Deep Hierarchical Multi-label Classification of Chest X-ray Images

Haomin Chen[1,2] HCHEN135@JHU.EDU
Shun Miao[1] SHWINMIAO@GMAIL.COM
Daguang Xu[1] DAGUANGX@NVIDIA.COM
Gregory D. Hager[2] HAGER@CS.JHU.EDU
Adam P. Harrison[1] ADAM.P.HARRISON@GMAIL.COM
[1] *NVIDIA AI-Infra, Bethesda, MD*
[2] *Department of Computer Science, Johns Hopkins University, Baltimore, MD*

Abstract

Chest X-rays (CXRs) are a crucial and extraordinarily common diagnostic tool, leading to heavy research for Computer-Aided Diagnosis (CAD) solutions. However, both high classification accuracy *and* meaningful model predictions that respect and incorporate clinical taxonomies are crucial for CAD usability. To this end, we present a deep Hierarchical Multi-Label Classification (HMLC) approach for CXR CAD. Different than other hierarchical systems, we show that first training the network to model conditional probability directly and then refining it with unconditional probabilities is key in boosting performance. In addition, we also formulate a numerically stable cross-entropy loss function for unconditional probabilities that provides concrete performance improvements. To the best of our knowledge, we are the first to apply HMLC to medical imaging CAD. We extensively evaluate our approach on detecting 14 abnormality labels from the PLCO dataset, which comprises 198,000 manually annotated CXRs. We report a mean Area Under the Curve (AUC) of 0.887, the highest yet reported for this dataset. These performance improvements, combined with the inherent usefulness of taxonomic predictions, indicate that our approach represents a useful step forward for CXR CAD.

Keywords: hierarchical multi-label classification, chest x-ray, computer aided diagnosis.

1. Introduction

Chest X-rays (CXRs) are the most frequently ordered image study (Folio, 2012). Commensurate with this importance, CXR Computer-Aided Diagnosis (CAD) has received considerable research attention, both prior to the popularity of deep learning (Jaeger et al., 2013), and afterwards (Wang et al., 2017; Yao et al., 2017; Guendel et al., 2018). These efforts have achieved notable successes, *e.g.*, Guendel et al. (2018) reporting very high mean Area Under the Curves (AUCs) on the Prostate, Lung, Colorectal and Ovarian (PLCO) dataset (Gohagan et al., 2000). Yet, pushing raw performance further will likely require models that depart from standard multi-label classifiers. Perhaps more importantly, standard multi-label classifiers are not able to leverage or align with domain knowledge. For instance, despite their importance to clinical understanding and interpretation, taxonomies of disease patterns are not typically incorporated into CXR CAD systems, or for other medical CAD domains for that matter. This observation motivates our work, which uses Hierarchical Multi-Label Classification (HMLC) to both push raw AUC performance further and also to provide more meaningful predictions that leverage clinical taxonomies.

Organizing diagnoses or observations into ontologies and/or taxonomies is crucial within radiology, *e.g.*, RadLex (Langlotz, 2006), with CXR interpretation being no exception (Folio, 2012; Demner-Fushman et al., 2015; Dimitrovski et al., 2011). This importance should also be reflected within CAD systems. For instance, when uncertain about fine-level predictions, *e.g.*, *nodules* vs. *masses*, a CAD system should still be able to provide meaningful parent-level predictions, *e.g.*, *pulmonary nodules and masses*. This parent prediction may be all the clinician is interested in anyway. Another important benefit is that observations are conditioned upon their parent being true, allowing fine-level predictors to focus solely on discriminating between siblings rather than on having to discriminate across all possible conditions. This can help improve classification performance (Bi and Kwok, 2015).

Because more than one abnormality can be observed on a CXR at the same time, a CAD system must operate in a multi-label setting. Prior work has well articulated the limitations of Binary Relevance (BR) learning (Dembczyński et al., 2012), *i.e.*, treating each label as an independent prediction. HMLC helps address this, by making predictions conditionally independent rather than globally independent. Inferring risk-optimal binary HMLC labels given a set of predictions is a surprisingly rich topic (Bi and Kwok, 2015), but here we focus instead on producing said predictions. In this way, our focus has similarities to recent deep neural network approaches for hierarchical *multi-class* classification of natural images (Redmon and Farhadi, 2017; Roy et al., 2018; Yan et al., 2014). A common approach is to simply train classifiers to predict conditional probabilities at each node. Within medical imaging, hierarchical classifiers have not received much attention for CAD, but there are works on HMLC medical image retrieval (Pourghassem and Ghassemian, 2008; Demner-Fushman et al., 2015; Dimitrovski et al., 2011).

We present a deep HMLC approach for CXR CAD. Our work departs from prior art in three important ways. First, like other deep approaches, we train a classifier to predict conditional probabilities. However, we also demonstrate that a second fine-tuning stage, trained using unconditional probabilities, can boost performance even further. Second, we formulate a numerically stable and principled loss function for unconditional probabilities that can handle the unstable multiplication of prediction outputs. Finally, we argue that in an HMLC setting, global metrics, such as AUCs, do not provide a complete picture. Instead, we advocate also investigating performance conditioned on a high-level node being true, *e.g.*, *one or more abnormalities*, providing a measure of model performance for different patient populations, some of which may be more clinically relevant depending on the application. We evaluate our HMLC approach on the PLCO dataset (Gohagan et al., 2000), reporting a mean AUC of 0.887, the highest yet reported for this dataset. To the best of our knowledge, we are the first to outline an HMLC CAD system for medical imaging.

2. Methods

We introduce a two-stage method for CXR HMLC, which we first overview in Section 2.1. This is followed by Sections 2.2 and 2.3, which detail our two training stages that use conditional probability and a numerically stable unconditional probability formulation, respectively.

2.1. Hierarchical Multi-Label Classification

The first step in creating an HMLC system is to create the label taxonomy. Without loss of generality, we focus on the labels and data found within the CXR arm of the PLCO dataset (Gohagan et al., 2000), a large-scale lung cancer screening trial that collected structured radiological reports

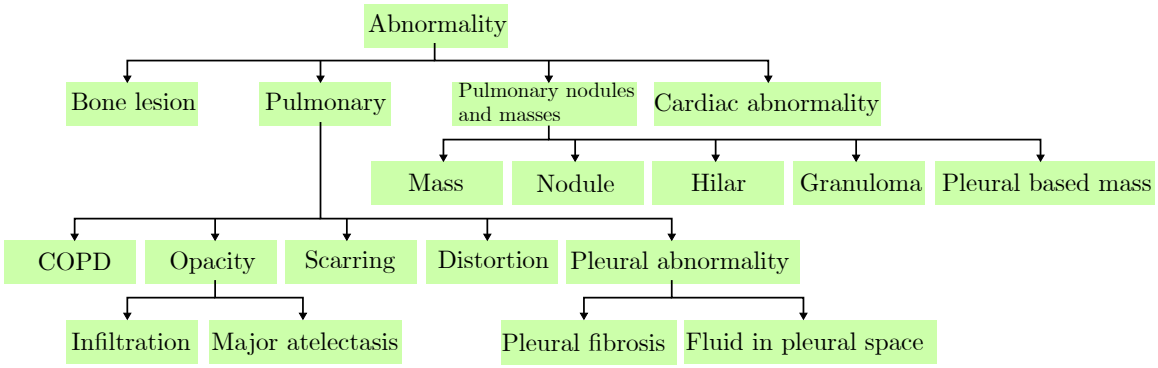

Figure 1: Constructed label hierarchy from the PLCO dataset.

of abnormalities obtained from multiple US clinical centers. From these fine-grained labels, we constructed a label taxonomy[1], which is shown in Figure 1. The hierarchical structure follows the PLCO trial's division of "suspicious for cancer" disease patterns vs. not, and is further partitioned using common groupings (Folio, 2012), totalling 19 labels. While care was taken in constructing the taxonomy and we aimed for clinical usefulness, we make no specific claim as such. We instead use the taxonomy to explore the benefits of HMLC, stressing that our approach is general enough to incorporate any appropriate taxonomy.

Because this is a multi-label setting, all or none of the labels in Figure 1 can be positive. The only restriction is that if a child is positive, its parent must be too. Siblings are not mutually exclusive. Finally, we assume that each image is associated with a set of fine-level labels and their antecedents, *i.e.*, there are no incomplete paths.

We use a DenseNet-121 (Huang et al., 2016) model as a backbone, connecting 19 fully connected layers to its last feature layer to extract 19 scalar outputs. Each output is assumed to represent the conditional probability (or its logit) given its parent is true. Thus, once the model is successfully trained, unconditional probabilities can be calculated from the output using the chain rule, *e.g.*, the unconditional probability of *scarring* can be calculated as

$$P(\text{Scarring}) = P(\text{Abnormality})P(\text{Pulmonary}|\text{Abnormality})P(\text{Scarring}|\text{Pulmonary}). \quad (1)$$

In this way, the predicted unconditional probability of a parent label is guaranteed to be greater than or equal to its children labels. We refer to the conditional probability in a label hierarchy as Hierarchical Label Conditional Probability (HLCP), and the unconditional probability calculated following the chain rule as Hierarchical Label Unconditional Probability (HLUP). The network outputs can be trained either conditionally or unconditionally, which we outline in the next two sections.

2.2. Training with Conditional Probability

Similar to prior work (Redmon and Farhadi, 2017; Roy et al., 2018; Yan et al., 2014), in the first stage of the proposed training scheme, each classifier is only trained on data conditioned upon its

1. Note, we merged "left hilar abnormality" and "right hilar abnormality" into "hilar abnormality", resulting in 19 labels.

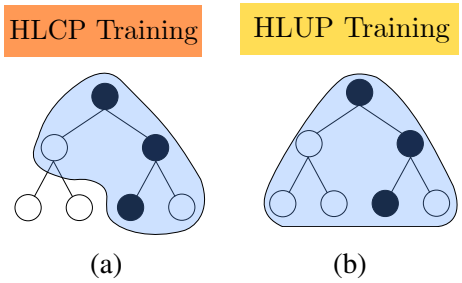

Figure 2: The HLCP and HLUP losses are depicted in (a) and (b), respectively, where black and white points are positive and negative labels, respectively. Blue areas indicate the activation area in the loss functions.

parent label being positive. Thus, training directly models the conditional probability. The shared part of the classifiers, *i.e.*, feature layers from the backbone network, is trained jointly by all the tasks. Specifically, for each image the losses are only calculated on labels whose parent label is also positive. For example, when an image with positive *Scarring* and no other positive labels is fed into training, only the losses of *Abnormality* and the children labels of *Pulmonary* and *Abnormality* are calculated and used for training.

Figure 2 (a) illustrates this training regimen, which we denote HLCP training. In this work, we use Cross Entropy (CE) loss to train the conditional probabilities, which can be written as

$$L_{HLCP} = \sum_{m \in M} CE\left(z_m, \hat{z}_m\right) * 1_{\{z_{a(m)}=1\}}, \tag{2}$$

where M denotes the set of all disease patterns, and m and $a(m)$ denote a disease pattern and its ancestor, respectively. Here $CE(\cdot,\cdot)$ denotes the cross entropy loss, and $z_m \in \{0,1\}$ denotes the ground truth label of m, with $\hat{z}_m$ corresponding to the network's sigmoid output.

Training with conditional probability is a very effective initialization step, as it concentrates the modeling power solely on discriminating siblings under the same parent label, rather than having to discriminate across all labels, which eases convergence and reduces confounding factors. It also alleviates the problem of low label prevalence because fewer negative samples are used for each label.

2.3. Fine Tuning with Unconditional Probability

In the second stage, we finetune the model using an HLUP CE loss. This stage aims at improving the accuracy of unconditional probability predictions, which is what is actually used during inference and is thus critical to classification performance. Another important advantage is that the final linear layer sees more negative samples. Predicted unconditional probabilities for label m, denoted $\hat{p}_m$, are calculated using the chain rule:

$$\hat{p}_m = \prod_{m' \in A(m)} \hat{z}_{m'}, \tag{3}$$

where $A(m)$ is the union of label m and its antecedents. When training using unconditional probabilities, the loss is calculated on every classifier output for every data instance. Thus, the HLUP CE

loss for each image is simply

$$L_{HLUP} = \sum_{m \in M} CE\left(z_m, \hat{p}_m\right). \tag{4}$$

Figure 2(b) visually depicts this loss.

A naive way to calculate (4) would be a direct calculation. However, such an approach introduces instability during optimization, as the training would have to minimize the product of network outputs. In addition, the product of probability values within $[0,1]$ can cause arithmetic underflow. For this reason, we outline a numerically stable formulation of (4), whose derivation can be found in Appendix A:

$$L_{HLUP} = \sum_{m' \in A(m)} \ell_{m'} + \gamma, \tag{5}$$

$$\ell_{m'} = -z_m \log\left(\frac{1}{1+\exp(-y_{m'})}\right) - (1-z_m)\log\left(1 - \frac{1}{1+\exp(-y_{m'})}\right), \tag{6}$$

$$\gamma = -(1-z_m)\left(\sum_{m' \in A(m)} y_{m'} + LSE\left(\left\{\sum_{j \in S} -y_j \quad \forall S \in \mathscr{P}\left(A(m)\right) \setminus \{\emptyset\}\right\}\right)\right), \tag{7}$$

where $\hat{y}_{m'}$ is the logit output for label m'. The expression in (6) is simply the CE loss given a logit input, which enjoys stable implementations within all popular deep learning software. For (7), $\mathscr{P}(\cdot)$ denotes the powerset, S enumerates all possible subsets of $\mathscr{P}(A(m))$, excluding the empty set, and $LSE(\cdot)$ is the LogSumExp function. Enumerating the powerset produces an obvious combinatorial explosion. However, for smaller-scale hierarchies, like that in Figure 1, it remains tractable. For larger hierarchies, an $O(|A(m)|)$ solution involves simply interpreting the LogSumExp as a smooth approximation to the maximum function, but we do not need that here. Numerically stable implementations of the LogSumExp, and its gradient, are well known. Thus, since both terms in (5) can be implemented stably, our formulation avoids the numerical issues faced by a naive calculation of (4).

3. Experiments

We validate our approach on the PLCO dataset (Gohagan et al., 2000), which contains 198,000 manually labeled CXRs. While the recent ChestXRay14 dataset (Wang et al., 2017) is extraordinarily valuable, we expect the PLCO structured labels to have greater reliability, especially in evaluation, over the former's text-mined labels. As noted in Section 2.1, after pre-processing the data is left with 14 leaf-node labels. We split the data into training, validation, and test sets, corresponding to 70%, 10%, and 20% of the data, respectively. Data is split at the patient level, and care was taken to balance the prevalence of each disease pattern as much as possible.

Our chosen network is DenseNet-121 (Huang et al., 2016), implemented using TensorFlow. We first train with the HLCP CE loss of (2) fine-tuning from a model pretrained from ImageNet (Deng et al., 2009). We refer to this model simply as *HLCP*. To produce our final model, we then finetune the HLCP model using the HLUP CE loss of (4). We denote this final model as *HLUP-finetune*.

While we do compare to a recent DenseNet121 BR approach (Guendel et al., 2018), we stress that direct comparisons of numbers are impossible, as Guendel et al. (2018) used different data splits and only evaluated on 12 leaf-node labels. For that reason, we also compare against three baseline

Table 1: Comparison of AUC and AP across tested models. Mean values across leaf-node and high-level disease patterns are shown, as well as leaf-node label conditioned on one or more abnormality being present.

	Leaf-node labels		**High-level labels**		**Leaf-node labels conditioned on abnormality**	
	AUC	AP	AUC	AP	AUC	AP
(Guendel et al., 2018)	0.874	N/A	N/A	N/A	N/A	N/A
BR-leaf	0.871	0.234	N/A	N/A	0.806	0.334
BR-all	0.867	0.221	0.852	0.440	0.808	0.323
HLUP	0.872	0.214	0.856	0.436	0.799	0.288
HLCP	0.879	0.229	0.857	0.440	0.822	0.329
HLUP-finetune	0.887	0.250	0.866	0.460	0.832	0.342

models, all using the same trunk network fine-tuned from ImageNet pretrained weights. The first, denoted *BR-leaf*, is trained using CE loss on the 14 leaf-node labels. This measures performance using a standard multi-label BR approach. The second, denoted *BR-all* is very similar, but trains a CE loss on all 19 labels independently, including high-level ones. In this way, *BR-all* measures performance when one wishes to naively output high-level abnormality nodes, without considering label taxonomy. Finally, we also test against a model trained using the HLUP CE loss, but not starting from the HLCP weights. As such, this baseline, denoted *HLUP*, helps reveal the impact of using a two-stage approach vs. simply training an HLUP classifier in one step. For all tested models, extensive hyper-parameter searches were performed on the NVIDIA cluster to optimize mean validation AUCs of leaf-node labels.

For all models, we evaluate the mean AUC and Average Precision (AP) on the test set. To start, we measure performance on both leaf-node as well as high-level patterns. The results are shown in the first two columns of Table 1. As the table demonstrates, the standard baseline BR-leaf model produces high AUC scores, in line with prior art (Guendel et al., 2018); however, it does not provide high-level predictions based on a taxonomy. Naively executing BR training on the entire taxonomy, *i.e.*, the BR-all model, does not improve performance. This indicates that if not properly incorporated, the label taxonomy does not benefit performance.

In contrast, the HLCP model is indeed able to match BR-leaf's performance on the leaf-node labels, despite also being able to provide high-level predictions. HLUP-finetune goes further by exceeding BR-leaf's performance, demonstrating that our two-stage training process can produce tangible improvements. This is underscored when comparing HLUP-finetune with HLUP, which highlights that without the two-stage training, HLUP training cannot reach the same performance. If we limit ourselves to models incorporating the entire taxonomy, our final HLUP-finetune model outperforms BR-all by 2% and 2.9% in leaf-node mean AUC and AP values, respectively. Figure 3 provides more details on these improvements, demonstrating that AUC values are higher for HLUP-finetune compared to the baseline method for all leaf-node and high-level disease patterns. Although not graphed here for clarity reasons, HLUP-finetune also outperformed the HLCP method for all

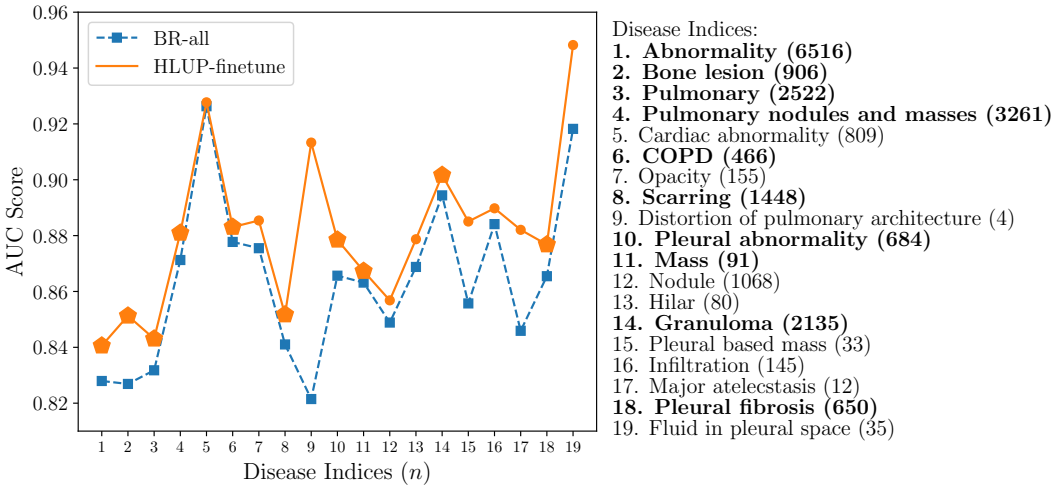

Figure 3: Comparison of AUC scores for all leaf-node and high-level disease patterns for the BR-all and HLUP-finetune models. The dashed line separates the leaf-node from the high-level disease patterns. Bolded labels and larger graph markers, denote disease patterns exhibiting statistically significant improvement ($p < 0.05$) using the StAR software implementation (Vergara et al., 2008) of the non-parametric test of DeLong et al. (1988).

disease patterns. Of note, is that significance values also respect the disease hierarchy, and if a child disease pattern demonstrates statistically significant improvement, so does its parent.

Because more than one label can be positive, multi-label classification performance has exponentially more facets for evaluation than single-label or even multi-class settings. Here, we explore one such facet, namely model performance conditioned on high-level nodes being positive. We restrict our focus to CXRs exhibiting one or more disease patterns, *i.e.*, *abnormality* being positive. As such, this sheds light on model performance when it may be critical to discriminate what combination of disease patterns are present, which is crucial for proper CXR interpretation (Folio, 2012). The last column of Table 1 depicts these results. As can be seen, in such settings, HLUP-finetune still exhibits increased performance over the baseline models and also the next-best hierarchical model. Importantly, if we compare the conditional AUCs between BR-all and HLUP-finetune, we see a 2.4% increase. As a result, in the critical setting of a CXR exhibiting at least one disease pattern, our HLUP-finetune still manages to provide key performance improvements.

Finally, we compare our numerically stable implementation of HLUP CE loss in (5) to: (a) the naive approach of directly optimizing (3); and (b) to a recent rescaling approximation, originally introduced for the multiplication of independent, rather than conditional probabilities, seen in multi-instance learning (Li et al., 2018). This latter approach rescales each individual probability multiplicand in (3) to guarantee that the product is greater than or equal to 1e-7. Similar to the naive approach, the product is then optimized directly using CE loss. Based on a maximum depth of four for our taxonomy, we implement this approach by rescaling each multiplicand in (3) to $[0.02, 1]$. As Table 2 demonstrates, regardless if we train from ImageNet or finetune from the HLCP model, our numerically stable formulation far outperforms this rescaling approximation. However, while our

Table 2: Comparison of AUCs produced using different HLUP CE loss implementations.

HLUP (naive)	HLUP (rescale)	HLUP (ours)	HLUP-finetune (naive)	HLUP-finetune (rescale)	HLUP-finetune (ours)
0.864	0.853	0.872	0.886	0.867	0.887

HLUP loss outperforms the naive implementation when training from ImageNet weights, it does not exhibit improvements when fine-tuning from the HLCP model. We hypothesize that the predictions for the HLCP are already at a good enough quality that the numerical instabilities of the naive HLUP CE loss are not severe enough to impair performance. Nonetheless, given the improvements when training from ImageNet weights, these results indicate that our HLCP CE loss does indeed provide tangible improvements in convergence stability. We expect these improvements to be greater given taxonomies of greater depth, and our formulation should also prove valuable to multi-instance setups which must optimize CE loss over the product of large numbers of probabilities, *e.g.*, the 256 multiplicands seen in Li et al. (2018).

4. Conclusion

We have presented a two-stage approach for deep HMLC of CXRs that combines conditional training with an unconditional probability fine-tuning step. To effect the latter, we introduce a new and numerically stable formulation for HLUP CE loss, which we expect would also prove valuable in other training scenarios involving the multiplication of probability predictions, *e.g.*, multi-instance learning. Through comprehensive evaluations, we report the highest yet mean AUC on the PLCO dataset, outperforming hierarchical and non-hierarchical alternatives. We also show performance improvements conditioned on one or more abnormalities being present, *i.e.*, predicting the specific combination of disease patterns, which is crucial for CXR interpretation. Experiments also demonstrate that HLUP fine-tuning is crucial in achieving these results. Future work should focus on characterizing improvements against the recently released CheXpert dataset (Irvin et al., 2019) and also on computer vision benchmarks. Additionally, another potential strength of the HMLC approach is handling incomplete labels, which also deserves further investigation. Finally, another interesting focus would be exploring whether using hierarchical features, rather than the shared ones of our approach, would improve results further.

5. Acknowledgements

We thank the National Cancer Institute (NCI) for access to NCI's data collected by the PLCO Cancer Screening Trial. The statements contained herein are solely ours and do not represent or imply concurrence or endorsement by NCI. We also thank Chaochao Yan for help on pre-processing the PLCO images and labels.

References

W. Bi and J. T. Kwok. Bayes-optimal hierarchical multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 27(11):2907–2918, Nov 2015. ISSN 1041-4347. doi: 10.1109/TKDE.2015.2441707.

Elizabeth R. DeLong, David M. DeLong, and Daniel L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, 44:837–845, 1988.

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence and loss minimization in multi-label classification. *Mach. Learn.*, 88(1-2):5–45, July 2012. ISSN 0885-6125. doi: 10.1007/s10994-012-5285-8. URL https://doi.org/10.1007/s10994-012-5285-8.

Dina Demner-Fushman, Sonya E. Shooshan, Laritza Rodriguez, Sameer Antani, and George R. Thoma. Annotation of chest radiology reports for indexing and retrieval. In Henning Müller, Oscar Alfonso Jimenez del Toro, Allan Hanbury, Georg Langs, and Antonio Foncubierta Rodriguez, editors, *Multimodal Retrieval in the Medical Domain*, pages 99–111, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24471-6.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Deroski. Hierarchical annotation of medical images. *Pattern Recogn.*, 44(10-11):2436–2449, October 2011. ISSN 0031-3203. doi: 10.1016/j.patcog.2011.03.026. URL http://dx.doi.org/10.1016/j.patcog.2011.03.026.

Les Folio. *Chest imaging: An algorithmic approach to learning*. Springer, 01 2012.

John K. Gohagan, Philip C. Prorok, Richard B. Hayes, and Barnett-S. Kramer. The prostate, lung, colorectal and ovarian (plco) cancer screening trial of the national cancer institute: History, organization, and status. *Controlled Clinical Trials*, 21(6, Supplement 1):251S – 272S, 2000. ISSN 0197-2456. doi: https://doi.org/10.1016/S0197-2456(00)00097-0. URL http://www.sciencedirect.com/science/article/pii/S0197245600000970.

Sebastian Guendel, Sasa Grbic, Bogdan Georgescu, Kevin Zhou, Ludwig Ritschl, Andreas Meier, and Dorin Comaniciu. Learning to recognize abnormalities in chest x-rays with location-aware dense networks, 2018. arXiv:1803.04565.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL http://arxiv.org/abs/1608.06993.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn L. Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *CoRR*, abs/1901.07031, 2019. URL http://arxiv.org/abs/1901.07031.

Stefan Jaeger, Alexandros Karargyris, Sema Candemir, Jenifer Siegelman, Les Folio, Sameer Antani, and George Thoma. Automatic screening for tuberculosis in chest radiographs: a survey. *Quantitative Imaging in Medicine and Surgery*, 3(2):89–99, April 2013. ISSN 2223-4292. doi: 10.3978/j.issn.2223-4292.2013.04.03.

Curtis P. Langlotz. Radlex: A new method for indexing online educational materials. *RadioGraphics*, 26(6):1595–1597, 2006. doi: 10.1148/rg.266065168. URL https://doi.org/10.1148/rg.266065168. PMID: 17102038.

Zhe Li, Chong Wang, Mei Han, Yuan Xue, Wei Wei, Li-Jia Li, and Li Fei-Fei. Thoracic disease identification and localization with limited supervision. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8290–8299, 2018.

Hossein Pourghassem and Hassan Ghassemian. Content-based medical image classification using a new hierarchical merging scheme. *Computerized Medical Imaging and Graphics*, 32(8):651 – 661, 2008.

J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 6517–6525, July 2017. doi: 10.1109/CVPR.2017.690. URL doi.ieeecomputersociety.org/10.1109/CVPR.2017.690.

Deboleena Roy, Priyadarshini Panda, and Kaushik Roy. Tree-cnn: A deep convolutional neural network for lifelong learning. *CoRR*, abs/1802.05800, 2018. URL http://arxiv.org/abs/1802.05800.

Ismael A Vergara, Tomás Norambuena, Evandro Ferrada, Alex W Slater, and Francisco Melo. StAR: a simple tool for the statistical comparison of ROC curves. *BMC bioinformatics*, 9:265–265, June 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-265. URL https://www.ncbi.nlm.nih.gov/pubmed/18534022.

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. *CoRR*, abs/1705.02315, 2017. URL http://arxiv.org/abs/1705.02315.

Zhicheng Yan, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Robinson Piramuthu. HD-CNN: hierarchical deep convolutional neural network for image classification. *CoRR*, abs/1410.0736, 2014. URL http://arxiv.org/abs/1410.0736.

Li Yao, Eric Poblenz, Dmitry Dagunts, Ben Covington, Devon Bernard, and Kevin Lyman. Learning to diagnose from scratch by exploiting dependencies among labels, 2017. arXiv:1710.10501.

Appendix A. Numerically Stable Formulation of HLUP CE Loss

Denoting the network's output logits as $\hat{y}_{(.)}$, the predicted unconditional probability of label m can be written as:

$$\hat{p}_m = \prod_{m'} \frac{1}{1+\exp(-y_{m'})}, \tag{8}$$

where we use m' to denote $m' \in A(m)$ for notational simplicity.

The HLUP CE loss is calculated as:

$$L_{HLUP} = -z_m \log(\hat{p}_m) - (1-z_m)\log(1-\hat{p}_m), \tag{9}$$

$$= -z_m \log\left(\prod_{m'} \frac{1}{1+\exp(-y_{m'})}\right) - (1-z_m)\log\left(1-\left(\prod_{m'} \frac{1}{1+\exp(-y_{m'})}\right)\right), \tag{10}$$

where z_m is the ground truth label of m.

We would like to break up the second term in (10) to produce the following formulation:

$$L_{HLUP} = -z_m \log\left(\prod_{m'} \frac{1}{1+\exp(-y_{m'})}\right) - (1-z_m)\log\left(\prod_{m'}\left(1-\frac{1}{1+\exp(-y_{m'})}\right)\right) + \gamma \tag{11}$$

$$= \sum_{m'}\left(-z_m \log\left(\frac{1}{1+\exp(-y_{m'})}\right) - (1-z_m)\log\left(1-\frac{1}{1+\exp(-y_{m'})}\right)\right) + \gamma, \tag{12}$$

which can be simplified to a sum of individual CE losses plus γ:

$$L_{HLUP} = \sum_{m'} \ell_{m'} + \gamma, \tag{13}$$

where ℓ_m are individual cross entropy terms, using z_m and $y_{m'}$ as the ground truth and logit input, respectively, and γ is the scalar quantity we want to formulate. Note that (13) allows us to take advantage of numerically stable CE implementations, *e.g.*, those within Tensorflow, to calculate $\sum_{m'} \ell_{m'}$.

To satisfy (12), we will need γ to satisfy:

$$-(1-z_m)\log\left(\prod_{m'}\left(1-\frac{1}{1+\exp(-y_{m'})}\right)\right) + \gamma = -(1-z_m)\log\left(1-\left(\prod_{m'}\frac{1}{1+\exp(-y_{m'})}\right)\right),$$

$$\log\left(\prod_{m'}\left(1-\frac{1}{1+\exp(-y_{m'})}\right)\right) - \frac{\gamma}{1-z_m} = \log\left(1-\left(\prod_{m'}\frac{1}{1+\exp(-y_{m'})}\right)\right),$$

$$\left(\prod_{m'}\left(1-\frac{1}{1+\exp(-y_{m'})}\right)\right)\exp(-\frac{\gamma}{1-z_m}) = 1-\left(\prod_{m'}\frac{1}{1+\exp(-y_{m'})}\right). \tag{14}$$

Denoting

$$\alpha = \exp(-\frac{\gamma}{1-z_m}), \tag{15}$$

we have:

$$\alpha\left(\prod_{m'}\left(1-\frac{1}{1+\exp(-y_{m'})}\right)\right)=1-\prod_{m'}\frac{1}{1+\exp(-y_{m'})}, \tag{16}$$

$$\alpha\left(\frac{\prod_{m'}\exp(-y_{m'})}{\prod_{m'}(1+\exp(-y_{m'}))}\right)=\frac{\prod_{m'}(1+\exp(-y_{m'}))-1}{\prod_{m'}(1+\exp(-y_{m'}))}, \tag{17}$$

$$\alpha=\frac{\prod_{m'}(1+\exp(-y_{m'}))-1}{\exp(\sum_{m'}-y_{m'})}. \tag{18}$$

Substituting the left side of (18) into (15) gives us:

$$\begin{aligned}\gamma&=-(1-z_m)\log(\alpha)\\&=-(1-z_m)\left(\sum_{m'}y_{m'}+\log\left(\prod_{m'}(1+\exp(-y_{m'}))-1\right)\right).\end{aligned} \tag{19}$$

If the log-product term of (19) is expanded, with 1 subtracted, it will result in

$$\gamma=-(1-z_m)\left(\sum_{m'}y_{m'}+\log\left(\sum_{S\in\mathscr{P}(A(m))\setminus\{\emptyset\}}\exp\left(\sum_{j\in S}-y_j\right)\right)\right), \tag{20}$$

where S enumerates all possible subsets of the powerset of $A(m)$, excluding the empty set. For example if there were two logits, y_1 and y_2, the summation inside the log would be:

$$\exp(-y_1)+\exp(-y_2)+\exp(-y_1-y_2). \tag{21}$$

The expression in (20) can be written as

$$\gamma=-(1-z_m)\left(\sum_{m'}y_{m'}+LSE\left(\left\{\sum_{j\in S}-y_j\quad\forall S\in\mathscr{P}(A(m))\setminus\{\emptyset\}\right\}\right)\right), \tag{22}$$

where *LSE* is the LogSumExp function. Many numerical packages, including TensorFlow, provide numerically stable implementations of *LSE*, and its derivative. By substituting (22) into (12), a numerically stable version of the HLUP CE loss can be calculated.

Should the cardinality of the powerset be too high, the LogSumExp expression can be approximated as a maximum function, which can be calculated using an $O(|A(m)|)$ scan of $y_{m'}$ values:

$$\gamma\approx-(1-z_m)\left(\sum_{m'}y_{m'}+\max\left(\left\{\sum_{j\in S}-y_j\quad\forall S\in\mathscr{P}(A(m))\setminus\{\emptyset\}\right\}\right)\right), \tag{23}$$

$$=\begin{cases}-(1-z_m)\left(\sum_{m'}y_{m'}+\sum_{j:y_j<0}-y_j\right), & \text{if } \exists y_{m'}<0\\ -(1-z_m)\left(\sum_{m'}y_{m'}+\max(\{-y_{m'}\})\right), & \text{otherwise}\end{cases}. \tag{24}$$