**SPH6004 Advanced Statistical Learning**

**Assignment 2**

**Saw Swee Hock School of Public Health**

**Ma Pei (A0305342R), A Janhavi (A0205165M),**

**He Yichen(A0236326H), Guan Tong(A0177881M),**

# Introduction

Medical imaging plays a critical role in the diagnosis and management of pulmonary diseases. With the growing availability of large-scale datasets and advancements in machine learning, there is increasing interest in developing automated tools to assist radiologists in interpreting chest x-rays (CXRs). In this project, we aim to develop a two-part system that leverages both image and text data to automatically identify and extract information about lung conditions from the MIMIC-CXR dataset.

Task 1 of this project focuses on building a multi-label classification model using chest X-ray images to predict the presence of 13 different lung conditions (excluding No Findings). The MIMIC-CXR dataset comprises over 370,000 radiographs in DICOM format, accompanied by corresponding labels indicating radiological findings. Due to the substantial storage and computational burden associated with the raw image data, we used generalised image embeddings extracted from a deep learning model trained to detect image abnormalities. These embeddings enable efficient training of downstream classification models when data size is constrained1. Task 2 of the project focuses on developing a language model to extract structured information from free-text radiology reports associated with each image. This component aims to identify the same 13 different lung conditions used in Part 1, thereby providing a complementary modality for diagnosis.

We began these tasks with preliminary data analysis to better understand the distribution and balance of condition labels. Given the substantial class imbalance present in the dataset, different class imbalance techniques were deployed. Additionally, performance metrics beyond accuracy, such as average precision score, were employed to more effectively evaluate model performance under imbalance. For Task 1, multiple classification methods were explored, including tree-based models, multilayer perceptrons, and transformers. For Task 2, a pipeline using simple logistic regression was explored after Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. Finally, we developed a fusion-based multimodal model that jointly leveraged visual and textual information for multi-label classification. Image and text features were first projected into a shared latent space and combined through a transformer-based fusion module to capture cross-modal interactions. The resulting integrated representation enabled the model to learn from complementary information across both modalities. An overall framework for our project is provided in Figure 1.

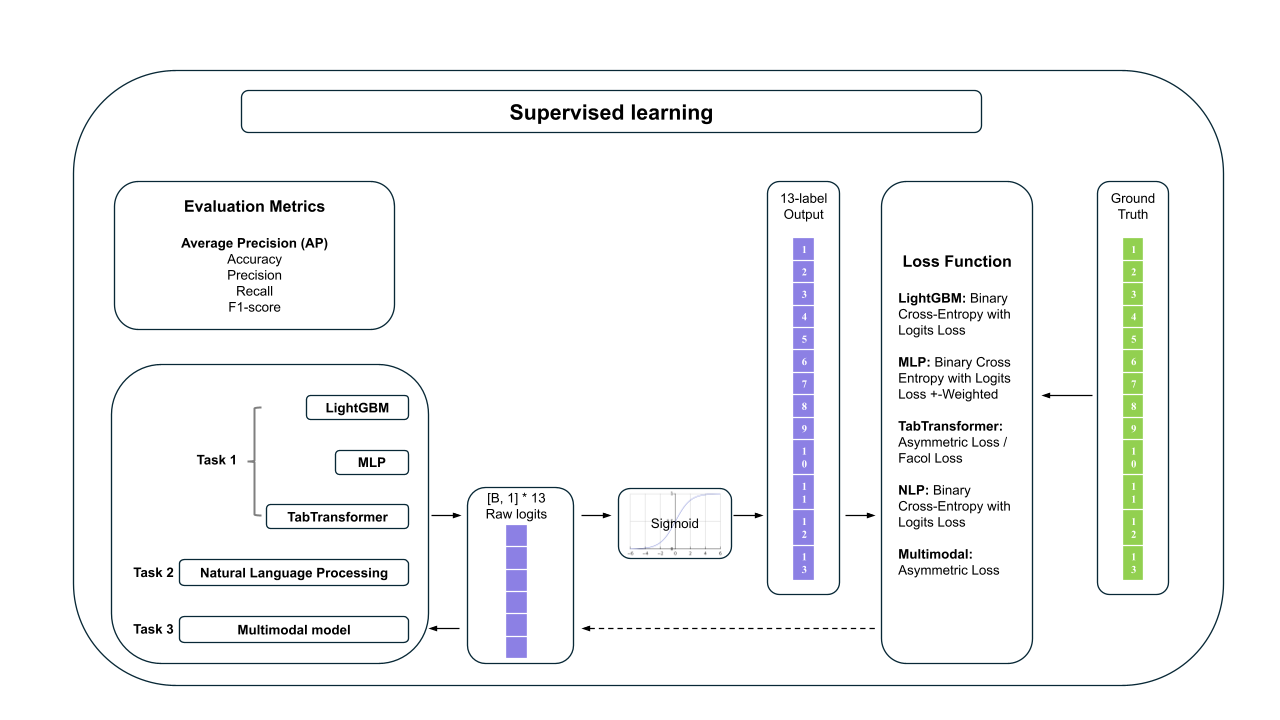


Figure 1: Overview of the methods used for our project

All codes have been uploaded in the public repository available on GitHub via <https://github.com/mapei0728/Group4_SPH-6004_Group-Assignment_AY24-25>.

# 

# Task 1: Classification using CXR images

## Methods

### Preliminary data analysis

An initial exploration of the training dataset revealed a pronounced class imbalance across all 13 lung condition labels. For every label, the proportion of samples where the condition is present (labelled as 1) is substantially lower than those where it is absent (labelled as 0). This trend is consistent across all labels, with the prevalence of lung conditions generally ranging from 1.05% to 11.8%, and the remaining samples marked as negative or missing (Refer to Figure 2).

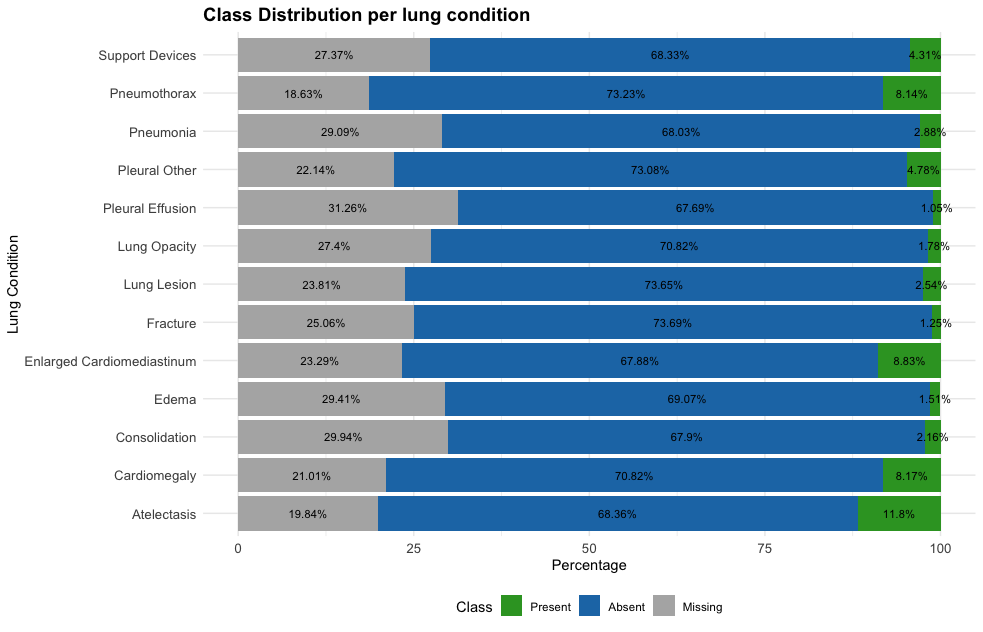


Figure 2: Class distribution for different lung conditions on the pre-fixed training data set. The total sample size is 36,502 (100%) for the different lung conditions.

To address this imbalance during model training, we applied a range of techniques, including the Synthetic Minority Over-sampling Technique (SMOTE) and alternative loss functions designed for imbalanced data. SMOTE works by generating synthetic examples for the minority class (the positive cases) through interpolation between existing samples, helping to balance the class distribution in the training set. In addition to resampling, we tested several loss functions to focus training on harder examples. These included a weighted Binary Cross Entropy Loss with logits objective, which increased the penalty for misclassifying positive samples, Masked Focal Loss (FocalL) 2, which concentrates learning on difficult-to-classify cases, and Masked Asymmetric Loss (ASL) 3, which adjusts the loss contribution based on the type of misclassification. Model performance was evaluated with and without these imbalance correction techniques to better discern their effects.

In addition to this imbalance, a substantial portion of the data for each label is missing, between 18.63% and 31.26% (Figure 2). Missing data indicates that the results from the CXR for the specific condition is uncertain1. Removing all patient records with any missing label would result in the loss of approximately 32.38% of the dataset and also cause a reduction in already limited positive samples across different labels. To avoid this, we adopted a masking strategy during training, treating each label independently. This approach allows us to make the most of the available data while accounting for label-specific missingness.

Across the tasks, we adopted a binary classification approach for each lung condition, rather than evaluating models based on how many conditions were correctly identified per image. This decision was guided by clinical reasoning. Each lung condition represents a distinct diagnosis with its own implications for patient care, so it might be more meaningful to assess how accurately a model can detect each condition individually. For example, correctly identifying cardiomegaly or pneumothorax can be critical even if other conditions are missed, and vice versa. Evaluating per-condition performance allows us to understand model strengths and limitations for each diagnosis, which is likely more aligned with how these tools would be used in practice, in assisting radiologists in identifying specific lung conditions.

Given the extreme class imbalance, traditional accuracy is a poor metric for model evaluation. A model that naively predicts all cases as negative would still achieve high accuracy due to the overwhelming number of negative samples. For instance, in a hypothetical dataset where 99% of samples are negative and only 1% is positive, a model that always predicts samples to be negative for a condition would still have 99% accuracy, even though it would be completely ineffective at identifying positive cases. As such, accuracy can give a misleading impression of model performance in imbalanced datasets, particularly for the underrepresented class.

Average Precision (AP) score was selected as the primary evaluation metric because it provides a threshold-independent measure of model performance, summarising how well the model ranks positive cases across all possible thresholds. This is especially important in our setting, where many lung conditions have highly imbalanced distributions. Metrics such as precision, recall, and F1-score depend on selecting a fixed classification threshold, typically 0.5, but this threshold may not be optimal for all lung conditions or models. Given that each model, and even each lung-condition, may require a different optimal decision threshold, individual thresholds for each condition and multiple models would be impractical and inconsistent. Additionally, we also considered other threshold-independant metrics, such as ROC-AUC. However, ROC-AUC considers the true negative rate, which dominates when positive cases are rare, which led model evaluations to be overly optimistic. In contrast, AP focuses on how well the model ranks true positives, making it more suitable for our classification task.

Not all models were trained using the same imbalance correction techniques or evaluation metrics during training. The choice of strategy was tailored to each model's architecture and training workflow, and is detailed in the respective model-specific sections of this report.

### Dimensionality Reduction

High-dimensional embeddings derived from medical images often contain significant redundancy and noise. Directly utilization of those raw embeddings in downstream machine learning models can increase the risk of overfitting and impose high computational costs. Therefore, to enhance the generalizability of the downstream model and reduce computation cost, dimensionality reduction needed to be applied to the embeddings.

Thus, to investigate whether specific regions or patterns within the image embeddings and remove redundant information, we explored the use of Uniform Manifold Approximation and Projection (UMAP)-Regularized Autoencoder4. The intuition was that certain shapes, textures, or intensities associated with those pathologies may be captured by specific embedding dimensions by filtering out noise, redundancy. Autoencoders are neural networks that learn to represent data in a reduced-dimensional space, preserving essential information for accurate reconstruction. However, standard autoencoders might not preserve important geometric relationships between similar features due to the stochastic nature of training.

To address this limitation, we introduced Uniform Manifold Approximation and Projection (UMAP) as a regularizer in the autoencoder training. UMAP explicitly maintains local and global geometric structures by preserving neighborhood relationships among data points. By integrating UMAP regularization, the autoencoder not only retains critical information for reconstruction but also ensures that similar pathological cases remain close in the embedding space.

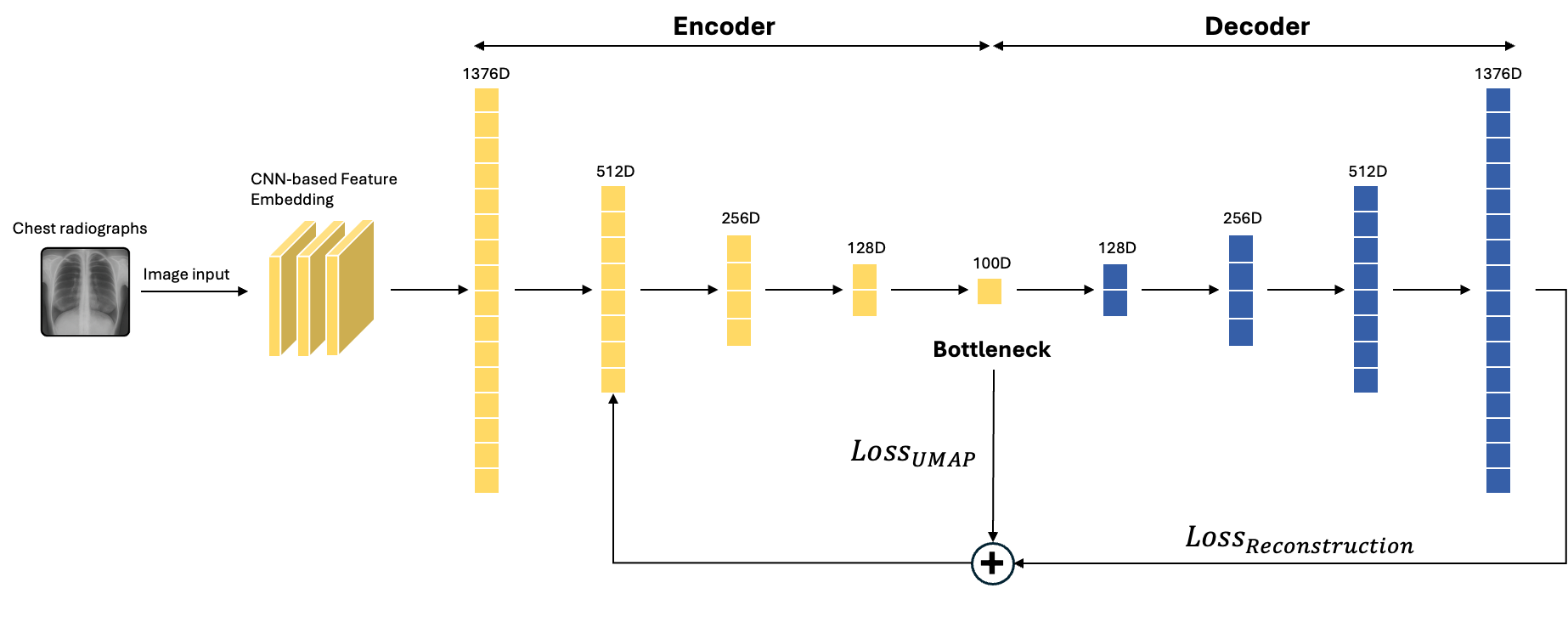
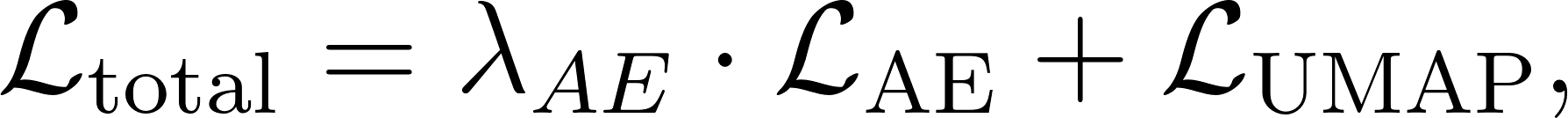
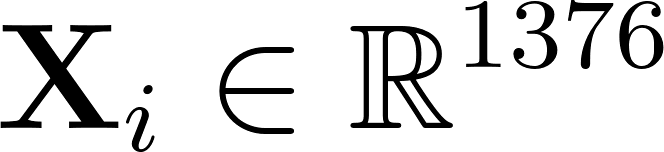
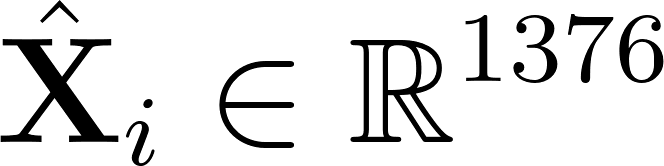
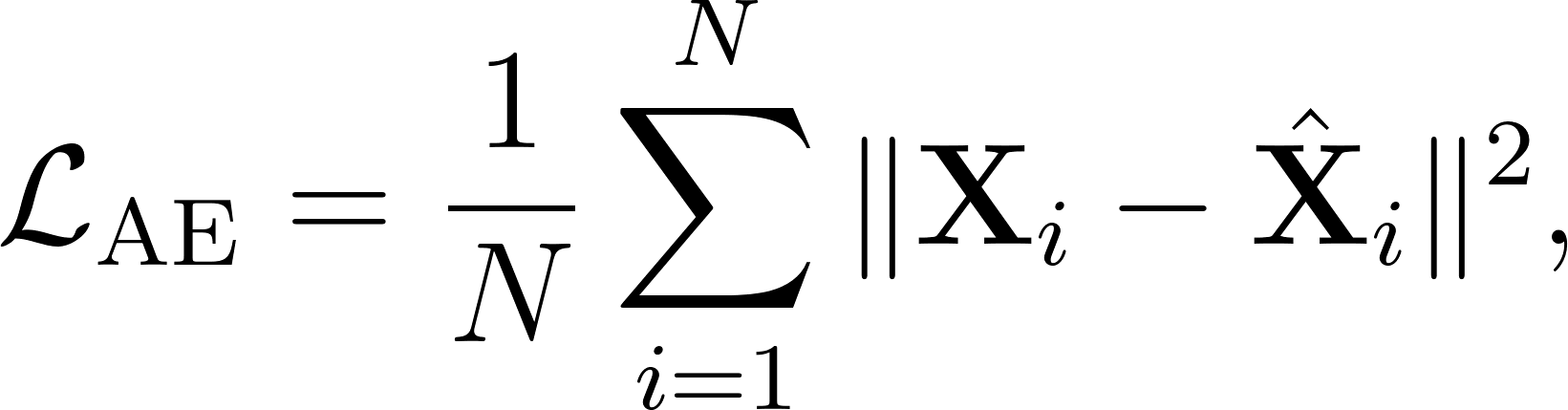


Figure 3: Diagram of the UMAP-Regularized Autoencoder

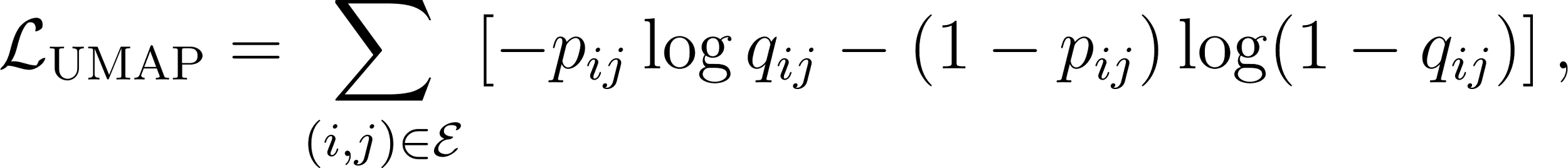
In general, the pipeline of UMAP-regularized autoencoder can be visualized using the diagram in Figure 3. Basically, the algorithm optimizes a joint loss function that combines the reconstruction loss of the autoencoder and the structural loss from UMAP. The total loss is defined as:

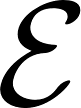
[](https://www.codecogs.com/eqnedit.php?latex=%5Cmathcal%7BL%7D_%7B%5Ctext%7Btotal%7D%7D%20%3D%20%5Clambda_%7BAE%7D%20%5Ccdot%20%5Cmathcal%7BL%7D_%7B%5Ctext%7BAE%7D%7D%20%2B%20%5Cmathcal%7BL%7D_%7B%5Ctext%7BUMAP%7D%7D%2C#0)

Where in our mode, [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathcal%7BL%7D_%7B%5Ctext%7BAE%7D%7D#0) is the mean squared error between the input raw image embeddings [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbf%7BX%7D_i%20%5Cin%20%5Cmathbb%7BR%7D%5E%7B1376%7D#0) and the reconstructed output [](https://www.codecogs.com/eqnedit.php?latex=%5Chat%7B%5Cmathbf%7BX%7D%7D_i%20%5Cin%20%5Cmathbb%7BR%7D%5E%7B1376%7D#0):

[](https://www.codecogs.com/eqnedit.php?latex=%5Cmathcal%7BL%7D_%7B%5Ctext%7BAE%7D%7D%20%3D%20%5Cfrac%7B1%7D%7BN%7D%20%5Csum_%7Bi%3D1%7D%5EN%20%5C%7C%20%5Cmathbf%7BX%7D_i%20-%20%5Chat%7B%5Cmathbf%7BX%7D%7D_i%20%5C%7C%5E2%2C#0)

and [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathcal%7BL%7D_%7B%5Ctext%7BUMAP%7D%7D#0) is the UMAP cross-entropy loss preserving local structure in the latent space:

[](https://www.codecogs.com/eqnedit.php?latex=%5Cmathcal%7BL%7D_%7B%5Ctext%7BUMAP%7D%7D%20%3D%20%5Csum_%7B(i%2Cj)%20%5Cin%20%5Cmathcal%7BE%7D%7D%20%5Cleft%5B%20-p_%7Bij%7D%20%5Clog%20q_%7Bij%7D%20-%20(1%20-%20p_%7Bij%7D)%20%5Clog%20(1%20-%20q_%7Bij%7D)%20%5Cright%5D%2C#0)

with [](https://www.codecogs.com/eqnedit.php?latex=p_%7Bij%7D#0) and [](https://www.codecogs.com/eqnedit.php?latex=q_%7Bij%7D#0) denoting the high- and low-dimensional edge weights defined on total edge sets [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathcal%7BE%7D#0) respectively, and [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7BAE%7D#0) controlling the contribution to the importance of autoencoder.

Furthermore, we chose not to rely on tree‑based feature selection such as XGBoost and Random Forests. Such methods select a subset of the original 1,376 features based on marginal predictive power but ignore the nonlinear manifold structure inherent in the embeddings.

Before training our compression model, we first inspected the separability of the original 1,376‑dimensional embeddings in 2D. Using standard UMAP on the raw features (Figure 4a), we observed that positive and negative cases for each pathology form loose but discernible clusters despite the heavy class imbalance. To mitigate this skew, we applied SMOTE to oversample the minority class and then reran UMAP (Figure 4b). The SMOTE‑augmented projection shows noticeably tighter clusters and clearer boundaries between positive and negative samples across nearly every label, confirming that even a simple neighborhood‑preserving projection can reveal disease‑specific structure when class balance is improved.

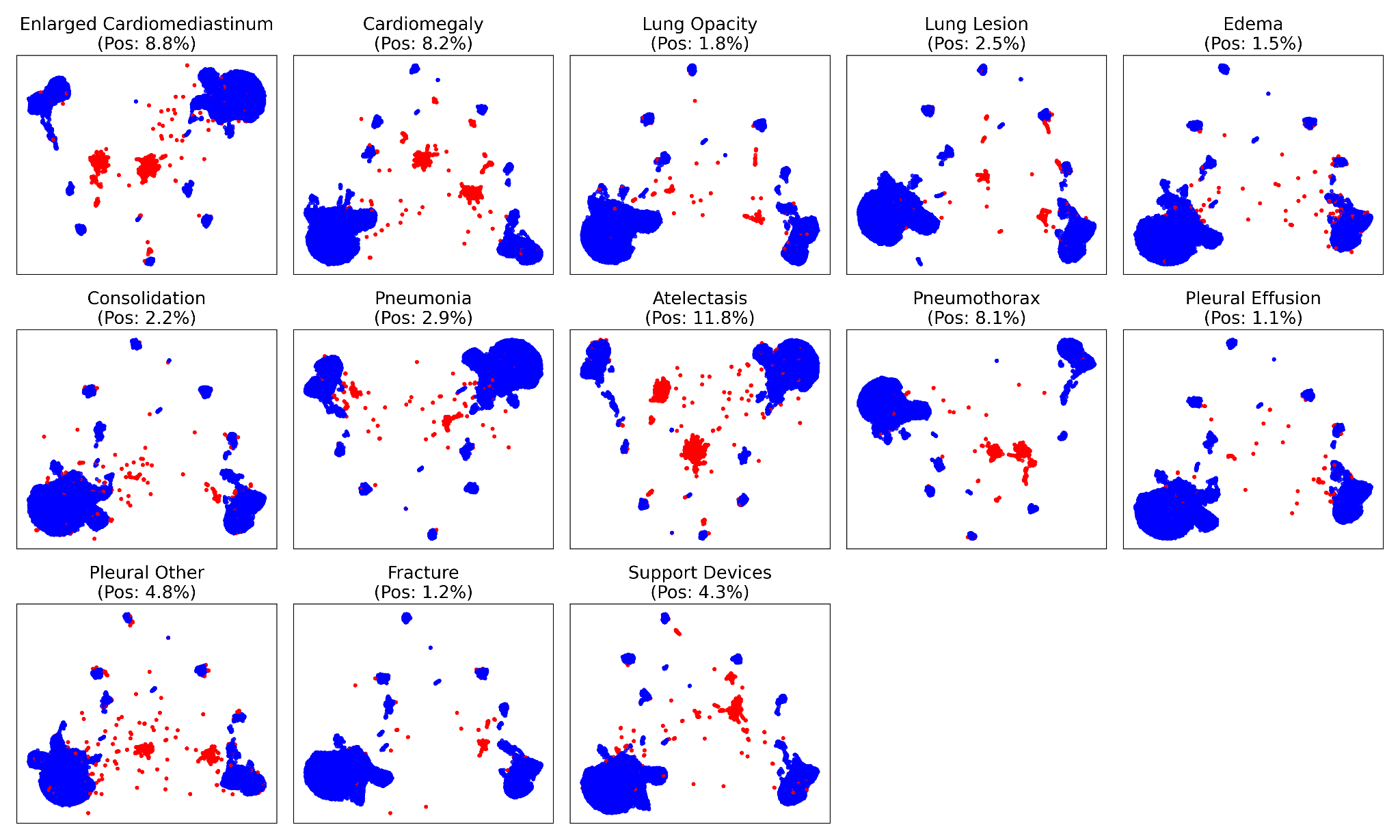
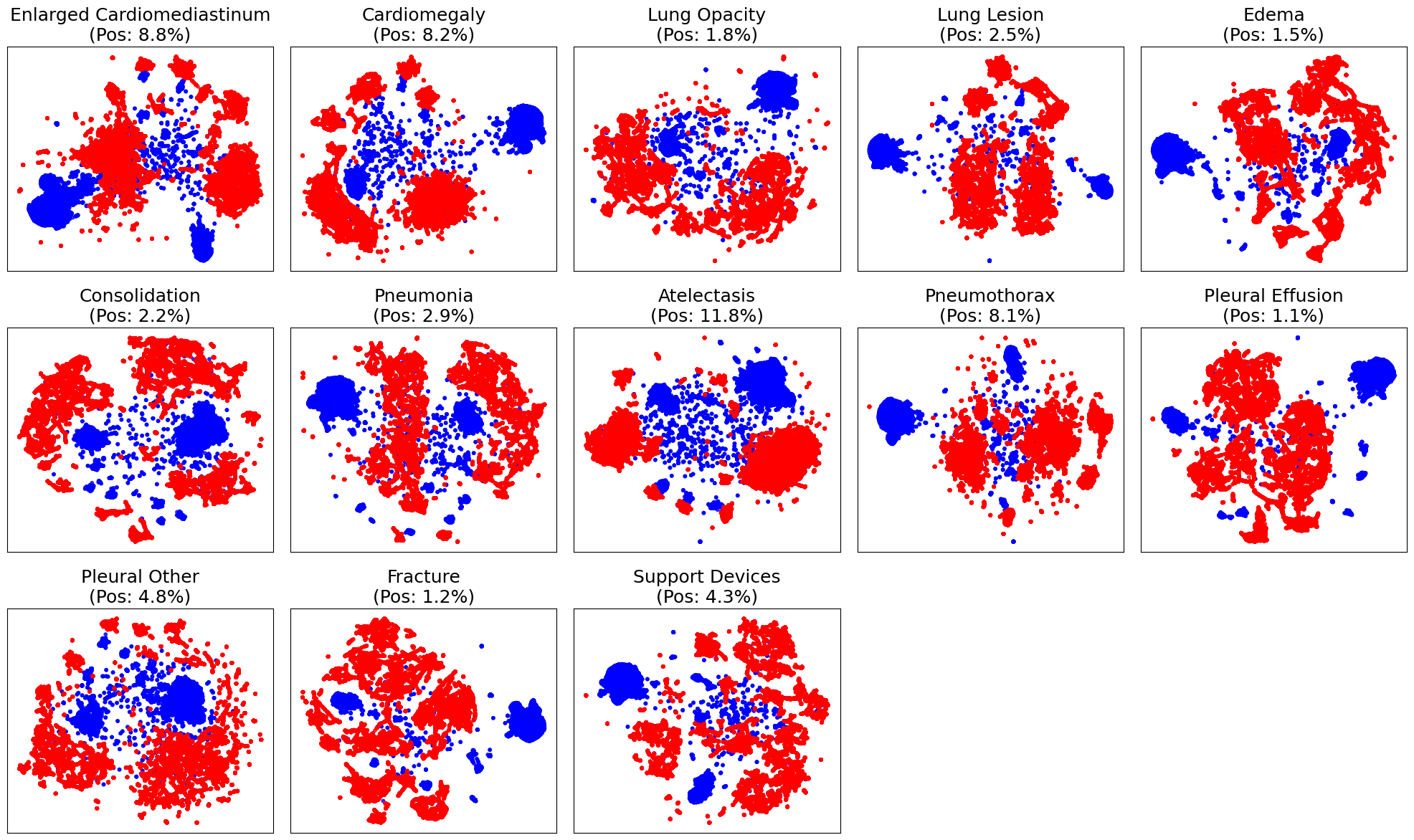
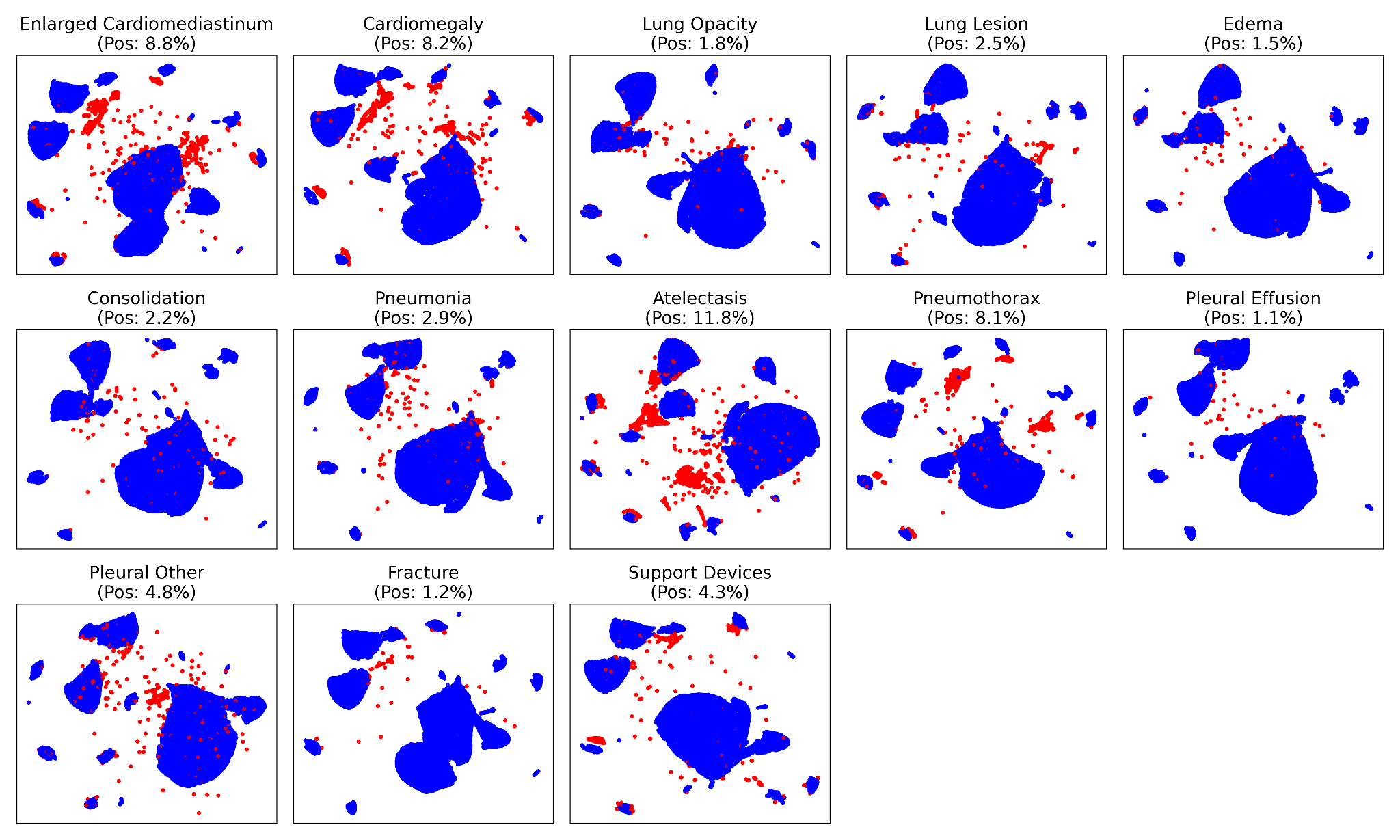
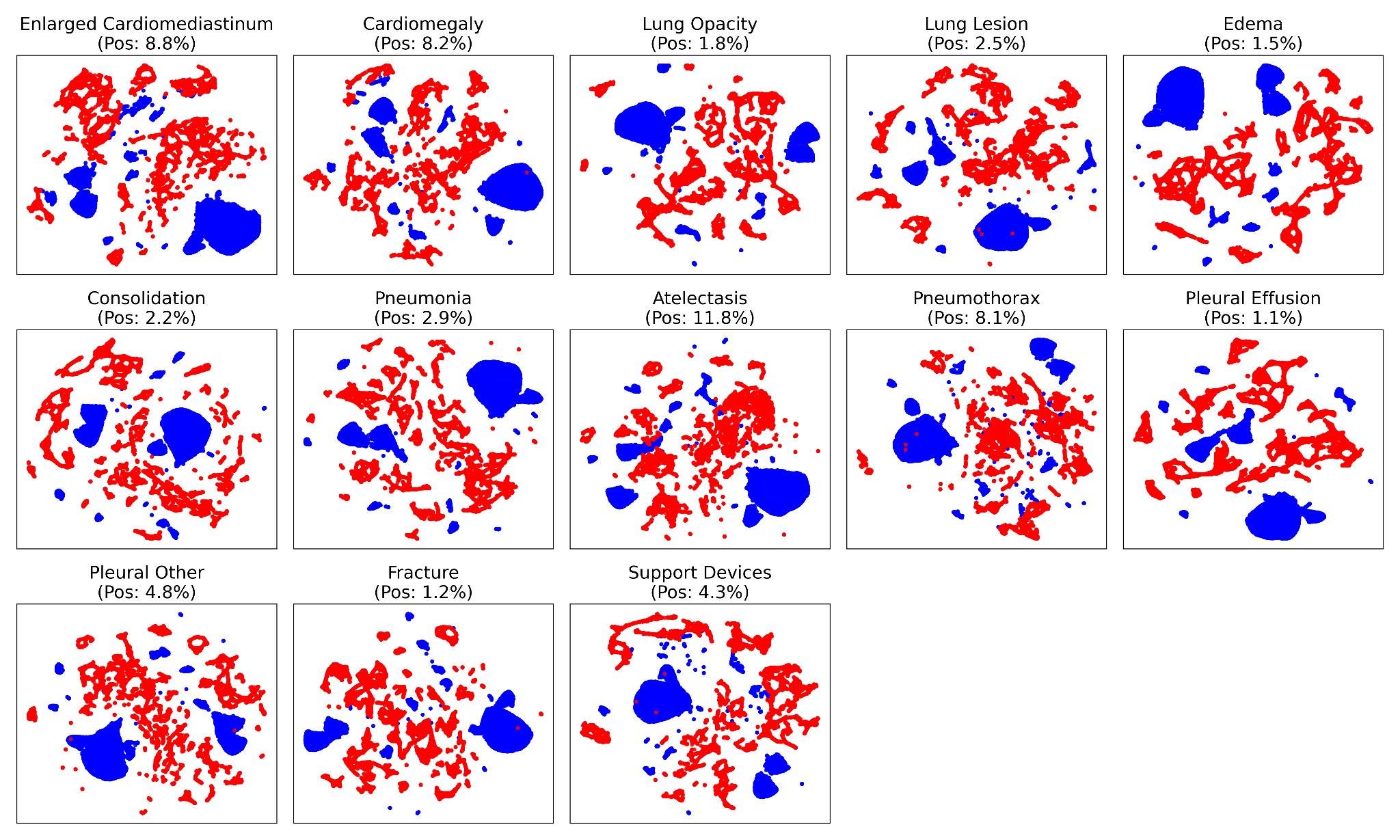
(a) Without SMOTE (b) With SMOTE

Figure 4: 2-D UMAP visualization of the raw 1376-dimensional embeddings

Next, we trained our UMAP‑Regularized Autoencoder on the training set. The encoder and decoder each comprised three dense layers (with hidden widths 512→128→100 and hidden widths 512->128->100), a UMAP loss weight of 0.5, a UMAP minimum distance of 0.1, and a batch size of 256. After fitting, we extracted the 100‑dimensional latent codes and applied a secondary, label‑wise UMAP to these codes for visualization (Figure 5). As we can tell from the two figures, it compares supervised UMAP projections of our 100-dimensional UMAP-AE before and after applying SMOTE. In Figure 5a, positive cases (red) for rare labels (e.g., Lung Opacity at 1.8%, Pleural Effusion at 1.1%) appear as isolated points scattered among the overwhelming blue negatives. Similarly like the pattern in Figure 3(a), though less discernible due to class imbalance, red points still circle a loose cluster shape. In contrast, Figure 5b shows that oversampling minority labels during autoencoder training yields tightly clustered red blobs for every condition, clearly delineated from negatives. These results illustrate that our UMAP-AE embedding retains the disease-relevant variation needed for classification, and that balancing rare classes with SMOTE helps the model learn a more expressive manifold. Hence, the emergence of coherent clusters justifies the existence of the low dimensional embedding of the high dimensional geometric pattern. Therefore, to further validate the effectiveness of our dimensionality reduction strategy, we conducted a comprehensive comparison of classifiers with the whole feature set and the dimension-reduced feature set.

(a) Without SMOTE (b) With SMOTE

Figure 5: 2-D UMAP visualization of the 100-dimensional embeddings after UMAP-Regularized Autoencoder

### Light Gradient Boosting Machine

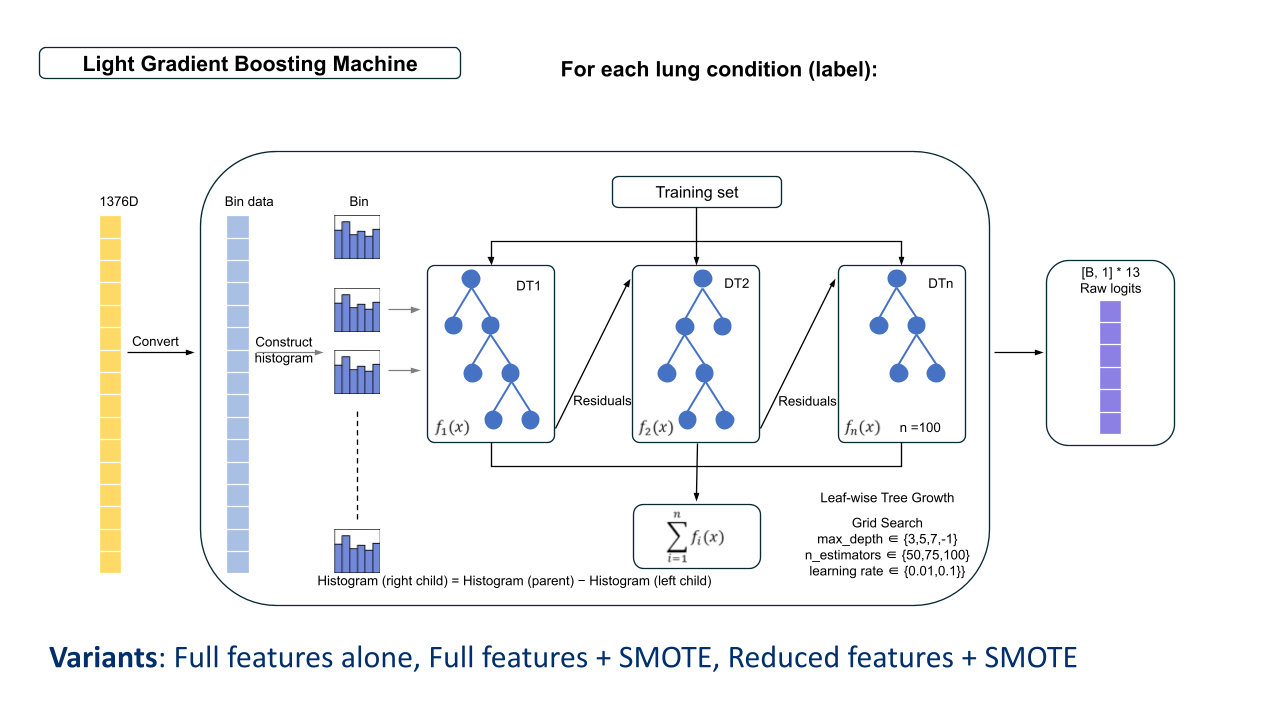


Figure 6: LightGBM model process overview

We explored the use of Light Gradient Boosting Machine (LightGBM) 5, a gradient boosting framework that uses tree-based learning algorithms, to perform multi-label classification on embedded CXR image data from the MIMIC-IV dataset. LightGBM was considered a suitable candidate for this task as the embedded CXR vectors, which were generated from pre-trained image models, are represented as high-dimensional numerical features. Unlike raw CXR images, the embeddings can be treated as tabular data, making them compatible with methods like LightGBM. Additionally, LightGBM’s efficiency and relatively fast training time made it feasible to conduct label-wise hyperparameter tuning across multiple classification tasks for each lung condition. Hence, our approach was to use a separate LightGBM model that was trained for each label, treating the presence or absence of each clinical finding as an independent binary classification problem. The overview of the LightGBM method is presented in Figure 6.

For each label, we conducted label-wise hyperparameter tuning, selecting the model that yielded the highest AP score on the validation set. The hyperparameter grid included learning\_rate ∈ {0.01, 0.1}, max\_depth ∈ {3, 5, 7, -1}, and n\_estimators ∈ {50, 75, 100}. The best-performing model for each label was then evaluated on a held-out test set. To investigate the impact of class imbalance and feature dimensionality, training and evaluation were repeated under three settings: (1) the full feature set with SMOTE applied, (2) the full feature set without SMOTE, and (3) a reduced feature set with SMOTE.

### Multi-Layer Perceptron

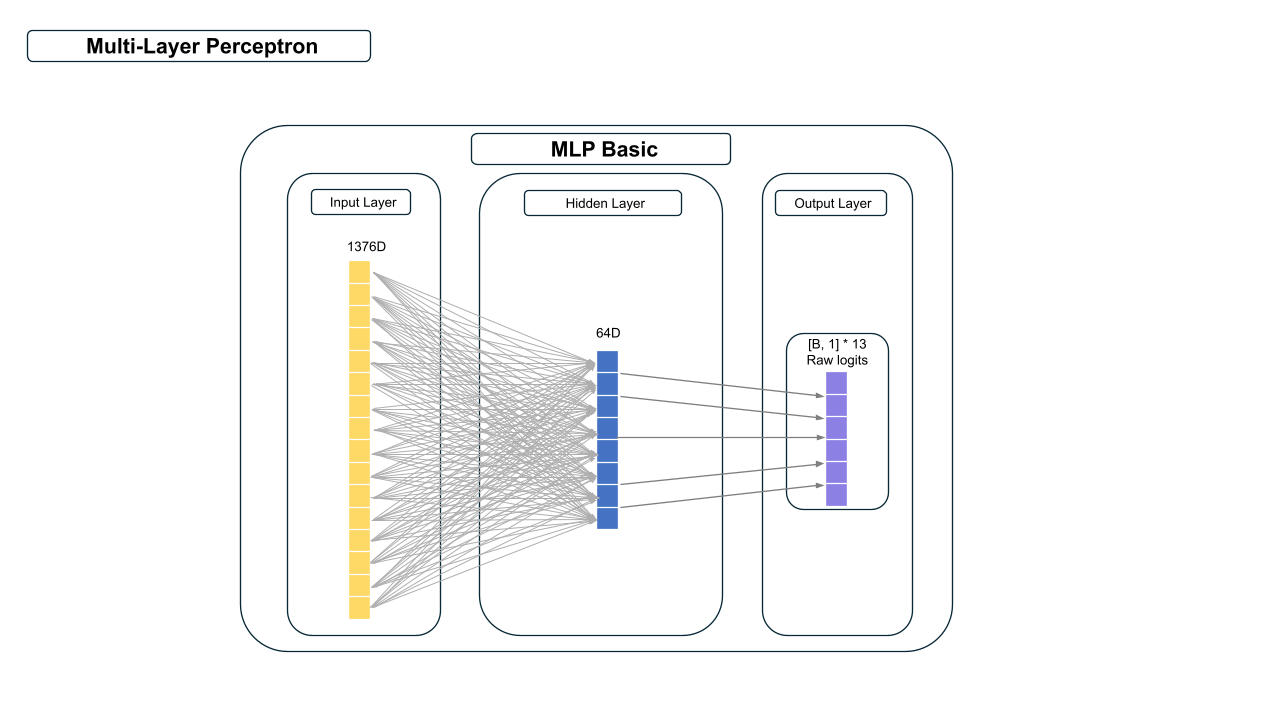


Figure 7: MLP process overview

A commonly used neural network, referred to as a Multi-Layer Perceptron (MLP) 6, was used to model the 13 labels using original features as well as a selected subset of features based on a UMAP method. This MLP is composed of multiple layers, including an input layer corresponding to the original size or selected subset feature size of 100, one hidden layer with 64 neurons with a Rectified Linear Unit (ReLU) activation function, and an output layer producing a separate prediction neuron for 13 target labels.

The network was trained using mini-batch gradient descent, drawing batches from a dataset that combined embeddings (inputs) with labels. Since some labels were missing, a masking routine excluded those entries from the loss function. During each training epoch, the model performed a forward pass on each mini-batch to produce predictions for all 13 labels, comparing them against valid (non-missing) labels through a Binary Cross Entropy with Logits7 objective. Individual label-specific losses were averaged before backpropagation. The Adam optimizer was employed over a maximum of 50 epochs, and an early stopping mechanism based on validation loss with a specified patience was introduced to curb overfitting. The overview of the MLP method is presented in Figure 7.

In addition to this basic setup, alternative variants were explored, including a more complex MLP composed of three successive hidden layers—first mapping the 1,376‑dimensional input to 256 units (ReLU), then to 128 units (ReLU), then to 64 units (ReLU)—before the final 13‑unit output layer; a weighted‑loss variant in which each label’s BCEWithLogitsLoss was initialized with a pos\_weight of 3.0 to up‑weight positive (minority) samples threefold during training, ensuring the model is not biased towards predicting the majority class; and a label-wise training approach using pre-mentioned SMOTE-based8 oversampling. After training, the models were switched to evaluation mode, with the output logits converted to probabilities via a sigmoid function and thresholded at 0.5 for binary predictions. Metrics such as accuracy, precision, recall, and average precision were computed separately for each label, to have an overview on how each version of the model managed the imbalanced nature of the data.

### Tabular Transformer

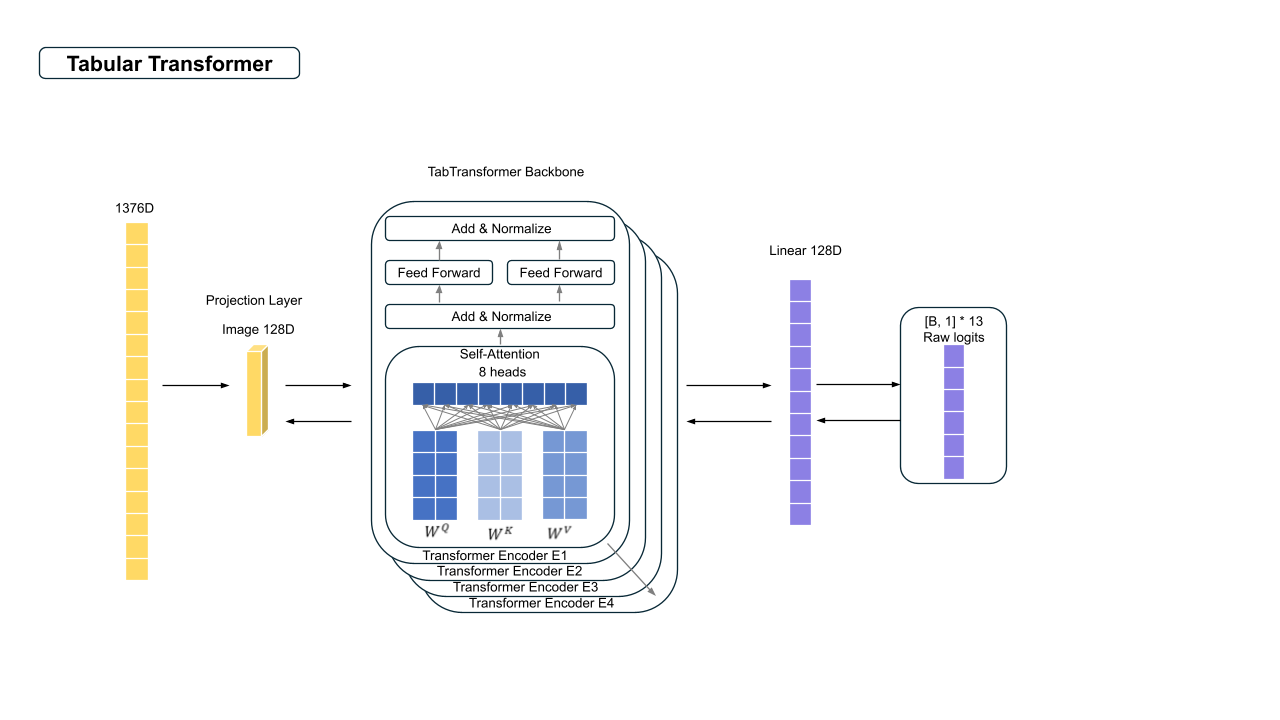
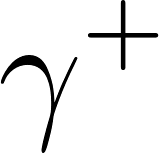
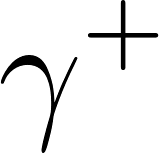
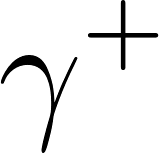


Figure 8: Tabular transformer overview

While several architectures were explored in this study, the TabTransformer was selected for detailed customization due to its capacity to model high-dimensional continuous features. The model projects the initial input image embedding vector into a lower-dimensional latent space via a linear transformation, followed by layer normalization to stabilize training. In the model, we stack four transformer encoder layers, each comprising eight-head self-attention and a gated feedforward network with GELU activations. Prior to entering the encoder stack, we apply Layer Normalization to stabilize feature distributions and improve convergence. The overview of the Tabular transformer method is presented in Figure 8. Label-wise models share the same architecture and training pipeline, but are optimized independently.

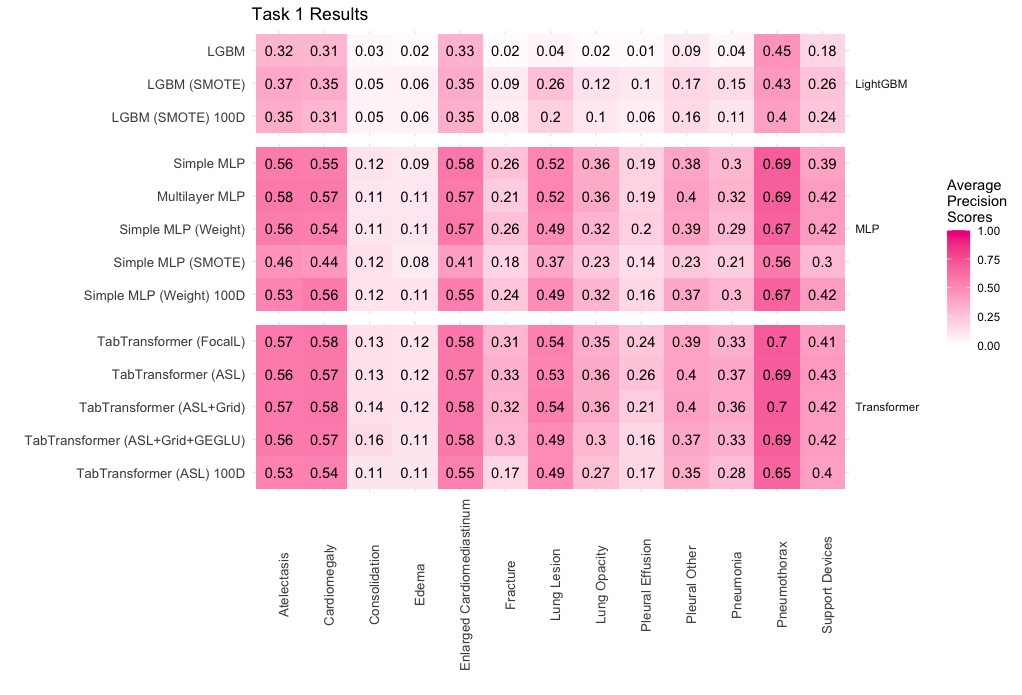
We conducted a series of design comparisons to determine the optimal model depth and attention width. Specifically, we evaluated encoder depths of 4, 8, and 16 layers. While increasing to 8 or 16 layers slightly increased model complexity, it did not yield consistent improvements in validation performance, and in some cases led to overfitting or training instability. Similarly, we tested 8-head and 16-head self-attention configurations, but found that 8 heads offered the best balance between representational power and computational efficiency. Therefore, we adopted a 4-layer, 8-head transformer configuration as our final architecture. Notably, we experimented with several activation mechanisms within the feedforward module, including the default GELU (Gaussian Error Linear Unit), and GEGLU (Gated Linear Unit with GELU activation), which theoretically offers advantages in non-monotonicity and smooth gradients.

Given the pronounced class imbalance in clinical diagnostic labels, we explored two loss functions designed to improve learning from rare positive cases: Masked Focal Loss (FocalL) and Masked Asymmetric Loss (ASL). While both approaches aim to address imbalance by emphasizing harder-to-classify examples, ASL introduces two tunable hyperparameters, [](https://www.codecogs.com/eqnedit.php?latex=%5Cgamma%5E%7B%2B%7D#0) and [](https://www.codecogs.com/eqnedit.php?latex=%5Cgamma%5E%7B-%7D#0), to asymmetrically adjust the penalization of false negatives and false positives. In our attempts, we compared the performance of TabTransformer models trained with FocalL and with ASL. The model trained with ASL consistently showed more stable and favorable results across multiple labels. In addition, a binary mask was applied during training to ignore missing or undefined labels, ensuring that the model focused only on valid entries for each instance. This masked formulation is particularly well-suited for medical datasets where partial labeling is common. Based on these findings, we adopted Masked ASL as the final loss function due to its superior robustness and compatibility with the clinical data setting.

Hyperparameter tuning was performed individually for each diagnostic label by conducting a grid search over ASL parameters [](https://www.codecogs.com/eqnedit.php?latex=%5Cgamma%5E%7B%2B%7D#0)∈ {0.0, 0.5, 1.0} and [](https://www.codecogs.com/eqnedit.php?latex=%5Cgamma%5E%7B-%7D#0)∈ {2.0, 3.0, 4.0, 5.0, 6.0}. For each hyperparameter combination, models were trained for 8 epochs on the training set, and the Average Precision (AP) on the validation set was used to select the optimal configuration. This training schedule was used exclusively for parameter selection. The final models were subsequently retrained using the best ([](https://www.codecogs.com/eqnedit.php?latex=%5Cgamma%5E%7B%2B%7D#0), [](https://www.codecogs.com/eqnedit.php?latex=%5Cgamma%5E%7B-%7D#0)) settings.

## Results

We used average precision (AP) score as the primary performance metric for model evaluation (Figure 9), as it is well-suited for imbalanced datasets and provides a summarised view of precision-recall performance across decision thresholds. Additional metrics including accuracy, precision, recall, F1-score were computed and are included in the appendix for completeness (Figure A3 - A6).

Figure 9: Average Precision Scores for models used in tasks 1. Models with the suffix “100D” refer to models trained with the reduced feature set that was determined from the UMAP-regularised autoencoder. SMOTE, Weight, ASL and FocalL refer to the different correction techniques used. Grid refers to the use of a grid search to select suitable weights for the loss function. GEGLU refers to the use of the gated linear unit in addition to GELU activation.

LightGBM models generally demonstrated lower performance, with AP scores under 0.5 across all lung conditions. Better performance was observed for Atelectasis, Cardiomegaly, Enlarged Cardiomediastinum, and Pneumothorax, which are all conditions with relatively higher prevalence in the dataset (above 8%, see Fig. 1). The application of SMOTE yielded slight improvements in AP scores, particularly for Lung Lesion, where the increase was most notable. However, when dimensionality reduction was combined with SMOTE, performance saw a marginal drop compared to using the full feature set, though it remained higher than the baseline model without imbalance correction or feature reduction.

MLP models consistently outperformed LightGBM across all lung conditions. Similar to LightGBM, MLP achieved better results on Atelectasis, Cardiomegaly, Enlarged Cardiomediastinum, Pneumothorax, and additionally on Lung Lesion, with AP scores frequently above 0.5. Particularly, Pneumothorax detection was strong, with AP scores between 0.6-0.7. Improvements for conditions like Consolidation and Edema remained minimal. Increasing model depth (via a multilayer MLP) only led to minute performance gains. The use of imbalance correction techniques showed mixed results. Model weighting yielded inconsistent effects, while SMOTE generally led to a drop in AP scores. When the weighted MLP was used with reduced features, the performance slightly decreased, though the drop in AP scores was limited.

TabTransformer models produced results comparable to the MLPs across all conditions, and slightly outperformed MLP in detecting Pneumothorax, with AP scores between 0.65–0.7. Experiments with different loss functions (focal loss, asymmetric loss) and activation mechanisms (GELU, GEGLU) resulted in only minimal variation in performance. The version using asymmetric loss was evaluated with a reduced feature set, leading to small decreases in AP across most conditions. However, a notable drop was observed for Fracture, where the AP score fell from 0.33 to 0.17.

## Assessing performance for Task 1 models

LightGBM consistently underperformed across all configurations, even with dimensionality reduction and class imbalance corrections. One likely reason for this could be that it may not be optimal for learning from high-dimensional, highly correlated embeddings derived from CXR images. Methodologically, LightGBM discretizes continuous features into bins to accelerate training, which can lead to significant information loss when applied to dense embeddings. Moreover, as it evaluates each feature independently when splitting, it likely struggles to capture the complex inter-feature relationships present in image data. Initial trials with another model also using a tree-based gradient boosting method, XGBoost (not shown), also showed similar poor performance. Thus, it is possible that these embedded features may contain complex relationships or spatial patterns that gradient-boosted decision trees fail to capture effectively.

In contrast, both MLP and TabTransformer models outperformed LightGBM, suggesting that neural networks are better suited to the nature of embedded image data. This likely arises because neural architectures can better learn the interdependent complex features end-to-end via gradient-based learning, and are not constrained by the limitations of lightGBM mentioned above. Both the TabTransformer and MLP performed similarly, though TabTransformer occasionally had the edge over MLP for generally poor performing lung conditions such as Pleural Effusion and Pneumothorax.

Various attempts were made to improve model performance through imbalance correction techniques, tuning, and feature selection, but these yielded limited success. Within the MLP models, the simple MLP and multilayer MLP achieved comparable performance. This suggests that the additional complexity introduced by deeper architectures did not yield substantial benefits, possibly because the embedded features already provided a low-complexity representation.

When comparing the effects of different imbalance correction techniques, SMOTE improved performance in the LightGBM models but not in the simple MLP. This discrepancy could be due to how each model handles synthetic data. For tree-based methods like LightGBM, the additional synthetic minority class samples generated by SMOTE might have helped refine decision boundaries during the tree-splitting process. In contrast, MLPs in high-dimensional settings could have been more sensitive to the quality of the training data, and thus, the interpolated samples introduced by SMOTE might introduce artefacts or noise that interfered with performance. In addition, SMOTE required label-wise training, thus 13 separated models were built, which may ignore the interaction between the labels, which would have hurt MLP performance. Instead, a weighted-loss variant, where positive (minority) samples were up-weighted during training, resulted in better results compared to using SMOTE. However, even this approach did not consistently improve performance across all lung conditions, and AP scores usually dipped compared to no imbalance correction. Similarly for TabTransformer, minimal differences in AP scores were detected despite testing different loss functions (focal, asymmetric) and activation functions (GELU vs. GEGLU). performance gains remained marginal.

Using the reduced feature set also did not improve performance. In fact, we generally observed a slight performance drop. While dimensionality reduction was motivated by the idea that only specific regions of a CXR carry meaningful disease signals, the full embedding may have retained more useful contextual information or patterns that are not easily isolated in some conditions, which could explain larger drops in AP scores in some instances. Collectively, these findings suggest that the primary bottleneck may not lie in the model architecture, imbalance correction strategy, or the large feature set.

The persistent difficulty in classifying certain lung conditions across the MLP and TabTransformer models suggests that the limitations may stem more from the quality of the feature representations than from the modelling approach itself. When the embedded features do not strongly distinguish between positive and negative cases, imbalance correction techniques or using different model architectures are unlikely to yield significant improvements. This challenge is compounded when the number of positive samples is small, making it harder for the model to learn consistent patterns. However, low prevalence alone does not entirely explain poor performance. For instance, Lung Opacity had a low positive prevalence (1.78%) but achieved a relatively better AP scores (~0.35) with both the simple MLP and TabTransformer models, outperforming conditions like Consolidation (2.16%) and Edema (1.51%), which consistently scored below 0.2. This suggests that the embedded representations for Lung Opacity may contain more distinctive features, allowing models to differentiate it more reliably despite limited data.

Altogether, this indicates that the problem is two-fold: limited positive examples reduce the statistical power available for learning, while weak or ambiguous representations within the image embeddings can further obscure differences between classes. This pattern is reflected in the performance across all Task 1 models. Conditions such as Atelectasis (11.8% positive), Cardiomegaly (8.17% positive), Enlarged Cardiomediastinum (8.83% positive), and Pneumothorax (8.14% positive) consistently achieved higher AP scores. These labels were both more prevalent and, presumably, easier to distinguish based on their visual features in the chest X-rays. Additionally, conditions like Consolidation and Edema, which performed poorly, may have poor distinguishability in the image embeddings. These conditions may also present more subtly in the PA-view images used. Thus, the visual signs of these conditions might not be well captured by the image encoders used to generate the embeddings. Future work could explore whether incorporating additional view positions would improve the discriminability of these conditions.

# Task 2: Multi-label classification from radiology reports

## Methods

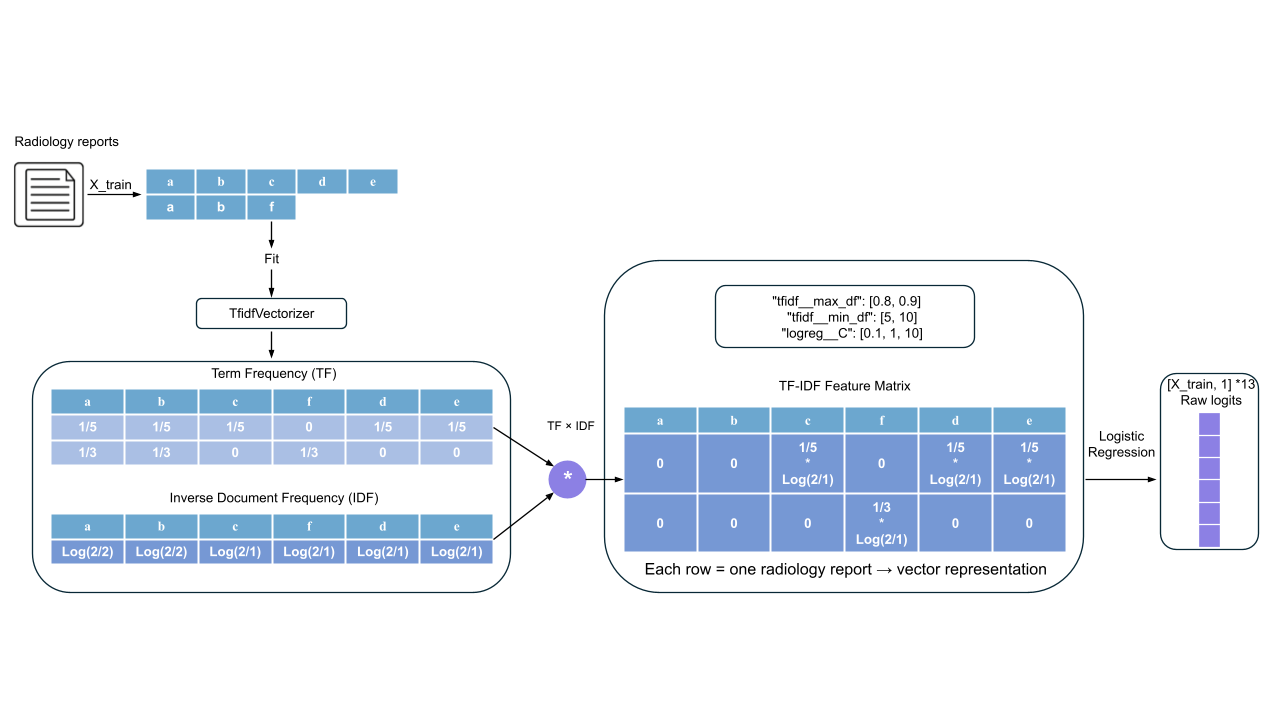


Figure 10: Diagram of NLP model framework

First, a function was implemented to link the different MIMIC data, where the only corresponding radiology reports for the image embedding data used in Task 1 were selected to ensure comparability.

These text data were then transformed using Term Frequency–Inverse Document Frequency (TF-IDF)9 vectorization, which converts free text into numerical feature vectors by multiplying the raw count of term in document, by the logarithm of the ratio of the total number of documents in the corpus, to the number of documents containing that term, thus emphasizing words that are frequent in a given report but rare across the corpus10. Compared with other vectorization methods, such as word2vec, doc2vec or using TF alone, TF-IDF usually performs better, specifically in book review sentiment classification, and thus it was selected for our analysis10.

The data was similarly split into training, validation, and test sets with a 8:1:1 ratio; afterwards, for each logistic regression model, a pipeline was built to chain the TfidfVectorizer with a LogisticRegression estimator. We wrapped this pipeline in a GridSearchCV (5‑fold CV) and optimized for average precision score to focus on the precision–recall trade‑off. The grid spanned maximum document frequency (tfidf\_max\_df) in [0.8, 0.9], minimum document frequency (tfidf\_min\_df\_ in [5, 10], thereby removing both very common tokens that appear in more than 80–90% of documents and very rare tokens that appear in fewer than 5–10 documents. This was done to limit noise from stop‑word‑like terms and infrequent words, while still capturing meaningful bi‑grams when present. In addition, the logistic regularization constant (logreg\_C) was varied in [0.1, 1, 10], to select the suitable training regularisation strength for each model. By selecting the parameter combination that maximized AP score on the validation folds, we ensured the final model balanced sensitivity and specificity in this imbalanced multi‑label setting. After model tuning, predictions were obtained on the test set to compute metrics such as accuracy, precision, recall, and F1 score.

## Results

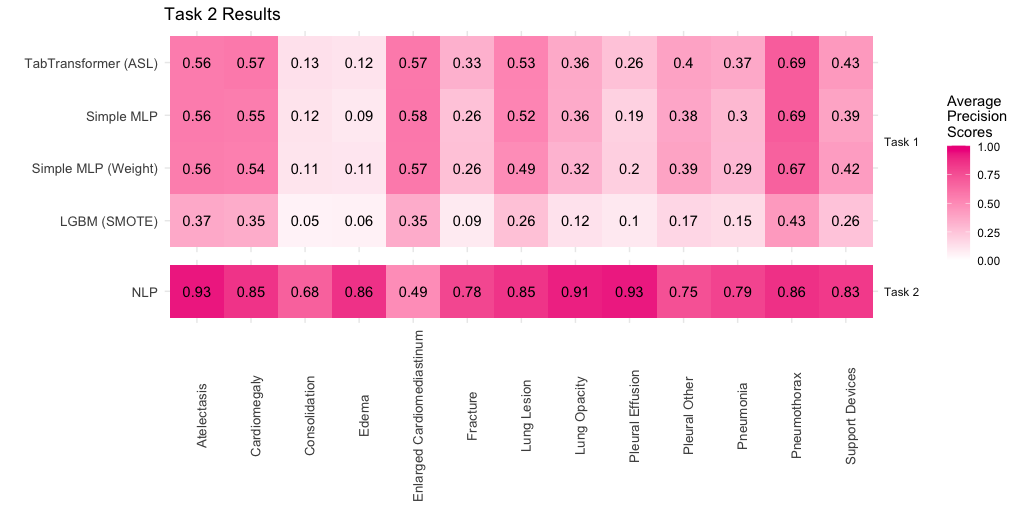


Figure 11: Average Precision Scores for NLP task in Task 2. Comparison with a subset of models from Task 1 are shown here, full comparison of all the models in Task 1-3 is provided in the Appendix (Figure A2).

Models trained on free-text radiology reports performed substantially better than the models trained on the generalised image embeddings of the CXR data across almost all lung conditions (Figure 11). AP scores were ≥ 0.75 for all labels except Enlarged Cardiomediastinum, where the NLP model underperformed compared to MLP and TabTransformer models. Nonetheless, for traditionally more challenging conditions like Consolidation, Edema, and Pleural Effusion, which had AP scores between 0.1-0.3 in Task 1 (Figure 9, 11), the NLP model achieved scores exceeding 0.65, marking a clear improvement.

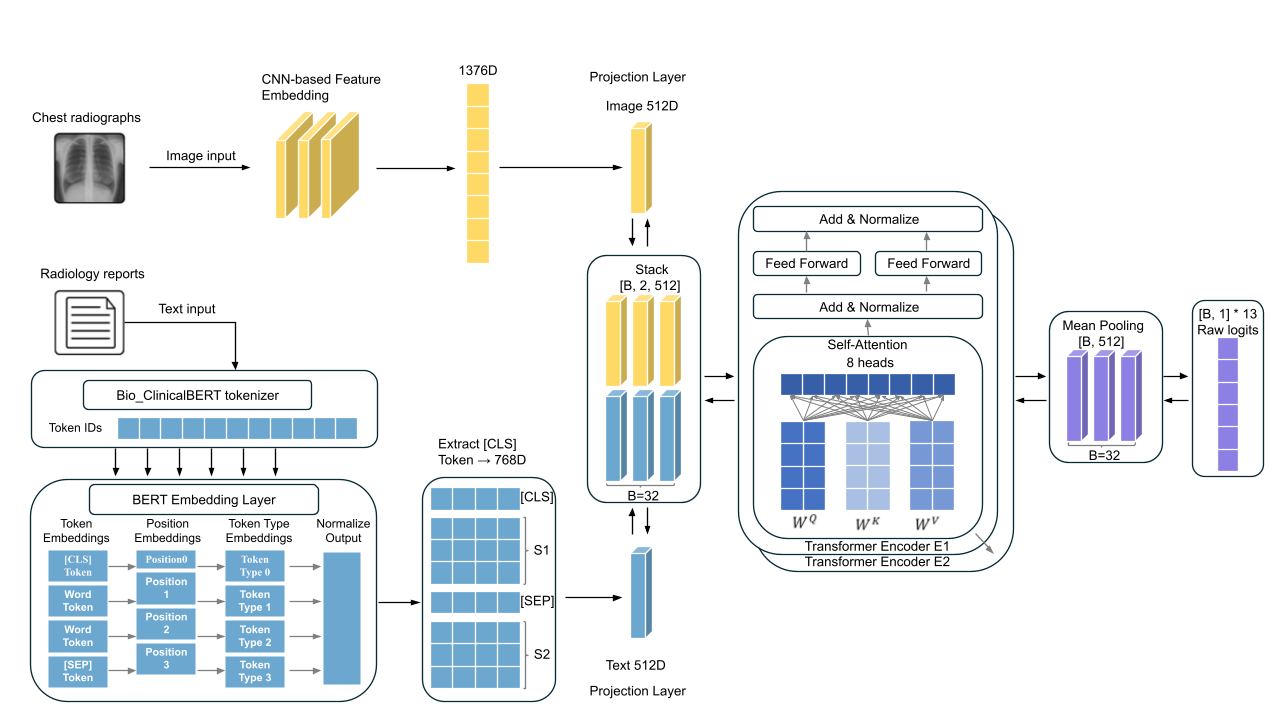
## Assessing performance for Task 2

The NLP model trained on free-text radiology reports achieved much better performance, with higher average precision scores for nearly all conditions compared to Task 1, except for Enlarged Cardiomediastinum. Given that the radiology report itself was used for label generation, this strong performance is expected. NLP models can leverage descriptive language that directly identifies or implies disease presence, especially for conditions that may be subtle in the CXR images. The relatively poor performance for Enlarged Cardiomediastinum is surprising, though it is possible that relevant phrases were missed during token processing, or that the condition was implied but not explicitly stated in the text.

# 

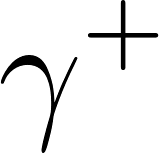
# Task 3: Multi-modal model

## Methods

Figure 12: Diagram of Multimodal model framework

To jointly leverage the image embedding data and the radiology report texts, we designed a multimodal deep learning model. Our dataset consisted of the image embeddings used in Task 1. Unlike Task 2, the textual inputs were preprocessed using the Bio\_ClinicalBERT tokenizer (instead of TF-IDF) with a maximum sequence length of 128. Among several pretrained biomedical transformers including standard BERT11, BioBERT12, and Bio\_ClinicalBERT13 variants, Bio\_ClinicalBERT was empirically selected due to its superior stability and downstream performance during preliminary comparisons. And since it is fine-tuned from BioBERT based on MIMIC-III clinical notes, the model should perform better for clinical-specific task. The final tokenized embeddings with special tokens like [CLS] and [SEP] output from the frozen Bio\_ClinicalBERT model was extracted as a 768-dimensional semantic representation of each report.

To combine image and text features, we first mapped each into a shared 512-dimensional space using separate linear layers. These were stacked into a tensor of shape [B, 2, 512] and fed into a two-layer TransformerEncoder (with GELU, 8 attention heads, and a 1024-dim feedforward layer) to capture cross-modal interactions. We then averaged the output across the two modalities to get a single feature vector, which was passed through a linear layer to predict the pathologies.

Training was conducted using the AdamW optimizer (learning rate =1e-4, batch size = 32) and a Masked Asymmetric Loss function ([](https://www.codecogs.com/eqnedit.php?latex=%5Cgamma%5E%7B%2B%7D#0) = 0.0, [](https://www.codecogs.com/eqnedit.php?latex=%5Cgamma%5E%7B-%7D#0) = 4.0) is used, which prioritizes hard negatives and ignores missing labels, making the model more robust to label imbalance and annotation noise

During the training procedure, ClinicalBERT parameters were kept frozen throughout training to reduce computational cost and training time, while also isolating the impact of the fusion module; this allowed us to more clearly evaluate how effectively the model integrates visual and textual representations without additional influence from fine-tuning the language encoder.

## Results



Figure 13: Average Precision Scores for NLP task in Task 2. Comparison with a subset of models from Task 1, and NLP (Task 2) are shown here, full comparison of all the models in Task 1-3 is provided in the Appendix (Figure A2).

The multimodal model, which integrated both the embedded data of the CXRs and radiology reports, generally outperformed Task 1 models but underperformed compared to the NLP-only model in Task 2 (Figure 12, Figure A2). While lower AP scores (< 0.3) were observed for Edema, Fracture, and Pleural Effusion, the model performed notably well on Enlarged Cardiomediastinum, achieving an AP score of 0.61 and surpassing both the Task 1 and Task 2 models for this condition.

## Assessing performance for Task 3

In Task 3, the multimodal model integrating image embeddings and text data outperformed Task 1’s image-only models, but not Task 2’s text-only model, in most cases. This suggests that textual data contributed more strongly to accurate label prediction than the image embeddings. However, for Enlarged Cardiomediastinum, the multimodal model achieved the best performance, outperforming both single-modality approaches. This suggests that combining data sources may compensate for limitations in each.

It is also possible that using different tokenization schemes, TF-IDF in the text-only Task 2 versus a BERT tokenizer in Task 3, introduced differences in how textual features were represented, thereby partially influencing comparative performance. Nevertheless, because Task 3 enforced a strict 1:1 weighting between text and image modalities, it is more plausible that this rigid weighting scheme is the primary factor constraining the model’s ability to optimally leverage each source. Future work should therefore explore both harmonizing tokenization approaches and learning condition-specific modality weights to fully exploit the complementary strengths of text and image data.

# Overall conclusion

Our evaluation of image-only, text-only, and multimodal methods for lung condition identification on MIMIC-IV CXR data shows clear trade-offs between feature types and model choices. Image embeddings alone gave limited performance, especially for conditions that are subtle or lack clear patterns on X-rays. Neural networks like MLP and TabTransformer outperformed tree-based models, showing their strength in learning complex feature interactions. However, text-based models still performed better overall by using the clear information in radiology reports. The multimodal model helped in some cases, especially for Enlarged Cardiomediastinum, by combining strengths from both image and text. Still, fixed 1:1 feature weighting and separate tokenization processes may have held back further gains in Task 3. Future work should focus on incorporating different CXR viewpoints, standardizing text preprocessing, and learning condition-specific feature weights. Using more imaging views and smarter dimensionality reduction may also improve results. In short, balancing data quality and model flexibility is key to building stronger clinical tools.

# References

1. Sellergren A, Kiraly A, Pollard T, et al. Generalized Image Embeddings for the MIMIC Chest X-Ray dataset. doi:10.13026/PXC2-VX69

2. Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal Loss for Dense Object Detection. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE; 2017:2999-3007. doi:10.1109/ICCV.2017.324

3. Ridnik T, Ben-Baruch E, Zamir N, et al. Asymmetric Loss For Multi-Label Classification. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE; 2021:82-91. doi:10.1109/ICCV48922.2021.00015

4. Sainburg T, McInnes L, Gentner TQ. Parametric UMAP embeddings for representation and semi-supervised learning. Published online 2020. doi:10.48550/ARXIV.2009.12981

5. Ke G, Meng Q, Finley T, et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: *Advances in Neural Information Processing Systems*. Vol 30. Curran Associates, Inc.; 2017. Accessed April 25, 2025. https://proceedings.neurips.cc/paper\_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

6. Chan KY, Abu-Salih B, Qaddoura R, et al. Deep neural networks in the cloud: Review, applications, challenges and research directions. *Neurocomputing*. 2023;545:126327. doi:10.1016/j.neucom.2023.126327

7. BCEWithLogitsLoss — PyTorch 2.7 documentation. Accessed April 25, 2025. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

8. Matharaarachchi S, Domaratzki M, Muthukumarana S. Enhancing SMOTE for imbalanced data with abnormal minority instances. *Mach Learn Appl*. 2024;18:100597. doi:10.1016/j.mlwa.2024.100597

9. TfidfVectorizer. scikit-learn. Accessed April 25, 2025. https://scikit-learn/stable/modules/generated/sklearn.feature\_extraction.text.TfidfVectorizer.html

10. Dept. of Computer Science, Jigawa State Colledge of Education, Gumel, Nigeria, Abubakar HD, Umar M, Dept. of Computer Sceince, Faculty of Science, Sokoto State University, Sokoto, Nigeria. Sentiment Classification: Review of Text Vectorization Methods: Bag of Words, Tf-Idf, Word2vec and Doc2vec. *SLU J Sci Technol*. 2022;4(1 & 2):27-33. doi:10.56471/slujst.v4i.266

11. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Published online 2018. doi:10.48550/ARXIV.1810.04805

12. Lee J, Yoon W, Kim S, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Wren J, ed. *Bioinformatics*. 2020;36(4):1234-1240. doi:10.1093/bioinformatics/btz682

13. Alsentzer E, Murphy JR, Boag W, et al. Publicly Available Clinical BERT Embeddings. Published online 2019. doi:10.48550/ARXIV.1904.03323

# Appendix

|  |  |
| --- | --- |
|  |  |

Figure A1: Training and validation loss for MLP, before and after early-stopping

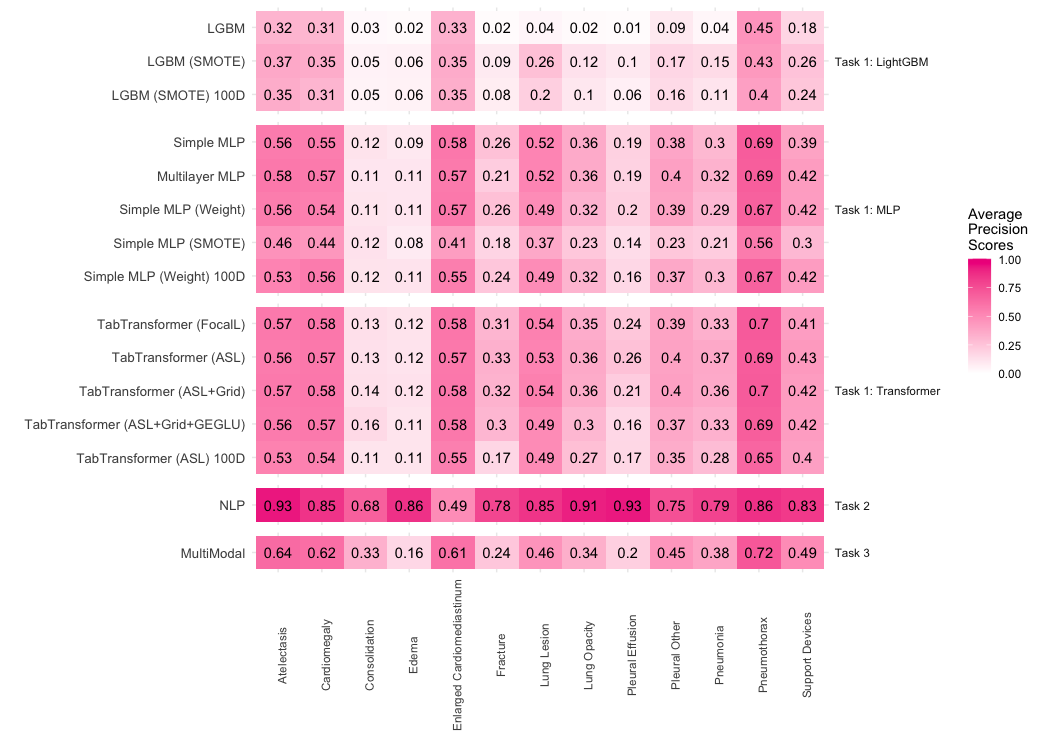


Figure A2: AP scores for models used in tasks 1-3. Models with the suffix “100D” refer to models trained with the reduced feature set that was determined from the UMAP-regularised autoencoder. SMOTE, Weight, ASL and FocalL refer to the different correction techniques used. Grid refers to the use of a grid search to select suitable weights for the loss function. GEGLU refers to the use of the gated linear unit in addition to GELU activation.

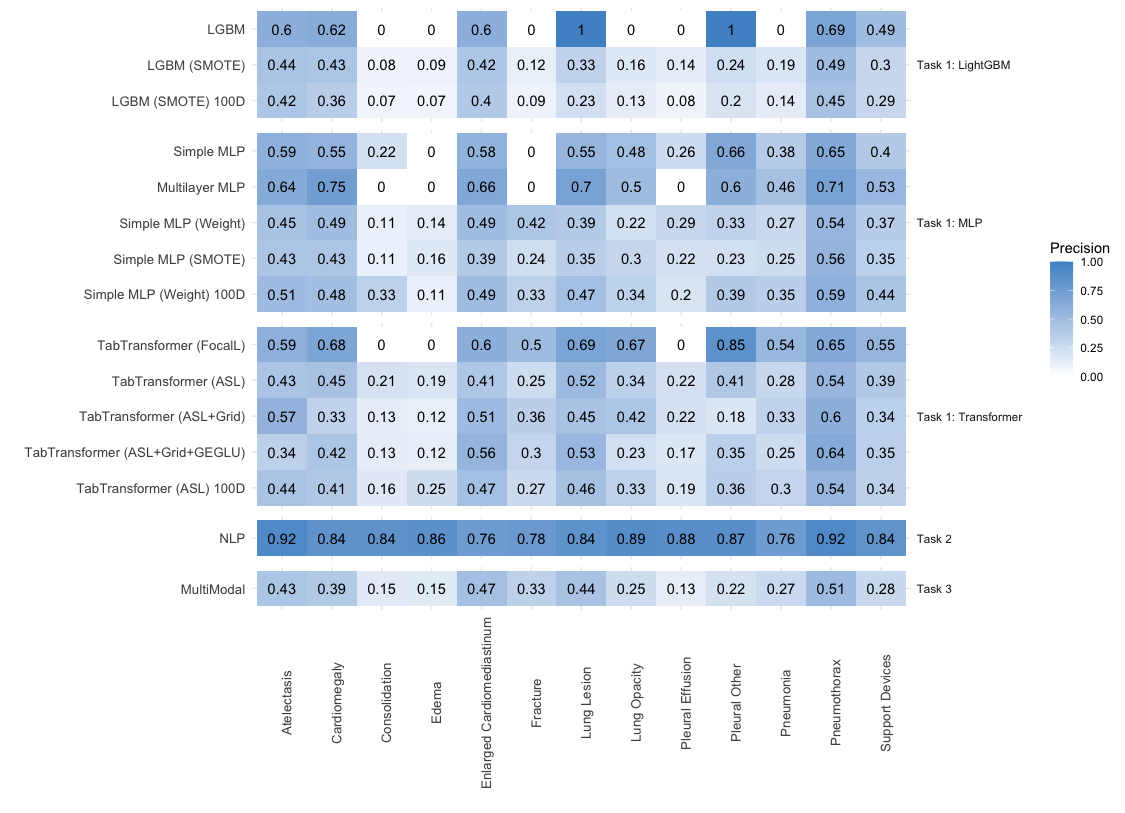


Figure A3: Precision for models used in tasks 1-3 for positive cases. Models with the suffix “100D” refer to models trained with the reduced feature set that was determined from the UMAP-regularised autoencoder. SMOTE, Weight, ASL and FocalL refer to the different correction techniques used. Grid refers to the use of a grid search to select suitable weights for the loss function. GEGLU refers to the use of the gated linear unit in addition to GELU activation.

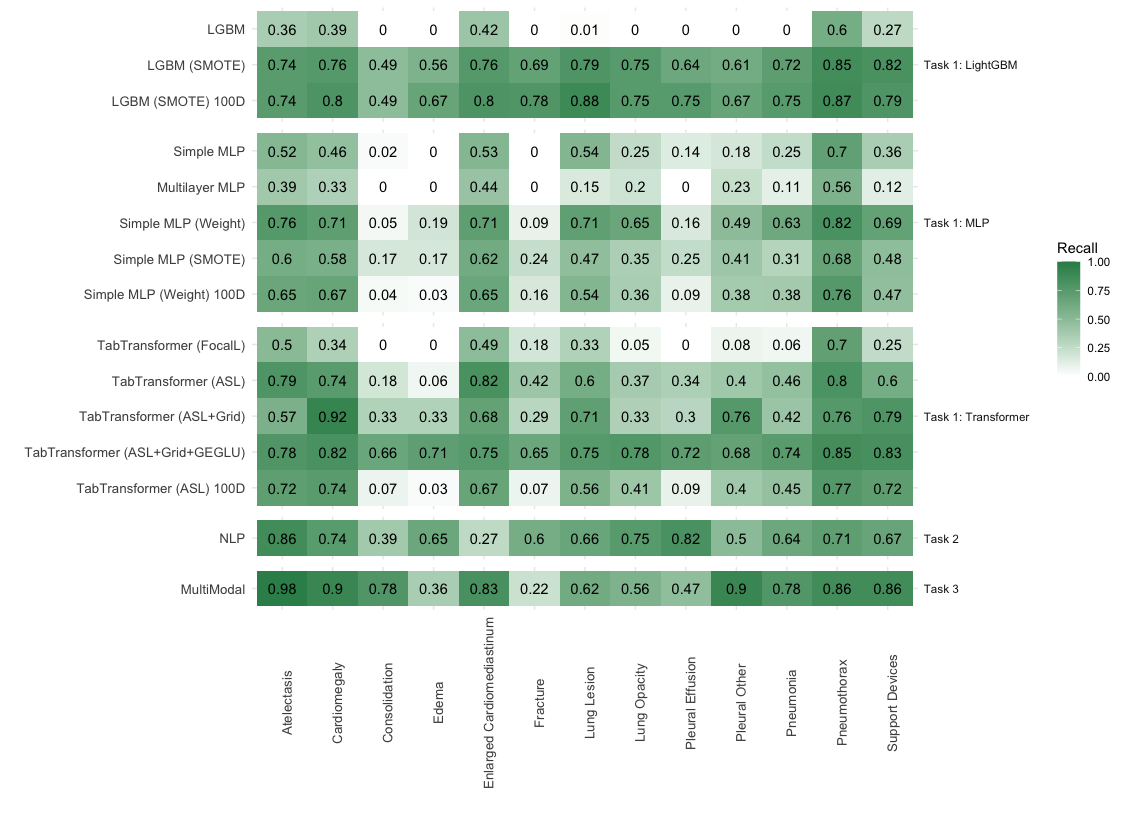


Figure A4: Recall for models used in tasks 1-3 for positive cases. Models with the suffix “100D” refer to models trained with the reduced feature set that was determined from the UMAP-regularised autoencoder. SMOTE, Weight, ASL and FocalL refer to the different correction techniques used. Grid refers to the use of a grid search to select suitable weights for the loss function. GEGLU refers to the use of the gated linear unit in addition to GELU activation.

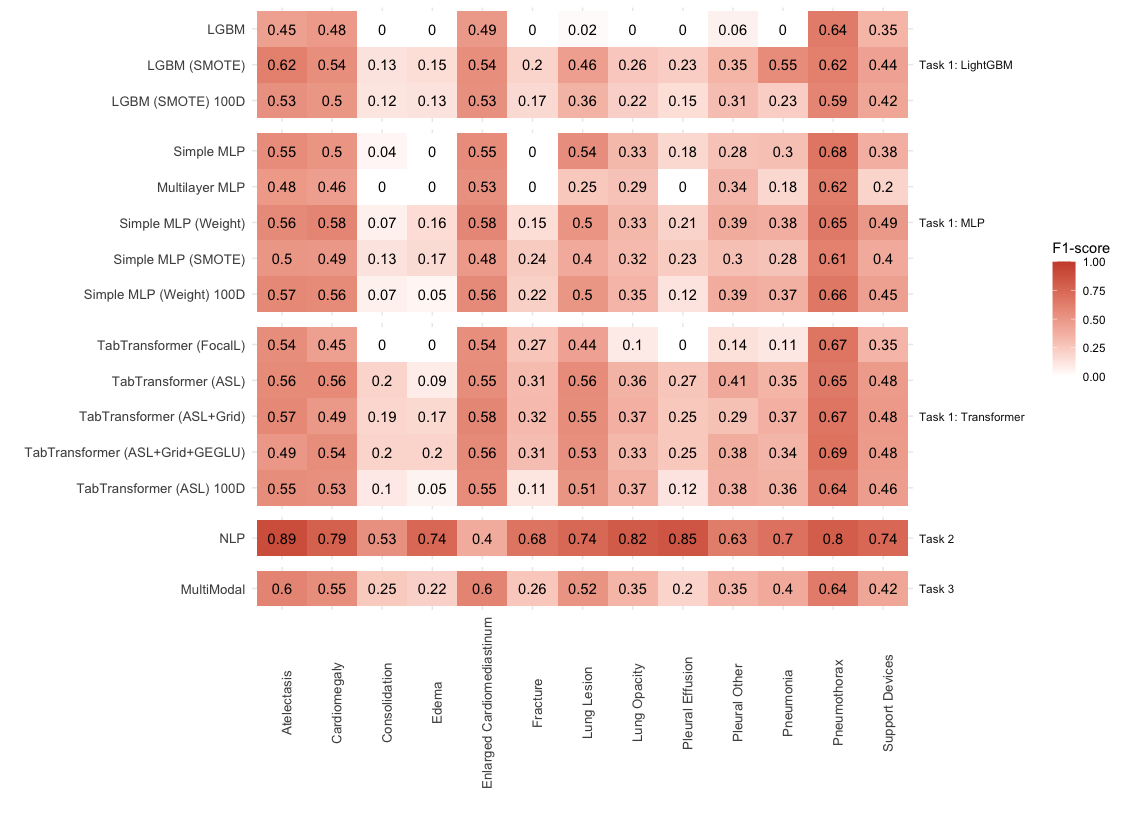


Figure A5: F1-score for models used in tasks 1-3 for positive cases. Models with the suffix “100D” refer to models trained with the reduced feature set that was determined from the UMAP-regularised autoencoder. SMOTE, Weight, ASL and FocalL refer to the different correction techniques used. Grid refers to the use of a grid search to select suitable weights for the loss function. GEGLU refers to the use of the gated linear unit in addition to GELU activation.

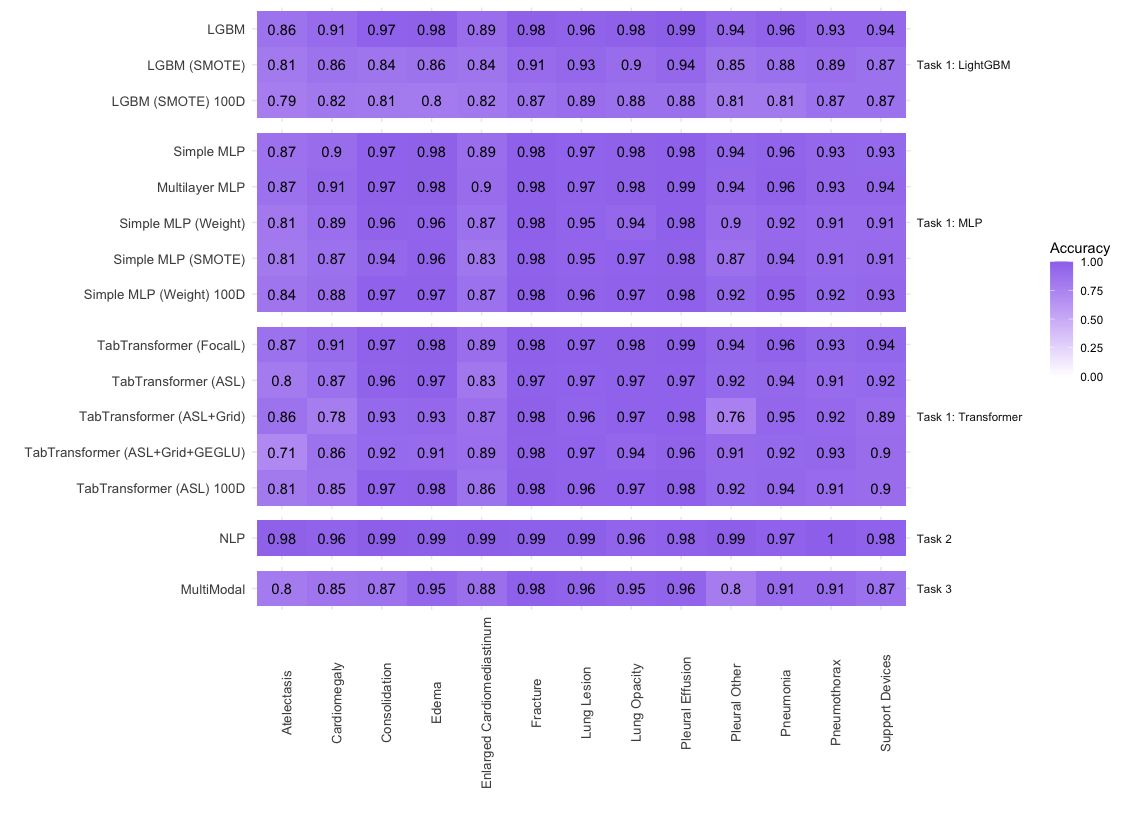


Figure A6: Accuracy for models used in tasks 1-3. Models with the suffix “100D” refer to models trained with the reduced feature set that was determined from the UMAP-regularised autoencoder. SMOTE, Weight, ASL and FocalL refer to the different correction techniques used. Grid refers to the use of a grid search to select suitable weights for the loss function. GEGLU refers to the use of the gated linear unit in addition to GELU activation.