at Stanford University. He is part of the Computational Intelligence and Machine Learning group (CIML). His interests are related to the area of machine learning for graphs and deep learning.

**Lorenzo Madeddu** is a Ph.D. student at the Department of Translational and Precision Medicine at Sapienza University of Rome with a Computer Science Master Degree. He received his master degree in Computer Science from "Sapienza" University of Rome in 2018. His research interests focus on machine learning, graph mining and Network Medicine. He is involved in interdisciplinary projects in the fields of Healthcare and Precision Medicine and is supported by the "Sapienza information-based Technology InnovaTion Center for Health - STITCH".

**Marco Podda** received a Ph.D. in Computer Science from the University of Pisa in 2021. He is part of the Computational Intelligence and Machine Learning group (CIML). His research interests include deep learning for graphs and generative models of graphs, with applications to the biomedical field. Currently, he studies Machine Learning methods for vaccine development with a joint grant between University of Pisa and GlaxoSmithKline (GSK).

**Giovanni Stilo** is an Associate Professor in the Department of Information Engineering, Computer Science and Mathematics at the University of L'Aquila. He received his PhD. in Computer Science in 2013, and in 2014 he was a visiting researcher at Yahoo! Labs in Barcelona. Between 2015 and 2018, he was a researcher in the Computer Science Department at La Sapienza University, in Rome. His research interests are in the areas of machine learning and data mining, and specifically temporal mining, social network analysis, network medicine, semantics-aware recommender systems, and anomaly detection. He has organized several international workshops, held in conjunction with top-tier conferences (ICDM, CIKM, and ECIR), and he is involved as editor and reviewer of top-tier journals, such as TITS, ECML-PKDD, TKDE, DMKD, AI, KAIS, and AIIM.