Zero-shot prediction of therapeutic use with geometric deep learning and clinician centered design

Kexin Huang[1,†,*], Payal Chandak[2,*], Qianwen Wang[1], Shreyas Havaldar[3], Akhil Vaid[3,4], Jure Leskovec[5], Girish Nadkarni[4], Benjamin S. Glicksberg[3,4], Nils Gehlenborg[1], and Marinka Zitnik[1,6,7,‡]

[1]Department of Biomedical Informatics, Harvard Medical School, Boston, MA 02115
[2]Harvard-MIT Program in Health Sciences and Technology, Cambridge, MA 02139
[3]Hasso Plattner Institute for Digital Health, Icahn School of Medicine at Mount Sinai, NY 10029
[4]Charles Bronfman Institute for Personalized Medicine, Icahn School of Medicine at Mount Sinai, NY 10029
[5]Department of Computer Science, Stanford University, Stanford, CA 94305
[6]Broad Institute of MIT and Harvard, Cambridge, MA 02142
[7]Harvard Data Science Initiative, Cambridge, MA 02138

† Present address: Department of Computer Science, Stanford University
⋆ Equal contribution
‡ Corresponding author: marinka@hms.harvard.edu

Of the several thousand diseases that affect humans, only about 500 have treatments approved by the U.S. Food and Drug Administration. Even for those with approved treatments, discovering new drugs can offer alternative options that cause fewer side effects and replace drugs that are ineffective for certain patient groups. However, identifying new therapeutic opportunities for diseases with limited treatment options remains a challenge, as existing algorithms often perform poorly. Here, we leverage recent advances in geometric deep learning and human-centered AI to introduce TxGNN, a model for identifying therapeutic opportunities for diseases with limited treatment options and minimal molecular understanding. TxGNN is a graph neural network pre-trained on a comprehensive knowledge graph of 17,080 clinically-recognized diseases and 7,957 therapeutic candidates. The model can process various therapeutic tasks, such as indication and contraindication prediction, in a unified formulation. Once trained, we show that TxGNN can perform zero-shot inference on new diseases without additional parameters or fine-tuning on ground truth labels. Evaluation of TxGNN shows significant improvements over existing methods, with up to 49.2% higher accuracy in indication tasks and 35.1% higher accuracy in contraindication tasks. TxGNN can also predict therapeutic use for new drugs developed since June 2021. To facilitate interpretation and analysis of the model's predictions by clinicians, we develop a human-AI explorer for TxGNN and evaluate its usability with medical experts. Finally, we demonstrate that TxGNN's novel predictions are consistent with off-label prescription decisions made by clinicians in a large healthcare system. These label-efficient and clinician-centered learning systems pave the way for improvements for many therapeutic tasks.

Introduction

A limited number of clinically-recognized diseases currently have approved treatments, underscoring the pressing need to develop therapies to address the healthcare demands of billions globally. Diseases often arise due to disrupting normal gene behavior and the molecular products they produce. Effective drugs can intervene against these diseases by restoring intended molecular behaviors. However, restoring the biological functions of disrupted genes through therapeutic interventions remains challenging for many diseases. Furthermore, most diseases are driven by alterations in more than one gene, wherein each gene can manifest with very different patterns of mutations across patients with the same disease. A powerful way to interpret these genetic events is to organize them into interactomes — networks of genes that participate in disease-associated processes and functions[1–3]. Machine learning has been used to analyze high-throughput molecular interactomes and electronic medical record data to unravel genetic architecture perturbed in disease[3,4] and help design therapies to target them[5]. To provide therapeutic predictions, geometric deep learning models optimized on protein interactomes[6] can extract disease signatures and match them to therapeutic candidates based on the proximity of their mechanisms to the disease-perturbed networks[6–8].

While successfully identifying therapeutic candidates for complex diseases[9,10], this approach considers diseases that already have extensive molecular signatures and existing treatment options. Despite the promising performance of this approach, its use in practice is hindered by the assumption that there are already known and similar drugs for a given disease of interest[11]. However, diseases with few or no therapies and limited molecular understanding have incomplete underlying interactomes, leading to poorly predictive disease signatures and drastic drops in the ability of deep learning models to identify therapeutic candidates[7,11]. The critical challenge is that models need to make zero-shot predictions — identifying therapies for diseases that have not been encountered during training; a model needs to extend therapeutic predictions to new diseases without seeing any prior therapeutic information for them. This learning capability is essential because several thousand diseases affect humans, of which only about 500 have any U.S. Food and Drug Administration-approved treatment (NIH's National Center for Advancing Translational Sciences). Out of 17,080 clinically recognized diseases examined in our study, only 1,363 diseases have directly indicated drugs, out of which 435 diseases have only one indicated medication, 182 have two indicated medications, and 128 have three indicated medications. Even for diseases with treatments, finding new drugs is clinically impactful because it can give alternative treatment options

with less severe side effects and replacements for drugs that are ineffective in subsets of patients.

Here, we introduce TxGNN, a geometric deep learning approach for therapeutic use prediction, focusing on neglected diseases with a limited understanding of molecular mechanisms and treatment options. TxGNN is trained on a therapeutics-centered graph overlaid with disease-perturbed networks targeted by existing treatments. This knowledge graph collects and organizes decades of biological research centering around 17,080 diseases, including complex and rare diseases. TxGNN uses a graph neural network model to embed therapeutic candidates and diseases into a latent representation space and is optimized to reflect the geometry of TxGNN's therapeutics-centered graph. To overcome the limitation of supervised deep learning in predicting therapeutic use for neglected diseases, TxGNN uses a metric learning module that operates on the latent representation space and can transfer TxGNN's model from diseases encountered during training to unseen diseases.

We evaluate TxGNN for predicting indication and contraindication therapeutic use on systematic, disease-area, and disease-centric hold-out datasets across 1,363 diseases. First, TxGNN shows consistently strong performance for neglected diseases in all settings. Across six settings, TxGNN gains up to 49.2% (average gain = 33.9%) in accuracy on predicting indications and up to 35.1% (average gain = 26.8%) in accuracy on predicting contraindications compared to a state-of-the-art graph neural network. Second, we evaluated TxGNN for diseases not encountered during machine learning model training. TxGNN correctly identifies therapies that recently (since June 2021) received U.S. Food and Drug Administration approval, ranking belzutifan among the top 4% of all therapeutic candidates for the treatment of a subset of von Hippel-Lindau cancers and faricimab in the top 2% of all therapeutic candidates for neovascular age-related macular degeneration and diabetic macular edema. Third, novel predictions made by TxGNN predictions were evaluated against a large EHR phenotyping dataset (across 1,272,085,403 patients, 480 diseases, and 1,290 drugs), showing a 107% enrichment of likelihood of usage in real-world adoption compared to non-prioritized predictions. Finally, TxGNN provides explanations for its predictions as reasoning paths that can guide human inquiry. We develop TxGNN Explorer, an interpretable and interactive human-AI explorer that visually presents these reasoning paths to support and evaluate its usability in a user study with a panel of 12 clinicians. TxGNN Explorer is available at http://txgnn.org.

Results

Overview of TxGNN's therapeutics-centered dataset bridging molecular and phenotypic scales. Reasoning about therapeutic action requires knowing what genes are perturbed by disease and how those genes can be targeted by therapy to restore the function of disrupted genes. To bring this information together, we consider a therapeutics-centered graph unified across 20 data resources (Figure 1a)[12] (Methods 1.1). Therapeutics-centered graph covers ten types of biological entities, including 27,671 protein-coding genes, 7,957 drugs, 17,080 clinically-recognized diseases, 14,035 anatomy concepts, 28,642 biological processes, 4,176 cellular components, 11,169 molecular functions, 818 exposures, 15,311 disease phenotypes, and 2,516 pathways. It also includes 29 types of relations among these entities, forming a connection map essential for understanding the mechanism of disease treatments. Further, the graph encodes physical protein-protein interactions, information on combinatorial drug action, membership of genes in molecular pathways, and gene phenotypes for a total of 8,100,498 edges (Methods 1).

The dataset comprises 9,388 indications representing approved therapies covering 1,363 diseases and 1,801 drugs. Further, it contains 30,675 contraindications split across 1,195 diseases, representing conditions when a particular medication must not be taken because of the harm it would cause the patient. On a median, a disease node is connected to 5 proteins, 14 phenotypes, 3 other diseases, and 2 exposures in the graph (Supplementary Figure 1). These connections to diverse entities enable machine learning models to fully determine the essential features of therapeutic action. However, 15,717 diseases (92% of diseases) currently have no indicated therapies, and 15,885 diseases (93% of diseases) have no known contraindications, highlighting the need for strategies to support research for diseases with no available therapies.

Machine learning has proved helpful in identifying therapeutic opportunities for diseases for which some therapies already exist[3–7,9,10]. The approach is to retrieve additional candidate therapies that are similar to existing ones across levels of biology[13]. However, this approach has limited applicability for diseases with incomplete biological mechanisms (Figure 1b) because a direct relationship between the disease and its candidate therapies does not exist[7,11]. Network proximity between diseases and candidate therapies is predictive of efficacy[9]. The challenge, however, is that diseases are on average multiple hops (avg=2.70 hops) away from their standard therapies in the knowledge graph — for example, herpes simplex virus (HSV) keratitis is found five hops away from idoxuridine in the TxGNN's knowledge graph, and in another example, the shortest path length between pityriasis simplex capitis and selenium sulfide is five; similarly Klebsiella pneu-

monia and clavulanic acid are five hops away from each other in the knowledge graph. While drugs indicated for diseases are significantly closer to each other in the graph than their random counterparts (the average shortest path length between each disease and random set of drug molecules is 3.35), this observation indicates that models must consider several hops of indirect relationships for accurate prediction.

Geometric biological priors in TxGNN for therapeutic use prediction. TxGNN operates on the principle that effective drugs target disease-perturbed networks in the protein interactome. TxGNN is a knowledge-grounded graph neural network (GNN) that maps therapeutic candidates and diseases (disease concepts) into the latent representation space, optimized to capture the geometry of TxGNN's knowledge graph (Methods 2.2). The latent representation model serves as a foundation model for therapeutic prediction tasks — given a therapeutic candidate and a disease, TxGNN transforms points in the latent space representing the candidate and disease into predictions about their relationship (Figure 1c). We consider two therapeutic tasks: indication prediction (*i.e.*, a disease that makes a particular treatment advisable) and contraindication prediction (*i.e.*, a disease that serves as a reason to avoid a certain medication due to the harmful adverse events that it would cause the patient).

When only limited molecular information exists for a disease of interest, we can leverage molecular mechanisms of other diseases in the knowledge graph to improve performance for the poorly annotated disease. In TxGNN, we obtain a disease signature vector for each disease based on the set of neighboring proteins, exposures, and other biomedical entities. A normalized dot product between two disease signatures accounts for the amount of molecular entities shared between diseases, serving as a disease similarity metric. These signatures can portray the disease similarity landscape, including many rare diseases (Figure 1d). Most disease pairs have low similarity scores since they do not share mechanisms (*e.g.*, T-substance anomaly and frontometaphyseal dysplasia have a score of 0.084). In contrast, the similarity is significant if two diseases score relatively high ($>$0.2). For example, Wells syndrome and pemphigus erythematosus have a similarity of 0.433. Both are skin diseases caused by autoimmune disorders. However, there are differences in phenotypes where Wells syndrome exhibits redness and swelling, and the pemphigus exhibits blisters. There are also a few disease pairs with extremely high similarity scores. For example, Pick's disease and Alzheimer's have a similarity of 0.909 since they share many of the same causes, but Pick's disease affects a specific part of the brain that controls emotions, behavior, personality, and language and thus have slightly different symptoms (*e.g.*, no delusions, earlier onset). Thus,

we can retrieve a set of similar diseases for each disease, quantified by the number of overlapping entities.

Leveraging the knowledge of retrieved diseases requires fusing it into the graph neural network model. To that end, given a target disease, TxGNN generates embedding for each retrieved disease. Then, it adaptively aggregates the embeddings in the metric learning module, weighing each retrieved disease by the target disease similarity. The aggregated output embedding is a compact summary of knowledge borrowed from similar diseases fused with the target disease embedding. With the unified modeling, large-scale GNN training enables TxGNN to process different downstream therapeutic tasks using shared drug and disease embeddings (Methods 2.3).

From a graph machine learning perspective, TxGNN can be considered a domain prior-guided graph rewiring technique (Supplementary Figure 2)[14]. Since TxGNN aggregates similar disease nodes to the target disease node in the latent space, it is equivalent to generating new edges between the target disease and similar diseases (*i.e.*, rewiring a set of disease nodes to the target disease node). These new edges allow relevant disease module information captured in the embeddings of similar diseases to flow into the target disease embedding, which enables zero-shot predictions. This graph rewiring is guided by a domain prior because we select similar diseases using the local hypothesis in network medicine, as discussed in the intuition above. Without these rewired edges, the target disease embedding is not meaningful since there are few connected molecular nodes and zero treatment nodes to learn the embedding from. Technically, the low degree of these disease nodes presents a network bottleneck, leading to difficulty in learning the embedding for the disease node and its neighboring nodes (*i.e.*, over-squashing)[14,15]. By rewiring disease nodes, we enlarge the bottleneck and thus circumvent the over-squashing issue and improve the learning of a meaningful representation.

Benchmarking TxGNN for zero-shot prediction of therapeutic use. We evaluate TxGNN for indication and contraindication prediction. The model is evaluated for zero-shot performance, meaning that it is asked to predict therapeutic use for diseases in the hold-out (test) set that are not seen during model training (*i.e.*, zero of each disease's indications or contraindications are available to the model during training), which is the intended use of TxGNN for neglected diseases, such as Stargardt disease[16] and hyperoxaluria[17] that have no existing therapies. We use three strategies to construct hold-out datasets (Methods 3): systematic hold-outs (diseases in the hold-out set are selected at random), disease area hold-outs (for each disease area, all diseases in the area are put in a hold-out set), and disease-centric hold-outs (for each disease, a separate hold-out set is

constructed).

Zero-shot performance on therapeutic use prediction on systematic hold-outs. We first consider systematic hold-out evaluation, where we exclude all known therapies for testing diseases from the dataset and ask the model to make predictions for these therapies. Biologically, the model is probed to predict diseases with no existing treatments, meaning no information about drug similarities is available. Each random selection spans 103 never-before-seen diseases with 495 indications and 1,729 contraindications.

We compare TxGNN to a state-of-the-art GNN[18] previously validated for therapeutic use prediction[6–8,19–21]. The GNN performs poorly on realistic yet challenging systematic data splits. TxGNN outperforms GNN by 0.180 on average in AUPRC when identifying new indications, which is a 33.92% performance gain in predicting new indications for previously approved drug molecules (Figure 2a). When identifying contraindications, TxGNN improves over the baseline by 0.133 in AUPRC, which is a 26.79% relative gain in identifying medical treatments that would cause harm to the patient (Figure 2b). Notably, in the systematic split where we test the model's generalizability to diseases with no treatments, TxGNN achieves an AUPRC of 0.874 (49.2% increase over GNN) when identifying indications and an AUPRC of 0.773 (35.1% increase over GNN) for contraindications.

Zero-shot performance on therapeutic use prediction for 5 disease areas. Biologically related diseases may have similar treatments[1]. For example, the same chemotherapy can be indicated for two types of cancers. Thus, if the model encounters the therapy for one cancer type during training, it can easily predict the same treatment for related cancer when the model is deployed[11]. This phenomenon is known as shortcut learning[22,23] and underlies many of deep learning's failures[24,25]: shortcuts are decision rules learned by the model that perform well on standard benchmarks but fail to transfer to challenging testing conditions[26], such as real-world scenarios when the model is asked to assist in the development of therapies for a new group of diseases (such as rare or neglected diseases) or diseases with no existing treatments.

To evaluate TxGNN in challenging testing conditions, we curate a stringent evaluation split where we evaluate model performance on a group of biologically related diseases referred to as a disease area. Given a set of diseases in a disease area, all their indications and contraindications are removed from the training dataset. Further, a large fraction (95%) of the connections from biomedical entities to these diseases are excluded from the training dataset. This removal simulates the diseases' limited molecular characterization while having no existing treatments. We

consider five disease areas (Table 2): the cell proliferation hold-out has 213 diseases with a total of 1,022 indications and 1,079 contraindications; the mental health hold-out has 60 diseases with 355 indications and 1,567 contraindications; the cardiovascular disease hold-out has 113 diseases with 453 indications and 4,242 contraindications; the anemia hold-out has 19 diseases with 88 indications and 752 contraindications; and lastly, the adrenal gland hold-out has 7 diseases with 41 indications and 374 contraindications.

While TxGNN consistently outperforms baselines on hold-out splits organized by disease areas, it performs exceptionally well on cell proliferation, anemia, and mental health-centric splits, where it achieves an absolute improvement of 0.270, 0.211, 0.162 on the AUPRC, which represents relative gains of 46.54%, 44.60%, 34.23% over the GNN baseline, respectively (Figure 2a-b). This finding demonstrates that TxGNN is broadly generalizable and produces accurate predictions even in the most challenging setting when the same disease or related diseases from the same area are not in the training set of the machine learning model. Notably, on cell proliferation diseases, TxGNN achieves an AUPRC of 0.851, showing an accurate prediction model for oncology diseases even though it has not seen any oncology disease during training.

Zero-shot performance for each of 1,363 diseases with indications and 1,195 diseases with contraindications. Another type of hold-out evaluation we consider are disease-centric tests. To evaluate the therapeutic prediction accuracy for each disease, we consider known indications and contraindications in the test set as hits and the rest of the drugs as negative samples and calculate performance metrics. TxGNN successfully predicts therapeutic use with high precision for diseases not seen by the model during training. For example, TxGNN flags nine indications (betamethasone, triamcinolone, prednisone, hydrocortisone, prednisolone, methylprednisolone, dexamethasone, cortisone acetate, hydrocortisone acetate) for adrenal insufficiency without seeing the disease in training with an AUPRC of 0.938. In the case of another disease, plasmablastic lymphoma has six indications. TxGNN assigns all six (carmustine, bleomycin, vincristine, prednisone, doxorubicin, dexamethasone) in the top 100 list with an AUPRC of 0.746. By contrast, anaplastic oligoastrocytoma has two contraindications, sirolimus, temsirolimus, and TxGNN identifies these contraindicated drugs correctly with an AUPRC of 0.833. Further, leiomyoma has 24 contraindications, including diethylstilbestrol and chlorotrianisene, and TxGNN annotates 22 of them in the top-100 list, achieving an AUPRC of 0.779. We have provided the disease and its performance metrics in Supplementary Tables 1 and 2.

Prioritizing indicated and contraindicated therapies with high sensitivity. Therapeutic use

prediction models can prioritize therapeutic candidates to maximize the experimental and clinical investigation yield by focusing only on top-ranked candidates. We generate a list of the top-100 predicted drugs for each disease and calculate the fraction of correct hits in the list (Recall@100). In addition to comparing with the GNN, we include a non-guided screening baseline, randomly selecting 100 drugs as predicted indications/contraindications for a disease. This strategy is similar to high-throughput screening without using machine learning for prioritization. The multi-relational GNN baseline has a similar prioritization performance as non-guided screening. In contrast, TxGNN performs consistently better across all evaluations. On predicting indications, TxGNN achieves, on average, 10.44 fold enrichment (from 5.94 - 13.11 folds) compared to non-guided screening (Figure 2c). Traditional GNN does not retrieve any hits for adrenal gland and anemia disease group splits and very few for mental health splits. On the other three splits (*i.e.*, no information about any disease and its treatment for the entire disease group is available to the model during training), TxGNN achieved 6.54 average fold enrichment over GNN. TxGNN can recall more than 70% of hits in the top 100 predictions for systematic disease splits, adrenal gland splits, and cell proliferation splits. In addition, it retrieves more than 45% hits for anemia and mental health splits, providing a generalist model for therapeutic use prediction.

On predicting contraindications, TxGNN achieves a 6.893 average folds enrichment over non-guided screening and a 4.33 fold enrichment over GNN (Figure 2d). In addition, TxGNN achieves Recall@100=0.865 and Recall@100=0.689 for adrenal gland and cell proliferation diseases, meaning it can retrieve most drugs that have harmful effects on adrenal gland and cell proliferation patients. For example, in the adrenal gland split, TxGNN prioritizes all 71 contraindications in the top 100 ranked list for familial glucocorticoid deficiency. Similarly, in the cell proliferation split, TxGNN correctly identifies sirolimus and temsirolimus as the top-2 contraindications for glioblastoma, and it predicts contraindicated indomethacin for colorectal cancer.

In 47 out of 189 cell proliferation diseases, TxGNN prioritizes all known treatments in the top-10 ranked list out of more than 7,000 drug candidates. Without seeing any treatment for the rare uterus tumor of adenosarcoma, TxGNN retrieves doxorubicin, dactinomycin, and vincristine as the 1st, 4th, and 5th most promising indications. This shows TxGNN's ability even for many rare diseases. On the mental health disease split, TxGNN achieves the best performance for narcolepsy, a chronic sleep disorder characterized by overwhelming daytime drowsiness and sudden sleep attacks. TxGNN ranked six treatments among the top 10 predicted indications, methylphenidate, modafinil, dextroamphetamine, armodafinil, pitolisant, and solriamfetol. Sepa-

rately, out of 53 diseases in the cardiovascular disease group split, TxGNN predicts all known indications for eight diseases in the top 50 list. Notably, for a rare heart disease tetralogy of fallot, caused by a combination of four heart defects, TxGNN assigns its first-line therapy alprostadil as the top-1 drug. On the anemia disease group split, TxGNN retrieves all treatments for 2 out of 12 anemia diseases in the top-10 list. For instance, for beta thalassemia, a blood disorder that reduces the production of hemoglobin, TxGNN predicts deferasirox, deferiprone, and luspatercept as the top-3 ranked indications.

Utility of geometric deep learning for therapeutic use prediction. We conduct a systematic study by modifying individual components of TxGNN to test their utility on the systematic disease split (Figure 2e). First, we remove the entire metric learning procedure, and it degrades to regular GNN ('No-Metric'). We find TxGNN has a 0.2884 AUPRC increase over the ablation for indication and 0.2008 AUPRC increase for contraindication. Then, we keep the metric learning procedure but remove pretraining ('No-Pretrain'). The ablation has 0.030 decrease in AUPRC and retrieves 7.5% fewer hits in the top 100 predictions for indication. We observe similar behaviors for predicting contraindicated use, where 'No-Pretrain' leads to a 0.044 decrease in AUPRC and recalls 7.7% fewer hits, showing that the biomedical knowledge-grounded pretraining strategy is valuable and leads to positive knowledge transfer. To test the utility of degree-based aggregation, we use a simple alternative by taking the average between the auxiliary and original disease embeddings ('Avg-Agg'). We find TxGNN has relatively similar performances in indication prediction but improves contraindication prediction by 0.022 in AUPRC and retrieves 1.8% more hits, showing the usefulness of this component. Lastly, we experiment with two alternative strategies to calculate the disease signature, one is only using protein nodes to calculate disease similarity ('Protein-Sig'), and another is a diffusion-based random walk signature ('Walk-Sig'). We find TxGNN retrieves 8.4%/5.4% more hits than 'Protein-Sig' and 9.6%/6.4% more hits than 'Walk-Sig' in indication/contraindication prediction, respectively, suggesting the importance of signature selection to characterize the similarity among diseases.

We find that each component is indispensable in the success of TxGNN. The deep metric learning module is the key factor that drives TxGNN performance, corroborating our hypothesis on disease similarity. To further understand the performance gain of the metric learning module from a machine learning standpoint, we explore the example of tonsillitis (Figure 4a). Diseases similar to tonsillitis (epiglottitis, peritonsillar abscess, nasopharyngitis, pharyngitis, vulvitis) are initially distant in the embedding space. Thus, by fusing distant disease embeddings, TxGNN es-

tablishes a long-range skip connection to the disease module of these similar diseases and provides complementary information missing from the local neighborhood around the target disease. This is especially beneficial in predicting therapeutic use for conditions with few or no treatments and limited molecular understanding (Figure 1b). TxGNN uses disease signatures as a learnable disease look-up catalog to identify the appropriate distant disease information that can be transferred to the underpowered target disease.

Evaluation of TxGNN predictions against recently developed therapies. To demonstrate that TxGNN is not driven by confirmatory bias, we consider ten recently introduced therapies that were approved after TxGNN's dataset and model development were completed. None of these therapies have direct relationships between their drug-disease nodes in the TxGNN dataset. We then asked TxGNN to make predictions for them (Table 1). We observe that TxGNN consistently prioritizes newly introduced drugs highly. On average, the drugs are found in the first third (30.19%) of the full-length prediction list. For example, TxGNN ranked Merck's belzutifan, an orphan drug that treats von Hippel-Lindau disease, among the top 4.11% therapeutic candidates. It ranked faricimab, a biologic developed for macular degeneration, among the top 2.25% chemicals. However, TxGNN ranked maribavir, an inhibitor of the cytomegalovirus (CMV) pUL97 kinase used to treat CMV infections in patients post-transplantation, in the 66.37% of prediction list — unsurprising, given that maribavir exerts its antiviral efficacy via an alternative protein target in CMV as compared to traditional CMV antivirals and that TxGNN dataset does not contain information about host-pathogen interactions.

Producing AI-based reasoning paths and visual explanations for predictions. Even though TxGNN can produce accurate predictions of therapeutic use, these predictions require critical examination and evaluation by clinicians. In addition to being accurate, predictive rationales extracted by TxGNN from the dataset must be concise and interpretable. TxGNN uses a self-explaining approach (GraphMask[27], Methods 2.6) to generate a sparse and sufficient subgraph relevant for therapeutic use prediction as the explanations. The approach produces an importance score ranging from 0 to 1 for every edge in the dataset, where 1 means that the edge is crucial for the prediction and 0 means that this edge is irrelevant to the prediction.

The self-explaining TxGNN model can extract high-quality sparse explanations around a drug-disease query in the dataset. We trained five models with different data splits using the systematic split. To measure the quality of model explanations, faithfulness is a theoretically-grounded metric[28,29] that measures the performance gap between predictions using a sparse ex-

plainable subgraph and the original predictions using the complete subgraph of a query drug-disease pair. A small gap suggests that the explainer accurately selects edges crucial for TxGNN to make a prediction, *i.e.*, the explanation is faithful to the rationale used by TxGNN predictor. The self-explaining TxGNN model achieved high faithfulness, where, on average, the AUPRC has only lost 0.98%. At the same time, the model extracts sparse and relevant information. For example, setting a 0.5 threshold on self-explaining edge importance scores, 86.1% of all edges in subgraphs are dropped. The user can also adjust the threshold to further filter to more important explanations. The evaluation results show that the extracted explanations filter out non-informative connections and contain a sparse set of meaningful relations that are necessary to make faithful drug-disease predictions (Figure 3a, Methods 2.6). Next, we asked an important and insufficiently studied question of how to represent best and visualize TxGNN-extracted explanatory rationales to support clinicians' reasoning about therapeutic use. In developing human-AI TxGNN Explorer, we took a user-centric approach by comparing three visual explanations for displaying GNN explanations, *i.e.*, neighbor nodes around the query disease, subgraphs, and paths (Figure 3a, Supplementary Figure 4). Our studies showed that path explanations improve user performance and satisfaction compared to neighbor and subgraph explanations[30].

Visual explanations have the potential to guide clinical inquiry into predictions. Improving the interaction between humans and machines is essential for successful collaboration between medical experts and AI[31,32]. This is crucial to prevent common pitfalls, such as over-reliance on model predictions without independent expert evaluation, displaying limited trust in model predictions even when they provide valuable information, and avoiding opaque judgments. Moreover, even when medical experts have an appropriate level of confidence in TxGNN's predictions, they may be unable to determine whether predictions are reliable. To evaluate and understand how clinicians use model predictions and explanations, we conduct a user study (Supplementary Figure 5) with 12 clinicians and summarize results in Figure 3c. We find that providing visual explanations can improve clinicians' performance in evaluating model predictions (*i.e.*, whether a predicted drug can be used to treat a disease). Compared with the no explanation baseline, TxGNN Explorer enables clinicians to more accurately assess a predicted drug-disease relation ($t(11) = 2.76, p < 0.05$). Since we showed study participants both correct and incorrect predictions, higher accuracy indicates not only that participants tend to trust model predictions but also that they can better distinguish between correct and incorrect predictions while using TxGNN Explorer. We also observed higher self-reported satisfaction levels when using TxGNN Explorer compared to the baseline.

Participants reported higher usability for path-based explanations than neighboring node-based or subgraph-based model explanations, as measured by User Trust, Model Helpfulness, Understandability, and Willingness to Use. Using TxGNN Explorer, 91.6 (11/12) participants (strongly) agreed that TxGNN's predictions and explanations are valuable. On the contrary, without explanation, 75.0% (8/12) participants (strongly) disagreed with using TxGNN's predictions. Participants also report a higher level of confidence in correct model predictions ($t(11) = 3.64, p < 0.01$) when using TxGNN Explorer over a baseline that does not use a human-AI explorer. Some participants stated that the path-based explanations provide helpful information for planning the downstream evaluation, such as examining biological mechanisms and possible adverse effects of predicted therapeutic candidates. TxGNN Explorer provides experimental evidence of effective human-AI collaboration for therapeutic use prediction.

Evaluation of most promising predictions in a large electronic health record system. The above evaluations show that TxGNN can retrieve promising therapeutic candidates by recovering the known indications. TxGNN's performance suggests that its novel predictions (*i.e.*, therapies that are not yet FDA-approved for a disease but are ranked high in TxGNN) could potentially have clinical value. Since these predictions are not approved treatment indications, there is no gold standard information to validate against. Motivated by prevalent off-label drug prescriptions in clinical practice, we use the enrichment of disease-drug pair co-occurrence in an extensive health system's electronic health records (EHR) as a proxy measure of a potential indication. In particular, we used data from the Mount Sinai Health System to curate a cohort of 1,272,085 patients with 480 diseases and 1,290 drugs. The cohort included all patients over 18 years of age with at least one drug and at least one diagnosis in the system until 2022. Diseases were included if at least one patient had a diagnosis in the dataset; drugs were included if at least ten patients were listed with the drug and an order date (Table 5 and Methods 3.7). The enrichment of disease-drug co-occurrence can be quantified by the ratio between the odds of using a particular drug given the disease and the odds of using that drug given other diseases. We obtain 619,200 log-odds ratios (Log-OR) for every disease and drug pair matched to the EHR system. We benchmark the Log-OR against the FDA-approved drug-disease pairs and found that approved pairs have a much larger log-odds ratio than the other pairs (Supplementary Figure 1), confirming that the log-odds ratio can serve as a feasible measure for evaluating the potential indication.

For every disease, we rank drug candidates based on TxGNN predictions, and we retrieve the top-1, top-5, top-5%, and bottom-50% novel candidates and calculate their respective average Log-

OR (Figure 4d). The top-1 novel TxGNN prediction has, on average, a 107% higher Log-OR than the bottom-50% predictions, suggesting that TxGNN top candidate has much higher enrichment in the EHR system and improved likelihood of being an appropriate indication. In addition, we see that the Log-OR increases as we loosen the fraction of retrieved candidates, suggesting that TxGNN prediction score is meaningful in capturing the likelihood of indication. In addition, while the overall average Log-OR is 1.09, we find that the top-1 therapeutic candidate by TxGNN has a Log-OR=2.26, which is close to the average Log-OR for FDA-approved indications, Log-OR=2.92. This observation suggests the potential utility of TxGNN's most promising predictions. Further, we show two examples of predicted novel indications for anaplastic astrocytoma, a rare malignant brain tumor, and Wilson's disease, a rare genetic disorder that causes excessive copper to accumulate in the organs (Figure 4e). The TxGNN's predicted likelihood is close to 0 for most candidate therapies, and only a few are likely to be indications. With an 80% predicted probability, lomustine is TxGNN's top-1 therapeutic candidate for anaplastic astrocytoma. This prediction is substantiated by the EHR records, where lomustine has a high off-label prescription rate for anaplastic astrocytoma with a Log-OR=10.64, further supported by evidence suggesting lomustine's efficacy towards anaplastic astrocytoma[33]. Similarly, TxGNN ranks deferasirox as the most promising therapy for Wilson's disease, a common cause of liver cirrhosis in children (Log-OR=5.26), suggesting that deferasirox's might have a similar efficacy as deferoxamine in removing hepatic iron[34]. These analyses demonstrate that TxGNN 's novel predictions are consistent with clinical decisions on off-label prescription.

Discussion

Thousands of diseases affect humans, and most of them lack effective treatments. Furthermore, multiple treatments may exist for some conditions, each with unique advantages and disadvantages. The optimal treatment for a patient will depend on various factors, including the patient's medical history and the specific condition being treated. Research is ongoing to develop therapies for various diseases. This entails identifying indications as conditions for which a candidate therapy might be appropriate. Additionally, research efforts include determining contraindications, which are conditions or circumstances under which therapy should not be used because of potential adverse interactions with other medications or medical conditions or due to safety concerns.

TxGNN has zero-shot predicting ability of therapeutic use for diseases without existing treatments and limited molecular understanding. This opens up new avenues for machine learn-

ing to identify therapeutic opportunities for diseases that are challenging to model using existing strategies as well as neglected and rare diseases[35] in urgent need of therapeutic innovation[36,37].

TxGNN achieves this goal by capturing and representing similar diseases, extracting relevant knowledge, and fusing it into the disease of limited knowledge. This is achieved through a network medicine principle that governs disease-treatment mechanisms[6] and involves finding diseases with a higher frequency of shared pathways, phenotypes, and pathologies in the latent representation space than expected. The set of retrieved similar diseases suggests that TxGNN is leveraging information from diverse disease partners. This principle of efficient retrieval of similar diseases from the latent space that drives TxGNN can be adapted to other problems, such as disease-target identification and phenotype modeling. These latent connections are captured by TxGNN, allowing for generalization to diseases with few treatment options. Unlike existing approaches, TxGNN can tackle indication and contraindication therapeutic use prediction in a unified formulation across 17,080 clinically-recognized diseases. It can perform zero-shot inference for new diseases without requiring additional fine-tuning for diseases using labeled data points.

Our results show that TxGNN can significantly improve therapeutic use prediction across a wide range of diseases, even under a real-world constraint of having zero known therapies for a given disease and extrapolating to a new disease area unseen during training. Additionally, TxGNN's predicted therapies strongly correlate with information in real-world electronic health records and can be used to test a large number of therapeutic hypotheses in parallel by identifying disease cohorts that either have or have not been prescribed a particular medication (indication vs. contraindication) using patient populations followed for several years. Finally, a clinician-centered TxGNN Explorer linked TxGNN with a self-explaining model to present TxGNN's predictions to an audience of clinicians and explore disease-treatment mechanisms leveraged by the model. The usability study of the clinician-centered design shows that researchers using the interactive TxGNN Explorer can reproduce machine learning models and more easily identify and debug failure points of models, highlighting the importance of clinician-centered design in shifting machine learning from development to biomedical implementation[38].

Data availability. The TxGNN's project website is at https://zitniklab.hms.harvard.edu/projects/TxGNN. Therapeutics-centered knowledge graph is available at Harvard Dataverse under a persistent identifier https://doi.org/10.7910/DVN/IXA7BM. We have deposited the knowledge graph and all relevant intermediate files in this repository. All clinical and electronic medical record data were deidentified, and the Institutional Review Board at Mount Sinai, New York City, U.S., approved the study.

Code availability. Python implementation of the methodology developed and used in the study is available via the project website at https://zitniklab.hms.harvard.edu/projects/TxGNN. The code to reproduce results, documentation, and usage examples are at https://github.com/mims-harvard/TxGNN. To facilitate the usage of the algorithm, we developed a TxGNN Explorer, a web-based app available at http://txgnn.org to access TxGNN's predictions.

Acknowledgements. K.H., P.C., and M.Z. gratefully acknowledge the support by NSF under No. IIS-2030459, US Air Force under No. FA8702-15-D-0001, and awards from Harvard Data Science Initiative, Amazon Research, Bayer Early Excellence in Science, AstraZeneca Research, and Roche Alliance with Distinguished Scientists. P.C. was supported, in part, by the Harvard Summer Institute in Biomedical Informatics. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders.

Authors contribution. P.C. retrieved, processed, and analyzed the therapeutics-centered knowledge graph. K.H. and P.C. developed and implemented new machine learning methods, benchmarked machine learning models, and analyzed model behavior, all together with M.Z. Q.W. and N.G. implemented the clinician-centered visual explorer of model predictions and performed a user study to evaluate its usability. S.H., A.V., G.N. and B.S.G. performed a validation study examining new predictions of therapeutic use through the electronic health record system. K.H., P.C, Q.W., S.H., A.V., J.L., G.N, B.S.G., N.G., and M.Z. contributed new analytic tools and wrote the manuscript. All authors discussed the results and contributed to the final manuscript. M.Z. designed the study.

Competing interests. The authors declare no competing interests.

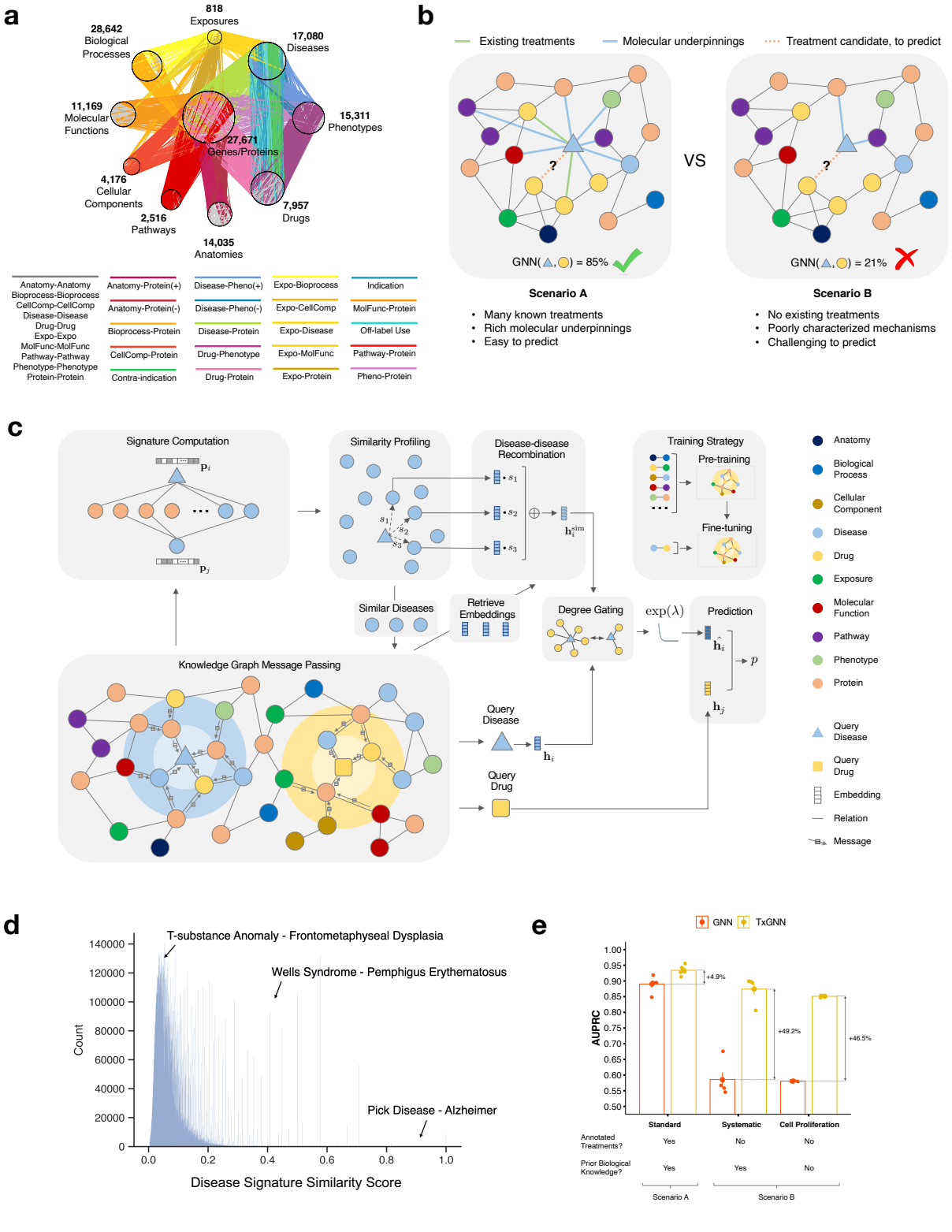

Figure 1: TxGNN is a geometric deep learning approach for discovering therapeutics across challenging diseases with no known treatments and limited molecular understanding. a. A large-scale therapeutics-driven knowledge graph integrates 17 primary data sources to depict a comprehensive landscape of biological mechanisms for therapeutics discovery. **b.** Commonly predicted diseases have many known treatments and rich molecular understandings and are thus easy to predict. However, diseases of active research and with the most therapeutic potentials are often the ones that have no existing treatments and poorly characterized mechanisms. The machine learning model is most useful in the latter case, but prevailing models fail. **c.** Detailed illustration of the TxGNN method. It follows three key steps. (1) TxGNN projects biological concepts into meaningful representations through knowledge graph neural network message passing on the KG (Methods 2.2). (2) It then designs a similarity disease search component to enrich molecularly uncharacterized diseases (Methods 2.4) and it has three modules (2.1) It computes a signature vector for each disease that captures the disease similarity. (2.2) Based on the signature vector distance, it profiles a set of similar diseases and retrieves their latent embeddings. (2.3) It then aggregates the different similar diseases into a powerful auxiliary embedding. (2.4) A gating mechanism is designed to control the effect between the original disease embedding and the auxiliary disease embedding since many well-characterized diseases have sufficient embeddings and do not need subsidies. (3) A decoder then maps the query drug and disease representation to predict the outcome (Methods 2.3). A pretext learning stage is devised to allow TxGNN to learn an initialized embedding that captures complex biological knowledge (Methods 2.5). **d.** The insight of TxGNN is that biology is a connected system where diseases are partially similar and can share multiple mechanisms. By identifying and fusing similar diseases to poorly characterized diseases, TxGNN can make an accurate prediction on these challenging diseases. TxGNN develops a disease signature that accurately depicts the disease relation landscape. **e.** State-of-the-art methods (GNN) break when inferring treatments for diseases in the wild, while TxGNN significantly improves over GNN and achieves accurate prediction for diseases with no treatments annotation and limited biological knowledge prior. The reported metric is AUPRC on indication prediction, average across five random runs.

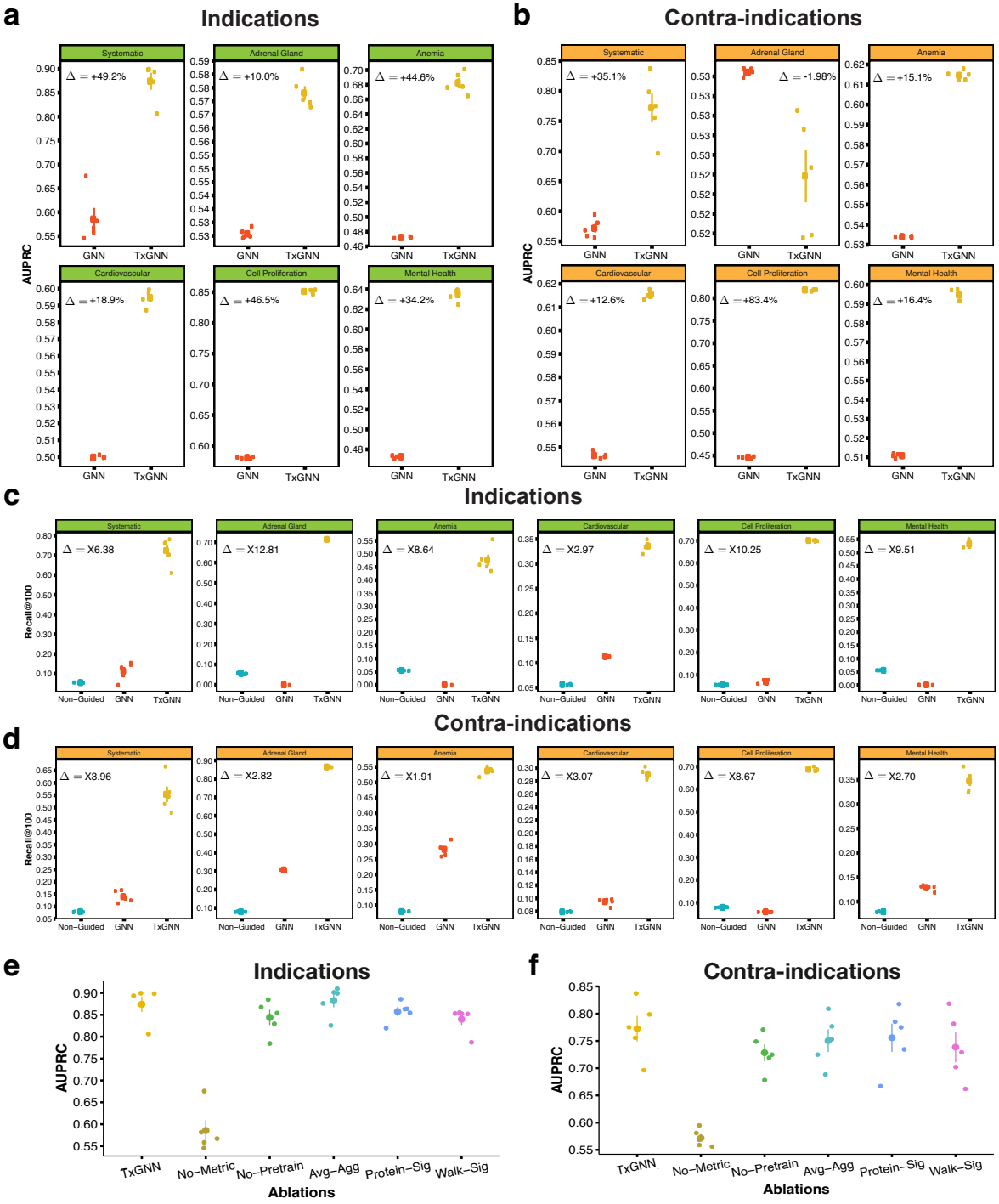

Figure 2: TxGNN predicts therapeutics indications and contraindications across disease areas with high precision. a. Performance of TxGNN and baseline GNN on indication prediction measured as the area under the precision-recall curve (AUPRC). A higher AUPRC is better. The result is reported with five random data splits. The mean is highlighted with the standard error as the error bar. Each panel is a type of disease split. **b.** Performance of TxGNN and GNN on contraindication prediction measured as AUPRC. **c.** Indication prioritization performance of TxGNN, GNN and Non-Guided as measured in Recall@100. Higher Recall@100 is better, with 1 retrieving every hit and 0 none retrieved. **d.** Contraindication prioritization performance of TxGNN, GNN and Non-Guided as measured in Recall@100. **e.** Performance of variants of TxGNN where we remove/modify each component of TxGNN to test its contribution to the model performance on indication prediction measured as AUPRC. **f.** Performance of variants of TxGNN on contra-indication prediction measured as AUPRC.

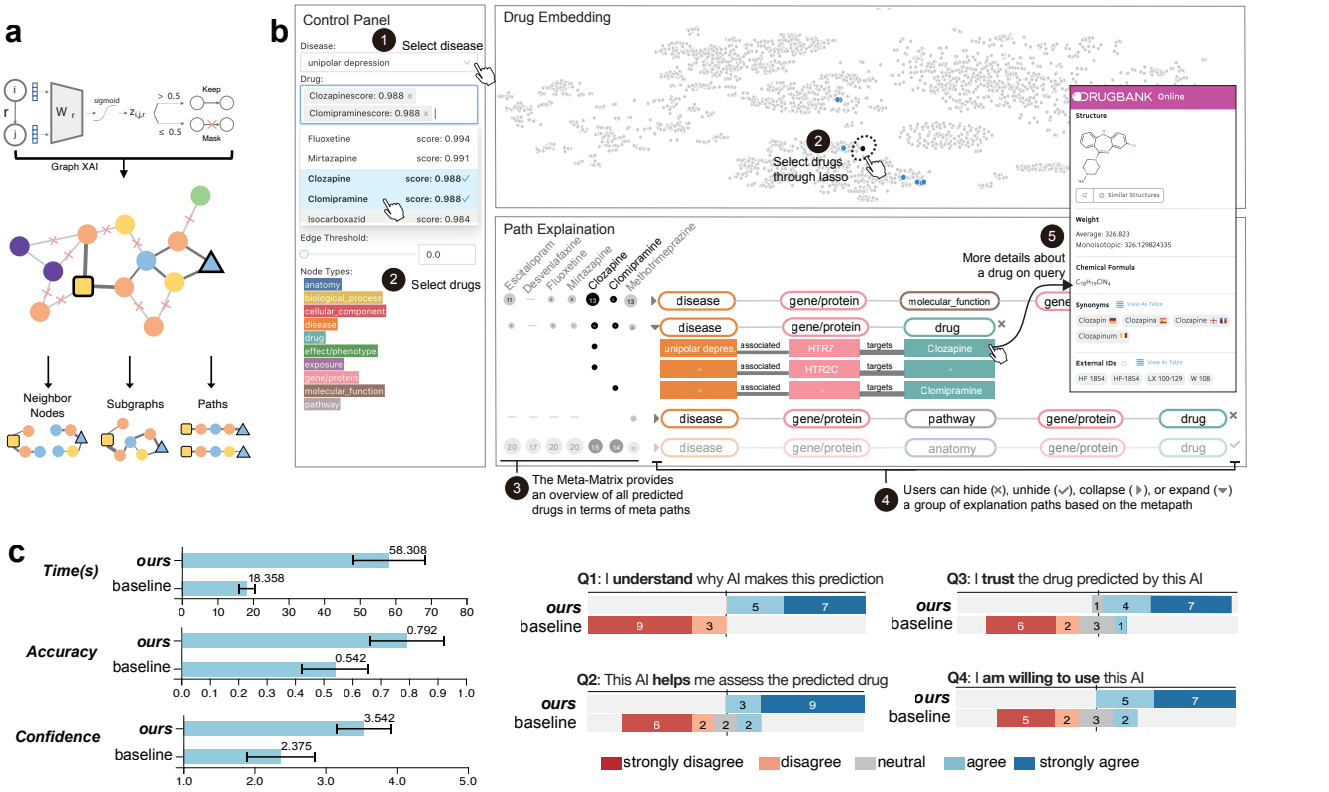

Figure 3: We generate, visualize, and evaluate explanations for TxGNN Predictions. a. We produce a sparse set biological relations to explain a drug-disease prediction by masking less informative relations in the neighborhood of the biological knowledge graph. **b.** DrugExplorer supports domain scientists in interpreting and interacting with model predictions and explanations. The 'Path Explanation' panel displays those biological relations that have been identified as critical for TxGNN's predictions about therapeutic use.**c.** We compare DrugExplorer with a no-explanation baseline in terms of user answer accuracy, exploration time, user confidence, and user agreement on 4 usability questions. Error bars indicate the 95% confidence intervals. Agree scores are placed to the right, disagree to the left.

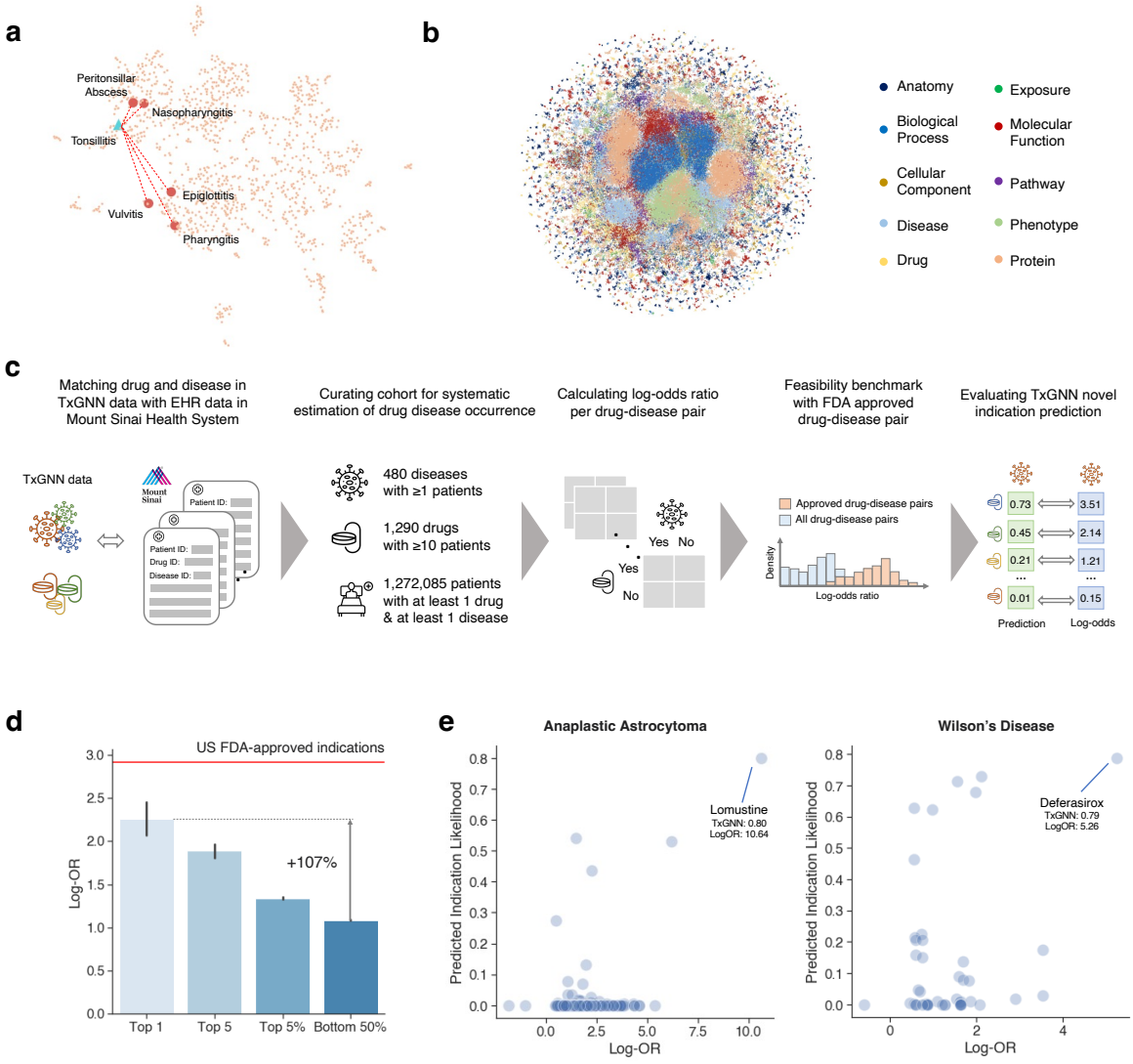

Figure 4: Understanding and evaluating the novel predictions made by TxGNN. a. The 5 most related diseases for Tonsillitis and their positions in the disease embedding space. We find the target disease is distant from the related diseases in the embedding manifold, suggesting TxGNN leverages a domain prior-guided selective aggregation scheme to enrich the target disease embedding. We use UMAP to generate the embedding landscape. **b.** TxGNN learn meaningful embeddings that capture biological hierarchy. **c.** Illustrated steps to evaluate TxGNN in large-scale EHR system at Mount Sinai. **d.** Evaluation of novel predictions made by TxGNN in Mount Sinai EHR system. The y-axis is the Log-OR of the disease-drug pairs, measuring the enrichment of the pair co-occurence, a proxy of indication. For every disease, we rank the TxGNN prediction and retrieve the top 1, top 5, top 5%, and bottom 50% of novel drug candidates and calculate the respective average Log-OR. The red horizontal line is the average Log-OR of FDA-approved indications. **e.** Examples of TxGNN predicted score against Log-OR for Anaplastic Astrocytoma and Wilson's disease. Each point represents a therapeutic candidate. The top 1 most probable candidate by TxGNN is highlighted with its TxGNN score and LogOR.

Online Methods

The Methods are structured as follows: 1) description of therapeutics-centered knowledge graph (Section 1), 2) description of machine learning approach (Section 2), and 3) outline of the experimental setup, benchmarking and evaluation (Section 3).

1 Therapeutics-centered knowledge graph

The knowledge graph is heterogeneous, with 10 types of nodes and 29 types of undirected edges. It contains 123,527 nodes and 8,063,026 edges. Tables 3 and 4 show a breakdown of nodes by node type and edges by edge type, respectively. The knowledge graph and all auxiliary data files are available via Harvard Dataverse at https://doi.org/10.7910/DVN/IXA7BM.

1.1 Primary data resources

The knowledge graph is compiled based on 17 primary knowledge bases that cover 10 types of biomedical entities and provide broad coverage of human disease, already-available drugs, and novel drugs in development. We briefly overview biological information retrieved from the knowledge bases, with details provided in Chandak *et al.*[12]: **Bgee.** Bgee[39] contains gene expression patterns across multiple animal species. Processing involved keeping only gold-quality calls and ensuring the anatomical entities were coded using the UBERON ontology. To extract only highly expressed genes in the anatomical entity, we filtered the data to keep data with an expression rank of less than 25,000. The processed data contains 1,786,311 anatomy-protein associations where gene expression was present or absent. **Comparative Toxicogenomics Database.** The Comparative Toxicogenomics Database (CTD)[40] focuses on environmental exposures' impact on human health. The processed data contains 180,976 associations between exposures and proteins, diseases, other exposures, biological processes, molecular functions, and cellular components. **DisGeNET.** DisGeNET[41] is a resource about the relationships between genes and human disease that experts have curated. The raw data contains 84,038 associations of genes with diseases and phenotypes. **DrugBank.** DrugBank[42] is a resource that contains pharmaceutical knowledge. Processing involved using the beautiful soup package to extract synergistic drug interactions. The processed data contains 2,682,157 associations. We also retrieved information about drug targets, enzymes, carriers, and transporters. The processed data contains 26,118 drug-protein interactions. **Drug Central.** Drug Central[43] curates information about approved drug indications and contraindications. The processed data contains 26,698 indication edges, 8,642 contraindication edges, and

1,917 off-label use edges. **Entrez Gene.** Entrez Gene[44] is a resource maintained by the NCBI that contains vast amounts of gene-specific information. Processing involved using the GOATOOLS package[45] to extract relations between genes and Gene Ontology terms. The processed data includes 297,917 associations of genes with biological processes, molecular functions, and cellular components. **Gene Ontology.** The Gene Ontology[46] network describes molecular functions, cellular components, and biological processes. Processing involved using the GOATOOLS package[45] to extract information for gene ontology terms and relations between go terms. The processed data contains 71,305 hierarchical associations between biological processes, molecular functions, and cellular components. **Human Phenotype Ontology.** The Human Phenotype Ontology[47] provides information on phenotypic abnormalities found in diseases. Processing involved parsing the ontology file to extract phenotype terms in the ontology, parent-child relationships, and cross-references to other ontologies. The processed data contains disease-phenotype, protein-phenotype, and phenotype-phenotype edges. We also obtained expertly curated annotations. The processed data includes 218,128 curated associations between diseases and phenotypes. **Mondo Disease Ontology.** Since the Mondo Disease Ontology[48] harmonizes diseases from a wide range of ontologies, including OMIM, SNOMED CT, ICD, and MedDRA, it was our preferred ontology for defining diseases. The processed data contains 64,388 disease-disease edges. **Orphanet.** Orphanet[49] is a database that gathers knowledge about rare diseases. The Orphanet portal has curated information about definitions, prevalence, management and treatment, epidemiology, and clinical description for 9348 rare diseases. **Physical protein-protein interactions.** Protein-protein interactions are composed of experimentally-verified interactions between proteins. The interactions we consider are diverse, including signaling, regulatory, metabolic-pathway, kinase-substrate, and protein complex interactions, which are unweighted and undirected. We use the human PPI network compiled by Menche *et al.*[1] as the starting resource. This resource integrates several protein-protein interaction databases, including TRANSFAC for regulatory interactions[50], MINT and IntAct for yeast to hybrid binary interactions[51,52], and CORUM for protein complex interactions[53]. Additionally, we retrieve protein-protein interaction information from BioGRID[54] and STRING[55] databases. We also consider the human reference interactome (HuRI) generated by Luck *et al.*[56], where we use the HI-union, a combination of HuRI and several related efforts to systematically screen for protein-protein interactions. The processed data contains 642,150 edges. **Reactome.** Reactome[57] is an open-source, curated database for pathways. The processed data contains 5,070 pathway-pathway and 85,292 protein-pathway edges. **Side Effect Resource.** The Side Effect Resource (SIDER)[58]

contains data about adverse drug reactions. We retrieved side-effect data and SIDER's drug to Anatomical Therapeutic Chemical (ATC) classification mapping. Processing involved extracting all side effects where the MedDRA term was coded at the "PT" or preferred term level and then mapping drugs from STITCH ID to ATC ID. The processed data 202,736 contains drug-phenotype associations. **UBERON.** UBERON[59] is an ontology containing human anatomy information. Processing involved extracting information about anatomy nodes and the relationships between them. The processed data includes 28,064 hierarchical relationships between anatomy nodes.

1.2 Building therapeutics-centered knowledge graph

We selected ontologies for each node type, harmonized primary data resources into a standardized format and resolved overlap across ontologies, and specified display names for relation/edge types to aid in the visualization of the knowledge graph by DrugExplorer and user studies.

Data standardization and ontologies. To harmonize primary data resources, we mapped them to common ontologies[12]. The node types 'drug', 'disease', 'anatomy', and 'pathway' are encoded as terms in DrugBank, Mondo, UBERON, and Reactome. Genes and proteins are treated as a single node type, 'gene/protein', and identified by Entrez Gene IDs. The node types 'biological process', 'molecular function', and 'cellular component' are defined using Gene Ontology terms. Disease phenotypes extracted from HPO and drug side effects extracted from SIDER are collapsed into a single node type, 'effect/phenotype', encoded using HPO IDs. Finally, 'exposure' nodes are defined using the ExposureStressorID field, which contains MeSH identifiers provided by the Comparative Toxicogenomics Database. Here, 'gene/protein' nodes are also referred to as protein nodes, and 'effect/phenotype' nodes are referred to as phenotype nodes interchangeably.

There was considerable overlap between phenotype and disease nodes across primary data resources. Overlapping nodes are effect/phenotype nodes in the Human Phenotype Ontology with the same ID number as disease nodes in Mondo Disease Ontology. They can be mapped from the Human Phenotype Ontology to Mondo using cross-references found in the Mondo. To resolve the overlap between phenotype nodes (Human Phenotype Ontology) and disease nodes (Mondo Disease Ontology), these overlapping phenotype nodes were converted to disease nodes by aligning edges across datasets as outlined in Chandak *et al.*[12].

Defining display relation names. To support the visualization of TxGNN's predictions, we added a 'display_relation' field, a descriptive version of the 'relation' field. When visualizing explanatory meta paths, the user sees two node types and a connecting relation name. For example, a

user would see a 'drug' connected to another 'drug' by the 'drug_drug' relation. Since the relation name becomes repetitive here, we introduced more meaningful descriptions through this display relation field. Some notable examples include converting 'drug_drug' to 'synergistic interaction', 'anatomy_protein_present' to 'expression present', and 'disease_phenotype_negative' to 'phenotype absent'. The display relation field does not map to the relation field one-one. For example, drug-protein relations in the knowledge graph can be displayed as 'target', 'enzyme', 'transporter', or 'carrier' depending on their specification in DrugBank. In the reverse, disease-disease and anatomy-anatomy relations have the display name 'is a' to indicate hierarchical relations.

Harmonizing graph structure and topology of therapeutics-centered knowledge graph. We merged the harmonized datasets into a heterogeneous knowledge graph and extracted its largest connected component using the approach outlined in Chandak *et al.*[12]. Since the knowledge graph is designed for therapeutic use prediction, we wanted to ensure that disease nodes in the graph were meaningful representations of diseases by collapsing disease nodes with nearly identical names into a single disease node. To this end, we adopted an approach previously validated[12] to group disease nodes with nearly identical names. First, disease groups were identified using automated string matching across disease names. This was achieved by selecting a starting disease via the ending-matching criteria and using the starting disease to find matches.

Matches included any diseases with the same initial phrase as the main disease name after deleting the ending word and any disease that contained all the words in the main disease name with no additional words, regardless of word order. Second, the intermediate disease groupings were tightened using ClinicalBERT[60] embedding similarities between disease names. The similarity between disease names was defined as the cosine distance between their ClinicalBERT embeddings. Finally, after applying an empirically chosen cutoff of similarity ≥ 0.98, we manually approved the suggested disease matches and assigned names to the new groups. After grouping, 22,205 diseases in the Mondo Disease Ontology were collapsed into 17,080 grouped diseases.

2 Geometric deep learning approach

Notation. We are given a heterogeneous knowledge graph (KG) $G = (\mathcal{V}, \mathcal{E}, \mathcal{T}_R)$ with nodes in the vertex set $v_i \in \mathcal{V}$, edges $e_{i,j} = (v_i, r, v_j)$ in the edge set $\mathcal{E}$, where $r \in \mathcal{T}_R$ indicates the relation type, v_i is called the head/source node and v_j is called the tail/target node. Each node also belongs to a node type set $\mathcal{T}_V$. Each node also has an initial embedding, which we denote as $\mathbf{h}_i^{(0)}$.

Problem definition. Given a disease i and drug j, we want to predict the likelihood of the drug being (1) indicated for the disease or (2) contraindicated for the disease. The goal is to inject factual knowledge from the KG into AI application to imitate important skills possessed by human experts, i.e., reasoning and understanding when forming hypotheses and making predictions about disease treatments.

2.1 Overview of TxGNN approach

TxGNN is a deep learning approach for mechanistic predictions in drug discovery based on molecular networks perturbed in disease and targeted by therapeutics. TxGNN is composed of four modules: (1) a heterogeneous graph neural network-based encoder to obtain biologically meaningful network representation for each biomedical entity; (2) a disease similarity-based metric learning decoder to leverage auxiliary information to enrich the representation of diseases that lack molecular characterization; (3) an all-relation stochastic pre-training followed by a drug-disease centric full-graph fine-tuning strategy; (4) a graph explainability module to retain a sparse set of edges that are crucial for prediction as a post-training step. Next, we expand each module in detail.

2.2 Heterogeneous graph neural network encoder

Given a knowledge graph, we aim to learn a numerical vector (i.e., network embedding) for each node such that it captures biomedical knowledge encapsulated in the neighboring relational structures. This is achieved by transforming initial node embeddings through several layers of local graph-based non-linear function transformations to generate embeddings[18,61]. These functions are optimized iteratively, given a loss function to gradually minimize the error of making poor therapeutic use predictions. Upon convergence, optimized functions generate an optimal set of node embeddings.

Step 1: Initialization. We denote the input node embedding $\mathbf{X}_i$ for each node i, which is initialized using Xavier uniform initialization[62]. For every layer l of message-passing, there are the following three stages:

Step 2: Propagating relation-specific neural messages. For every relation type, first calculates a transformation of node embedding from the previous layer $\mathbf{h}^{(l-1)}$, where the first layer $\mathbf{h}^{(0)} = \mathbf{X}$. This is achieved via applying a relation-specific weight matrix $\mathbf{W}_{r,M}^{(l)}$ on the previous layer

embedding:

$$\mathbf{m}_{r,i}^{(l)} = \mathbf{W}_{r,M}^{(l)} \mathbf{h}_i^{(l-1)} \tag{1}$$

Step 3: Aggregating local network neighborhoods. For each node v_i, we aggregate on the incoming messages $\{\mathbf{m}_{r,j}^{(l)} | j \in \mathcal{N}_r(i)\}$ from neighboring nodes of each relation r denoted as $\mathcal{N}_r(i)$ by taking the average of these messages:

$$\widetilde{\mathbf{m}^{(l)}}_{r,i} = \frac{1}{|\mathcal{N}_r(i)|} \sum_{j \in \mathcal{N}_r(i)} \mathbf{m}_{r,j}^{(l)} \tag{2}$$

Step 4: Updating network embeddings. We then combine the node embedding from the last layer and the aggregated messages from all relations to obtain the new node embedding:

$$\mathbf{h}_i^{(l)} = \mathbf{h}_i^{(l-1)} + \sum_{r \in \mathcal{T}_R} \widetilde{\mathbf{m}^{(l)}}_{r,i} \tag{3}$$

After L layers of propagation, we arrive at our encoded node embeddings $\mathbf{h}_i$ for each node i.

2.3 Predicting drug-disease relationships

Given the disease embedding and the drug embedding, we can predict the interaction between a disease-drug pair. As we have three relation types to predict for each disease-drug pair, we use a trainable weight vector $\mathbf{w}_r$ for each relation type. We then use DistMult[63] to calculate the interaction likelihood for that relation. Formally, for disease i, drug j, and relation r, we calculate the predicted likelihood p:

$$p_{i,j,r} = \frac{1}{1 + \exp(-\text{sum}(\mathbf{h}_i \cdot \mathbf{w}_r \cdot \mathbf{h}_j))}. \tag{4}$$

2.4 Similarity Disease Search to Enrich Molecularly Uncharacterized Disease Embedding

Diseases receive various degrees of research, given their prevalence, complexity, and so on. For example, we know very little about the molecular underpinnings of many rare diseases[64,65]. Nevertheless, these diseases usually present the most promising therapeutic opportunities[66]. Due to the lack of understanding of these diseases, machine learning predictions have become more important than ever. However, the limited research on these diseases is reflected by the scarcity of

relevant nodes and edges around these diseases in our biological knowledge graph. Because of this sparsity, the graph embedding tends to be lower quality. For example, if a disease has zero connections in the KG (i.e., no existing knowledge), then the disease embedding will be the random initialization. Empirically, we see that prevailing GNN approaches have drastically lower predictive performance on our disease-centric splits to simulate this realistic property of diseases compared to random splits (Figure 1g).

We hypothesize that the obtained network embedding for these diseases is not meaningful due to this limited prior in the KG. Thus, a model must subsidize and augment the network embedding for these molecularly uncharacterized diseases. Our key insight is that human physiology is a connected system where diseases are similar across dimensions (e.g., lung cancer is similar to brain cancer in the dimension of cancer diseases, while lung cancer is similar to asthma in the dimension of lung diseases). Therefore, if we could borrow useful information from a set of similar diseases that are relatively well-characterized in the KG through the model, we could augment the embedding of the candidate disease and improve the prediction.

To do that, we propose a three-step procedure: (1) a disease signature vector that captures the intricate disease similarities; (2) an aggregation mechanism that integrates the different similar diseases into a robust auxiliary embedding that can subsidize original disease embedding; (3) a gating mechanism to control the effect between the original disease embedding and the auxiliary disease embedding since many well-characterized diseases have sufficient embeddings and do not need subsidies. We discuss each of the three steps in detail below.

Network-based Disease Signature Profiling. The overall goal for this module is to obtain a signature vector $\mathbf{p}_i$ for every disease i. There are numerous ways to calculate the similarity between two diseases. As disease representations generated by the graph neural network alone are not sufficient to characterize the candidate disease, they ideally should not be directly used to calculate similarity. Instead, we resort to graph theoretical techniques that are rooted in the field of network science[5]. We consider the following three types of signature functions:

- **Protein signatures (PS):** The mechanism of actions for small molecule drugs is to act upon protein targets in the disease pathway[67]. Thus, the ideal disease signature should preserve similarity in the protein target space. If two diseases have similar proteins in their corresponding disease pathways, they are more likely to have a similar treatment mechanism[1,68]. This key observation motivates the protein signature[69]. We have a bit vector for each disease where each bit corre-

sponds to a specific protein. A bit is flipped to one if the bit corresponds to a protein in the disease pathway. Formally, for disease i, the protein signature is defined as:

$$\mathbf{p}_i^{\text{PS}} = [\ \text{p}_1\ \cdots\ \text{p}_{|\mathcal{V}_\text{P}|}\ ], \tag{5}$$

where

$$\text{p}_j = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i^{\text{P}} \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

and $\mathcal{N}_i^{\text{P}}$ is the set of proteins that lie in the 1-hop neighborhood of disease i and $|\mathcal{V}_\text{P}|$ the number of total available proteins. To calculate similarity between two diseases i, j, we use dot product:

$$\text{sim}^{\text{PS}}(i,j) = \mathbf{p}_i^{\text{PS}} \cdot \mathbf{p}_j^{\text{PS}} = |\mathcal{N}_i^{\text{P}} \cap \mathcal{N}_j^{\text{P}}|. \tag{7}$$

The similarity directly measures the number of intersecting proteins in the disease pathway of i, j. If the similarity is high, we know these two diseases have a larger number of intersecting diseases, which increases the probability of similar treatment mechanisms.

- **All-node-types signatures (AT):** Human knowledge about disease pathways are vastly incomplete. Thus, some diseases may not have complete protein pathways in the knowledge graph, which leads to biased protein signatures. Additional biological knowledge about diseases could potentially benefit. In the knowledge graph, other node types connect to diseases, including effect/phenotype, exposure, and disease. Since the local neighborhood can define some characteristics of diseases, we can extend the principle of protein signature, such that if two diseases share the same nodes in these additional node types, they have similar biological underpinnings. We call these all-node-types signatures. Formally, for disease i, the protein signature is defined as:

$$\mathbf{p}_i^{\text{AT}} = [\ \text{p}_1 \cdots \text{p}_{|\mathcal{V}_\text{P}}\ \text{ep}_1 \cdots \text{ep}_{|\mathcal{V}_{\text{EP}}|}\ \text{ex}_1 \cdots \text{ex}_{|\mathcal{V}_{\text{EX}}|}\ \text{ep}_1 \cdots \text{ep}_{|\mathcal{V}_{\text{EP}}|}\ \text{d}_1 \cdots \text{d}_{|\mathcal{V}_\text{D}|}\ ], \tag{8}$$

where

$$\text{p}_j = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i^{\text{P}} \\ 0 & \text{otherwise} \end{cases}, \text{ep}_j = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i^{\text{EP}} \\ 0 & \text{otherwise} \end{cases}, \text{ex}_j = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i^{\text{EX}} \\ 0 & \text{otherwise} \end{cases} \text{d}_j = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i^{\text{D}} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and $\mathcal{N}_i^{\text{EP}}, \mathcal{N}_i^{\text{EX}}, \mathcal{N}_i^{\text{D}}$ is the set of effect/phenotype, exposure, diseases nodes lie in the 1-hop neighborhood of disease i and $|\mathcal{V}_{\text{EP}}|, |\mathcal{V}_{\text{EX}}|, |\mathcal{V}_{\text{D}}|$ the number of total available effect/phenotype, exposure, diseases respectively. We also adopt the dot product as the similarity measure, which means the similarity is the sum of all shared nodes across the four node types:

$$\text{sim}^{\text{AT}}(i,j) = \mathbf{p}_i^{\text{AT}} \cdot \mathbf{p}_j^{\text{AT}} = |\mathcal{N}_i^{\text{P}} \cap \mathcal{N}_j^{\text{P}}| + |\mathcal{N}_i^{\text{EP}} \cap \mathcal{N}_j^{\text{EP}}| + |\mathcal{N}_i^{\text{EX}} \cap \mathcal{N}_j^{\text{EX}}| + |\mathcal{N}_i^{\text{D}} \cap \mathcal{N}_j^{\text{D}}|. \quad (10)$$

- **Diffusion signatures (DS):** The above two signatures rely on the first-hop neighbor of the diseases, while higher-hop neighbors may contain useful molecular characterization. Diffusion signature simulates many random walks, where each random walk is a path of length h starting from the disease i: $w = v_i \xrightarrow{e_{i,1}} v_1 \cdots v_{h-1} \xrightarrow{e_{h-1,h}} v_h$[70]. The set of visited nodes in the k-th random walk from disease node i is denoted as $\mathcal{W}_i^k$. $\cap_k \mathcal{W}_i^k$ represents the total set of visited nodes across all walks, and we can calculate the normalized visitation probability for visited node j as:

$$f_j = \frac{\sum_k \sum \mathbb{1}_{\mathcal{W}_i^k = j}}{\sum_k |\mathcal{W}_i^k|} \quad (11)$$

These nodes correspond to a multi-hop snapshot of molecular interactions centering around the diseases, and the visitation probability corresponds to the influence level. Given this probability score, we can obtain the diffusion signature for disease node i:

$$\mathbf{p}_i^{\text{DS}} = [\, f_1 \ \cdots \ f_{|\mathcal{V}_{\text{P}}|} \,]. \quad (12)$$

For diffusion signature, we still use the dot product:

$$\begin{aligned} \text{sim}^{\text{DS}}(i,j) = \mathbf{p}_i^{\text{DS}} \cdot \mathbf{p}_j^{\text{DS}} &= \sum_u^{|\mathcal{V}_{\text{P}}|} \frac{\left(\sum_k \sum \mathbb{1}_{\mathcal{W}_i^k = u}\right) \cdot \left(\sum_k \sum \mathbb{1}_{\mathcal{W}_j^k = u}\right)}{(\sum_k |\mathcal{W}_i^k|)^2} \\ &\sim \sum_u^{|\mathcal{V}_{\text{P}}|} \left(\sum_k \sum \mathbb{1}_{\mathcal{W}_i^k = u}\right) \cdot \left(\sum_k \sum \mathbb{1}_{\mathcal{W}_j^k = u}\right). \end{aligned} \quad (13)$$

Note the denominator $(\sum_k |\mathcal{W}_i^k|)^2 = (|k| * h)^2$ is a constant. Intuitively, the similarity between diseases i and j is higher when two diseases visit more shared nodes at a higher frequency.

Given the selected signature for diseases and calculated similarities among the diseases, for

a query disease, we can then obtain k most similar diseases for a query disease i:

$$\mathcal{D}_{\text{sim},i} = \text{argmax}^{k}_{j \in \mathcal{V}_{\text{D}}} \text{sim}(i, j) \tag{14}$$

Disease-disease metric learning. Given the set of similar diseases, we aim to obtain an embedding that fuses different similarity dimensions into a single embedding sufficient to enhance the query disease that might be sparsely annotated. We use a weighted scheme, where the similarity score weights each disease as follows:

$$\mathbf{h}_i^{sim} = \sum_{j \in \mathcal{D}_{\text{sim}}} \frac{\text{sim}(i, j)}{\sum_{k \in \mathcal{D}_{\text{sim}}} \text{sim}(i, k)} * \mathbf{h}_j. \tag{15}$$

Embedding gating. The final step is to update the original disease embedding $\mathbf{h}_i$ with the disease-disease metric learning embedding $\mathbf{h}_i^{sim}$ through a gating mechanism. The gating mechanism consists of a scalar $c \in [0, 1]$ that balances between these two types of embeddings. Note that this requires special treatment because for a disease well-characterized in the knowledge graph, we do not need the disease-disease metric learning embedding, and it potentially can even bias the final embedding. The disease-disease metric learning embedding is most useful for uncharacterized diseases since the original disease embedding is insufficient to characterize molecular mechanisms. Note that the learnable attention mechanism to select whether or not to attend original/augmented embedding does not work well because the training examples are usually the most characterized, which makes the attention weight assign high importance to the original embeddings and leaves the subsidy embedding unused. Instead, we propose a heuristic algorithm that assigns weight based on the node degree for the drug-disease relation type that is under calculation: $|\mathcal{N}_i^r|$. The higher the degree, the more well-characterized the disease is, and the less weight should be assigned to the disease-disease metric learning embedding and vice-versa. Also, this scalar should have a very high value when the node degree is minimal (0 or 1) and decreases quickly when the node degree increases. To approximate this effect, we use an inflated exponential distribution density function with $\lambda = 0.7$:

$$c_i = 0.7 * \exp(-0.7 * |\mathcal{N}_i^r|) + 0.2 \tag{16}$$

We observe the result is not sensitive to λ (Supplementary Figure 6). Finally, we use parameter

search and find optimal $\lambda = 0.7$. Then, we can finally obtain an augmented disease embedding:

$$\widehat{\mathbf{h}}_i = c_i * \mathbf{h}_i^{sim} + (1 - c_i) * \mathbf{h}_i \tag{17}$$

We then use this augmented disease embedding to feed into the DistMult decoder[63] described in Section 2.3.

2.5 Training TxGNN deep graph models

Objective function. The training objective is to accurately predict whether or not a relation holds given two entities in the knowledge graph. This can be formulated as a binary classification task for each relation. The positive samples consist of all pairs (i, j) with diverse relation types $r \in \mathcal{T}_R$. We denote this as $\mathcal{D}_+$ and the label $y_{i,r,j} = 1$. Similarly, for each pair, we generate negative counterparts through sampling described in Section 3, denoted as $\mathcal{D}_-$. For each pair i, j and its relation type r, the model predicts the likelihood $p_{i,j,r}$ and the training loss is calculated via binary cross entropy loss:

$$\mathcal{L} = \sum_{(i,r,j) \in \mathcal{D}_+ \cup \mathcal{D}_-} y_{i,r,j} * \log\left(p_{i,r,j}\right) + \left(1 - y_{i,r,j}\right) * \log\left(1 - p_{i,r,j}\right) \tag{18}$$

Previous work has focused on knowledge graph completion, leading them to optimize over the entire set of relations in the knowledge graph[71]. However, since we are only interested in drug-disease relations, training on all relation types could move the model capacity toward capturing knowledge we are not interested in. Conversely, since complicated biological mechanisms drive drug-disease relations, the vast array of biomedical relations in the knowledge graph presents a unique information source that holistically describes biological systems. Thus, the challenge is to ultimately do well on a small set of relations while also transferring knowledge positively from the larger relation set.

To solve this challenge, TxGNN uses a pre-training strategy. During pre-training, TxGNN is trained to predict relations across the entire set of relation types in the KG using stochastic mini-batching. This process allows TxGNN to distill biomedical knowledge into enriched node embeddings. Next, during fine-tuning, TxGNN zooms in and trains only on the drug-disease relations to obtain more granular drug-disease-specific embeddings that optimize for the best therapeutic outcomes.

Pre-training. TXGNN is first pre-trained on millions of biomedical entity pairs across the entire set of relations. As there are millions of edges, full-graph training is computationally infeasible. Thus, we use stochastic mini-batching to train only on a set of pairs in each training step. Each epoch goes through all pairs of data in the training knowledge graph. During pre-training, degree-adjusted disease augmentation is turned off since it is unavailable for other node types. All relations are treated equally. The weights of the trained encoder model are then used to initialize the encoder model weights during fine-tuning. Note that the weight in the decoder DistMult $\mathbf{w}_r$ is reinitialized before fine-tuning to discourage the effect of negative transfer.

Fine-tuning. After pre-training, we have an initialization that captures general biological knowledge. Next, we focus on optimizing drug-disease relation prediction. To do that, we only use the samples of all drug-disease pairs (i, j) with relation types $r \in \{\text{indication, contraindication, rev\_indication, rev\_contraindication}\}$. The rest of the relations are discarded in the training objective but are included in the knowledge graph for messaging the passing of drug and disease nodes. During fine-tuning, the degree-adjusted inter-disease embedding is turned on.

The complete TXGNN model is pre-trained and fine-tuned in an end-to-end manner. The best-performing model on the validation set is then used for performance evaluation on the test set and downstream machine-learning analyses.

2.6 Explaining model predictions

Distilling model predictions into mechanisms of molecular networks perturbed in disease and targeted by therapeutics. A machine learning model can provide accurate disease treatment predictions. However, for domain scientists' adoption, prediction alone is not sufficient. Thus, a model is expected to generate why it outputs this prediction in a form familiar to domain experts' decision-making. In the case of treatment prediction, an ideal form of explanation is to simulate how drug developers approach drug-disease relation — that is, to understand how a drug perturbs the local biological system such that it creates a therapeutic effect on the disease pathway. As TXGNN leverages the large-scale biological knowledge graph, we can probe into the local neighborhood around a query drug-disease node and pinpoint the exact mechanism contributing to the prediction. However, as a biological network is complex, making meaningful explanations requires a model to prune most uninformative edges and extract a sparse version of the local neighborhood. This can be formulated as a graph explainability problem where we try to identify a sparse set of edges where the model can make a faithful prediction using these edges[28]. To achieve it, we

develop a post-training graph explainability module, adapted from GraphMask approach[27], that can drop spurious edges from the dataset and retain a sparse set of edges that contribute most towards the prediction. Next, we describe the mathematical formulation of GraphMask as used by TxGNN.

Local explanation subgraphs through pruning superfluous biomedical relations. Given a trained disease treatment prediction model, for each target node j and one of the neighbor source node i with edge $e_{i,j}$ at layer l, we have intermediate messages $\mathbf{m}_{r,i}^{(l)}$, $\mathbf{m}_{r,j}^{(l)}$ given a relation r. Given these two embeddings, we concatenate them and feed them into a relation-wise single-layer neural network parameterized by $\mathbf{W}_{g,r}^{(l)}$ to predict the likelihood of masking the message from source node i when we compute the target node j embedding, followed by a gate consisting of a sigmoid layer to squeeze the likelihood into 0 to 1 and an indicator function to decide whether or not to drop the edge:

$$z_{i,j,r}^{(l)} = \mathbf{1}_{[\mathbb{R}>0.5]}\left(\text{sigmoid}\left(\mathbf{W}_{g,r}^{(l)}\left(\mathbf{m}_{r,i}^{(l)} \| \mathbf{m}_{r,i}^{(l)}\right)\right)\right), \tag{19}$$

such that $z_{i,j,r}^{(l)} \in 0, 1$. In practice, we add a location bias of 3 to the sigmoid function at initialization. This ensures that for initialized inputs, the biased sigmoid outputs are close to 1, meaning that the gates are open at initialization, and the model can adaptively close the gates to mask edges in the subgraph. This step is crucial as random initialization starts by dropping random edges. The gap between the original and updated predictions is big, so the model minimizes the gap instead of balancing the two objectives. Next, instead of simply removing the message when the gate outputs 0, the message is replaced with a learnable baseline vector $\mathbf{b}_r^{(l)}$ for each relation r and layer l. Thus, the updated message from source node i to target node j becomes:

$$\hat{\mathbf{m}}_{i,r}^{(l)} = z_{i,j,r}^{(l)} \cdot \mathbf{m}_{i,r}^{(l)} + (1 - z_{i,j,r}^{(l)}) \cdot \mathbf{b}_r^{(l)} \tag{20}$$

Then, we can proceed with the standard message aggregation and update steps to compute the updated node embedding (Section 2.2), feed to inter-disease augmentation (Section 2.4), and generate the updated predictions $\hat{p}$ between a drug and a disease (Section 2.3). The GraphMask gate weights are optimized with two objectives. The first objective is faithfulness, where the updated predictions after masking are encouraged to be the same as the original prediction outcome. The second objective is to promote the model to mask as much as possible. These two objectives present a trade-off since larger amounts of masking would lead to a more significant gap between

updated/original predictions. This can be formulated as constrained optimization using Lagrange relaxation, where we strive to maximize the Lagrange multiplier λ for constraint while minimizing the main objective. Formally, we use the loss function below:

$$\max_{\lambda} \min_{\mathbf{W}_g} \sum_{k=1}^{L} \sum_{(i,j,r)\in\mathcal{D}_+\cup\mathcal{D}_-} \mathbf{1}_{[\mathbb{R}\neq 0]} z_{i,j,r}^{(k)} + \lambda \left(\|\hat{p}_{i,j,r} - p_{i,j,r}\|_2^2 - \beta \right), \tag{21}$$

where β is the margin between the updated and original prediction. After training, we can remove edges (i, j, r) that have $z_{i,j,r}^{(k)} = 0$ and use the retained edges as the explanations. We can also use the value calculated before the indicator function to measure the level of contributions to the prediction and can be used as adjustments of more granular differences.

Necessary adaptations of GraphMask approach for biomedical knowledge graphs. We modify GraphMask[27] in the following manner to generate meaningful local explanation subgraphs of the knowledge graph. (1) Instead of a complex gate that outputs scores close to 0/1, we adopt a smooth sigmoid gate where predictions are uniform across 0 to 1. This is important because we find hard concrete map edges to 1 as long as they affect the model prediction. However, this still keeps many edges that preclude us from making acceptable medical explanations. The sigmoid gate instead allows us to distinguish the intensity of contributions and provides a flexible framework. By setting a threshold, we remove large amounts of positive edges and only retain ones crucial for the model prediction. (2) Second, while GraphMask has a single learnable weight for every edge in the dataset, we adopt a separate weight for each relation. Since embeddings across relations are different, the model assigns uniformly high scores for all edges of a given relation type despite edges varying in relevance. Using relation-specific weights allows the model to capture the importance scores of individual edges.

3 Experimental setup and implementation details

Next, we outline the experimental setup, including information on performance evaluation and dataset splits. We also provide details on the practical implementation of TxGNN deep graph models.

3.1 Creating dataset splits for rigorous performance evaluation

Our dataset presents well-studied information and includes the vast majority of existing treatments. As a result, it is easy to predict treatments for diseases with various pre-existing treatments. How-

ever, for zero-shot prediction of therapeutic use, we need to make good predictions on conditions with few or no current treatments available. The classical random split of edges of the knowledge graph into training and testing sets would not simulate this application. In the random split, for diseases with many known indications, the model would view some of these drug-disease edges in training and thus easily predict therapeutic use based on drug similarity. However, this would prevent the model from assimilating meaningful biological knowledge. Therefore, we consider the following dataset splits into training and test sets:

- **Disease area splits:** Many diseases of therapeutic interest have no existing treatments and lack significant biological knowledge. To evaluate whether TxGNN would be robust to predicting drug-disease relationships for such diseases, we develop data splits that simulate well-studied diseases as molecularly uncharacterized diseases. We cannot directly test on molecularly uncharacterized diseases, such as rare diseases, because the treatments are too few to build a confident machine learning model. We select five disease groups: cell proliferation, mental health, cardiovascular diseases, anemia, and adrenal gland diseases, and then extract groups for these diseases from the Disease Ontology hierarchy such that group includes the disease and all its children. Since these well-studied diseases have many drug-disease relationships, we can easily evaluate the model's performance during the simulation.

  For each disease, we create a separate data split as follows. First, all the drug-disease edges connected to the diseases in the group are moved to the test set. As a result, TxGNN has no information about existing indications and contraindications use edges for the chosen disease group during training. This simulates the lack of existing treatments encountered with molecularly uncharacterized diseases. Next, we remove a significant fraction (5% of the knowledge graph size) of the local 1-hop subgraph neighborhood for the disease group. Again, this simulates the limited biological understanding of molecularly uncharacterized diseases. Dataset statistics of each disease area split is provided in Table 2.

- **Systematic dataset splits:** The deployed machine learning model should excel at predicting diseases without known treatments. Predicting new treatments for diseases that already have treatments is easier than predicting diseases without treatments. This is because information about existing treatments can directly illuminate the molecular mechanism, and drug similarity can help infer new treatments. Thus, to robustly test our model, we design this split to systematically study prediction on novel unseen diseases. To do that, we first randomly split the entire set

of diseases. Then, we take all drug-disease relations associated with the testing set of diseases to the test set such that there are no known treatments during training and the testing set consists of novel diseases. The testing set has around 100 different diseases in each randomly seeded run.

- **Disease-centric dataset splits:** We adopt a disease-centric evaluation to simulate realistic usage of drug candidate prioritization. First, for each disease in the test set, we pair it with all other drugs in the KG, except the drug-disease relations in the training set. Then, we make predictions for all pairs and rank based on the likelihood of interaction. We then retrieve the top K drugs and compute the recall (*i.e.*, how much drug and disease in the testing set are in the top K). Finally, we build a baseline of random screening where we randomly sample top K drugs from the drug set and compute the recall.

3.2 Modeling molecular and clinical relationships

In graphs, each edge typically has a direction and points from the source to the target node. However, in our biological knowledge graph, edges are bidirectional. For example, a drug A indicated for disease B is represented in TxGNN by a tuple $(A, \text{indication}, B)$. Similarly, disease B can be treated by drug A, corresponding to a tuple $(B, \text{rev\_indication}, A)$. For homogeneous relation type (*e.g.*, protein-protein interactions) where the head and tail node belongs to the same node types, there is no additional reverse relation type as the reverse edges are collapsed into the original relation type. Thus, we add these reverse relation types to the knowledge graph, following standard practice. For the sake of notation, when the reverse relation has a different relation type from the original type r, we denote the reverse relation type as r^c.

3.3 Negative sampling for training TxGNN models

As we only have positive data, negative data are constructed via sampling. The sampling from the unobserved simulates the realistic constraint where most possible drugs do not interact with the disease. For each relation type, we fix the source nodes and permute the target nodes through either random sampling from the set of nodes associated with this relation type's target nodes or a weighted sampling based on the degree of the target nodes. As we conduct reverse relation type construction, the source node type would also be shuffled and included in the negative samples when we do sampling for the reverse relation type. This concept of negative sampling based on shuffling target nodes is crucial. For example, suppose we want to study drugs A that can treat disease B, then we narrow down to the relation $(B, \text{rev\_indication}, A)$ instead of the

$(A, \text{indication}, B)$.

3.4 Hyperparameter tuning

We conduct hyperparameter tuning using Hyperband on validation set micro AUROC using complex disease split following two stages. The first is to optimize the parameters for pre-training and fix fine-tuning parameters, where we conduct a sweep of grid search with a learning rate of $\{1e-4, 5e-4, 1e-3\}$, batch size of $\{1024, 2048\}$, and epoch size of $\{1, 2, 3\}$. Next, we fix the pre-training parameters and do a grid search for fine-tuning parameters with the hidden size of $\{64, 128, 256, 512\}$, input size of $\{64, 128, 256, 512\}$, output size of $\{64, 128, 256, 512\}$, number of inter-disease prototypes of $\{3, 5, 10, 20, 50\}$ and learning rate of $\{1e-4, 5e-4, 1e-3\}$. We obtain a final set of hyperparameters with a pre-training learning rate of $1e-3$, batch size of 1024, epoch size of 2, the fine-tuning learning rate of $5e-4$, hidden size of 512, input size of 512, output size of 512, number of prototypes 3.

3.5 Implementation details

The TxGNN is implemented using DGL[72] and PyTorch[73] Python deep learning frameworks. We use Pandas[74], Numpy[75] for data processing and computing; scikit-learn[76] for evaluation metrics; seaborn[77], matplotlib[78], UMAP[79] for visualization; Weights and Bias (https://www.wandb.ai) for training monitoring and hyperparameter tuning. We train the model with one NVIDIA Tesla V100 GPU in a server. TxGNN Explorer is implemented in JavaScript using React.js[80], D3.js[81], and Ant Design[82]. The graph data is managed using Neo4j database[83]. TxGNN Explorer communicates with TxGNN through a Python web server built with Flask[84].

3.6 Usability study of TxGNN with medical experts

We designed and developed TxGNN Explorer following a user-centric design study process[30], which compared three visual presentations of GNN explanations from users' perspectives and motivated the implementation of path-based explanations based on user feedback. We evaluated the usability of TxGNN Explorer by comparing it with a non-explanation baseline that shows drug predictions and corresponding confidence scores. Twelve medical experts (7 males, 5 females, avg. age=34.25) were recruited for the usability study through personal contacts, Slack channels, and email lists in collaborating institutions. We conducted the evaluation on Zoom due to COVID-19-related restrictions. Each participant logged in to the user study system (Supplementary Figure S5) using their computers and shared their screens with the interviewer. The order of predictions

and the order of two conditions (TxGNN Explorer or baseline) were randomized and counterbalanced across participants. For each drug assessment task, the participants were asked to 1) decide whether this drug prediction is correct (*i.e.*, the drug can potentially be used to treat the disease) and 2) give a confidence score for their decision using a 5-point Likert scale (1=not confident at all, 5=completely confident). The study system automatically recorded the completion time for assessing each prediction. After assessing all predictions, participants provided subjective ratings for the two conditions in terms of *Trust*, *Helpfulness*, *Understandability*, and *Willingness to use* via a 5-point Likert scale (1=strongly disagree, 5=strongly agree).

3.7 Evaluations within a large healthcare system

We leveraged patient data from the Mount Sinai Health System's electronic health records (EHR) in New York City, U.S., to assess patterns from predictions in clinical practice. All clinical data were deidentified, and the Mount Sinai Institutional Review Board approved the study. The cohort consisted of over 10 million patients and was filtered for patients over 18 years of age with at least one drug and at least one diagnosis on record, leaving 1,272,085 patients. This cohort was 40.1 percent male, and the average age was 48.6 years (SD: 18.6 years). Table 5 shows the dataset's racial breakdown.

All disease and medication data were captured using the Observational Medical Outcomes Partnership (OMOP)[85,86] standard data model. We produce predictions for the 1,363 diseases with indications by training the full knowledge graph with only 5% of randomly selected drug-disease pairs as a validation set for early stopping. This experiment does not evaluate zero-shot performance for all 17,080 diseases since the model has more confidence in conditions with known indications. Disease names in the TxGNN prediction dataset were matched to SNOMED or ICD-10 codes and finally mapped to OMOP concepts in the Mount Sinai data system. We included only diseases with at least one patient diagnosis in the dataset, leaving 480 conditions. Medication names in the TxGNN prediction were matched to DrugBank ID, which was then mapped to RxNorm IDs and OMOP concepts. We included only medications with at least one patient order in the dataset, leaving 1,290 medications. Next, we included drug-disease pairs for which at least one patient was listed with both the drug and the disease, leaving 1,236 drugs and 470 diseases. For each drug-disease pair, we created a contingency table. Using the SciPy[87] library's Fisher exact function, we computed 2-sided odds ratios and p-values for each pair. Finally, we used the statsmodels[88] Python library's multi-test function to apply a two-sided Bonferonni correction on

the previously generated p-values. Finally, we noted statistically significant drug-disease pairs as those with $p < 0.005$.

Drug name	Active ingredient	Disease	Approval	FDA Number	Orphan	TxGNN	Percentile
Vabysmo	Faricimab	Macular degeneration	01/28/2022	BLA761235	No	0.938	2.25%
Welireg	Belzutifan	von Hippel-Lindau disease	08/13/2021	NDA215383	Yes	0.720	4.11%
Mounjaro	Tirzepatide	Type 2 diabetes mellitus	05/13/2022	NDA215866	No	0.286	12.50%
Ztalmy	Ganaxolone	CDKL5 disorder	03/18/2022	NDA215904	Yes	0.335	18.73%
Leqvio	Inclisiran sodium	Familial hypercholesterolemia	12/22/2021	NDA214012	No	0.301	19.32%
Tezspire	Tezepelumab-ekko	Asthma	12/17/2021	BLA761224	No	0.233	32.41%
Vtama	Tapinarof	Psoriasis	05/23/2022	NDA215272	No	0.261	32.70%
Adbry	Tralokinumab	Atopic dermatitis	12/27/2021	BLA761180	No	0.040	50.37%
Vonjo	Pacritinib citrate	Myelofibrosis	02/28/2022	NDA208712	Yes	0.011	63.14%
Livtencity	Maribavir	Cytomegalovirus infection	11/23/2021	NDA215596	Yes	0.033	66.37%

Table 1: Evaluation of novel TxGNN predictions against recently developed therapies. Out of 7,957 therapeutic candidates, TxGNN ranked recent FDA-approved drugs high.

Disease area	Number of diseases	Number of indications	Number of Contraindications
Adrenal gland	7	41	374
Anemia	19	88	752
Cardiovascular diseases	113	453	4,242
Diseases of cell proliferation	213	1022	1079
Mental health diseases	60	355	1,567

Table 2: Statistics on disease-area-based dataset splits used to evaluate the zero-shot prediction of therapeutic use. Given all diseases in a given disease area, all indications and contraindications were removed from the dataset used to train machine learning models. Additionally, a large fraction (95%) of the connections between biomedical entities to these diseases were removed from the therapeutics-centered knowledge graph. Disease-area splits were curated to evaluate model performance on diseases with limited molecular understanding and no existing treatments.

Node Type	Count	Percent (%)
Biological process	28,642	22.1
Protein	27,671	21.4
Disease	17,080	13.2
Phenotype	15,311	11.8
Anatomy	14,035	10.8
Molecular function	11,169	8.6
Drug	7,957	6.2
Cellular component	4,176	3.2
Pathway	2,516	1.9
Exposure	818	0.6
Total number of nodes	129,375	100.0

Table 3: Statistics on nodes in the therapeutics-centered knowledge graph.

Relation	Count	Percent (%)
Anatomy – Protein (present)	3,036,406	37.5
Drug – Drug	2,672,628	33.0
Protein – Protein	642,150	7.9
Disease – Phenotype (positive)	300,634	3.7
Biological process – Protein	289,610	3.6
Cellular component – Protein	166,804	2.1
Disease – Protein	160,822	2.0
Molecular function – Protein	139,060	1.7
Drug – Phenotype	129,568	1.6
Biological process – Biological process	105,772	1.3
Pathway – Protein	85,292	1.1
Disease – Disease	64,388	0.8
Drug – Disease (contraindication)	61,350	0.8
Drug – Protein	51,306	0.6
Anatomy – Protein (absent)	39,774	0.5
Phenotype – Phenotype	37,472	0.5
Anatomy – Anatomy	28,064	0.3
Molecular function – Molecular function	27,148	0.3
Drug – Disease (indication)	18,776	0.2
Cellular component – Cellular component	9,690	0.1
Phenotype – Protein	6,660	0.1
Drug – Disease (off-label use)	5,136	0.1
Pathway – Pathway	5,070	0.1
Exposure – Disease	4,608	0.1
Exposure – Exposure	4,140	0.1
Exposure – Biological process	3,250	<0.1
Exposure – Protein	2,424	<0.1
Disease – Phenotype (negative)	2,386	<0.1
Exposure – Molecular function	90	<0.1
Exposure – Cellular component	20	<0.1
Total number of edges	8,100,498	100.0

Table 4: Statistics on edges in the therapeutics-centered knowledge graph.

Racial group	Count	Percent (%)
Asian	60,041	4.7
Black	162,102	12.7
White	534,305	42.0
Unknown	241,998	19.0
Other	273,639	21.5
Total number of patients	1,272,085	100.0

Table 5: Demographics of the electronic health record dataset at Mount Sinai Health System in New York City used to validate TxGNN's hypotheses on therapeutic use prediction.

References

1. Menche, J. *et al.* Uncovering disease-disease relationships through the incomplete interactome. *Science* **347** (2015).

2. Zitnik, M., Feldman, M. W., Leskovec, J. *et al.* Evolution of resilience in protein interactomes across the tree of life. *Proceedings of the National Academy of Sciences* **116**, 4426–4433 (2019).

3. Ruiz, C., Zitnik, M. & Leskovec, J. Identification of disease treatment mechanisms through the multiscale interactome. *Nature Communications* **12**, 1–15 (2021).

4. Goh, K.-I. *et al.* The human disease network. *Proceedings of the National Academy of Sciences* **104**, 8685–8690 (2007).

5. Barabási, A.-L., Gulbahce, N. & Loscalzo, J. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics* **12**, 56–68 (2011).

6. Li, M. M., Huang, K. & Zitnik, M. Graph representation learning in biomedicine and healthcare. *Nature Biomedical Engineering* 1–17 (2022).

7. Gysi, D. M. *et al.* Network medicine framework for identifying drug-repurposing opportunities for covid-19. *Proceedings of the National Academy of Sciences* **118** (2021).

8. Zitnik, M., Agrawal, M. & Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **34**, i457–i466 (2018).

9. Guney, E., Menche, J., Vidal, M. & Barábasi, A.-L. Network-based in silico drug efficacy screening. *Nature Communications* **7**, 1–13 (2016).

10. Cheng, F., Kovács, I. A. & Barabási, A.-L. Network-based prediction of drug combinations. *Nature Communications* **10**, 1–11 (2019).

11. Guney, E. Reproducible drug repurposing: When similarity does not suffice. In *Pacific Symposium on Biocomputing 2017*, 132–143 (World Scientific, 2017).

12. Chandak, P., Huang, K. & Zitnik, M. Building a knowledge graph to enable precision medicine. *Scientific Data* **10**, 67 (2023).

13. Duran-Frigola, M. *et al.* Extending the small-molecule similarity principle to all levels of biology with the chemical checker. *Nature Biotechnology* **38**, 1087–1096 (2020).

14. Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X. & Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature. *ICLR* (2022).

15. Arnaiz-Rodríguez, A., Begga, A., Escolano, F. & Oliver, N. Diffwire: Inductive graph rewiring via the lov\'asz bound. *LoG* (2022).

16. Tanna, P., Strauss, R. W., Fujinami, K. & Michaelides, M. Stargardt disease: clinical features, molecular genetics, animal models and therapeutic options. *British Journal of Ophthalmology* **101**, 25–30 (2017).

17. Cochat, P. & Rumsby, G. Primary hyperoxaluria. *New England Journal of Medicine* **369**, 649–658 (2013).

18. Schlichtkrull, M. *et al.* Modeling relational data with graph convolutional networks. In *ESWC*, 593–607 (Springer, 2018).

19. Ding, Y., Jiang, X. & Kim, Y. Relational graph convolutional networks for predicting blood–brain barrier penetration of drug molecules. *Bioinformatics* **38**, 2826–2831 (2022).

20. Wang, W., Yang, X., Wu, C. & Yang, C. Cginet: graph convolutional network-based model for identifying chemical-gene interaction in an integrated multi-relational graph. *BMC bioinformatics* **21**, 1–17 (2020).

21. Shu, J., Li, Y., Wang, S., Xi, B. & Ma, J. Disease gene prediction with privileged information and heteroscedastic dropout. *Bioinformatics* **37**, i410–i417 (2021).

22. Bickel, S., Brückner, M. & Scheffer, T. Discriminative learning under covariate shift. *Journal of Machine Learning Research* **10** (2009).

23. Schölkopf, B. *et al.* On causal and anticausal learning. *ICML* 1255–1262 (2012).

24. Niven, T. & Kao, H.-Y. Probing neural network comprehension of natural language arguments. *Proc. 57th Annual Meeting of the Association of Computational Linguistics* 4658–4664 (2019).

25. Zech, J. R. *et al.* Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine* **15**, e1002683 (2018).

26. Geirhos, R. *et al.* Shortcut learning in deep neural networks. *Nature Machine Intelligence* **2**, 665–673 (2020).

27. Schlichtkrull, M. S., De Cao, N. & Titov, I. Interpreting graph neural networks for NLP with differentiable edge masking. *International Conference on Learning Representations* (2021).

28. Agarwal, C., Queen, O., Lakkaraju, H. & Zitnik, M. Evaluating explainability for graph neural networks. *Scientific Data* **10** (2023).

29. Agarwal, C., Zitnik, M. & Lakkaraju, H. Probing GNN explainers: A rigorous theoretical and empirical analysis of gnn explanation methods. In *International Conference on Artificial Intelligence and Statistics*, 8969–8996 (2022).

30. Wang, Q., Huang, K., Chandak, P., Zitnik, M. & Gehlenborg, N. Extending the nested model for user-centric xai: A design study on gnn-based drug repurposing. *IEEE Transactions on Visualization and Computer Graphics* **29**, 1266–1276 (2023).

31. Gaube, S. *et al.* Do as AI say: susceptibility in deployment of clinical decision-aids. *NPJ digital medicine* **4**, 31 (2021).

32. Tschandl, P. *et al.* Human–computer collaboration for skin cancer recognition. *Nature Medicine* **26**, 1229–1234 (2020).

33. Chamberlain, M. C. Salvage therapy with lomustine for temozolomide refractory recurrent anaplastic astrocytoma: a retrospective study. *Journal of Neuro-Oncology* **122**, 329–338 (2015).

34. Seetharaman, J. & Sarma, M. S. Chelation therapy in liver diseases of childhood: Current status and response. *World Journal of Hepatology* **13**, 1552 (2021).

35. Alsentzer, E. *et al.* Deep learning for diagnosing patients with rare genetic diseases. *medRxiv* 2022–12 (2022).

36. O'Connell, D. Neglected diseases. *Nature* **449**, 157–157 (2007).

37. Tambuyzer, E. *et al.* Therapies for rare diseases: therapeutic modalities, progress and challenges ahead. *Nature Reviews Drug Discovery* **19**, 93–111 (2020).

38. Zhang, A., Xing, L., Zou, J. & Wu, J. C. Shifting machine learning for healthcare from development to deployment and from models to data. *Nature Biomedical Engineering* 1–16 (2022).

39. Bastian, F. B. *et al.* The Bgee suite: integrated curated expression atlas and comparative transcriptomics in animals. *Nucleic Acids Research* **49**, D831–D847 (2021).

40. Davis, A. P. *et al.* Comparative Toxicogenomics Database (CTD): update 2021. *Nucleic Acids Research* **49**, D1138–D1143 (2021).

41. Piñero, J. *et al.* The DisGeNET knowledge platform for disease genomics: 2019 update. *Nucleic Acids Research* gkz1021 (2019).

42. Wishart, D. S. *et al.* DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Research* **46**, D1074–D1082 (2018).

43. Avram, S. *et al.* DrugCentral 2021 supports drug discovery and repositioning. *Nucleic Acids Research* **49**, D1160–D1169 (2021).

44. Maglott, D., Ostell, J., Pruitt, K. D. & Tatusova, T. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research* **39**, D52–D57 (2011).

45. Klopfenstein, D. V. *et al.* GOATOOLS: A Python library for Gene Ontology analyses. *Scientific Reports* **8**, 10872 (2018).

46. The Gene Ontology Consortium *et al.* The Gene Ontology resource: enriching a GOld mine. *Nucleic Acids Research* **49**, D325–D334 (2021).

47. Köhler, S. *et al.* The Human Phenotype Ontology in 2017. *Nucleic Acids Research* **45**, D865–D876 (2017).

48. Shefchek, K. A. *et al.* The Monarch Initiative in 2019: an integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Research* **48**, D704–D715 (2020).

49. Weinreich, S., Mangon, R., Sikkens, J., Teeuw, M. E. e. & Cornel, M. [orphanet: a european database for rare diseases]. *Nederlands tijdschrift voor geneeskunde* **152**, 518—519 (2008).

50. Matys, V. *et al.* TRANSFAC ® : transcriptional regulation, from patterns to profiles. *Nucleic Acids Research* **31**, 374–378 (2003).

51. Ceol, A. *et al.* MINT, the molecular interaction database: 2009 update. *Nucleic Acids Research* **38**, D532–D539 (2010).

52. Aranda, B. *et al.* The IntAct molecular interaction database in 2010. *Nucleic Acids Research* **38**, D525–D531 (2010).

53. Giurgiu, M. *et al.* Corum: the comprehensive resource of mammalian protein complexes—2019. *Nucleic acids research* **47**, D559–D563 (2019).

54. Oughtred, R. *et al.* The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions. *Protein Science* **30**, 187–200 (2021).

55. Szklarczyk, D. *et al.* The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Research* **49**, D605–D612 (2021).

56. Luck, K. *et al.* A reference map of the human binary protein interactome. *Nature* **580**, 402–408 (2020).

57. Jassal, B. *et al.* The reactome pathway knowledgebase. *Nucleic Acids Research* gkz1031 (2019).

58. Kuhn, M., Letunic, I., Jensen, L. J. & Bork, P. The SIDER database of drugs and side effects. *Nucleic Acids Research* **44**, D1075–D1079 (2016).

59. Mungall, C. J., Torniai, C., Gkoutos, G. V., Lewis, S. E. & Haendel, M. A. Uberon, an integrative multi-species anatomy ontology. *Genome Biology* **13**, R5 (2012).

60. Alsentzer, E. *et al.* Publicly Available Clinical BERT Embeddings 7.

61. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In *ICML*, 1263–1272 (PMLR, 2017).

62. Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 249–256 (2010).

63. Yang, B., Yih, W.-t., He, X., Gao, J. & Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *ICLR* (2015).

64. Griggs, R. C. *et al.* Clinical research for rare disease: opportunities, challenges, and solutions. *Molecular Genetics and Metabolism* **96**, 20–26 (2009).

65. Boycott, K. M., Vanstone, M. R., Bulman, D. E. & MacKenzie, A. E. Rare-disease genetics in the era of next-generation sequencing: discovery to translation. *Nature Reviews Genetics* **14**, 681–691 (2013).

66. Thomas, S. & Caplan, A. The orphan drug act revisited. *Jama* **321**, 833–834 (2019).

67. Yıldırım, M. A., Goh, K.-I., Cusick, M. E., Barabási, A.-L. & Vidal, M. Drug—target network. *Nature Biotechnology* **25**, 1119–1126 (2007).

68. Agrawal, M., Zitnik, M. & Leskovec, J. Large-scale analysis of disease pathways in the human interactome. In *Proceedings of the Pacific Symposium on Biocomputing*, 111–122 (2018).

69. Kovács, I. A. *et al.* Network-based prediction of protein interactions. *Nature Communications* **10**, 1–8 (2019).

70. Raj, A., Kuceyeski, A. & Weiner, M. A network diffusion model of disease progression in dementia. *Neuron* **73**, 1204–1215 (2012).

71. Lin, Y., Liu, Z., Sun, M., Liu, Y. & Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In *AAAI* (2015).

72. Wang, M. *et al.* Deep graph library: Towards efficient and scalable deep learning on graphs. (2019).

73. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *NeurIPS* **32**, 8026–8037 (2019).

74. McKinney, W. *et al.* pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing* **14**, 1–9 (2011).

75. Harris, C. R. *et al.* Array programming with numpy. *Nature* **585**, 357–362 (2020).

76. Pedregosa, F. *et al.* Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

77. Waskom, M. L. Seaborn: statistical data visualization. *Journal of Open Source Software* **6**, 3021 (2021).

78. Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in science & engineering* **9**, 90–95 (2007).

79. McInnes, L., Healy, J., Saul, N. & Grossberger, L. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software* **3**, 861 (2018).

80. Inc, F. React.js. https://github.com/facebook/react.

81. Bostock, M., Ogievetsky, V. & Heer, J. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics* **17**, 2301–2309 (2011).

82. Team, A. D. Ant design. https://github.com/ant-design/ant-design/.

83. Neo4j, I. Neo4j graph data platform. https://neo4j.com. Accessed: 2020-10-01.

84. Grinberg, M. *Flask web development: developing web applications with python* (" O'Reilly Media, Inc.", 2018).

85. Stang, P. E. *et al.* Advancing the science for active surveillance: rationale and design for the Observational Medical Outcomes Partnership. *Annals of Internal Medicine* **153**, 600–606 (2010).

86. Klann, J. G., Joss, M. A., Embree, K. & Murphy, S. N. Data model harmonization for the All Of Us Research Program: Transforming i2b2 data into the OMOP common data model. *PloS ONE* **14**, e0212463 (2019).

87. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* **17**, 261–272 (2020).

88. Seabold, S. & Perktold, J. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, vol. 57 (2010).