Improving Generalized Zero-shot Learning Using Knowledge Graphs for Multi-label Chest X-ray Classification

Chinmay Prabhakar[1] CHINMAY.PRABHAKAR@UZH.CH
Anjany Sekuboyina[1,2] ANJANY.SEKUBOYINA@UZH.CH
Johannes C. Paetzold[2] JOHANNES.PAETZOLD@TUM.DE
Hongwei Bran Li[1,2] HONGWEI.LI@TUM.DE
Tamaz Amiranashvili[1] TAMAZ.AMIRANASHVILI@UZH.CH
Jens Kleesiek[3] JENS.KLEESIEK@UK-ESSEN.DE
Bjoern Menze[1] BJOERN.MENZE@UZH.CH
[1] *Department of Quantitative Biomedicine, University of Zurich*
[2] *Department of Computer Science, Technical University of Munich*
[3] *Institute for AI in Medicine (IKIM), University Hospital Essen*

Editors: Under Review for MIDL 2022

Abstract

Generalized zero-shot learning (*GZSL*) aims to develop models that can reliably label classes not encountered during training, while maintaining a good performance on the *seen* ones. This becomes especially challenging in the realm of *multi-label* chest X-ray image classification, due to the presence of numerous unknown disease-types and the limited information inherent to x-ray images. In this work, we present a knowledge graph-based approach to *GZSL*. Our method directly injects the semantic relationships between *seen* and *unseen* disease classes by making use of the Unified Medical Language System (*UMLS*). Specifically, we use the *UMLS* as a knowledge base and device a principled approach of parsing and processing it, conditioned on the task at hand. We show that our method matches the labelling performance of the state-of-the-art while outperforming it on *unseen* classes (AUROC **0.68** vs. **0.66**). We also demonstrate that embedding the disease-specific knowledge as a graph provides inherent explainability, which allows us to understand the multi-label relation and model decision. The code is available at https://github.com/chinmay5/ml-cxr-gzsl-kg

1. Introduction

In recent years, deep learning-based computer-aided diagnostic systems (*CAD*) have achieved expert-level performances in some challenging tasks (Rajpurkar et al., 2017; Esteva et al., 2017; De Fauw et al., 2018). However, existing systems typically rely on large-scale annotated datasets, are often single-modal, and are limited to the concepts visible during training. Such a limitation magnifies in the scenario of novel and rare diseases. This is especially the case in multi-label x-ray image classification task where multiple diagnoses (labels) per image exist, and it is infeasible to collect the annotations for every label. Consequently, existing CAD systems are limited by the expressivity of their training annotations and are invariably unable to predict *unseen* diseases. On the other hand, Radiologists do not rely on a single information source and integrate all available information(e.g., medical literature, prior experience, symptomatic correlations, etc.) to recognize such *unseen* diseases.

Zero-shot learning aims to address the issue of annotation scarcity (Zhang et al., 2017; Yu et al., 2018; Changpinyo et al., 2017). The models are trained to classify certain diseases (i.e., *seen* classes) and during inference, they are expected to classify *only* unobserved diseases (*unseen* classes). Generalized zero-shot learning (GZSL) is a more practical setup. It enables the model to classify both *seen* and *unseen* diseases during inference. In other words, the models are expected to perform well at classifying *new* diseases while retaining their performance on the *seen* ones. One critical step to achieve *GZSL* is to incorporate 'clinical knowledge' to model the relation between the *seen* and *unseen* diseases, using available natural language models e.g. *Word2Vec* (Goldberg and Levy, 2014; Zhang et al., 2019), *BERT* (Devlin et al., 2018), or the domain-specific *BioBERT* (Alsentzer et al., 2019).

We suppose that such natural language based methods might not always explicitly encode 'knowledge' (Schick and Schütze, 2020), more so in a clinical setting. As an alternative, we propose to exploit a more explicit knowledge representation called the Unified Medical Language System (*UMLS*) (Bodenreider, 2004). *UMLS* is a relational database of medical knowledge represented as a knowledge graph, consisting of millions of medical entities (or nodes, e.g., diseases, anatomical locations, medicines etc.) and the relations between them. This rich source of curated, medical knowledge can be employed as a critical component to enhance *GZSL*. However, *UMLS* constitutes an ultra large database and lacks efficient ways of parsing and processing, thus making its *ad-hoc* usage challenging in practice.

In this work, we attempt to classify *multi-label* chest x-rays in a *GZSL* setting by incorporating semantic clinical knowledge from the *UMLS* in the form of a graph on the multi-disease labels. Thanks to the universality of *UMLS*, our framework can be readily extended to any diagnosis task on any medical data. Specifically, our contributions are four-fold:

1. We propose a principled approach towards parsing the *UMLS* and using it as a source of semantic information.
2. We utilize the parsed knowledge from *UMLS* for multi-label disease classification in chest x-rays in a *GZSL* setting, improving upon state-of-the-art methods.
3. We validate our approach to two chest x-ray datasets with non-identical disease labels, thus confirming the utility of *UMLS*.
4. Since incorporating semantic knowledge as a graph offers inherent explainability, we explore to use the *GNNExplainer* (Ying et al., 2019) to draw medical intuitions.

2. Related Work

GZSL with knowledge graphs. In the natural image domain, knowledge graphs can effectively bridge the semantic gap between *seen* and *unseen* classes, thus, they are an essential component in *GZSL* (Wang et al., 2018; Zhao et al., 2017; Xian et al., 2017; Li et al., 2020). The graphs are constructed with nodes representing individual classes and edges indicating a semantic relation between these classes. In the medical domain, Chen *et al.* (Chen et al., 2020a) proposed to use label co-occurrences that appeared in the training set to generate a knowledge graph. However, this approach is not applicable in the *GZSL* setting as the unseen co-occurrences are not a part of the graph, and parsing is limited to

specific applications. Zhou *et al.* (Zhou et al., 2021) mine the radiology reports to generate a chest radiology graph. However, the graph uses only the MIMIC-CXR dataset (Johnson et al., 2019) and is tightly coupled to it. Instead, we construct a semantically rich graph by parsing the *UMLS* and extend its applicability to different diagnosis tasks.

Generalized zero-shot learning for multi-label tasks. In the multi-label setting, the *GZSL* aims to classify a given image associated with multiple labels, a setup relatively unexplored in chest radiographs. Paul *et al.* (Paul et al., 2021) propose a trait-guided multi-view semantic embedding strategy but assumes the availability of radiology reports along with the radiographs. Hayat *et al.* (Hayat et al., 2021) propose to create an end-to-end network that jointly learns visual representations from radiographs and aligns them to the semantic features by using *BioBERT* embeddings (Devlin et al., 2018). The method aligns the visual features with their semantic label embeddings. In contrast, we show that the relational clinical information from *UMLS* can be a better embedding than using only *BioBERT* embeddings.

3. Method

3.1. Problem formulation

Consider a multi-label set $\mathcal{Y}$ consisting of C classes. Of these C classes, only S classes are *seen* and U classes are *unseen*. Let $\mathcal{Y}^S$ and $\mathcal{Y}^U$ denote the label sets for the *seen* and *unseen* classes, respectively. Note that $\mathcal{Y}^S \cap \mathcal{Y}^U = \emptyset$ i.e. training images contain onyl *seen* labels. Thus, $\mathcal{Y}^C = \mathcal{Y}^S \cup \mathcal{Y}^U$, where $\mathcal{Y}^S = \{y_1, y_2, ..., y_S\}$ and $\mathcal{Y}^U = \{y_{S+1}, y_{S+2}, ..., y_C\}$. The label vector $y_i \in \{0, 1\}^S$ indicates the presence of every *seen* class. During training in a *GZSL* setting, images containing only the *seen* labels $\mathcal{Y}^S$ are given. During inference, given an image x_{test}, the model is supposed to correctly predict the labels from both *seen* or *unseen* classes, $y_{test} \in \mathcal{Y}^C$.

3.2. Training procedures for GZSL

The training procedure has three stages, as summarized in Figure 1. First, the image processing module is trained using the instances of *seen* classes, to learn the *visual classifier weights*. Second, a Graph processing module (*GPM*), responsible for processing the *UMLS* and generating node features for the disease labels, is aligned to the *visual classifier weights*. Finally, the *GPM* weights replace the *visual classifier weights* and the image-processing module is fine-tuned based on the enriched weights using the labeled data from seen classes.

Image processing module. The module is trained on the labeled 2D radiographs from *seen* classes. *DenseNet121* (Huang et al., 2018) is employed as the backbone and trained to extract visual features. A fully-connected layer with 1024-dimension is used as a classification head, as shown in Step 1 in Figure 1. The *visual classifier weights*, denoted as $W_\phi \in R^{1024 \times C}$, are considered to be the image representation of a radiograph. The j^{th} column of W_ϕ, denoted by w_ϕ^j is the representation of the j^{th} disease learnt from the images. *Note that, only weights of seen classes are semantically rich while the unseen weights are random. They act as placeholders that are replaced by GPM weights in Step 3 (Figure 1).* The *GZSL* task can then be expressed as predicting a new set of weights for each

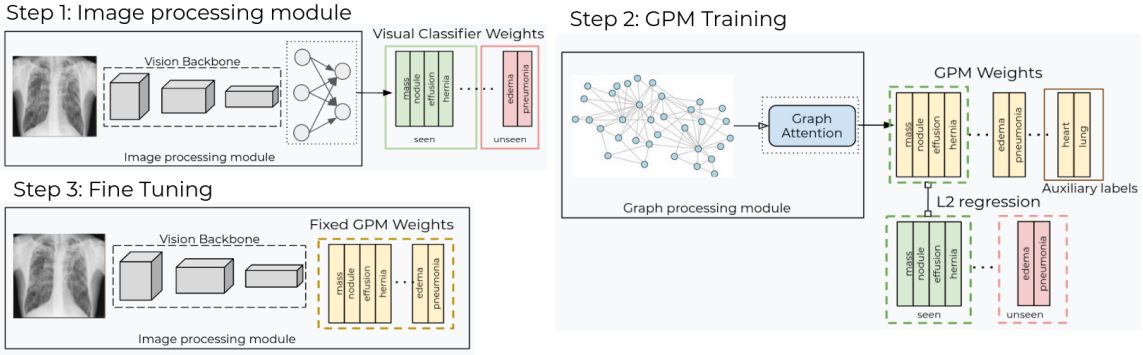

Figure 1: The proposed training pipeline. First, the vision backbone is trained with samples of the *seen* classes. This generates *Visual Classifier Weights* W_ϕ for each of the target labels. In the second step, the Graph Processing Module (*GPM*) is trained using a normalized L2 regression loss (Eq. **??**) between the *Visual Classifier Weights* and weights learned by final layer of *GPM* (referred to as *GPM Weights* W_G) using only the *seen* class' weights. In the final step, the *GPM* weights W_G replace the classification head of the Image processing module. We fix these *GPM* weights and fine tune the image processing module .

of the unseen classes to extend the output of the backbone. The image processing module is trained using weighted multi-label classification loss $\mathcal{L}_{cls}$(Eq. 1) (Chen et al., 2020a) which re-weights the positive and negative samples in the mini-batches to handle potential data-imbalance issues.

$$\mathcal{L}_{cls} = -\omega_p \sum_{l_i=1} log(\sigma(p_i)) - \omega_n \sum_{l_i=0} log(1 - \sigma(p_i)) \tag{1}$$

where p_i is the model logit, l_i is the corresponding label, $|P|$ and $|N|$ are the total number of positive and negative samples per mini-batch. Thus, $\omega_p = \frac{|P|+|N|+1}{|P|+1}$ and $\omega_n = \frac{|P|+|N|+1}{|N|+1}$ are the balancing factors to handle data imbalance.

Graph construction. We use the *UMLS* to obtain semantic clinical information to enhance *GZSL*. However, a naive parsing of the entire *UMLS* is neither feasible nor beneficial owing to its large database size and superfluous information. Thus, we parse only the *relevant* part for our specific task, resulting in a **subgraph** of *UMLS*.

Figure 2 summarizes the three steps to parse this subgraph. First, we extract the entities (nodes) corresponding to the label set C. Starting from each of these entities, we extract its *5-hop* neighbourhood, resulting in a first noisy *UMLS subgraph*. Please refer to the appendix for details about parsing a *k-hop* neighbourhood. This subgraph is then trimmed using *all-pair-shortest path* of the label set. The parsing is restricted to *UMLS* entries which are in English. Additionally, we only include relationships that indicate either **a label hierarchy** or **an anatomical dependence**, viz. `inverse_isa`, `finding_site_of`, `part_of`, `is_associated_anatomic_site_of`, and `has_member`. Once the *UMLS-subgraph* is ready, all the nodes that now include the nodes corresponding to the label-set and the nodes that lie on the all-pairs shortest path (referred to as *auxiliary nodes*) are initialized with *BioBERT* embeddings (Alsentzer et al., 2019) creating G_{UMLS}.

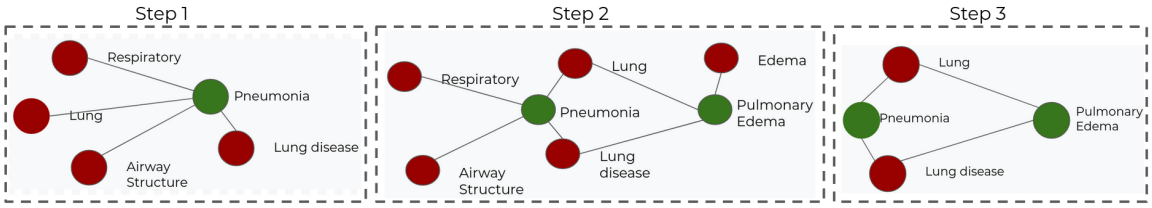

Figure 2: Parse logic of *UMLS* for a 1-hop neighborhood. The *label-set* (in green) act as seed points. In the first step, an element of the label-set is chosen at random, and its directly connected relations are extracted from the *UMLS*. This might produce entities not part of the label-set (in red). Next, the same process is repeated for the remaining entries of the *label-set*. In the final step, we prune the resulting graph by retaining the nodes and edges that are part of *All Pair Shortest Path* with respect to the *label-set*.

Graph processing module (GPM). The GPM aims to enrich features of G_{UMLS}. Assume w_G^j denotes the representation of the j^{th} node in G_{UMLS} This representation is enriched using (GATv2 (Brody et al., 2021)) layers in thethe following layout: $w_G^j \rightarrow$ **GATv2** $\rightarrow$ **LeakyRELU** $\rightarrow$ **GATv2** $\rightarrow$ **LeakyRELU** $\rightarrow$ **GATv2** $\rightarrow \hat{w}_G^j \in R^{1024}$. We now extract enriched representations corresponding to the C classes in our dataset, resulting in $W_G \in R^{1024 \times C}$. These disease representations based on the graph are referred to as the *GPM weights*. Next, we align the weights of the seen classes between the *GPM* and the *visual classifier weights*, i.e.

$$\mathcal{L}_{reg} = \sum_{j \in seen} ||w_\phi^j - \hat{w}_G^j||^2 \tag{2}$$

Now, owing to message passing in the graph convolution layers, the node features of *unseen* classes are also enriched. Once trained, the *GPM weights* are semantically richer for *unseen* classes compared to *visual classifier weights*.

Fine tuning. The final step involves replacing the classification head W_ϕ with the *GPM* weights W_G and fine-tuning the vision backbone using the labeled data from *seen* classes. Since W_ϕ and W_G are **not** identical for the *seen* classes, the step is essential to ensure no performance degradation. Hence, the semantic gap between the *seen* and *unseen* classes is bridged while not compromising on the knowledge about *seen* classes.

4. Experiment Setup

We evaluate our method on two public chest X-ray datasets: a) The NIH Chest X-ray dataset (Wang et al., 2017), and b) The Indiana Univ Chest X-ray dataset (Shin et al., 2016). Radiographs with multi-label annotations are provided for both datasets.

NIH Chest X-ray. 112,120 frontal X-ray images are split into training (70%), validation (10%) and test sets (20%). Each image is associated with 14 class labels. We use *Atelectasis, Effusion, Infiltration, Mass, Nodule, Pneumothorax, Consolidation, Cardiomegaly, Pleural Thickening, and Hernia* as the *seen* classes while *Edema, Pneumonia, Emphysema, and Fibrosis* are the *unseen* classes (same as (Hayat et al., 2021)), resulting in 30,758 training images, 4,474 validation images and 10,510 test images.

	k=2			k=3			AUROC		
Method	**p@k**	**r@k**	**f1@k**	**p@k**	**r@k**	**f1@k**	**S**	**U**	**HM**
NIH Chest X-ray									
CNN	0.28	0.34	0.30	0.23	0.43	0.29	**0.80**	0.52	0.63
CXR-ML-GZSL	0.33	**0.36**	0.32	0.28	**0.47**	0.34	**0.79**	0.66	0.72
Ours	**0.38**	0.33	**0.35**	**0.31**	0.43	**0.36**	0.79 ±0.001	**0.68** ± 0.002	**0.73** ± 0.001
Indiana University Chest X-ray									
CNN	0.23	0.25	0.24	0.27	0.34	0.30	**0.69**	0.78	0.73
CXR-ML-GZSL	**0.33**	0.26	**0.29**	0.27	0.35	0.31	0.683	0.79	0.73
Ours	0.28	**0.28**	0.28	**0.28**	**0.36**	**0.32**	0.68 ±0.001	**0.80** ±0.002	0.74 ±0.001

Table 1: Performance Evaluation on the NIH Chest X-ray and Indiana University Chest X-ray dataset. We report the results using Precision@k, Recall@k , F1@k for k $\in \{2, 3\}$. We also report AUROC for *seen (S) & unseen (U) classes and the Harmonic Mean (HM)*. CXR-ML-GZSL refers to (Hayat et al., 2021) and *CNN* is DenseNet121 trained on only the seen classes. We report the mean and standard deviation value across five runs of the model. Please refer to the appendix for more details.

Indiana University Chest X-ray. We used a similar setup as the NIH dataset. We split the frontal X-ray images into training (70%), validation (10%) and test sets (20%). Each image is associated with 17 class labels. We use *Cardiomegaly, Scoliosis, Effusion, Thickening, Pneumothorax,Hernia, Calcinosis, Atelectasis, Cicatrix, Opacity, Lesion,Airspace disease, and Hypoinflation* as the *seen* classes while *Edema, Pneumonia, Emphysema, Fibrosis* are the *unseen* classes, resulting in 1014, 145, and 408 for training, validation, and test sets respectively.

Evaluation metrics. We report overall precision, recall, and f1 scores for the top k predictions (where $k \in 2, 3$) and the average area under the receiving operating characteristic curve (AUROC) for *seen* and *unseen* classes and their harmonic mean.

4.1. Results

Comparison with state-of-the-art. We summarize our results in Table 1. Our model performs better than the baseline for *unseen* classes while performing comparably on the *seen* classes. Since our proposed solution relies on a universal knowledge graph (*UMLS*) and is not tightly coupled with the dataset we operate on, the extension of our method to different datasets with different numbers of target labels is almost trivial. Verifying this, we evaluate the baseline and our proposed method on the Indiana University Chest X-ray dataset. Note that another UMLS sub-graph has to be created as the label set changes. The remaining modules, however, remain unchanged. Observe the improvement over baseline performance, showcasing our methodś extensibility with minimal changes.

Ablation Study. To hightlight our contributions and evaluate different components, we run the ablation experiment on the NIH Chest X-ray dataset. Please refer to Appendix G for discussions about the *CNN* baseline method.

BioBERT embeddings vs. knowledge graph. It is known that *BioBERT* embeddings are semantically rich in text representation. However, they might not sufficiently capture clinical relation information in the *GZSL* setting. We ran an experiment using the *BioBERT* embeddings but without the graph structure. The nodes are initialized with *BioBERT* em-

beddings and passed through several fully connected layers, processing nodes independently without any inter-node interaction. The semantic richness ensures decent performance on the *unseen* classes (AUROC 0.60), obtaining a HM of 0.68 overall. However, the performance is still considerably worse than our proposed graph for *unseen* classes (**0.60** vs. **0.68**), indicating that the *BioBERT* embeddings are insufficient to bridge the semantic gap.

Learned graph vs. random graph. To analyze the importance of graph structure, we replace the *UMLS* graph with different random graphs (Stochastic Block Model, Planted Partition Model and Erdos Renyi random graph model (Newman et al., 2002)). As can be seen in Table 2, all random graph models perform worse than the *BioBERT* embedding model. We attribute this to an incoherent graph structure in random graphs, leading to a *negative knowledge transfer* between the nodes. The decrease in performance is especially steep in the case of *unseen* classes. This is expected since the learned graph structure passes essential semantic knowledge to classify *unseen* diseases and it indicates that the graph structure is critical for the overall performance.

The importance of node embeddings. To evaluate the importance of node embeddings, we initialize the G_{UMLS} nodes using BioWord2Vec embeddings (Zhang et al., 2019), instead of *BioBERT* embeddings. On average, the model performs better than the independently processed *BioBERT* embeddings. Still, the performance is much worse compared to the proposed solution (**0.68** vs. **0.61** for *unseen* classes). These experiments corroborate the importance of graph structure and strong feature representation for the node embeddings. Hence, the proposed solution uses UMLS graph structure and *BioBERT* embeddings.

The effect of the depth of GAT layers. The GPM module uses GATv2 convolutions to process node embeddings. We experimented with a different number of convolution layers, and results are shown in Figure 3. As we can observe, the AUROC value is maximum when using three GATv2 layers with an HM of 0.73. From Table 3 in the appendix, we can see that the maximum distance between any two target nodes is four. Hence, with 3 layers, neighbourhood aggregation covers the entire graph and additional layers lead to performance degradation due to the over-smoothing effect (Chen et al., 2020b).

Model interpretability. Deep learning models are often accused of being "black-boxes" and lacking inherent interpretibility. However, with methods such as Grad-CAM (Selvaraju et al., 2017) and GNNExplainer (Ying et al., 2019), we can visualize what regions of the input a model focuses on, to make its decisions. Thus, the decision making process of the network is made more transparent. We discuss Grad-CAM in this section and refer the readers to the Appendix F for experiments on GNNExplainer.

Figure 4 shows some of the visualizations obtained using Grad-CAM on samples containing *unseen* classes in the test set. As we can see, our model focuses on radiograph regions most likely responsible for the diseases. We have included more visualizations in Appendix F.

5. Conclusion

We propose a promising solution for parsing, storing and processing the comprehensive *UMLS* knowledge graph to improve *GZSL*. We also show that our method can be easily extended to multiple datasets with minimal effort. We find that *UMLS* provides a very

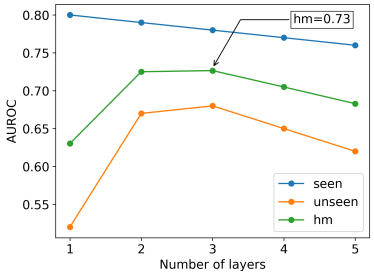

Figure 3: Plots of AUROC values vs. the number of GATv2 layers in the Graph Processing Module (GPM). The Harmonic Mean (HM) of AUROC for *seen & unseen* classes tends to increase first reaching a maximum value of 0.73 for three GATv2 layers and then decreases.

	AUROC		
Method	**S**	**U**	**HM**
CNN	0.80	0.52	0.63
BERT	0.78	0.60	0.68
Random Graph -ER + BERT	0.77	0.58	0.66
Random Graph -SBM + BERT	0.78	0.57	0.66
Random Graph -PAM + BERT	0.77	0.51	0.61
UMLS + Word2Vec	0.78	0.61	0.69
UMLS + BERT	**0.79**	**0.68**	**0.73**

Table 2: Ablation study. The **CNN** model is trained only based on the *seen* classes. **BERT** model used *BioBERT* embeddings for the nodes but assumes no graph structure. **Random Graph + BERT + *** uses a graph created from random graph generation algorithms and uses *BioBERT* embeddings for its nodes. **UMLS + Word2Vec** uses G_{UMLS} but initializes the node embeddings using Bio-Word2Vec. **UMLS + BERT** uses the G_{UMLS} (UMLS parsing + BioBERT node embeddings).

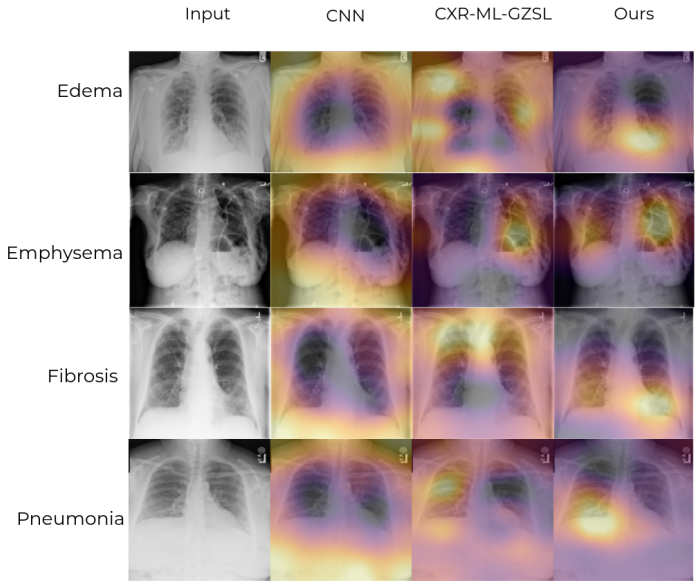

Figure 4: Saliency map visualization for the *unseen* classes. Each row contains one of the unseen diseases and the Grad-CAM output of the three models. We have included the original input image in the first column for reference.The model focuses on regions that are relevant for diagnosis of the individual diseases.

rich source of semantic information that can be used for *GZSL* applications. One limitation of this work is that we have only used the *structural* information from the *UMLS* and considered it as a homogeneous graph. As such, we had to hand-pick relations that we deemed helpful in our scenarios. In the future, we aim to treat *UMLS* as a heterogeneous graph (i.e., treating different relations independently), thereby removing the relationship selection step.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-1909. URL https://www.aclweb.org/anthology/W19-1909.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks?, 2021.

Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical image analysis*, 66:101797, 2020.

Soravit Changpinyo, Wei-Lun Chao, and Fei Sha. Predicting visual exemplars of unseen classes for zero-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Bingzhi Chen, Jinxing Li, Guangming Lu, Hongbing Yu, and David Zhang. Label co-occurrence learning with graph convolutional networks for multi-label chest x-ray image classification. *IEEE journal of biomedical and health informatics*, 24(8):2292–2302, 2020a.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3438–3445, 2020b.

Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

Ronald L Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43–57, 1985.

Nasir Hayat, Hazem Lashen, and Farah E. Shamout. Multi-label generalized zero shot learning for the classification of disease in chest radiographs, 2021.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

Aoxue Li, Zhiwu Lu, Jiechao Guan, Tao Xiang, Liwei Wang, and Ji-Rong Wen. Transferrable feature and projection learning with class hierarchy for zero-shot learning. *International Journal of Computer Vision*, 128:2810–2827, 2020.

Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the national academy of sciences*, 99(suppl 1):2566–2572, 2002.

Angshuman Paul, Thomas C Shen, Sungwon Lee, Niranjan Balachandar, Yifan Peng, Zhiyong Lu, and Ronald M Summers. Generalized zero-shot chest x-ray diagnosis through trait-guided multi-view semantic embedding with self-training. *IEEE Transactions on Medical Imaging*, 2021.

Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

Timo Schick and Hinrich Schütze. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774, 2020.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

Hoo-Chang Shin, Kirk Roberts, Le Lu, Dina Demner-Fushman, Jianhua Yao, and Ronald M Summers. Learning to read chest x-rays: Recurrent neural cascade model for automated image annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2497–2506, 2016.

Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. *Advances in Neural Information Processing Systems*, 34, 2021.

Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnn explainer: A tool for post-hoc explanation of graph neural networks. *arXiv preprint arXiv:1903.03894*, 2019.

Yunlong Yu, Zhong Ji, Jichang Guo, and Zhongfei Zhang. Zero-shot learning via latent space encoding. *IEEE transactions on cybernetics*, 49(10):3755–3766, 2018.

Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2021–2030, 2017.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific data*, 6(1):1–9, 2019.

Bo Zhao, Xinwei Sun, Yuan Yao, and Yizhou Wang. Zero-shot learning via shared-reconstruction-graph pursuit. *arXiv preprint arXiv:1711.07302*, 2017.

Yi Zhou, Tianfei Zhou, Tao Zhou, Huazhu Fu, Jiacheng Liu, and Ling Shao. Contrast-attentive thoracic disease recognition with dual-weighting graph reasoning. *IEEE Transactions on Medical Imaging*, 40(4):1196–1206, 2021.

Appendix A. Training Details

The training of our model happens in three steps. In the first step, the vision backbone is trained using Adam optimizer with a learning rate of 1e-3 and weight_decay of 1e-5 for a total of 40 epochs. The learning rate is decreased by a factor of 10 when the validation loss does not decrease for three epochs. The vision backbone is trained for 40 epochs.

Next we train the GPM. It is composed of three Gatv2 layers. The first GATv2 layer has $in\_channels = 1024, out\_channels = 768$ and next GATv2 layer uses $in\_channels =$

768, *out_channels* = 768 a final GATv2 uses *in_channels* = 768, *out_channels* = 1024. The *GPM* uses leaky_relu non-linearity with *negative_slope* = 0.2.

The *GPM* module is trained to align W_ϕ and W_G. The objective is to align the weights of $W_G \in R^{1024\times C}$ with the weights of $W_\phi \in R^{1024\times C}$ using a normalized L2 regression loss (Eq. **??**). For training, we use Adam optimizer with learning rate of 1e-3 and weight_decay of 5e-4 and run for 1000 iterations.

The Fine-tuning step for the NIH Chest X-ray dataset uses an Adam optimizer with a learning rate of 1e-4 and weight decay of 1e-3, run for a total of 40 epochs. The learning rate is reduced by 10 when the validation loss does not decrease for 3 epochs. In the case of the Indiana University Chest X-ray dataset, the setup is similar, but it uses a weight decay of 1e-5 and is run for 50 epochs. The experiments are conducted using the PyTorch Geometric libray (Fey and Lenssen, 2019) on a NVIDIA GeForce RTX 3090 machine.

Data pre-processing. The Indiana University Chest X-ray dataset was preprocessed by clipping top and bottom 0.5% DICOM pixel values, scaling pixel values linearly to fit in a range of 0-255 and resizing images to 2048 on the shorter side. For both (NIH-Chest X-ray and Indiana University Chest X-ray) datasets, we used only the frontal images.

Training procedure. We ran 5 instances of the model in the same setup. We compute the mean and standard deviation of the results and report it along with the numbers in Table 1.

Random Graph Generation. The graph generated using *Erdos-Renyi model* uses an *edge_probability* of 0.2. The *Stochastic Block model* uses *block-size* of $\frac{1}{num\_classes}$ and an *edge_probability* of 0.2. For the *Planted Partition Model* we use the Barabasi- Albert-Graph generation (Barabási and Albert, 1999). In all the cases, the number of nodes is the same as G_{UMLS}.

Model behaviour on Indiana University Chest X-ray dataset. We observe a high AUROC of the *unseen* classes for the CNN baseline. Our motivation for using this dataset was to show the easy extensibility of our proposed method. However, due to the strict rule of selecting only "Seen" classes during training and removing all the instances containing any of the "Unseen" classes, coupled with the small dataset size, we ended up with a very skewed dataset. The final test set consisted of only 74 samples with unseen classes. Partly because of the small size of the test set, our method only slightly outperform the CNN baseline on the *unseen* classes (0.80 vs. 0.79). In summary, the dataset is not expressive enough to warrant large statistical performance differences; thereby, we use the NIH Chest X-ray dataset to conclude the model behavior in our ablation study.

Appendix B. Graph Exploration

Here we summarize some of the most essential properties of our parsed graph structure.

B.1. K-hop neighbourhood generation

Our parsing method can be extended to parse the k-hop neighbourhood of the graph. Figure 5 shows an example of parsing the 2-hop neighbourhood. We start with elements from the label set (*Pneumonia* and *Cardiomegaly* shown in the figure). We keep track of the entities

directly connected to these nodes forming our 1-hop neighbourhood. In the next step, we find the directly connected neighbours of the nodes obtained at end of step-1 excluding the relations already parsed. This is our 2-hop neighbourhood. We can repeat the process k times to get the k-hop neighbourhood. Finally, we would compute the *all pair shortest path* between the *label-set* elements and remove the nodes that are not part of the shortest paths.

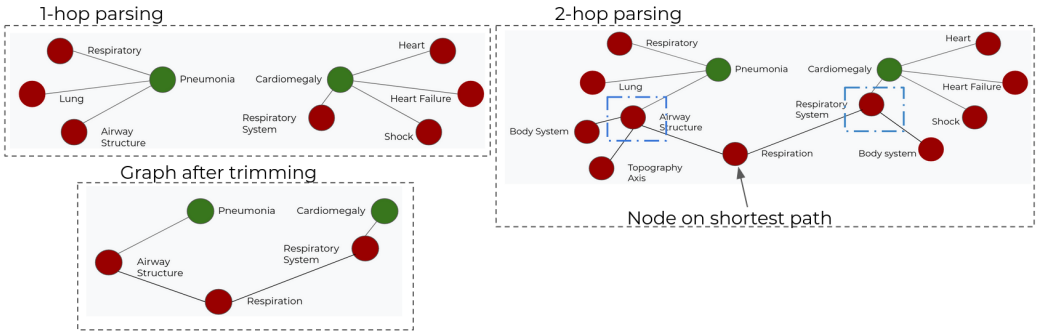

Figure 5: The k-hop parsing process for the *UMLS*. We start with the relations directly connected to elements of the *label-set*. These relations form the *1-hop* neighbourhood. In the next step, the same process is repeated on the nodes generated in step-1. This creates the *2-hop* neighbourhood. We can repeat the step to get *k-hop* neighbourhood. The final pruning will ensure only nodes in the *all-pair-shortest-path* are retained.

B.2. Graph Visualization

We visualize the parsed graph using a *spring layout* in Figure 6. The *label-set* nodes are annotated with green color. The auxiliary nodes are shown in red.

B.3. Shortest Distance between the nodes

Table 3 summarizes the pair-wise distance between all the target labels for the NIH Chest X-ray dataset. As we can see, there are **no self-loops** in the graph and the maximum distance between two target labels is **4**. Hence, the *GPM* should produce the best result for 3 conv layers. We observe the same in Figure 3.

B.4. Graph Properties

Table 4 summarizes a few of the important properties of the parsed graph. The graph **does not have** any isolated nodes and **does not contain** self-loops. The graph is parsed to be **directed** and has ∞ **diameter**. An ∞ diameter indicates that not all nodes can be reached from each other. We also plot the node degree distribution for the graph in Figure B.4. The plot shows our graph contains certain highly connected hub nodes, and the degree distribution almost follows a linear plot on the log-log scale.

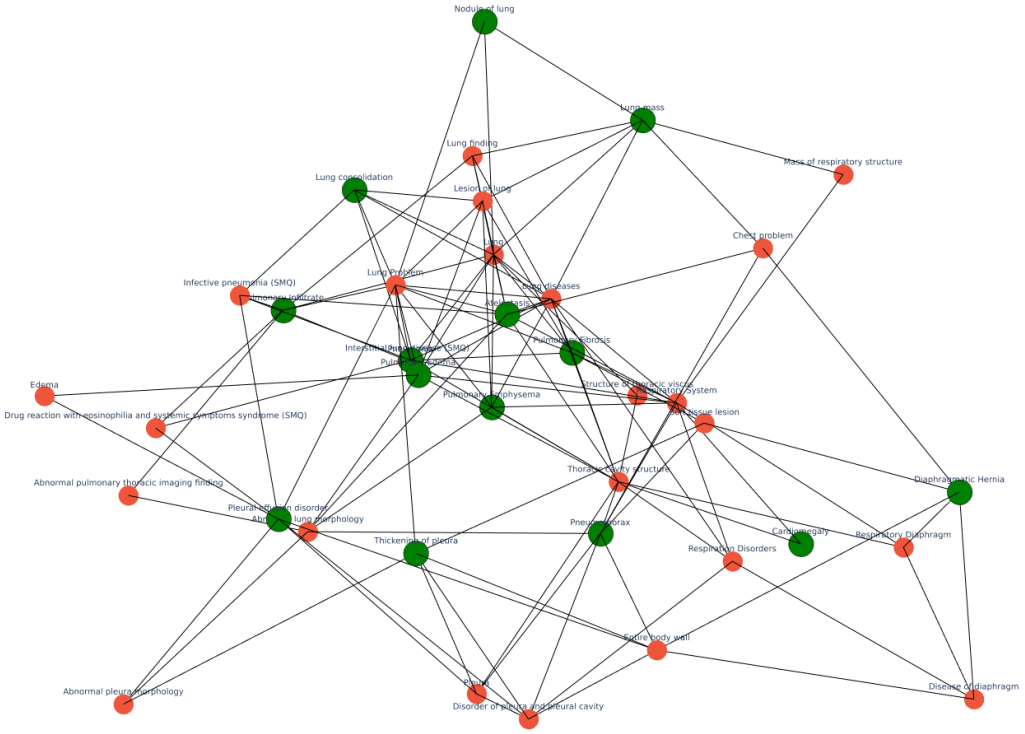

Figure 6: A visualization of the parsed graph. We have used the *spring layout* to plot the graph. The nodes colored in green are the target labels for the NIH Chest X-ray dataset, while those colored in red are the extra labels obtained by parsing the *UMLS*.

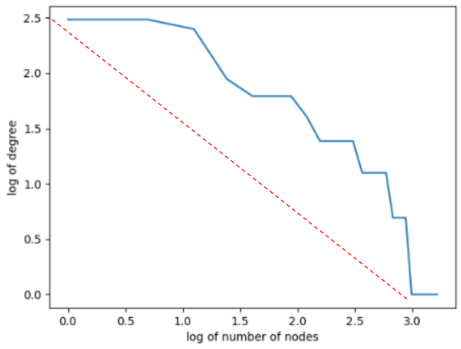

Figure 7: Plot of log of node degree vs. the number of nodes in the parsed graph. Similar to graphs prevalent in nature, there are few nodes in the graph that have a high node degree (*hub nodes*). At the same time, the connectivity gradually decreases, reaching a minimum of 0, indicating that the node is connected to only one other node (please remember this is a graph on the log-log scale. The red-dashed line represents a perfect linear relation on the log-log scale).

Appendix C. Class-wise AUROC comparison

Table 5 shows the per-class AUROC value for the test-set. As we can see, our method tends to perform better for the *unseen* classes and is quite close to the baseline for the samples from *seen* classes.

Node	**Node**													
	A	C	PED	PI	LM	N	Pn	LC	Pt	PEdm	PEpy	PF	T	D
A	0	2	2	2	2	2	2	2	2	2	2	2	2	2
C	2	0	3	**4**	3	**4**	2	3	2	2	2	2	3	3
PED	2	3	0	2	2	2	2	2	2	2	2	2	2	2
PI	2	**4**	2	0	2	2	2	2	3	2	2	2	2	**4**
LM	2	3	3	2	0	1	2	2	2	2	2	2	3	2
N	2	**4**	2	2	1	0	2	2	3	2	2	2	2	3
Pn	2	2	2	2	2	2	0	1	2	2	2	2	2	3
LC	2	3	2	2	2	2	1	0	3	2	2	2	2	**4**
Pt	2	2	2	3	2	3	2	3	0	2	2	2	2	2
PEdm	2	2	2	2	2	2	2	2	2	0	2	2	2	3
PEpy	2	2	2	2	2	2	2	2	2	2	0	2	2	3
PF	2	2	2	2	2	2	2	2	2	2	2	0	2	2
T	2	3	2	2	3	2	2	2	2	2	2	2	0	2
D	2	3	2	**4**	2	3	3	**4**	2	3	3	2	2	0

Table 3: All pair shortest path between the target label nodes for NIH Chest X-ray dataset. **A** represents Atelectasis, **C** represents Cardiomegaly, **PED** represents Pleural Effusion Disorder, **PI** represents Pulmonary Infiltrate, **LM** represents Lung Mass, **N** represents Nodule of lung, **Pn** represents Pneumonia, **LC** represents Lung Consolidation, **Pt** represents Pneumothorax, **PEdm** represents Pulmonary Edema, **PEpy** represents Pulmonary Emphysema, **PF** represents Pulmonary Fibrosis, **T** represents Thickening of pleura, **D** represents Diaphragmatic Hernia. As we can see, the maximum distance between the target label nodes is **4** and thus, using 4 convolution layers would lead to an oversmoothing effect for the *label-set nodes*.

Property	Value
Has Isolated Node?	**No**
Has Self-loops?	**No**
Is Directed?	**Yes**
Diameter	∞

Table 4: Properties of the parsed graph. The graph is *directed*, has *no self-loops*, *does not contain isolated nodes* and has ∞ diameter.

Method	Atelectasis	Cardiomegaly	Pleural Effusion Disorder	Pulmonary Infiltrate	Lung Mass	Nodule of lung	Pneumothorax	Lung Consolidation	Thickening of pleura	Diaphragmatic Hernia	**Pneumonia**	**Pulmonary Edema**	**Pulmonary Emphysema**	**Pulmonary Fibrosis**
CNN	0.77	**0.91**	0.83	**0.71**	0.80	0.77	0.84	0.72	**0.74**	**0.96**	0.51	0.51	0.45	**0.60**
CXR-ML-ZSL	0.76	0.90	0.83	0.70	0.80	0.75	0.83	0.69	0.72	0.90	0.62	0.67	0.74	**0.60**
Ours	**0.79**	0.90	0.83	**0.71**	**0.82**	**0.79**	**0.85**	**0.73**	0.67	0.81	**0.66**	**0.70**	**0.80**	0.58

Table 5: The Class-wise AUROC comparison across all disease classes in the test set. As we can see, our method tends to obtain the best results for the *unseen* classes (marked in bold) while being comparable to the *seen* classes.

Disease	**Nearest Neighbour (*BioBERT*)**	**Nearest Neighbour (GPM)**
Atelectasis	Lung Problem	Pneumonia
Cardiomegaly	Chest problem	Diaphragmatic Hernia
Pleural effusion	Pleural Diseases	Thickening of pleura
Pulmonary Infiltrate	Lower respiratory tract structure	Pneumonia
Lung mass	Lung diseases	Abnormal pleura morphology
Nodule of lung	Lesion of lung	Thickening of pleura
Pneumonia	Lung Problem	Pulmonary Edema
Pneumothorax	Pulmonary Emphysema	Pulmonary Emphysema
Lung consolidation	Lung diseases	Interstitial lung disease (SMQ)
Pulmonary Edema	Lung Problem	Pneumonia
Pulmonary Emphysema	Pulmonary Fibrosis	Diaphragmatic Hernia
Pulmonary Fibrosis	Pulmonary Emphysema	Diaphragmatic Hernia
Thickening of pleura	Disorder of pleura and pleural cavity	Pulmonary Edema
Diaphragmatic Hernia	Respiratory Diaphragm	Pulmonary Emphysema

Table 6: Comparing the 1-nearest neighbours in the embedding space for *BioBERT* vs. *GPM* feature space embeddings. While *BioBERT*'s embedding space is valid but generic, the *GPM* feature space is aligned to learn the relationship between different diseases based on the *UMLS* structure.

Appendix D. Feature Space Lookup

Nearest Neighbour lookup in the feature space is an efficient way to decipher the predictions made by a Deep Learning model. In Table 6 we explore the feature space of original *BioBERT* embeddings and the embeddings produced by *GPM*. We use an L2 distance-based 1-Nearest Neighbour (*NN*) lookup. The *BioBERT* feature space has a lot of semantic information, but it does inherently know the relationship between different diseases. For instance, in its embedding space, *NN* of *Pleural Effusion* is *Pleural disease*. Although this is valid but the information does not include relations between these diseases. The *GPM*, on the other hand, brings *Thickening of Pleura* closer to *Pleural Effusion* in the embedding space, thereby explicitly learning a relationship between the two. This demonstrates that a feature space with rich semantic features and efficacious encoding between diseases is learned by our model.

Appendix E. Discussions On Potential Improvements

Heterogeneous Graph The current work treats the generated graph as a homogeneous one. Specifically, the different "kinds" of nodes are not distinguished. We plan to treat the graph as a heterogeneous one in our future work. For instance, the graph node *pneumonia*, the nodes *lung*, *chest*, *breathing_problem* and *azithromycin* are directly connected. We expect this would bring increment performance gain and better model expressivity. By realizing that *azithromycin* is a medicine for the disease while *lung* is an organ and *breathing_problem* a symptom, the model can more smoothly draw correlations to other diseases that are treated using the same medicine. The model can recognize if the neighbouring nodes are diseases or medicines and thus, treat the nodes differently by learning different weights and transformations for different kinds of nodes and edges.

Pruning Techniques We provide a principled approach to prune the knowledge graph to ensure the model has a good inductive bias. We used the *All-Source-Shortest Path* algorithm for the pruning. There are alternative, equally valid, approaches of pruning such as *Longest Path*, *Minimum Spanning Tree* algorithm etc (Graham and Hell, 1985). We plan to explore other pruning approaches and evaluate if these alternative approaches lead to performance gain.

Open World Hypothesis States that a knowledge graph is never "complete". The same holds for UMLS as well. We have a two-fold approach towards this incompleteness: (1) we plan to research into 'what' relationships to parse so that the missing relationships can be learnt using the transitive property of a knowledge graph. e.g: a relation '(A,is_son_of,B)' can be learnt from '(A,is_brother_of,C)' and '(C,is_daughter_of,B)', provided we have enough triples of each kind. (2) UMLS is continuously being updated. Since our approach is not limited to any particular relase of the UMLS, it can seamlessly adapted to the updated versions.

Calibration In our current work we have used the GNNExplainer (Ying et al., 2019) framework to provide model interpretability. The GNNExplainer generates a small subgraph centred around each of the target nodes and gives us an idea about the importance of these connected nodes for the inference. Using confidence calibration (Wang et al., 2021) to build trust into model predictions is another line of work that can provide model interpretability.

Appendix F. Model interpretability

GNNExplainer Since a graph provides inherent explainability, we determine what nodes and edges in the graph are considered relevant for predictions using the *GNNExplainer* (Ying et al., 2019) framework. The GNNExplainer would produce a subgraph G_S by pruning some of the nodes of the original graph. X_S^F are the node features of the resulting subgraph. We compute the mean square error between the original GPM node features x^{jd} and the resulting subgraph node features $\tilde{x_S}^{jd}$ (referred to as the *feature_regression_loss*). We define $H(Y|G = G_S, X = X_S^F)$ as the entropy of the subgraph. It encodes how much information is "lost" by removing the nodes (& their associated features) from the original graph. We aim to find such nodes that can be removed with minimal change in the expressivity of the model. Conversely, these nodes play a minimal role in the model decision and hence, for understanding the model behavior, we should not focus on them (Ying et al., 2019). Removing such nodes would lead to minimal changes to the entropy & the *feature_regression_loss*. Thus, to select only the consequential nodes in the graph, we optimize

$$\mathcal{L}_{exp} = \lambda \cdot \frac{1}{D} \sum_{j} \sum_{D} (x^{jd} - \tilde{x_S}^{jd})^2 + H(Y|G = G_S, X = X_S^F) \tag{3}$$

We empirically set λ to 10^3 to ensure that all loss terms are approximately of the same scale. Figure 8 visualizes some nodes in the *UMLS* graph. We observe that *seen-class* node connectivity is aggressively pruned. In contrast, *unseen-class* nodes and their connectivity are mostly conserved, thereby expressing a reliance on the graph structure for learning features of the *unseen* classes.

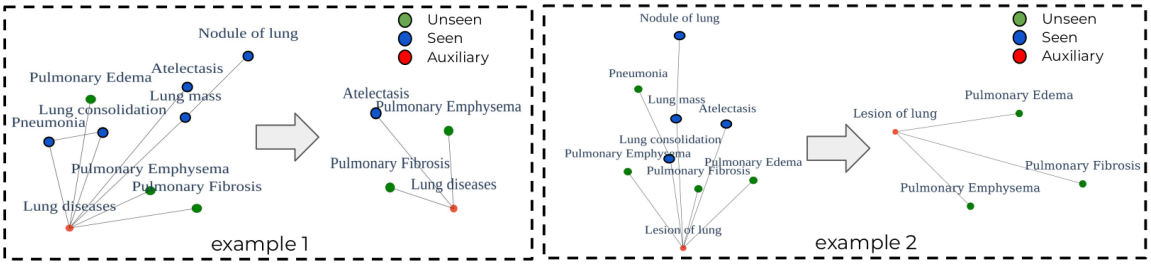

Figure 8: Visualization of two example subgraphs. Nodes colored in blue and green are the target seen and unseen labels for the NIH Chest X-ray dataset, while the nodes in red represent the extra labels obtained by parsing the *UMLS*. The *seen* class nodes are more aggressively pruned compared to the *unseen* class nodes showing a reliance on graph structure for semantic information.

Grad-CAM We include more Grad-CAM visualization from our model. We plot scenarios where our method performs well 9 and where our model struggles 10

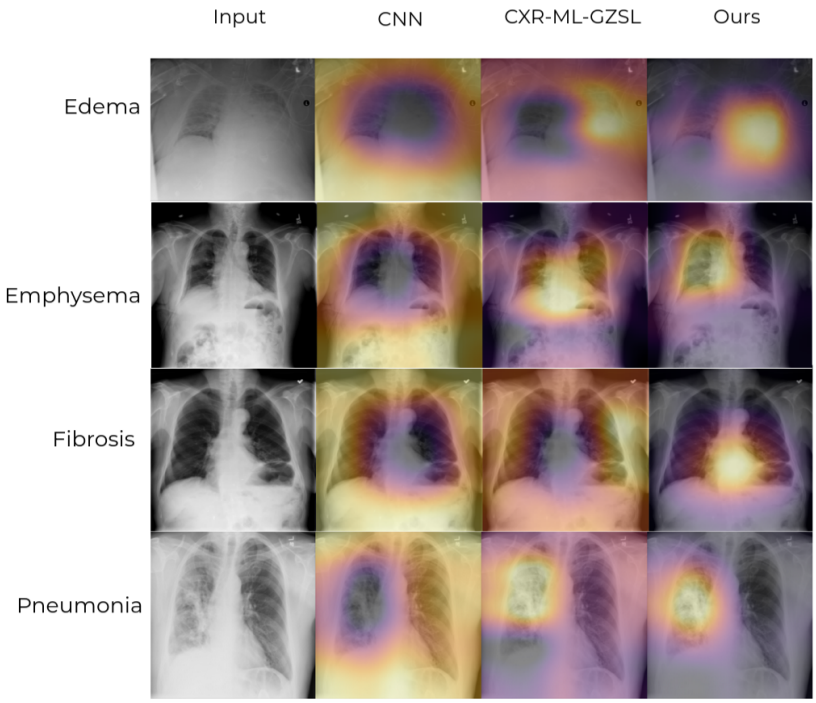

Figure 9: Saliency map visualization for the *unseen* classes. Each row contains one of the unseen diseases and the Grad-CAM output of the three models. These are the samples where our model performs **decently**. We have included the original input image in the first column for reference.The model focuses on regions that are relevant for diagnosis of the individual diseases.

Appendix G. Further Experiments

Parsing the PadChest graph In our experiments, we used NIH Chest X-ray and Indiana University Chest X-ray datasets. They contain 14 and 17 target labels respectively.

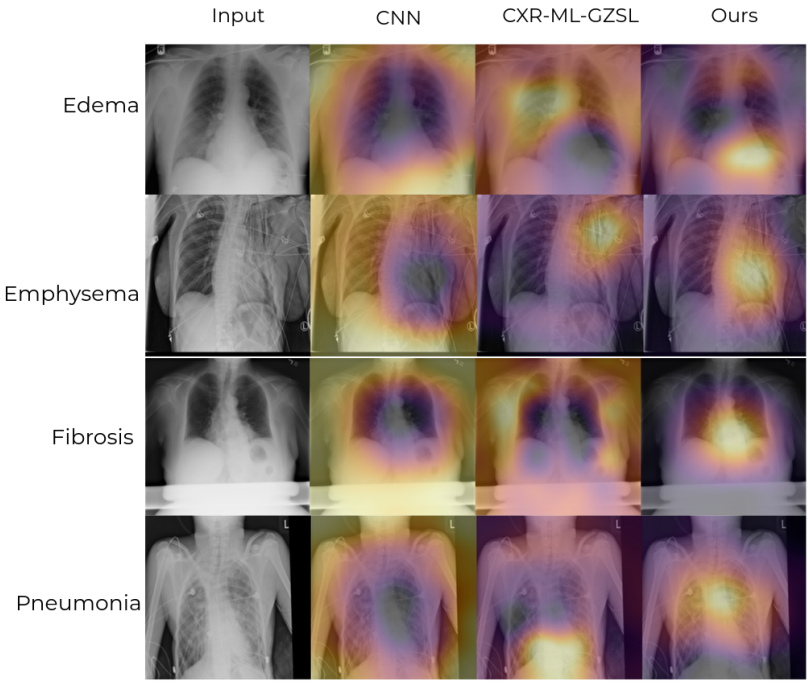

Figure 10: Saliency map visualization for the *unseen* classes. These are the samples where our model performs **poorly**. Each row contains one of the unseen diseases and the Grad-CAM output of the three models. We have included the original input image in the first column for reference.The model focuses on regions that are relevant for diagnosis of the individual diseases.

As such, we can handle the scenario with 14 or 17 diseases easily using hand-crafted features. However, if we have more labels, like a typical clinical setting, it would be tedious to handcraft the graphs. To test the extensibility of our method to such scenarios, we ran experiments to *parse* the *UMLS* based on the PadChest(Bustos et al., 2020) labels. The dataset contains 89 labels of interest. The final graph generated by our method, after pruning based on the *All-Source-Shortest Path* contains 755 nodes and 2000 edges. We plan to include experiments on PadChest as part of our future work.

The effectiveness of semantic knowledge. We train the *DenseNet-121* backbone for only the *seen* classes and evaluate it for both *seen* and *unseen* classes to check if the task is trivial enough to be solved without any domain-specific semantic knowledge. As expected, the model performs well for the *seen* classes with an AUROC of 0.80 while struggles with the novel *unseen* classes with an AUROC of 0.52) (see Table 2), suggesting that semantic knowledge is essential.