THE UNIVERSITY OF CHICAGO

INVESTIGATION OF HEAD CT AI METHODS IN TRAUMATIC BRAIN INJURY INCLUDING SHORTCUT LEARNING EFFECTS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE BIOLOGICAL SCIENCES
AND THE PRITZKER SCHOOL OF MEDICINE
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMMITTEE ON MEDICAL PHYSICS

BY
JOEL FAUBRICIO TOLEDO-UREÑA

CHICAGO, ILLINOIS
DECEMBER 2026

© 2026 by Joel Faubricio Toledo-Ureña

ORCID iD: https://orcid.org/0000-0002-7251-759X

"Would you be more proud of accomplishing something that is difficult or something easy?"

-Dr. Joelle Murray

TABLE OF CONTENTS

LIST OF FIGURES vi

LIST OF TABLES ix

ACKNOWLEDGMENTS x

ABSTRACT xii

1 INTRODUCTION 1
1.1 Computed Tomography 2
1.2 Artificial Intelligence in Medical Imaging 3
1.2.1 Convolutional Neural Networks and Vision Transformers 4
1.2.2 AI Training Methods 6
1.2.3 Embeddings 8
1.2.4 Foundation Models 10
1.3 Machine Learning Algorithms Employed 11
1.3.1 U-Net 11
1.3.2 ResNet 13
1.3.3 RAD-DINO 15
1.3.4 CT-FM 16

2 DEEP LEARNING METHODS FOR BRAIN EXTRACTION ON PENETRATING AND NON-PENETRATING TRAUMATIC BRAIN INJURY CT SCANS 20
2.1 Introduction 20
2.2 Methods 22
2.2.1 Data collection and Image processing 22
2.2.2 Manual Segmentation 24
2.2.3 Deep Learning U-Net algorithms 24
2.2.4 Statistical Analysis 26
2.3 Results 26
2.4 Discussion 28
2.5 Conclusion 30

3 DEVELOPMENT OF CLUSTER ANALYSIS METHODS IN THE IDENTIFICATION OF SHORTCUT INFLUENCES 32
3.1 Introduction 32
3.2 Databases 34
3.2.1 Iris Dataset 34
3.2.2 TBI Dataset 34
3.3 Methods 35
3.3.1 Clustering Algorithms 35
3.3.2 Dimensionality Reduction Algorithms 38

3.3.3 Visualization Tools . . . 40
3.3.4 Cluster Analysis Metrics . . . 40
3.4 Use Cases . . . 43
3.5 Discussion . . . 54
3.6 Conclusion . . . 55

4 EVALUATIONS OF EMBEDDINGS FROM DEEP LEARNING MODELS AND SUBSEQUENT TASK-ORIENTED FINETUNING . . . 57
4.1 Introduction . . . 57
4.2 Methods . . . 59
4.2.1 Datasets . . . 59
4.2.2 Deep Learning Architectures and Embedding Extraction . . . 62
4.2.3 Dimensionality Reduction . . . 65
4.2.4 Cluster Analysis . . . 67
4.2.5 Subsequent Use in Predictive Stage . . . 68
4.2.6 Role of SinoCT in Finetuning Each Pretrained Architecture . . . 69
4.2.7 Performance Evaluation and Statistical Analyses . . . 71
4.3 Results . . . 71
4.3.1 Dimensionality Reduction . . . 72
4.3.2 Cluster Analysis . . . 77
4.3.3 TBI GOSE at 6 Month Prediction . . . 79
4.4 Discussion and Conclusion . . . 83

5 SUMMARY AND FUTURE DIRECTIONS . . . 87

REFERENCES . . . 92

LIST OF FIGURES

1.1 Example of window level and width and the resulting images. A) shows a brain windowed HCT which highlights soft tissues. B) highlights the same image but with a bone window. Figure from reference [1] 2
1.2 Convolution process for input image. Kernel is moved across the image where convolution operations are performed. The output is called the feature map. Following the convolutions, a ReLU activation function is applied to all elements of the resulting feature map 4
1.3 Vision transformer architecture. Image gets split into patches followed by a linear projection of flattened patches. Each embedding gets a positional embedding added prior to input into the transformer encoder. 5
1.4 Example of HCT slice being encoded into an embedding. First input data (HCT slice) is prepared and input into pretrained deep learning architecture. The high dimensional data is then converted into an N-dimensional embedding that represents the input. 9
1.5 General foundation model principals. Large scale dataset such as electronic health records, medical images, or other data are used to pretrained foundation models. With subsequent finetuneing, foundation models can be adapted for a variety of downstream tasks within the design constraints of the model. 10
1.6 U-Net model architecture. The values at the top of the block indicate the feature map depth while the values to the left indicate resolution. Operations are shown with the corresponding arrow colors. 12
1.7 Residual connection implemented in ResNet models. Shows how the input is also incorporated to the output following two convolutions. This allows gradients to flow through these shortcuts, reducing the vanishing gradient problem [2–4]. . . 14
1.8 DINOv2 training schematic. Blue shows the student network while green shows the teacher network. Both architectures are the same, but student learns via standard backpropagation and the teacher network learns via an exponential moving average of the student weights. 16
1.9 CT-FM training regime. A) shows standard SimCLR methods commonly used with real images. These images are augmented and then processed through a SegResNet encoder. A contrastive loss function learns to group similar augmentations together while repelling different images. B) Intra-sample contrastive learning. CT patches within a scan are chosen and then augmented. This process helps the model learn differences within the CT in addition to difference between scans. Note Figure only depicts two patches where CT-FM uses 20 patches per scan during training. 19

2.1 Example of how models produce predictions. First, the sequence input is introduced to the model. Then the 8 U-Nets each make a prediction, these predictions are averaged, and then a threshold is applied to filter out weak predictions. . . . 26

2.2 DSC vs %HD for each of the slices in the independent testing set. Individual slices shown are examples of extremes, with the green outline showing the manual ground truth and red showing the model prediction. Best segmentations are those with a low %HD and higher DSC, in the upper left-hand corner. These are predictions from Model C. Independent testing group set is shown as either a blue circle (np-TBI) or green cross (p-TBI) 28
2.3 Example model predictions on testing sets. A) shows an example of a HCT image without artifacts and calvarial fractures. B) shows an example of a HCT image of a p-TBI patient with large amounts of artifacts and damage. Image i) shows the original CT image, image ii) shows the ground truth, image iii) shows Model A prediction, image iv) shows Model B prediction, and image v) shows the Model C prediction. 31

3.1 Overview of CAT. Extracted embeddings get uploaded into the software. Dimensionality reduction and clustering metrics are calculated to evaluate embedding clusters 33
3.2 Processing of 3D HCT scan using a 2D ResNet50 model pretrained with ImageNet. Slices were processed sliced by slice, producing slice-level embeddings. All slice-level embeddings were then mean-pooled to produce a scan-level representation. 35
3.3 Overview of main interface of CAT 44
3.4 Backend parameters for clustering and dimensionality reduction algorithms. . . 45
3.5 Example of tool output showing PCA plots of Iris flower features reduced to two components (i) and three components (ii). A) Points colored by K-Means cluster assignment. B) Points colored by ground-truth species label. 46
3.6 UMAP plots of Iris flower features. A) Points are colored by Fuzzy C-Means Clusters. B) Points are colored by Iris species. 47
3.7 Cluster metric summary produced by CAT. Results were for agreement between K-Means clustering with 3 groupings and ground truth species label. 48
3.8 PCA plots of Iris features reduced to two dimensions. A) Colors plots by K-Means cluster (2 clusters). B) Colors plots by Iris Setosa or the grouped classification (both Iris Versicolor and Viriginica) 48
3.9 Cluster metric summary produced by CAT. Results are for agreement between K-Means clustering with 2 groupings and binarized species label. 49
3.10 UMAP plots of TBI embeddings reduced to two dimensions. A) Colors plots by K-Means cluster (2 clusters). B) Colors plots by mortality at 6 months 50
3.11 Cluster metric summary produced by CAT. Results are for agreement between K-Means clustering with 2 groupings and TBI patient mortality at 6 months. . . 51
3.12 UMAP plots of TBI embeddings reduced to two dimensions. A) Colors plots by K-Means cluster (3 clusters). B) Colors plots by scanner vendor. 52
3.13 Cluster metric summary produced by CAT. Results are for agreement between K-Means clustering with 3 groupings and scanner vendor. 52
3.14 Cluster Composition from CAT. Shows how each cluster produced by clustering algorithm is comprised of reference label (scanner manufacturer). 53

4.1 TXA TBI Dataset exclusion criteria for embedding analysis. Out of all the available data with Scans (766), the two sites with the two most prevalent scanner vendors were chosen. Total number of scans was further reduced if scans did not have clinical data available. 61
4.2 Preprocessing of HCTs for the different architectures in this study. A) Shows the different HU windows applied to the HCT scan. After windowing, each stack of the same brain scan gets concatenated to be processed by ResNet50 or CT-FM. B) Single blood window is applied with a higher resolution but a decrease in depth for the RAD-DINO inputs. 65
4.3 Flow chart of methods for embedding evaluation. First HCT embeddings are extracted from deep learning architectures to obtain scan-level representations. Qualitative and quantitative analysis is done on these extracted embeddings to determine the how cases relate to one another. For the TXA TBI Dataset, embeddings are used to make the clinical prediction of GOSE ≤ 3 at 6 months. Processes can be redone following sequential finetuning of pretrained deep learning architectures. 66
4.4 2D UMAPs of the extracted embeddings from the COVID-19 dataset using pretrained (left) and sequentially finetuned (right) models. From Top down the models are ResNet50, RAD-DINO, and CT-FM. Cases (data points) are colored by scanner manufacturer. UMAP 1 and UMAP 2 represent the reduced dimensions of each embedding produced by UMAP dimensionality reduction. 73
4.5 2D UMAPs of the extracted embeddings from the COVID dataset using pretrained (left) and sequentially finetuned (right) models. From Top down the models are ResNet50, RAD-DINO, and CT-FM. Cases (data points) are colored by scanner manufacturer.UMAP 1 and UMAP 2 represent the reduced dimensions of each embedding produced by UMAP dimensionality reduction. 75
4.6 2D UMAPs of the extracted embeddings from the TBI dataset using pretrained (left) and sequentially finetuned (right) architectures. From Top down the architectures are ResNet50, RAD-DINO, and CT-FM. Cases (data points) are colored by acquisition site. UMAP 1 and UMAP 2 represent the reduced dimensions of each embedding produced by UMAP dimensionality reduction. 76
4.7 Paired slope plots indicating the metric changes following sequential finetuning of the models. Architectures are ResNet50 (blue), RAD-DINO (green), and CT-FM (orange). A) shows the cluster metrics for the COVID-19 dataset and B) shows cluster metrics for the TXA TBI dataset. Data points are shown with the corresponding 95% CI. Arrows indicate the desired trajectory for each metric, indicating reduction in clustering following sequential finetuning. 79
4.8 Logistic regression model prediction probabilities plotted against GOSE. Prediction probabilities are colored by outcome (red GOSE ≤ 3, green GOSE > 3). Kendall's τ for each respective test is shown above plot 82

LIST OF TABLES

1.1 Corresponding HU Values for different tissues. 3
1.2 Training methods overview for SL, unsupervised learning, and SSL. 8

2.1 GCS Components and Scores 21
2.2 Mean DSC and HD performance metrics with the 95% CI of all models on the n-TBI and p-TBI test sets. The 95% CI was calculated using bootstrapping with 1000 repetitions [CI lower, CI upper]. n-TBI: non-penetrating. N is number of total slices. All testing sets are independent testing sets. Asterisks denote statistically significant differences when compared to Model A on the p-TBI testing set ($p<0.05$, $p<0.01$, $p<0.001$). 27

3.1 TBI dataset demographics. 34

4.1 Breakdown of the different HCTs datasets used in the study. These HCTs came from (i) the UChicago Datamart, (ii) a multi-institutional clinical trial dataset, or (iii) the labeled SinoCT dataset of normal and abnormal HCT scans. 59
4.2 UChicago COVID-19 Data Mart Dataset Patient Characteristics 60
4.3 TXA TBI Dataset Patient Characteristics 62
4.4 GOSE scores and respective outcomes 68
4.5 Training Parameters 70
4.6 Classification head hyperparameters for each model. 70
4.7 Cluster Metrics, including the ARI, DBI, and Purity Score, for the different experimental arms and architectures for the COVID dataset. Arrows indicate desired change for metrics evaluating extracted embeddings from pretrained to sequentially finetuned. Bold values indicate statistically significant differences (95% CI excludes zero). 77
4.8 Cluster Metrics, including the ARI, DBI, and Purity Score, for the different experimental arms and architectures for the TXA TBI dataset. Arrows indicate desired change for metrics evaluating extracted embeddings from pretrained to sequentially finetuned. Bold values indicate statistically significant differences (95% CI excludes zero). 78
4.9 Model performance comparison (AUC with 95% CI) for RAD-DINO, ResNet50, and CT-FM on the TXA TBI dataset. AUCs reported for embeddings from pretrained models and sequentially finetuned models. 80
4.10 Pairwise comparisons of AUC across architectures. Bold values indicate statistically significant differences (Bonferroni adjusted $p < 0.05$) 80

ACKNOWLEDGMENTS

Throughout my time at the University of Chicago there have been many individuals who have contributed to my growth on a professional and personal level.

First and foremost, I would like to thank my advisor Dr. Maryellen Giger for her mentorship. I am beyond honored to have had such an accomplished scientist offer her guidance to me during this journey.

The members of my committee have also contributed greatly to this work. Drs. Ingrid Reiser and Samuel Armato who shared their expertise in medical physics, image analysis, and artificial intelligence. Dr. Ali Mansour for his clinical expertise in traumatic brain injury and applications of AI in medicine. I greatly appreciate the time, insightful discussions, and guidance from each of my committee members. Their insight and feedback has helped shape and strengthen this work.

I would like to thank all the members of Giger Lab who have been instrumental in my professional development all while making my time in this lab unforgettable. To Karen Drukker, Hui Li, Chun-Wai Chan, Feng Li, Li Lan, Sasha Edwards, John Papaioannou, and Heather Whitney who were very welcoming and at many times offered insight and support that has been monumental to this work. A very special shoutout to Jordan Fuhrman who has given more than his fair share of time helping me with many aspects of my projects. His insight and advice will always be something I carry going forward.

To all the members of the GPMP that I have had the pleasure of getting to know. In particular, a special thanks to Joseph Cozzi, Geneva Anderberg, and Jason Meier who made my time as a student of the GPMP an amazing experience. Starting graduate school at the height of COVID would have been impossible if it were not for them. I would also like to take the time to further thank Jason Meier who has become more of a brother to me during this time. Over the last 6 years we have shared many experiences that were special. Whether it was playing video games, going fishing at ridiculous hours, getting into expensive

hobbies, traveling to another country, or eating in China Town, the moments and bond that we developed will last a lifetime.

To my family, who has always supported me. I know it wasn't easy to uproot my life and leave Oregon, go long stretches of time without going home, and missing holidays and birthdays. Know that I really appreciate the ever lasting guidance and motivation you all gave me. The sacrifices that my parents made to get me into sports and activities laid the groundwork for me getting to where I am today.

Lastly, my biggest appreciation goes to my fiancée, Madeleine Glenn. From the moment we met back when we were teenagers, she has been my biggest supporter, motivator, cheerleader, and best friend. Without her unwavering support who knows where I would have ended up. Even though she was also busy finishing graduate school and starting her career, she has never failed to be there for me when I needed it most. Words cannot describe the joy I feel knowing that I have her by my side. From moving to Chicago, getting our dog Blue and our cat Frankie, eating meals together after a long day, going to try new experiences, pushing each other to try new things, and even just silly dance breaks in the middle of the day, every moment we shared gave me the strength to continue and I can not thank her enough.

ABSTRACT

Computed tomography (CT) of the brain is an essential tool for evaluating various pathologies. In emergent care, a head CT (HCT) scan is often the preferred imaging modality for assessing a patient due to the rapid acquisition speed, which allows for quick assessment and timely decision making in critical situations. This imaging modality is specifically beneficial in traumatic brain injury (TBI) cases, where imaging data are vital to determining the patient's treatment plan and care, as well as for potentially investigating neurological symptoms in patients experiencing long COVID-19. However, interpreting and evaluating HCT scans can take time and require trained experts to ensure accurate assessment. Thus, by developing automated computer techniques, we aim to improve HCT scan assessment to ultimately enhance patient treatment quality and outcomes.

Since the 1980's, computer-aided diagnosis techniques have been developed to aid healthcare professionals in evaluating patient data in a more effective scope. Recent advancements in artificial intelligence (AI) have significantly improved diagnostic accuracy and streamlined clinical workflows, demonstrating effectiveness and efficiency across various healthcare applications. One such advancement is that of foundation models, which encode input data for a variety of downstream tasks. Therefore, we propose the use of machine learning methods to quantitatively evaluate HCT scans acquired from patients with TBI and COVID-19.

HCT scans often include extraneous information that can have an impact on the performance of machine learning models, such as causing an algorithm to reach an erroneous prediction. Such non-patient content may include medical devices irrelevant to the task, the CT bed, and other foreign bodies. Manual segmentation of brain anatomy is rarely performed in clinical settings due to its impracticality. However, if computational techniques could automate this process, it could be a useful tool in an overall AI system for enhancing patient evaluation. Therefore, we developed a U-Net based cross-validation ensemble method to segment the brain in TBI patient CT scans, including accurate management of

areas with high tissue damage and/or image artifacts, allowing for consistent downstream evaluation focusing only on the brain matter.

Deep learning architectures encode input data into embeddings, which can be used for clinically relevant downstream tasks. However, deep learning architectures are susceptible to shortcut learning, in which the model encodes spurious information, degrading downstream performance. Assessing learned embedding representations using cluster analysis techniques, such as dimensionality reduction and cluster metrics, allows evaluation of underlying patterns and shortcuts in the data. Despite the availability of various dimensionality reduction techniques and clustering metrics, there is a lack of consensus and consistency among methods for evaluating extracted embeddings and underlying patterns. Thus, we developed the Cluster Analysis Tool (CAT) to streamline the analysis of extracted embeddings from deep learning architectures to dimensions not accessible to the human eye. CAT was capable of performing dimensionality reduction and calculating cluster analysis metrics to provide both qualitative and quantitative assessment. To evaluate CAT's ability to handle inputs of varying complexity and structure, we tested the tool on both the Iris dataset, a set of low-dimensional features, and a TBI clinical dataset, consisting of high-dimensional embeddings extracted from deep learning architectures.

HCT scans are a vital diagnostic tool that captures 3D information that allows for life-saving decisions to be made. There are many different scanner models, vendors, acquisition parameters, and patient populations, and AI architectures may encode spurious correlations rather than clinically relevant information. Thus, the final aim of this dissertation was to assess extracted embeddings from three different pretrained models: ResNet50, pretrained on ImageNet; RAD-DINO, pretrained on chest radiographs; and CT-FM, pretrained on CT scans. Embeddings were extracted from HCT scans in the COVID-19 Data Mart and TXA TBI Trial datasets. We used the clustering methods employed by CAT to qualitatively and quantitatively assess the extracted embeddings. Furthermore, we evaluated

the impact of finetuning on embedded representations by utilizing the SinoCT dataset. All pretrained models exhibit vendor-level embedding clustering, both qualitatively and quantitatively. Finetuning reduced vendor level embedding clustering for the COVID-19 Data Mart scans, while the TXA TBI Trial dataset embeddings still clustered by vendor, i.e., still demonstrated a shortcut behavior.

The work in this dissertation demonstrates how AI methods can be applied for data preparation and embedding evaluation in HCT scans of TBI and COVID-19 patients. This included applying deep learning to segment complex presentations of TBI HCT scans, developing CAT to streamline qualitative and quantitative embedding cluster analysis, and using CAT methods to evaluate vendor-based shortcuts in HCT embeddings, demonstrating how finetuning reshaped embeddings and influenced downstream classification. Methods developed in this work highlight the importance of upstream analysis in supporting the development of reliable and safe applications of AI in medicine.

CHAPTER 1

INTRODUCTION

Medical imaging artificial intelligence (AI) has grown exponentially over the last 20 years [5–8]. With the introduction of new models and methods each year, performance on clinically relevant tasks such as segmentation, pathology detection, report generation, image reconstructions, and outcome related predictions has improved. As such, more of these models are being integrated into everyday clinical workflows [9–11]. Advancements in medical imaging AI could potentially help radiology work flows since image interpretation remains a time-consuming process that is vulnerable to radiologist error.

There are many factors that contribute to radiology errors including, but not limited to, perceptual errors, cognitive errors, mental health, burnout, and visual fatigue [12]. The role of AI in medical imaging analysis has been shown to reduce errors across the radiology clinical workflow by reducing missed findings, false positives, and improving disease classifications [13]. However, performance increases with the aid of these AI tools does not always translate in clinical practice where data can vary from controlled research studies [14]. Thus, the increasing need for reliable and safe AI tools is paramount to ensure both an improvement in clinical workflows while being robust enough to translate into different environments.

In this dissertation, we develop and evaluate machine learning methods for head computed tomography (HCT) segmentation and embeddings analysis of patients who have had traumatic brain injury (TBI) or COVID related complications. These methods address current limitations in medical AI that can result in poor performing models that can negatively impact patient treatment and outcomes. We first provide an overview of imaging and machine learning technologies utilized in this dissertation.

1.1 Computed Tomography

Computed tomography (CT) is a valuable tool that was invented in 1967, with the first successful medical scan of a live patient in 1971 [15,16]. CT can provide meaningful diagnostic information due to its slice-based image reconstruction. In some cases, CT is the preferred imaging modality over MRI due to its fast acquisition and lower cost [17, 18]. Briefly, CT images are acquired by rotating an x-ray source and detector around the body, generating a series of projections. These projections are processed using reconstruction algorithms to generate cross-sectional slices which can then be further rendered into 3D representations [15]. Due to the speed of acquisition and high spatial resolution, CT is especially useful for time-sensitive diagnostic tasks such as those found in the emergency departments.

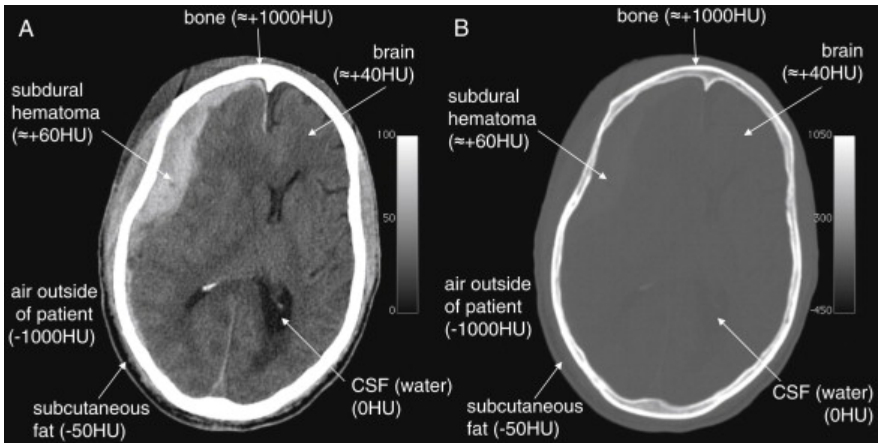


Figure 1.1: Example of window level and width and the resulting images. A) shows a brain windowed HCT which highlights soft tissues. B) highlights the same image but with a bone window. Figure from reference [1]

.

A useful feature of CT is that every reconstructed voxel relates to anatomy through Hounsfield Units (HU). This is because HU provides a quantitative measure of x-ray attenuation, which changes depending on the degree to which a given tissue absorbs the incident photons. Thus, with CT, we get different HU value ranges indicating different tissues. The HU values corresponding to different tissues are listed in Table 1.1 [19–21]. As such, by adjusting the range of these values, one can either highlight or suppress different HU values,

allowing radiologists to detect several pathologies and abnormalities as shown in Figure 1.

Table 1.1: Corresponding HU Values for different tissues.

HU Value	**Corresponding Tissues**
-1000	Air
-950 to -650	Lungs
-115 to -100	Subcutaneous Fat
0	Water
30	Gray Matter
40	White Matter
> 1000	Cortical Bone
> 3000	Metal Implants

1.2 Artificial Intelligence in Medical Imaging

AI assistance for medical imaging has been around since the 1980s, with tasks like computer-aided diagnosis (CAD) laying down the framework on which modern models are built [22,23]. These models have a several uses in radiology, including pathology classification, prognosis prediction, workflow optimization, and image segmentation [24]. These machine learning algorithms can extract quantitative information from clinical data (e.g., medical images, charts, history, etc.) and identify patterns. In medical imaging AI, convolution neural networks (CNNS) play a vital role as the backbone of many successful architectures for clinically relevant tasks. Furthermore, how these models learn to recognize patterns depends on their pretraining approach (described in Section 1.2.2) and the resulting learned patterns are often represented as vector representations called embeddings [25]. Recently, large deep learning architectures called foundation models make use of large unlabeled datasets and specific pretraining regimes to produce embeddings that allow for multiple downstream tasks [26]. In principle, foundation models are designed to be easily adaptable to a variety of downstream tasks. However, in practice, adaptation of a foundation model for a specific task is not always straightforward. The following sections further elaborate on these topics.

1.2.1 Convolutional Neural Networks and Vision Transformers

A CNN is the backbone of many deep learning models designed to process grid-like data such as images and videos. These networks are capable of various tasks such as classification, image segmentation, and image reconstruction [27–29]. The convolutional layer is responsible

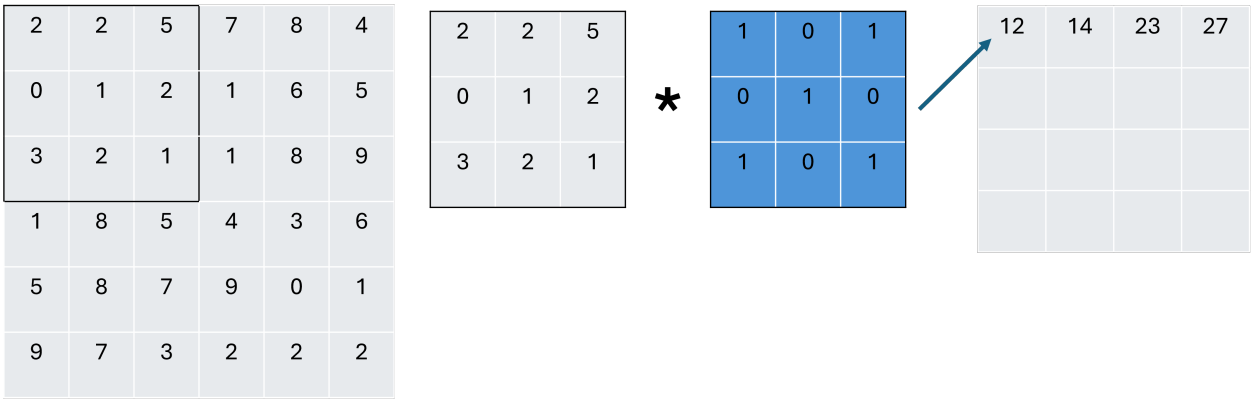


Figure 1.2: Convolution process for input image. Kernel is moved across the image where convolution operations are performed. The output is called the feature map. Following the convolutions, a ReLU activation function is applied to all elements of the resulting feature map

for learning feature representations of the input. These features can include edges, textures, and gradients. Early layers tend to capture low-level image qualities such as edges and intensity gradients. Later layers capture more complex patterns. To extract these features, a kernel is moved along the image, performing convolution operations as shown in Figure 1.2. The output of these operations is called a feature map, where the depth corresponds to the number of filters in the layer. After each convolution, CNNs apply a non-linear activation function to each element in the feature maps. The most common non-linear function applied is rectified linear unit (ReLU), which zeros out negative values and keeps positive values. This process introduces non-linearity and allows the model to learn complex patterns. Following the convolution and activation, a pooling operation (max-pooling) is applied, downsampling the feature maps. The resulting feature maps are then flattened and passed through a fully connected layer to produce the final class probabilities. While CNNs have several inductive

biases that make them effective for image related tasks, they may struggle in situations where these assumptions fail. In medical imaging, this presents in tasks that require relating spatially different areas of the input [30, 31].

To address CNN inductive bias, Vision transformers (ViTs) were developed, adapting the self-attention mechanisms from transformers originally developed for natural language processing [32]. As shown in Figure 1.3, ViTs process images differently compared to CNNs. While CNNs perform convolutional operations, assuming that nearby pixels are related, ViTs convert the input image into a sequence of smaller patches. Image patches are subsequently converted into tokens with an additional positional embedding added to preserve spatial information.

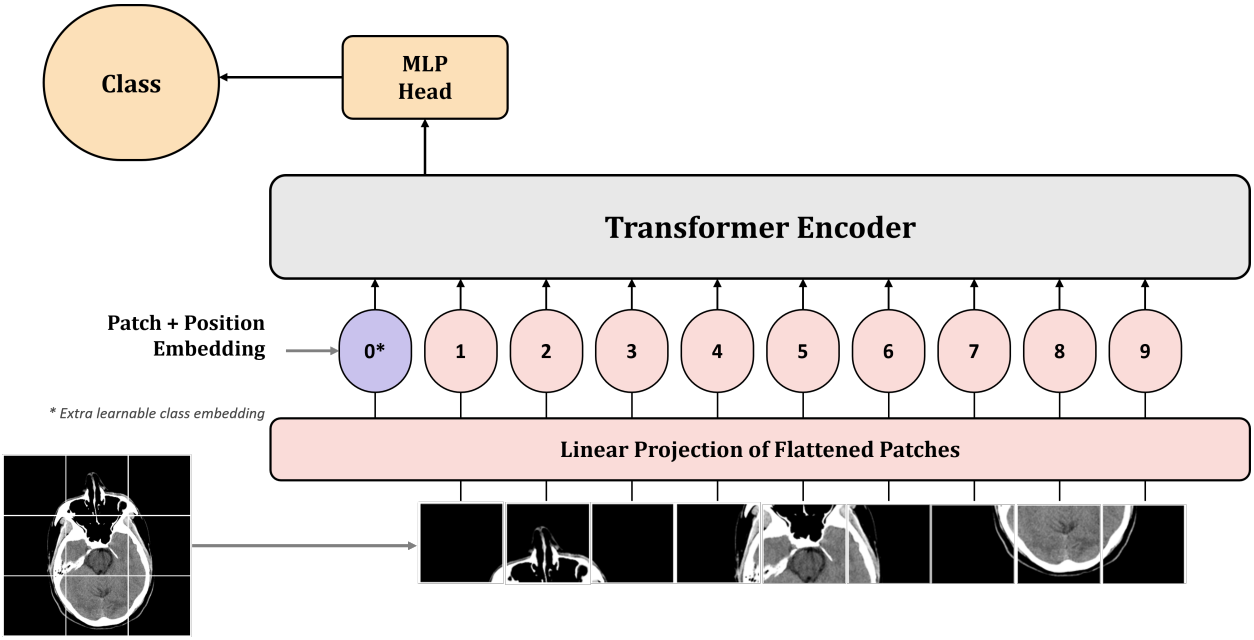


Figure 1.3: Vision transformer architecture. Image gets split into patches followed by a linear projection of flattened patches. Each embedding gets a positional embedding added prior to input into the transformer encoder.

The key feature of ViTs is the self-attention mechanism in the transformer encoder. Briefly, general attention allows a model to focus on different parts of another sequence, which is useful in models used for translation [31, 32]. Introduced by Vaswani et al., self-attention

allowed transformers to weigh different sections of the input data relative to one another for natural language processing [32]. For ViTs, self-attention allows vectors to encode not only image patch and position information, but also information regarding their relationships with other patches. This allows ViTs to capture both local and global information without being restricted to nearby pixels, as CNNs are. However, ViTs require larger amounts of training data to outperform CNNs and require larger amounts of compute, making development and deployment a much more difficult task [31, 33].

1.2.2 AI Training Methods

While standard deep learning architectures learn via backpropagation, the training paradigm can differ from model to model. The training regimes can be categorized by the use of labeled or unlabeled data.

Architectures using supervised learning (SL) techniques make use of labeled data. The model produces an output which then gets compared to the ground truth label. During training, the loss function is driven by how correct the model prediction is, with parameters being updated to fit the function [34]. As such, the architecture is directly optimized for specific tasks. While this training is well suited for classification and prediction models, the amount of labeled training data required is often expensive or scarce in medical imaging.

In supervised learning (SL), the model predictions are compared to the ground-truth labels. Architectures using SL techniques make use of labeled data. The model produces an output which then gets compared to the ground truth label. During training, the loss function is driven by how correct the model prediction is, with parameters being updated to fit the function [34]. As such, the architecture is directly optimized for specific tasks. While this training is well suited for classification and prediction models, the amount of labeled training data required is often expensive or scarce in medical imaging.

No labels are available for unsupervised learning approaches, thus the loss is calculated

by using the structure of the data itself (e.g., reconstruction error in autoencoders) [35]. The loss is then backpropagated through the network to compute gradients. To update model parameters to minimize the loss function, the gradients are used by an optimizer. To reduce the risk of overfitting, different regularization techniques can be used, such as dropout and data augmentation [36]. The goal of unsupervised learning is to train an architecture by using unlabeled data. In contrast to SL, the loss function in unsupervised learning makes use of the data structure itself [34]. An example is how an autoencoder learns by producing a reconstruction of the input. The reconstruction is then compared against the original [37]. While the model requires no labels, enabling the use of large datasets, models cannot distinguish if features are clinically relevant, potentially reducing downstream analysis and tasks.

Another learning technique is self-supervised learning (SSL) that leverages unlabeled data [38]. Depending on the training regime, different types of information are encoded when inputs are processed through the models. Unlike SL and unsupervised learning approaches, SSL involves a training method that makes use of pseudo-labels from the training data rather than external labels [38]. This approach enables architectures to train in a SL manner. For example, SSL can utilize contrastive learning, where the input data goes through augmentations. The architecture is trained to group similar augmentations close together in latent space while pushing away differing inputs [39]. Another example involves patching the input image and the model learns to predict the missing content [40]. Most of the current foundation models are pretrained with the help of SSL.

Table 1.2 below shows the comparisons between training methods, highlighting key advantages and disadvantages for each.

Table 1.2: Training methods overview for SL, unsupervised learning, and SSL.

	Supervised Learning	**Unsupervised Learning**	**Self-Supervised Learning**
Data Label	Labeled	Unlabeled	Unlabeled
Learning Signal	Human annotated labels	Relies on inherent structure of data itself	Pseudo-labels from the data itself
Advantages	- Learned features are directly aligned with task of interest	- Can use large amounts of unlabeled data - no annotation cost	- Can use large amounts of unlabeled data - no annotation cost - Leverages supervised learning techniques with no labels
Disadvantages	- Labeling of data is a time consuming and expensive process.	- Harder to evaluate since there is not a clear target answer - Needs human intervention to interpret what clusters mean	- Computationally expensive. - Pseudo-labeling task (how to mask or augment data) must be well designed
Examples	CNN trained for classification task	K-means clustering, PCA	Large Language Models, Foundation models

1.2.3 Embeddings

In machine learning, an embedding is a learned vector of scalar values that represents input data such as images, audio, and text [41, 42]. Medical imaging AI models take in imaging inputs such as radiographs, CTs, ultrasounds, and MRIs and encode image-specific information into embeddings. Thus, an embedding is a lower-dimensional representation of the input data that makes downstream tasks computationally efficient.

Typically, embeddings are extracted from the final layer of a deep learning model, or, when a classification head is appended, from the penultimate layer. For CNNs like ResNet50, embeddings are extracted from the global average pooling layer [43]. Alternatively, embeddings can be constructed from the intermediate layers of a deep learning architecture. An

example being where outputs from each pooling layer are concatenated [25, 44]. In contrast, vision transformer (ViT) architectures attach a classification (CLS) token to the input patches. The CLS token aggregates information across all patches via self-attention, and the final representation is commonly used as a compact embedding for downstream tasks [45].

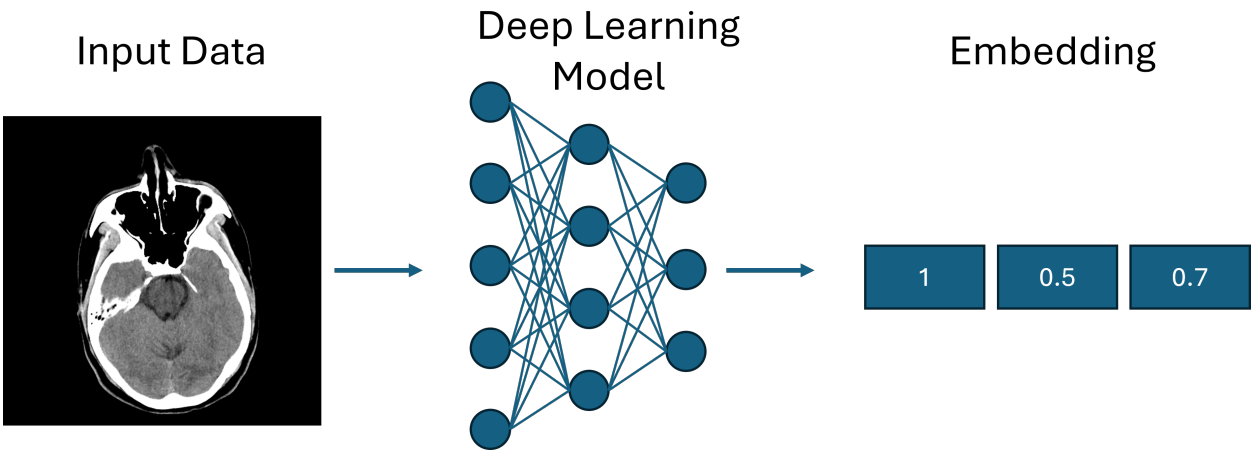


Figure 1.4: Example of HCT slice being encoded into an embedding. First input data (HCT slice) is prepared and input into pretrained deep learning architecture. The high dimensional data is then converted into an N-dimensional embedding that represents the input.

Regardless of the architecture, embeddings should capture clinically meaningful features of the input data such that similar cases cluster together while dissimilar cases remain separated in embedding space. Typically, these embeddings are used for downstream tasks as previously mentioned, such as classification, prediction, subgroup clustering, and segmentation [25, 44].

However, deep learning models are designed to minimize the loss function during training without any notion for clinical relevance. As such, they can also encode spurious information, such as scanner vendor or acquisition site [46]. If a spurious feature happens to correlate with clinical labels, the model classification performance can be misleading. For example, if all "positive" cases in a dataset come from one specific scanner, model can learn features of that specific scanner rather clinically meaningful features. Shortcut learning can degrade downstream performance and fail to generalize to new populations. To address these limitations,

foundation models have been developed.

1.2.4 Foundation Models

Foundation models are large deep learning models that are trained on vast datasets [26]. Briefly, the pretrained models extract embeddings, numerical representations of input data, to then perform various downstream tasks following adaptation. These models emerged from transformer architectures, where the self-attention mechanism introduced allowed the model to weight the input elements simultaneously [32]. As a result, the algorithm can effectively model long-range dependencies while preserving contextual information. This work laid down the foundation for BERT and GPT-3 models that demonstrated that pretraining with large unlabeled data using self-supervised learning could achieve strong performance across a wide range of downstream tasks [47, 48].

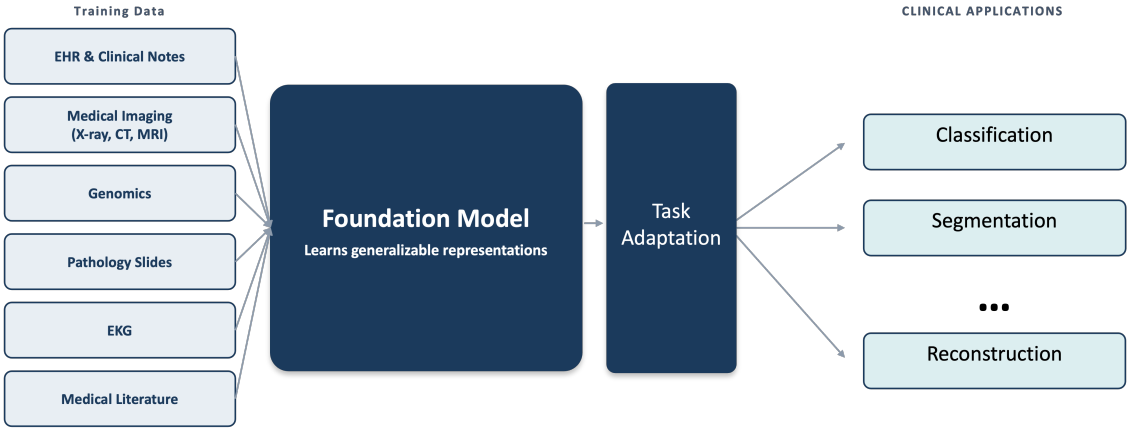


Figure 1.5: General foundation model principals. Large scale dataset such as electronic health records, medical images, or other data are used to pretrained foundation models. With subsequent finetuneing, foundation models can be adapted for a variety of downstream tasks within the design constraints of the model.

Foundation models have been at the forefront of AI research in medicine. As shown in Figure 1.5, depending on the model, a user can leverage a variety of inputs such as images, electronic health records, lab results, etc. and be able to generate predictions, segmentations, radiology reports, and image reconstructions [49]. This contrasts with standard deep learning

models, that are trained to do a specific task and can fail to generalize to new inputs without substantial retraining.

1.3 Machine Learning Algorithms Employed

In the following subsections, the different models that were used in this dissertation will be discussed.

1.3.1 U-Net

U-Net is a convolutional neural network that was specifically developed for semantic segmentation of biomedical images [50]. Previous iterations of fully convolutional networks (FCNs) were capable of pixel-wise segmentation but struggled to keep spatial information due to the progressive downsampling native to their architectures [51]. Figure 1.6 shows the U-Net model architecture.

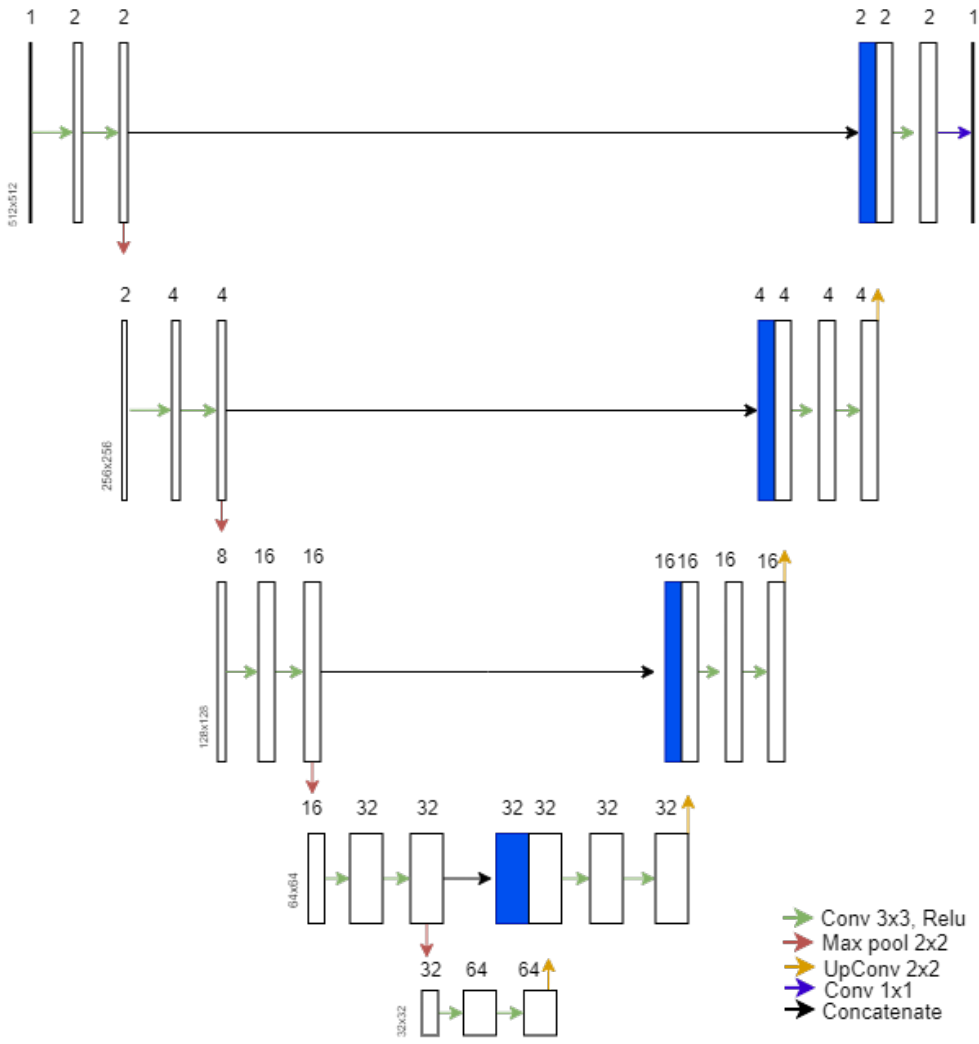


Figure 1.6: U-Net model architecture. The values at the top of the block indicate the feature map depth while the values to the left indicate resolution. Operations are shown with the corresponding arrow colors.

The key feature of U-Net is the encoder - decoder design. The encoder path takes in the input image and applies two convolutions using a 3x3 kernel followed by a ReLu which is then downsampled using a maxpooling operation. For each step, the number of feature channels increase while reducing the spatial resolution. The decoder path mirrors these operations, the feature maps are upsampled and combined with the encoder features, then passed through two convolutions with a ReLu operation. Each step restores spatial resolution and decreases the number of feature channels. An important aspect of U-Net is the

skip-connections, that are shown as black arrows in Figure 1.6. These skip-connections are between the corresponding encoder-decoder layers and preserve spatial information that was lost during downsampling of the input. This architecture is particularly useful in biomedical segmentation, where accurate localization of small structures is important. In this work, we use U-Net as the backbone to our cross-validation ensemble model in Chapter 2 to segment the brain from HCTs of patients with Traumatic Brain Injury (TBI).

1.3.2 ResNet

Residual neural networks (ResNet) were introduced in 2015 and won the ImageNet Large Scale Visual Recognition Challenge of that year [43, 52]. Deep networks prior to ResNet have a common issue known as the vanishing gradient problem, whereas training progresses, gradients diminished exponentially [2–4]. This problem causes the earlier layers in the model to barely update or stop learning completely. ResNet addresses this issue by introducing residual connections in the architecture. These residual connections essentially bypass one or more layers of the model, adding the original input to the output of said layers. This residual connection can be seen in Figure 1.6.

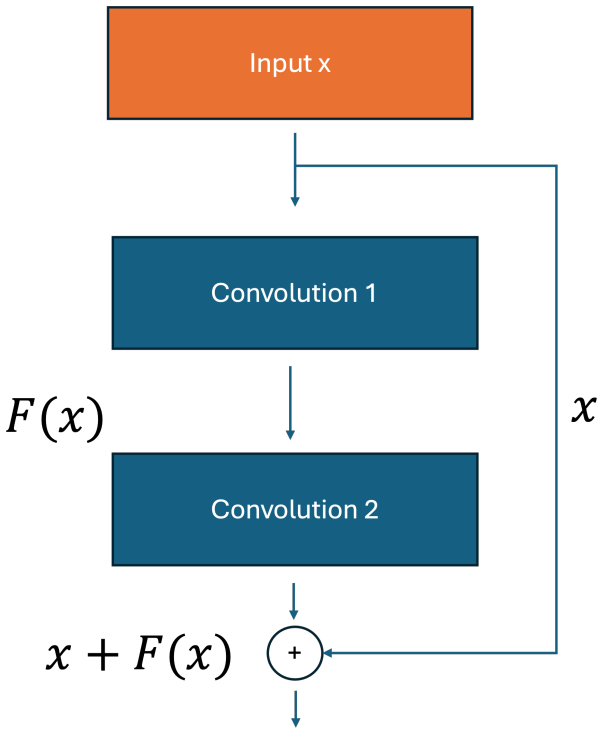


Figure 1.7: Residual connection implemented in ResNet models. Shows how the input is also incorporated to the output following two convolutions. This allows gradients to flow through these shortcuts, reducing the vanishing gradient problem [2–4].

Let the underlying mapping be denoted by $H(x)$, which traditional models try and approximate. ResNet reformulates this by learning the residual function, defined as $F(x) = H(x) - x$. Thus, the output of the residual block is now $H(x) = F(x) + x$. The authors propose that learning the residual function is more straight forward than learning from the original mapping. The best case scenario would be where $H(x) \approx x$ and thus $F(x) = H(x) - x \approx 0$. It is easier to learn a function which gives output near zero rather than having multiple nonlinear layers learn the identity mapping.

The original ResNet paper tested various ResNet models. The building blocks of each ResNet model is consistent, allowing even their deepest network, ResNet 152, to be lower complexity compared to other models such as VGG-19. Because of the size, ResNet50 has been a popular choice for medical imaging applications since it allows for high performance on relatively small datasets, as commonly found in medical imaging research [53–57].

Despite the development of newer architectures such as ViTs, ResNet remains a relevant model in both natural and medical image applications. Newer architectures, such as transformers, incorporate the same residual connection mechanism within their attention and MLP blocks [45]. Furthermore, ResNet offers a balance between computational cost and performance relative to newer and larger models, making it feasible to train on standard modern GPUs in resource-limited settings [58]. Lastly, due to its extensive use, it has become a reliable baseline for benchmarking newer models.

1.3.3 RAD-DINO

The development of RAD-DINO was primarily motivated due to the limitations in training deep vision-language models. These models make use of textual supervision, the use of natural language in the training data [59,60]. While this approach produces high performing models, the translation into medical imaging AI introduces complications. Due to the limited nature of medical datasets, several challenges emerge restricting the use of image-text pair datasets when scaling to larger datasets. Some of these challenges include limited number image-text datasets, privacy concerns with patient protected information, and incomplete text annotation. To address these issues, RAD-DINO was developed, an image encoder pretrained on chest radiographs using SSL introduced by DINOv2 [40, 61]. This allowed the model to circumvent the reliance on paired image-text data entirely, instead learning rich and transferable visual representations directly from a large and diverse collection of approximately 838k chest radiographs.

RAD-DINO employs a Siamese student-teacher network as shown in Figure 1.7. Both the student and the teacher have the same architecture, but the student updates its weights via standard backpropagation, while the teacher uses an exponential moving average of the students' weights. There are two main objectives during training. The first is masked image modeling (MIM). With MIM, the student is fed images with randomly masked patches. The

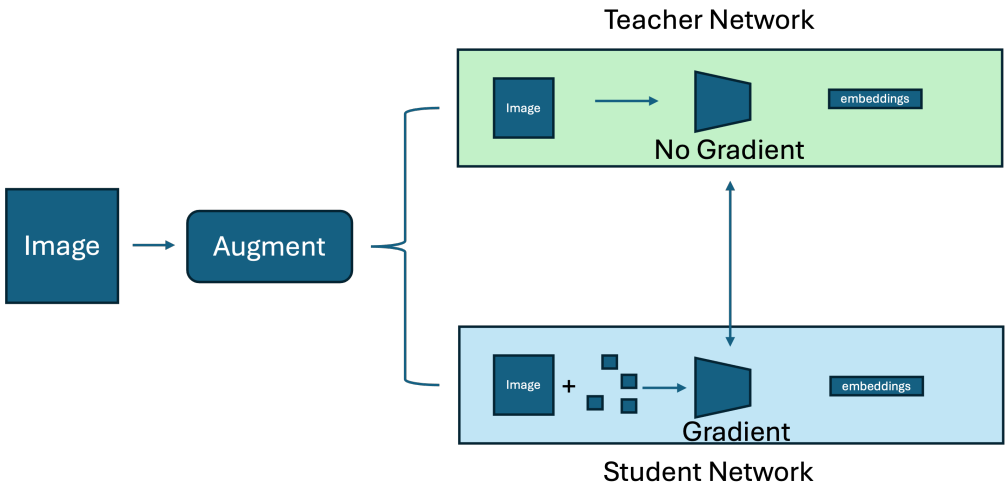


Figure 1.8: DINOv2 training schematic. Blue shows the student network while green shows the teacher network. Both architectures are the same, but student learns via standard backpropagation and the teacher network learns via an exponential moving average of the student weights.

objective is to essentially predict the teacher's feature representations for the missing patches. The second objective is self-supervised instance discrimination. In this task, the student is fed multiple local and global crops of an image and must align its representation with the teacher's global view of the same image. Through these two tasks, the model learns to process shapes, textures, and spatial information via MIM while learning global information from self-supervised instance discrimination. RAD-DINO has shown great performance across different tasks such as segmentation and classification of lung nodules [40, 62]. A different study also validated the model's robustness across different datasets [63]. This dissertation explores the model's capabilities across a different imaging modality and anatomy in Chapter 4, evaluating classification performance and susceptibility to vendor bias.

1.3.4 CT-FM

Foundation models have also started showing promise for 3D inputs. As mentioned in Section 1.1, CT is a 3D imaging modality. When 2D models are used to evaluate CT

scans, the models must process each slice and produce the corresponding slice level feature representations. The slice level representations then need to be aggregated to produce a scan level embedding [44, 64, 65]. Some aggregation methods include mean pooling, max pooling, and LogSumExp. Recently multiple instance learning approaches have been used to aggregate slice-level representations by using learned attention weights to obtain a final weighted scan level representation [65]. Regardless of the approach, 3D information, which can contain relevant context, can be lost or inadequately captured. Thus, 3D models offer an alternative approach that attempts to retain axial information at the cost of increased computational complexity.

One such model is CT-FM, a foundation model trained using over 140,000 CT scans [39]. These CT scans were acquired from different parts of the body including the abdomen, head, chest, and extremities. CT-FM was trained using Simple Framework for Contrastive Learning Representations (SimCLR) using a SegResNet encoder [66]. This is a contrastive learning approach that takes an input image and splits it into two different augmentations of the same image. Both views are passed through a shared encoder followed by a multilayer perceptron. The contrastive loss maximizes the agreement between augmented views of the same object while pushing them away from different images in latent space.

The study modified SimCLR to be better suited for 3D medical images. As shown in Figure 1.7b, the authors added an intra-sample objective. This approach not only contrasts different scans, but contrasts within the same scan as well. The CT scan is broken down into 3D patches, containing a cube of size 12.8cm x 12.8cm x 6.4cm (all images were resampled to 3mm slice thickness and 1mm in-plane resolution in). A patch is then chosen and transformed into two different augmented views. Augmentations for CT scans are categorized as standard contrastive learning choices and medical imaging specific choices. Standard contrastive learning choices include random resize and crop. Medical imaging choices include histogram shifting, random affine transformations, and intensity scale variations. This

change allows the encoder to not only distinguish differences from different scans but also learn that different areas within the image are distinct from one another.

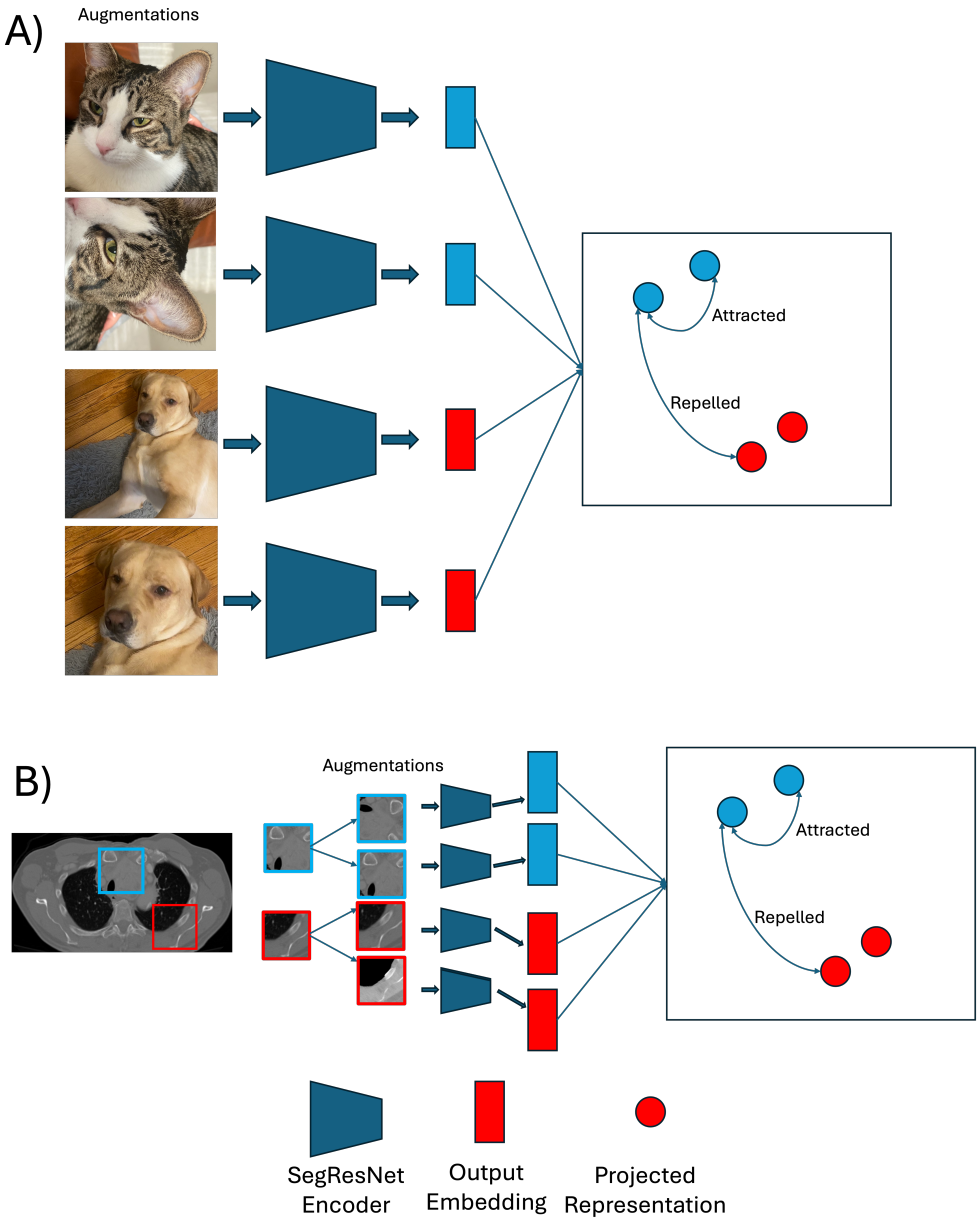


Figure 1.9: CT-FM training regime. A) shows standard SimCLR methods commonly used with real images. These images are augmented and then processed through a SegResNet encoder. A contrastive loss function learns to group similar augmentations together while repelling different images. B) Intra-sample contrastive learning. CT patches within a scan are chosen and then augmented. This process helps the model learn differences within the CT in addition to difference between scans. Note Figure only depicts two patches where CT-FM uses 20 patches per scan during training.

CHAPTER 2

DEEP LEARNING METHODS FOR BRAIN EXTRACTION ON PENETRATING AND NON-PENETRATING TRAUMATIC BRAIN INJURY CT SCANS

2.1 Introduction

Computed tomography of the brain is an invaluable diagnostic modality that can provide vast quantitative information. This information is vital not only for diagnosis and treatment planning, but also in the development of new techniques and methods [67–69]. HCTs contain both brain and non-brain information including but not limited to, skulls, CT beds, and medical devices. Thus, for tasks specifically involving brain tissue analysis, an important preprocessing step in obtaining quantitative information from HCTs is the removal of the skull, otherwise known as brain extraction [70]. Several studies have shown that convolutional neural networks may incorrectly classify images based on the presence of such information and, because of this, extraction of brain matter may improve both performance and confidence in CNN-based image evaluation tools. This is particularly relevant in conditions like TBI where imaging plays a vital role in patient planning and treatment.

To classify TBI, physicians use the Glasgow Coma Scale (GCS), a score meant to measure a person's level of consciousness following a TBI [71]. The scale is comprised of three components including eye opening, verbal response, and motor response. Table 2.1 shows the different components of the scale.

Scores ranging from 3-8 are classified as "Severe", 9-12 as "Moderate", and 13-15 and "Mild". While the GCS is considered the "gold standard" for clinical TBI evaluation, the metric has several drawbacks, ranging from but not limited to, subjectivity, correlation to prognosis, unsuitability for all patients, and barriers caused by patient disability and spoken language [72]. Despite being a strong predictor of outcome [73, 74], the severity classes are

Table 2.1: GCS Components and Scores

Component	**Score**	**Response**
Eye Opening (1–4)	4	Opens eyes spontaneously
	3	Opens eyes to verbal command
	2	Opens eyes to pain
	1	No eye opening
Verbal Response (1–5)	5	Oriented, converses normally
	4	Confused, disoriented conversation
	3	Inappropriate words
	2	Incomprehensible sounds
	1	No verbal response
Motor Response (1–6)	6	Obeys commands
	5	Localizes pain
	4	Withdraws from pain
	3	Abnormal flexion (decorticate posturing)
	2	Extension (decerebrate posturing)
	1	No motor response
Total GCS Score	**3 (minimum) – 15 (maximum)**	

not sufficient for describing a patient's response to treatment and their functional outcomes. To address these limitations, researchers have developed a more robust methodology for classifying TBI called CBI-M [75]. The new framework combines clinical assessments, blood-based biomarkers, imaging, and modifiers (e.g., medical history, mental health conditions, how injury occurred) to give a comprehensive categorization to a patient's TBI. This work focuses on the imaging pillar of CBI-M to leverage AI and imaging data to improve patient stratification and disease evaluation.

Brain extraction tool (BET) is software that is used in functional MRI studies to provide researchers with data containing only the brain tissue [76]. While BET is used as the gold standard for brain extraction in MRI studies, it is not suitable for CT images without proper modification of the data since BET was originally designed around MRI imaging characteristics instead of HU used in CT [77, 78]. Several AI-based segmentation tools have also been developed for various pathologies including cancer, hemorrhages, and stroke [79–81]. Despite advances in brain extraction methods, there is a lack of generalizable

AI-based brain segmentation algorithms that can encapsulate complex features present in traumatic brain injury (TBI) HCTs. The introduction of irregularities and imaging artifacts due to fractures, bleeds, and other trauma present in TBI HCT can complicate the task of segmentation. By training and testing models using both penetrating TBI (p-TBI) and non-penetrating TBI (np-TBI) cases, we aim to develop a U-Net-based model capable of segmenting HCT with calvarial fractures, bleeding, and metal artifacts, addressing a major limitation of prior work. Furthermore, due to the introduction of p-TBI data, the model is expected to generalize to more severe np-TBI cases. In this study, we propose a segmentation model capable of automatically segmenting brain regions in HCTs of both p-TBI and np-TBI patients using a cross-validation ensemble approach.

2.2 Methods

2.2.1 Data collection and Image processing

This study used two different databases: 1) a np-TBI dataset from a randomized control trial, The Prehospital Tranexamic Acid for TBI Trial [82] and 2) a p-TBI data collected retrospectively from local level 1 trauma center.

The Prehospital Tranexamic Acid (TXA) for TBI trial was a randomized, double-blind, 3-group, multicenter phase II trial designed to examine the efficacy and safety of out-of-hospital administration of tranexamic acid compared with placebo in participants with moderate or severe TBI who were not in shock. It included clinical and imaging data from multiple institutions across North America (United States and Canada) [83–85]. The np-TBI dataset consisted of 30 cases acquired from multiple clinical sites.

The p-TBI dataset included 20 retrospectively collected HCT scans from a Level 1 trauma center under IRB approval (IRB24-0307). These cases involved gunshot wounds to the head and served in our study as training, model fine-tuning, and evaluation. All scans were

de-identified.

Thus, 50 HCT scans were used in this study. Exclusion criteria included severe motion artifacts, incomplete imaging, and patients under the age of 18 years old. For all studies, tissue HCT reconstructions were chosen over bone due to their superior soft-tissue visualization, which is critical for assessing the extent of brain injury. Preliminary evaluations indicated the injury severity is less discernible in bone reconstructions. All scans were axial reconstructions of either 2.5 mm or 5.0 mm thickness and windowed with a standard clinical brain window width of 80 and level of 40 [86]. Images were resized to 512 x 512 using bilinear interpolation and normalized to zero mean and unit variance if necessary.

To independently evaluate the performance of the models developed in this study, we separated the cohorts into training (N= 38 cases) and testing (N=12 cases) subsets using the MIDRC generalized stratified sampling algorithm [87]. This stratification algorithm ensured that both training and testing sets have equal distributions of specific variables. For the np-TBI testing set, the data were stratified to match GCS across the training and testing sets. Alternatively, the matched attributes for sampling the p-TBI dataset also included the level of artifacts in the scans given the penetrating effect of the injury. The level of artifacts was rated from low, medium, or high, determined by evaluating the amount of metal artifacts and how much the artifacts obscure brain content within the slices of a scan. As noted in Section 2.1, the GCS has several drawbacks related to subjectivity and can be unsuitable for patients who are sedated or intubated. Stratification using GCS may introduce some noise into the training and testing splits. Due to the different number of cases in both cohorts, different training, validation, and testing splits were used to ensure that the independent testing sets contained the same value of cases (6 np-TBI and 6 p-TBI). For np-TBI, the 24 cases underwent 8-fold cross-validation (70% training, 10% validation), while the p-TBI dataset was split into 50/20/30% for training, validation, and testing, respectively.

2.2.2 Manual Segmentation

An in-house segmentation software was developed using Python 3.11 to allow manual delineation of binary masks of the brain matter on each CT slice to serve as the reference standard. For this investigation, brain matter was defined as tissue from the base of the cerebellum through the entire cerebrum, including gray and white matter as well as ventricular cerebral spinal fluid, but excluding skull, extracerebral hemorrhage, and scalp tissues. If no brain matter was present in a slice, a blank mask was generated.

All segmentations were manually delineated by a single trained rater, with any discrepancies resolved by consensus. The mean segmentation time per scan was approximately 30 minutes.

2.2.3 Deep Learning U-Net algorithms

This study developed 3 U-Net models using a combination of the np-TBI and p-TBI datasets: A) trained only on np-TBI images, B) initially trained on np-TBI images and then fine-tuned with p-TBI images, and C) trained using the combined training set of np-TBI and p-TBI images. This training strategy allowed for the comparison between models trained under varying tissue damage conditions (including artifacts, fracturing, and bleeding), thus evaluating the optimal training strategy for cases with extreme damage and obstruction.

The models used in the cross-validation ensemble were U-Net architectures built from scratch using PyTorch v2.0.0. The total number of trainable parameters was 121,869. Details of model architecture can be found in the project GitHub repository (`https://github.com/jtoledo255/Brain-CT-UNET-Segmentation-Model`).

Using an 8-fold cross-validation ensemble, eight models were trained on different training/validation splits, and the final predictions were aggregated via averaging to produce the final outputs. This method leverages all data in the training set while reducing variance associated with any split, producing a more stable and generalizable model. Figure 2.1 shows

a schematic of this method. Note that the use of cross validation was not just for evaluation, but also for ensemble infrastructure.

Model A was trained using only the np-TBI dataset. Each fold consisted of 21 cases for training and 3 cases for validation. Since 8-fold cross validation was used, each case produced 8 different predictions (one per fold). These 8 predictions were first averaged and then a threshold of 0.5 was applied to the averaged mask. Model A performance was directly evaluated on the held-out np-TBI test set. Transfer evaluation was then performed on the held-out p-TBI test set to assess the increasing pathological complexity.

Alternatively, Model B was developed by fine-tuning the last 4 layers of each of the folds of Model A with the p-TBI training set. The training set consisted of 10 cases for training and 4 cases for validation. The performance of Model B was evaluated using both the held-out np-TBI and p-TBI test sets.

Model C was trained by combining both the np-TBI and p-TBI training sets together and employing the same 8-fold cross validation ensemble method. For folds 1–6, the validation sets were manually constructed to include three np-TBI cases and two p-TBI cases. Due to the number of available cases, folds 7 and 8 included three np-TBI cases and one p-TBI case in their validation sets.

Figure 2.1 shows how the trained models made predictions on the held-out testing sets. For this study, the input to the model was a sequence of 7 consecutive 2D slices with the central slice as the segmentation target as shown in Figure 2.1. Compared to traditional 3D models which can be computationally expensive, this approach uses 2D convolutions while treating the neighboring slices as different channels [88, 89]. The outputs from each model in the ensemble were averaged and a threshold of 0.5 was applied to achieve a binary segmentation.

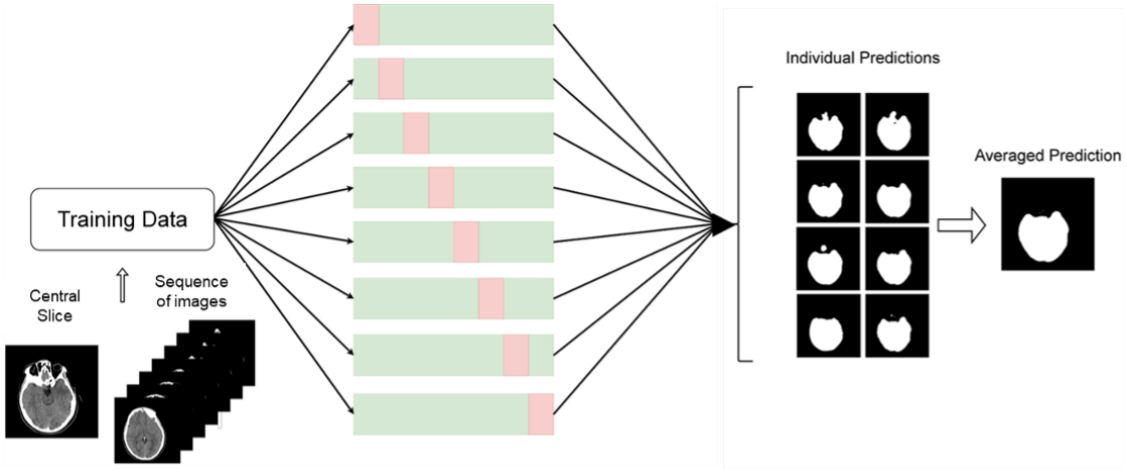


Figure 2.1: Example of how models produce predictions. First, the sequence input is introduced to the model. Then the 8 U-Nets each make a prediction, these predictions are averaged, and then a threshold is applied to filter out weak predictions.

2.2.4 Statistical Analysis

The metrics used to evaluate the model performances were the Dice-Sørensen Coefficient (DSC) and Hausdorff Distance (HD) [90, 91]. All evaluations were conducted using the held-out independent testing sets.

The 95% confidence intervals were calculated using a bootstrapping method with 1000 repetitions [92]. For comparing different models on an independent test set, a paired t-test was used to determine statistical significance [93]. Comparisons of models on different test sets used an unpaired t-test. To determine if the variance between the results was equal, a Levene's test was used [94].

2.3 Results

Mean DSC and HD metrics and 95% confidence intervals (CI) are shown in Table 2.1. In general, high performance was achieved for all models when testing on the independent np-TBI test set. While a performance reduction was observed in both DSC and HD for the p-TBI independent test set, the performance remained relatively high (DSC > 0.9).

Table 2.2: Mean DSC and HD performance metrics with the 95% CI of all models on the n-TBI and p-TBI test sets. The 95% CI was calculated using bootstrapping with 1000 repetitions [CI lower, CI upper]. n-TBI: non-penetrating. N is number of total slices. All testing sets are independent testing sets. Asterisks denote statistically significant differences when compared to Model A on the p-TBI testing set ($p<0.05$, $p<0.01$, $p<0.001$).

Model	**Training set type**	**Finetuned with**	**Testing set type**	**DSC**	**HD (mm)**
Model A	np-TBI $N = 1{,}118$	–	np-TBI $N = 176$	0.956 [0.938, 0.969]	9.76 [8.37, 11.2]
Model B	np-TBI $N = 1{,}118$	p-TBI $N = 524$	np-TBI $N = 176$	0.963 [0.950, 0.973]	9.52 [8.21, 11.0]
Model C	np-TBI + p-TBI $N = 1{,}642$	–	np-TBI $N = 176$	0.968 [0.958, 0.976]	8.86 [7.74, 9.99]
Model A	np-TBI $N = 1{,}118$	–	p-TBI $N = 154$	0.930 [0.906, 0.949]	16.7 [14.3, 19.1]
Model B	np-TBI $N = 1{,}118$	p-TBI $N = 524$	p-TBI $N = 154$	0.943 [0.926, 0.956]**	16.3 [14.0, 18.7]
Model C	np-TBI + p-TBI $N = 1{,}642$	–	p-TBI $N = 154$	0.952 [0.933, 0.967]***	12.3 [10.6, 14.1]***

When Model A was evaluated on the independent p-TBI testing set, a significant decrease in performance was observed in both the DSC and the HD ($p = 0.04$ and $p = 3.9x10^{-7}$, respectively).

Model C achieved the strongest performance and demonstrated a statistically significant improvement compared to Model A. When tested on the np-TBI cohort, DSC of 0.968 [0.958, 0.976] and an HD of 8.86 mm [7.74, 9.99] was achieved. Results for testing on the independent p-TBI test set showed a DSC of 0.952 [0.933, 0.967] and an HD of 12.3 mm [10.6, 14.1]. Both were a statistically significant improvement compared to the Model B trained on solely np-TBI data, thus suggesting including extreme cases is helpful for model performance. Model A, trained only with the np-TBI training set, performed worst on the independent test p-TBI cases, with a DSC of 0.930 [0.906, 0.949] and an HD of 16.7 mm [14.3, 19.1]. Finetuning with p-TBI did not significantly improve model performance when tested on either np-TBI or p-TBI independent test sets. This suggests that the more complex pathological conditions in p-TBI scans make it more challenging for the models to segment brain matter accurately. However, training with the combined np-TBI and p-TBI training sets improved segmentation performance on the independent p-TBI test cases, particularly in reducing HD.

To further explore the extremes of these results, the %HD was calculated and compared to the DSC as shown in Figure 2.2. Briefly, the %HD is the percentage of the effective

diameter from the manual segmentation to the HD of that slice.

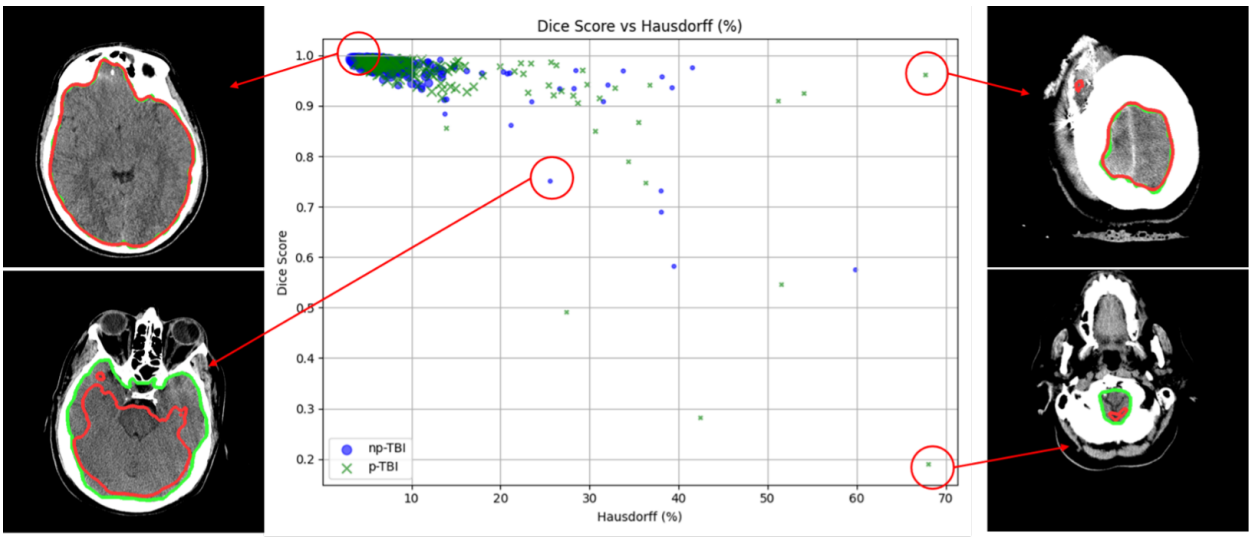


Figure 2.2: DSC vs %HD for each of the slices in the independent testing set. Individual slices shown are examples of extremes, with the green outline showing the manual ground truth and red showing the model prediction. Best segmentations are those with a low %HD and higher DSC, in the upper left-hand corner. These are predictions from Model C. Independent testing group set is shown as either a blue circle (np-TBI) or green cross (p-TBI)

2.4 Discussion

The study results indicate that a U-Net segmentation model, using a cross-validation ensemble technique, can effectively segment the brain matter of TBI HCT images. Performance of the models on np-TBI testing set was overall better than the models tested on the p-TBI testing set, as indicated by both the DSC and the HD. This performance loss when evaluating on the p-TBI independent test set, is likely due to the increase in artifacts, calvarial fractures, and deformities seen on these scans, which aren't as prevalent in the np-TBI cases.

As demonstrated with Model C, performance on both testing sets increased significantly when the models are trained with both p-TBI and np-TBI cases as compared to finetuning a model trained with only np-TBI images. This result is consistent with the literature showing that training with more data produces higher performing networks compared to

finetuning [95–98]. In this study, Model C was trained with a total of 1,642 images, whereas Model A and Model B were trained with 1,118 np-TBI images and Model B further finetuned with 524 p-TBI images. The finetuning process of Model B was only conducted using the last four layers of the U-Net. This approach was chosen to preserve the low-level features in the encoder, which capture generic image representations. By freezing these early layers and finetuning the later ones, the model could adapt its high-level representations to better capture injury patterns and imaging artifacts.

The optimal location for performance is in the upper left corner of Figure 2.2, with a low %HD and a high DSC. Model C predictions grouped primarily in the upper left corner of the plot, where both the manual segmentation and the model prediction aligned closely. On the far-right corner of the plot, the model's performance had a high DSC but also a high %HD. As shown, the segmentation and prediction align well, with a grouping of pixels also being incorrectly classified as brain. For predictions in this category, simple post-processing can be applied to the mask to adjust the output [99]. Finally, the lower-right example represents a poor prediction with both low overlap and large spatial discrepancy between boundaries, resulting in the lowest DSC and highest HD values.

While the overall performance of the models was high, their segmentation accuracy deteriorated when evaluating HCT images with substantial artifacts, as illustrated in Figure 2.3b. Qualitatively, both Model A and Model B have very similar predictions but have a large majority of brain structure missing. In contrast, Model C demonstrates the improved performance shown in Table 1. Although Model C's prediction is not the best overall, it clearly segmented more brain matter compared to Model A and Model B. This is an indication that Model C, trained with both np-TBI and p-TBI, can segment more effectively when there is a higher prevalence of artifacts.

2.5 Conclusion

This study evaluated three U-Net-based segmentation models for extracting brain matter from np-TBI and p-TBI HCT images. While all models performed well on held-out test sets, performance decreased on p-TBI cases due to artifacts. The model trained jointly on both cohorts demonstrated improved robustness to artifacts, highlighting the benefit of diverse training data. Results indicate that all models exhibited high performance on the held-out testing sets, with the performance decreasing when models were evaluating p-TBI cases. Model C, which was trained with both cohorts of data, showed improvement in segmentation performance when evaluating p-TBI data, indicating that this model can effectively segment brain matter even when present with some artifacts.

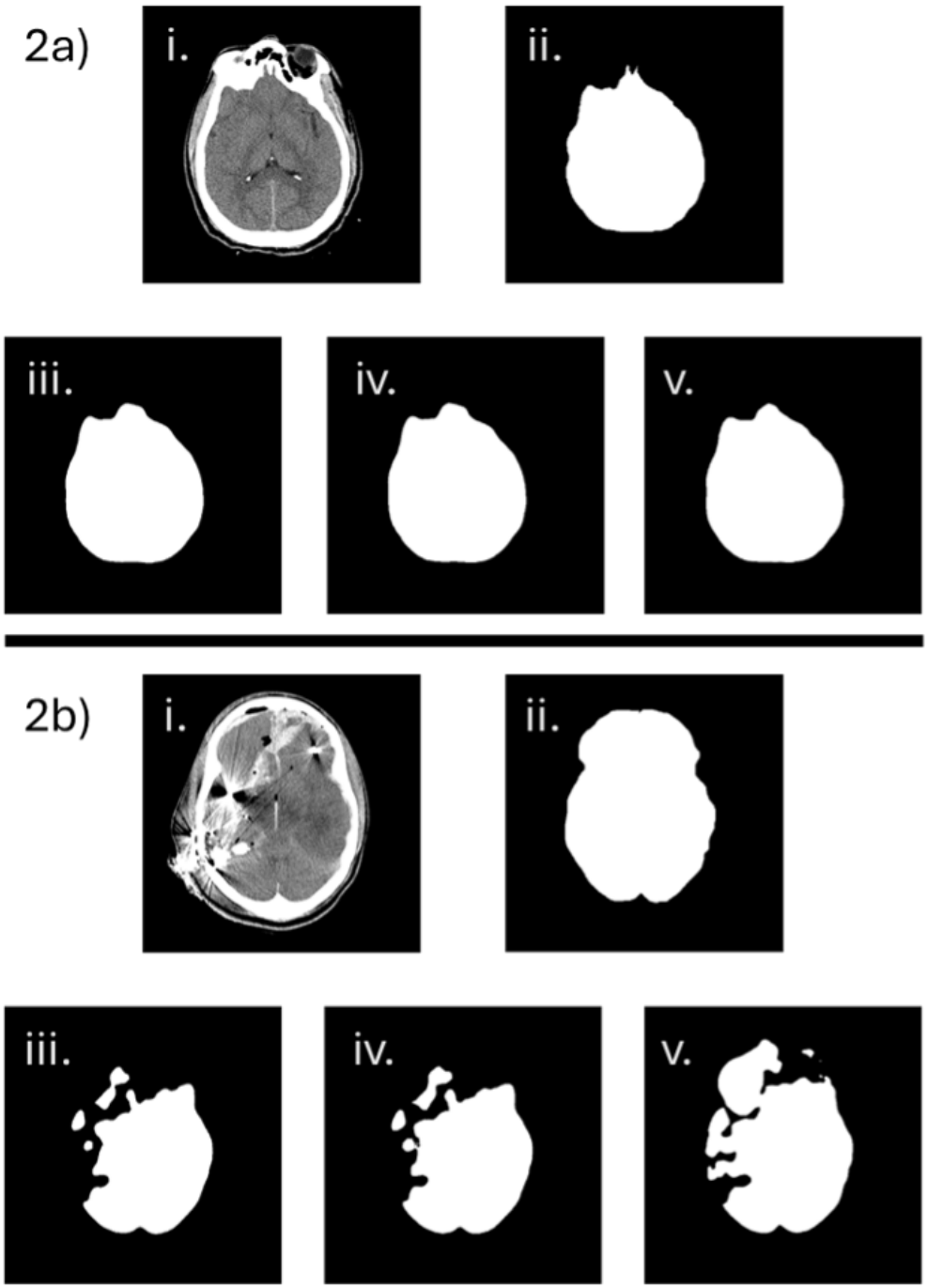


Figure 2.3: Example model predictions on testing sets. A) shows an example of a HCT image without artifacts and calvarial fractures. B) shows an example of a HCT image of a p-TBI patient with large amounts of artifacts and damage. Image i) shows the original CT image, image ii) shows the ground truth, image iii) shows Model A prediction, image iv) shows Model B prediction, and image v) shows the Model C prediction.

CHAPTER 3

DEVELOPMENT OF CLUSTER ANALYSIS METHODS IN THE IDENTIFICATION OF SHORTCUT INFLUENCES

3.1 Introduction

Cluster analysis is an unsupervised learning technique that is meant to find underlying patterns in the data [100–102]. Previously, this method has been used in medicine to discover disease subtypes, stratify patient risks, allocate resources, and analyze genomic data [103–106]. Due to the nature of clustering, underlying patterns are often used to guide further analysis rather than serve as a final result. Briefly, these techniques identify structures in the data by grouping datapoints whose features are similar to one another.

Clinical datasets often combine different data types ranging from EHR to medical images. Given the complexity of the data, cluster analysis is a useful tool for identifying clinically relevant patterns and structures. For example, cluster analysis has been used to identify distinct phenotypes from patients with sepsis due to bloodstream infections [107]. Another group used hierarchal clustering to identify patient subgroups of people diagnosed with rheumatoid arthritis [108]. Cluster analysis in AI is practical since many models convert imaging data into embeddings.

In histopathology, convolutional autoencoders have been used to group histology images, with resulting clusters interpreted as cancer subtypes [109]. Similarly, in oncology imaging, researchers have benchmarked multiple medical imaging foundation models to analyze their performance in deriving deep learning-based radiographic phenotypes for quantitative tumor imaging [110]. In ophthalmology, autoencoders trained on retinal optical coherence tomography images and color fundus photographs generate embeddings that, when clustered, reveal links to cardiovascular and neurodegenerative diseases. [111].

Despite the growing use of cluster analysis in medical imaging AI, there is no standard-

ized approach to evaluate clusters. While studies mentioned rely on measures like Calinski-Harabasz and silhouette scores to assess cluster separation in feature space, others depend on external validation using clustering algorithms and comparing those clusters to ground truth labels [112]. However, it is still common to assess clusters primarily on visual inspection without applying any quantitative criteria [113–116]. Due to the number of available dimensionality reduction techniques and clustering algorithms, and the lack of consensus on which evaluation metrics should accompany them, this process remains inconsistent across studies. Thus, it is difficult to compare clustering results across different datasets, assess the reliability of identified subtypes, or determine which combination of methods is most appropriate for a given imaging dataset. Therefore, we developed the Cluster Analysis Tool (CAT) which is designed to standardize and streamline cluster analysis for medical imaging AI and other high dimensional data. Figure 3.1 shows an overview of CAT workflow.

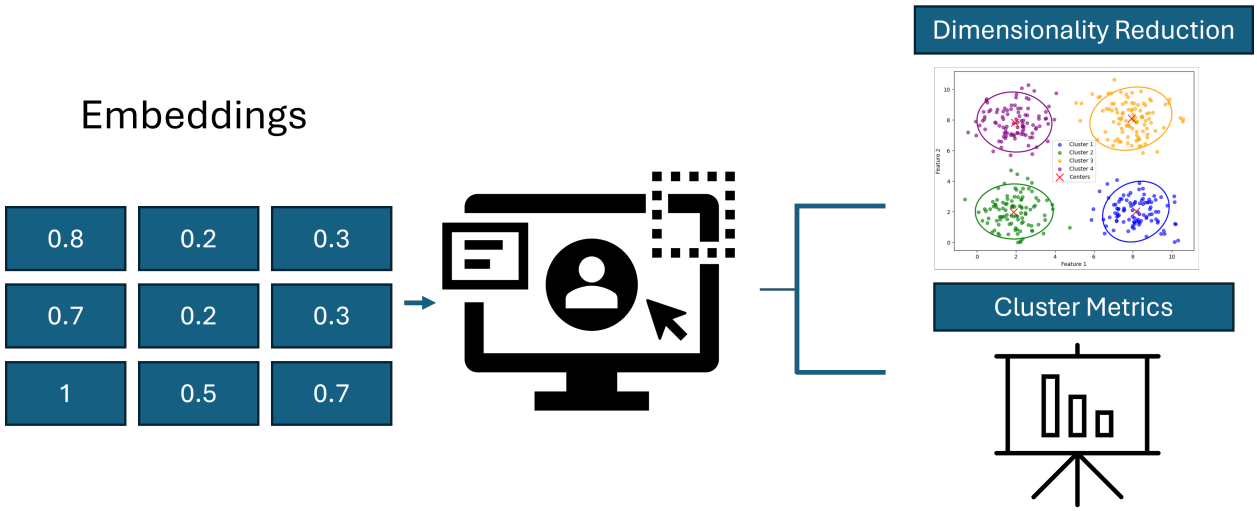


Figure 3.1: Overview of CAT. Extracted embeddings get uploaded into the software. Dimensionality reduction and clustering metrics are calculated to evaluate embedding clusters

3.2 Databases

3.2.1 Iris Dataset

The Iris Dataset is a database that was constructed by R.A. Fischer in 1936 [117]. This database contains measurements of three different species of Iris flowers, Iris Setosa, Iris Versicolor, and Iris Virginica. Measurements of 50 flowers from each species was taken, measuring the sepal and petal width and length. This dataset has been a staple in evaluating different techniques, including but not limited to, K-Means clustering, principal component analysis (PCA), logistic regression, and hierarchical clustering [118]. The Iris Dataset was used in our study as a benchmark dataset to demonstrate the functionality of CAT.

3.2.2 TBI Dataset

To assess the tools' ability to translate to clinical data, we employed the previously mentioned non-penetrating TBI dataset from The Prehospital Tranexamic Acid for TBI Trial [82]. This dataset contains about 1,000 cases of patients who underwent this trial that studied the effect of Tranexamic Acid and its efficacy for TBI patients. Of the patients enrolled in the trail, 673 cases had both imaging and clinical data and were included in this analysis. Details of the resulting dataset can be found in Table 3.1.

Table 3.1: TBI dataset demographics.

Number of Cases	673
Age	Age (mean ± std): 40.84 ± 18.39 Range: 18 – 94
Sex (M/F)	76%/24%
GCS (Severe/Moderate/Mild)	57%/40%/3%
Outcomes (Survived/Deceased)	86%/14%

For preprocessing, images were resampled to 2mm x 2mm x 5 mm with zero padding and

center cropped to 128 x 128 x 48. Finally, to incorporate multiple imaging windows (blood, subdural, bone, and stroke), the 4 windowed images were concatenated along the z-axis so that the model received a single-channel input (128 x 128 x 192). HCT scans from patients were processed using a 2D ResNet50 that was pretrained on ImageNet as shown in Figure 3.2. Since HCT scans are a 3D imaging modality, each slice was input into the architecture separately, thus creating a slice-level embedding. This process was repeated for each slice along the z-axis. All slice-level embeddings were then mean-pooled to obtain a final scan level embedding.

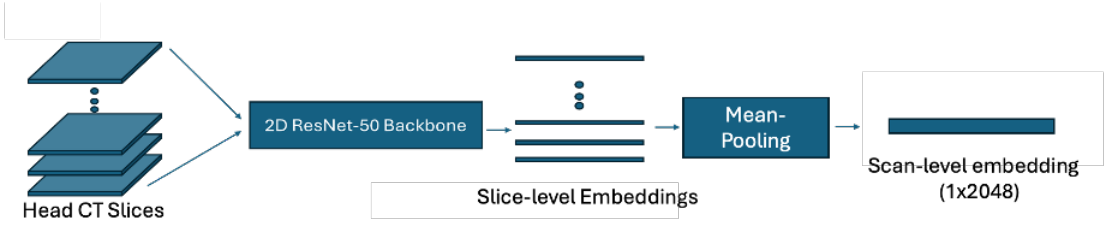


Figure 3.2: Processing of 3D HCT scan using a 2D ResNet50 model pretrained with ImageNet. Slices were processed sliced by slice, producing slice-level embeddings. All slice-level embeddings were then mean-pooled to produce a scan-level representation.

CAT takes in multiple file formats including csv and excel sheets. Input data can consist of embeddings, human engineered features, or a combination of both. Furthermore, relevant labels of interest can be included for further downstream analysis.

3.3 Methods

3.3.1 Clustering Algorithms

After loading data into CAT, users can select which features to include in the data analysis. Features serve as the input to a clustering step where users can select from K-Means, Fuzzy C-Means, or DBSCAN.

K-Means

K-Means clustering is one of the most popular clustering algorithms. Proposed by multiple researchers during the 1950's and 60's, the algorithm consists of 5 different components [119, 120].

1. Selecting the number of clusters, k.
2. Randomly select k points corresponding to selected number of clusters for the number of initial centers(if number of clusters is 3, then 3 points selected as center).
3. Using the Euclidean distance, calculate the distance between object and cluster center, then assign the object to the cluster with the smaller distance.
4. Recalculate the centroid of each cluster by taking the average position of each data point in that cluster. Update this as the new cluster center.
5. Repeat steps 3 and 4 until the same objects are found in the same clusters in consecutive iterations.

CAT allows users to set the parameters for K-Means such as number of clusters and random seed for replication. K-Means is computationally efficient, making it an ideal choice for large datasets. Although it's widely used for cluster analysis, results can vary from run to run due to centers being chosen at random. This algorithm is also sensitive to outliers because cluster centroids can end up being pulled towards the outlier direction. Additionally, K-Means struggles when clusters are not spherical or similarly sized, which is often the case with high dimensional biomedical or imaging data.

Fuzzy C-Means

In contrast to K-Means clustering, Fuzzy C-Means (FCM) is soft clustering algorithm. With FCM, every point is assigned a degree of membership, allowing that point to belong to

multiple clusters [121]. This is achieved by minimizing an objective function that represents the sum of the weighted distances between data points and the corresponding cluster centers.

1. Select number of clusters and fuzziness parameter (m >1, where higher values result in more overlapping assignments).

2. Randomly initialize a membership matrix and assign each point a membership for each cluster (0-1). The sum for each point across all clusters should be unity.

3. Calculate average position of all points, where all points are weighted by their corresponding membership value in that cluster. This is called the fuzzy centroid [121].

4. The membership degree of each point is then updated depending on its distance from the calculated fuzzy centroids.

5. Repeat steps 3 and 4 until the change in membership values between consecutive iterations falls below a defined threshold.

The membership degree for each point is well suited for clinical data, especially for points near cluster boundaries, where points can fall in separate clusters. While this flexibility can be useful in some settings, when it comes to interpretation, FCM clustering becomes more complex since final cluster assignments have to be derived from the membership matrix.

DBSCAN

Rather than cluster points relative to a centroid, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) assigns clusters based on point-to-point proximity. Proposed by Ester et al. in 1996, the algorithm consists of the following steps [122]:

1. Select ϵ and MinPts. ϵ is the maximum distance that a point can be to be considered in the neighborhood. MinPts defines minimum number of points in a neighborhood for a point to be considered a core point.

2. A random unvisited point is selected.

3. Calculate all distances to other points in the dataset and evaluate how many points fall within the ϵ.

4. If number of points that are within ϵ is greater than or equal to MinPts then that point of interest is considered a core point. If it's less than MinPts, then point of interest is temporarily considered noise.

5. Expand the cluster by iteratively adding all points within ϵ of the core point and checking whether any of those neighboring points are also core points, in which case their neighbors are added to the cluster as well.

6. Repeat 2-5 until every point is visited and has been labeled a core points or noise.

DBSCAN differs from K-Means and FCM because instead of assigning the number of clusters, that number is determined by the algorithm and the structure of the data. Another difference is how DBSCAN deals with outliers and noise. While K-Means and FCM are susceptible to outliers due to them being forced into clusters, DBSCAN can label them as noise.

3.3.2 Dimensionality Reduction Algorithms

Dimensionality reduction algorithms are useful for cluster analysis. Features and embeddings from models can vary greatly in length, with some models producing embeddings on the order of thousands of dimensions. As such, different dimensionality reduction techniques can aid in visualization and downstream analysis by projecting high-dimensional data into lower dimensional space. These techniques can vary different from one another; thus, we incorporated PCA, t-Distributed Stochastic Neighbor Embedding (t-SNE), Uniform Manifold Approximation and Projection (UMAP), and Factor Analysis of Mixed Data (FAMD) into CAT.

PCA

PCA is a linear dimensionality reduction algorithm that attempts to reduce the number of dimensions of the dataset while preserving as much important information as possible [123]. Essentially, PCA projects the data onto a set of orthogonal axes called principal components. These principal components are ordered depending on the amount of variance that they encapsulate with the first component being the highest, reducing variance with each subsequent component. By retaining the first few principal components, PCA can simplify high-dimensional datasets to lower dimensions. When reduced to two or three dimensions, we can visualize the data structure and how they relate to one another. While PCA is computationally efficient, it is a linear technique which may not capture complex non-linear structures present in the data.

t-SNE

Unlike PCA, t-SNE is a non-linear dimensionality reduction technique that is primarily used for visualizing high dimensional data into two or three dimensions [124]. t-SNE focuses on preserving local relationships between points rather than preserving the variance like PCA. Briefly, the algorithm converts pair-wise similarities between points into probabilities and then optimizes the lower dimension representation by matching the similarities. This approach reveals local structures within the data, making it well suited for visualization. However, this method is computationally expensive and non-deterministic, thus different runs can result in different representations.

UMAP

Another non-linear dimensionality reduction technique is UMAP, which aims to preserve local structure but in addition preserves global structure of the data [125]. UMAP uses manifold learning techniques and topological data analysis to reduce dimensions. Compared

to t-SNE, UMAP is computationally efficient, making it substantially faster. This is notable with larger scale datasets as those found in medicine. As such, UMAP has become a popular tool in machine learning, bioinformatics, and data exploration for visualizing complex datasets and generating informative low-dimensional representations.

FAMD

Since some features that users may want to analyze can be categorical, CAT also contains Factor Analysis of Mixed Data [126]. FAMD is a dimensionality reduction technique designed for datasets containing both numerical and categorical variables. FAMD generalizes PCA for continuous variables and Multiple Correspondence Analysis (MCA) for categorical variables, treating both variable types in a balanced way so that neither dominates the resulting projection.

3.3.3 Visualization Tools

Following clustering and/or dimensionality reduction, CAT allows users to visualize the reduced embeddings in either 2D or 3D, depending on the chosen dimension parameters. The plot uses matplotlib library to visualize results in a pop-up window. The initial plot following dimensionality reduction shows the datapoints color coded by the clusters from the selected clustering algorithm. To further explore datasets, CAT also allows users to color code datapoints from a selected variable.

3.3.4 Cluster Analysis Metrics

While clustering algorithms group data and dimensionality reduction provide a strong qualitative foundation for visualization of these groupings, cluster metrics can complement evaluation by providing additional quantitative measures. These metrics can range from internal

measures, which assess structures to the data itself, to external measures, which evaluate clusters against a known truth label.

Even though CAT can compute a range of different clustering metrics, the following section covers three representative metrics that cover both categories of quantitative measures. The Adjusted Rand Index (ARI) and Purity scores are external metrics that capture how well the cluster assignments compare to ground truth labels and how clean each cluster is, respectively [127, 128]. The Davies-Bouldin Index (DBI) is an internal metric that geometrically evaluates how compact and how separated each cluster is. These three metrics together provide a comprehensive overview of how well the data clusters not only geometrically, but in relation to labels that can be clinically relevant.

Adjusted Rand Index

The original Rand Index (RI), proposed by Rand in 1971, is a metric that evaluates the pair-wise agreement between data-driven clusters (i.e., from clustering algorithm) and a set of ground truth labels [129]. Following Equation 3.1, where a is the agreeing pairs in the same cluster and b are agreeing pairs in different clusters, we can calculate the rand index. The denominator $\binom{n}{2}$ gives the total number of pairwise combinations across the n points in the dataset. While the base RI gives us a general idea of how well the data-driven clusters correspond to the ground truth, it fails to account for agreement that would be expected by random chance [127].

The ARI addresses this limitation by adjusting the observed agreement using its expected value under a generalized hypergeometric model. Specifically, ARI is computed from a contingency table that summarizes the overlap between the two partitions. Thus, the ARI takes the form shown in Equation 3.2, which is bounded by 1 and equals 0 when the expected value is the same as the RI. It's worth noting that ARI can also be negative, indicating worse than random agreement. Like the RI, the ARI is symmetric, meaning that the score

is unchanged by the ordering or labeling of the clusters.

$$RI = \frac{a+b}{\binom{n}{2}} \tag{3.1}$$

$$ARI = \frac{RI - Expected(RI)}{max(RI) - Expected(RI)} \tag{3.2}$$

Davies-Bouldin Index

To assess the geometry of the clusters, the DBI evaluates both the intra-cluster spread and the distance between clusters. Introduced by David L. Davies and Donald W. Bouldin in 1979, this metric evaluates the clustering algorithm directly [130]. Unlike the ARI, the DBI does not require ground truth for comparison. Rather, it relies on the structure of the clustered data. As shown in Equation 3.3, for each cluster i, the similarity measure $R_{i,j}$ is computed with respect to every other cluster j, and the maximum value is retained. The DBI is then obtained by averaging these maximum similarity values across all (k) clusters. Equation 3.4 shows the cluster similarity, where the numerator indicates the intra-cluster spread of clusters i and j, and the denominator is the distance of the centroids between cluster i and j. As such, a lower DBI indicates both a well separated and tightly compact clusters, ranging from 0 – inf. Here S_i (S_j) denotes the average distance between the centroid $i(j)$ and each point of cluster $i(j)$. $M_{i,j}$ is the distance between centroid of clusters i and j.

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} \left(R_{i,j}\right) \tag{3.3}$$

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}} \tag{3.4}$$

Purity Score

The Purity Score is another external cluster evaluation metric that measures how "clean" clusters are by checking the most frequent label class in each cluster [128]. To calculate the Purity Score, each cluster is assigned a class that depends on the most frequent ground truth label in that cluster. After each cluster gets assigned a label, the number of correctly assigned data points whose ground truth label matches the dominant label of their cluster gets counted. The final score is the sum of correctly assigned points across all clusters divided by the total number of points as shown in Equation 3.5. Because each cluster will always have at least one dominant class capturing some fraction of the points, the theoretical minimum for is 1/k, where k is the number of clusters from the clustering algorithm.

$$\text{Purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j| \tag{3.5}$$

Here, $\Omega = [\omega_1, \omega_2, ..\omega_k]$ is the set of clusters and $C = [c_1, c_2, ..c_j]$ is the of classes. N is the total number of points in the dataset.

3.4 Use Cases

Overview of CAT Usage

CAT provides a simple yet flexible interface for users to perform clustering, dimensional reduction, and visualization. As mentioned in Section 3.2.2, the input file must contain feature data. As shown in Figure 3.3, the main UI prompts users to select the data that they want to use and preview it. The preview feature ensures users that they are using the correct dataset.

Prior to clustering and dimensionality reduction, it is important to set the parameters for each process respectively. While default settings are functional in most cases, CAT also

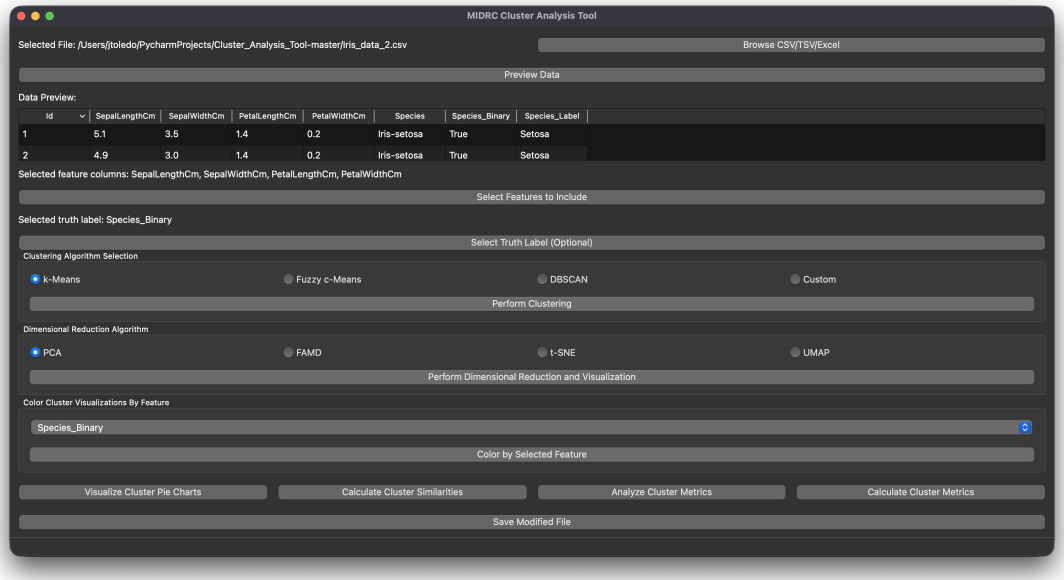


Figure 3.3: Overview of main interface of CAT

allows users to manually set parameters through a backend parameters window that contains both clustering and dimensionality reduction settings. Parameters related to clustering and dimensional reduction include random seeds, data standardization, and number of clusters/number of reduced dimensions. Selections in this window may have downstream effects that are automatically inferred for later processes (e.g., if 3 dimensions are chosen for DR, the plot will automatically plot in 3D).The backend parameters window is shown in Figure 3.4

On the main interface, users must select which features they want to use (i.e., which columns to include) for analysis. As discussed in Section 3.3.4, some metrics use truth labels and thus users have the option to select which column will serve as the reference label. Following selection of variables, by running 'Perform Clustering' and 'Perform Dimensional Reduction and Visualization' for their respective choices will populate the interface with the corresponding clustering assignments and reduced-dimensional visualizations. The resulting plot is that of the reduced dimensions color coded by clusters from the clustering algorithm chosen. By selecting 'Calculate Cluster Metrics' on the bottom right of the interface, a

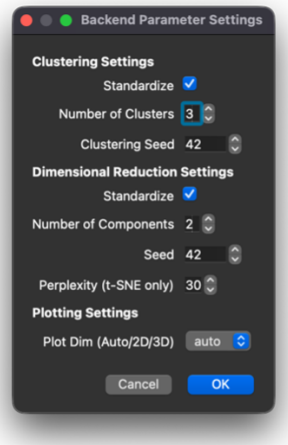


Figure 3.4: Backend parameters for clustering and dimensionality reduction algorithms.

separate window appears with a comprehensive set of evaluation metrics, with ARI, Davies-Bouldin Index, and Purity prominently shown at the top of the list for quick reference. This combination of visual and quantitative features in CAT allows for in-depth cluster analysis, addressing the limitations of relying on either approach alone. Example use cases demonstrating this workflow are discussed in the following sections.

Iris Dataset Example

The Iris Dataset was loaded into CAT as shown in Figure 3.3. The Preview Window shows the features of the Iris flowers. For this example, all four measurements were selected for cluster analysis. Since there are three species of Iris flowers, the number of clusters selected in the backend parameters window was set to 3. Selecting K-Means clustering and PCA for dimensionality reduction, the 2D plot for 2 component reduction (Figure 3.5Ai and Figure 3.5Bi) and 3D plot for 3 component reduction is shown (Figure 3.5Aii and Figure 3.5Bii).

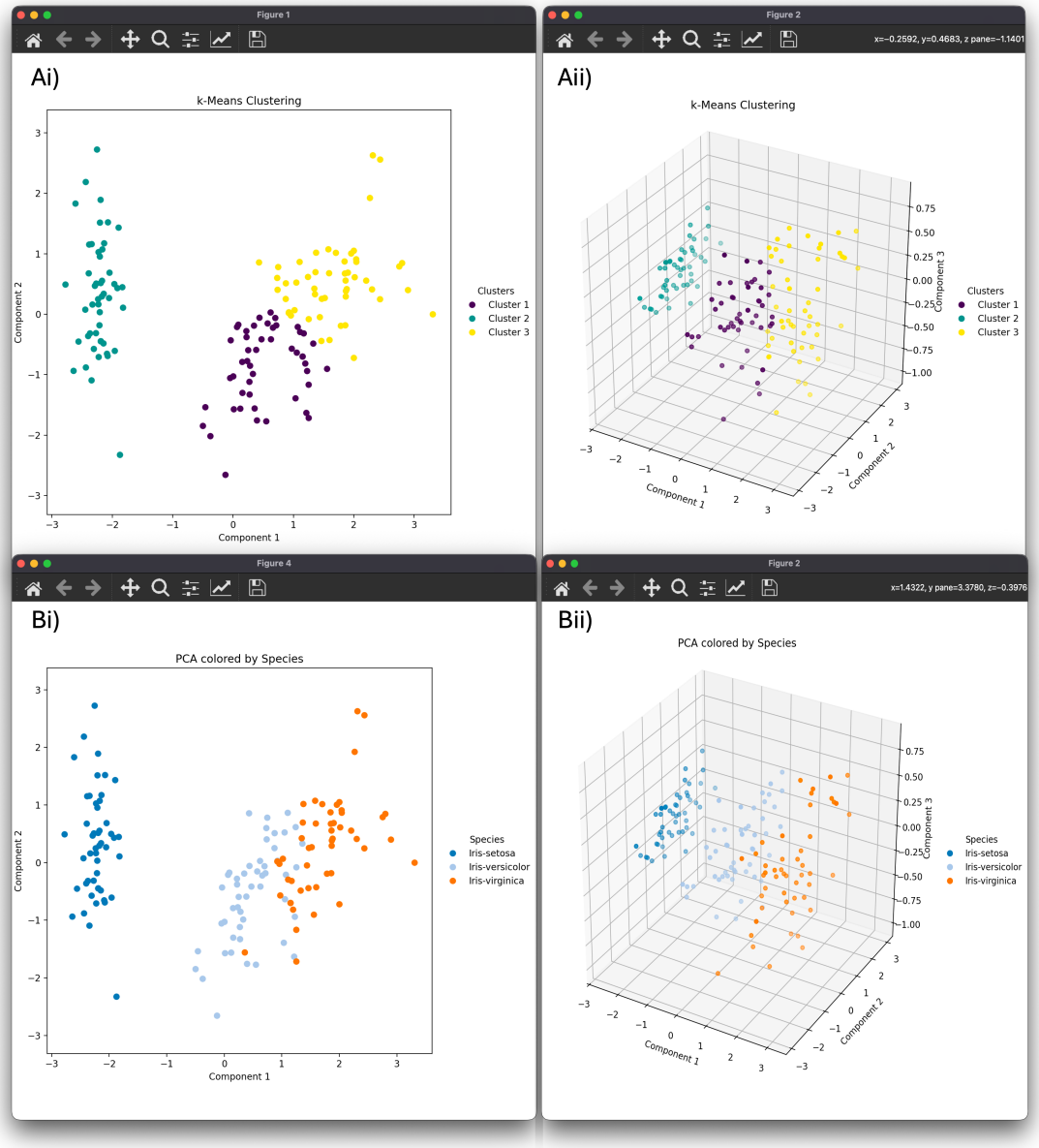


Figure 3.5: Example of tool output showing PCA plots of Iris flower features reduced to two components (i) and three components (ii). A) Points colored by K-Means cluster assignment. B) Points colored by ground-truth species label.

Figure 3.5 shows the resulting PCA plots of the Iris flowers features produced by CAT. Figure 3.5A shows the predicted label assignment from K-Means clustering and Figure 3.5B shows the same PCA plot colored by ground truth. By coloring datapoints by both K-Means clustering and the known species, we can directly compare the two and see if the clustering

algorithms cluster by features relevant to the known species. Iris Setosa formed a distinct grouping with no overlap with the other two species. Iris Versicolor and Setosa overlap as shown in Figures 3.5A(i) but overall show a boundary separating the two species. The separation between the features was also captured in the K-Means clustering groupings. CAT also supports visualization using different clustering algorithms and dimensionality reduction techniques as shown in Figure 3.6 where the same features undergo fuzzy c-means clustering and UMAP dimensionality reduction.

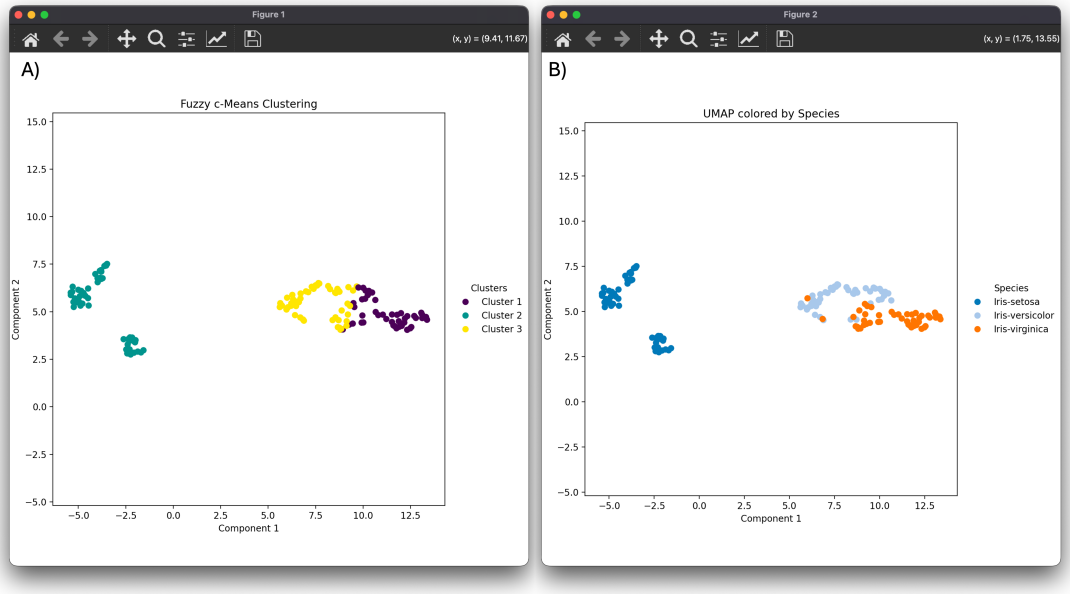
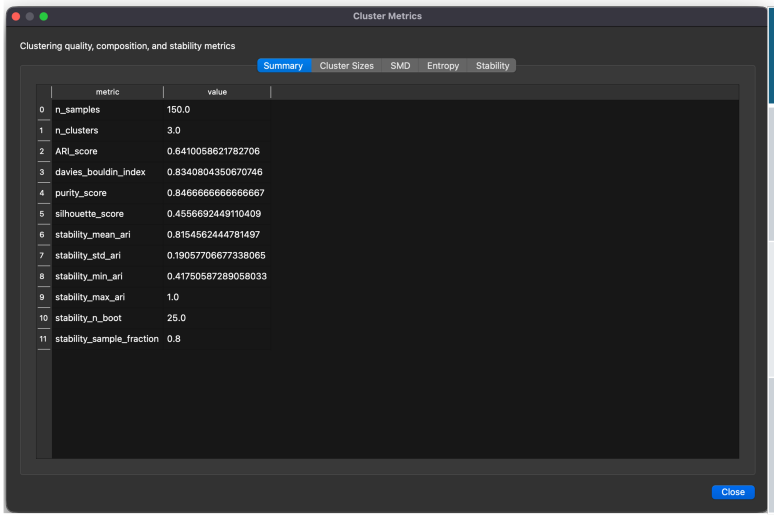


Figure 3.6: UMAP plots of Iris flower features. A) Points are colored by Fuzzy C-Means Clusters. B) Points are colored by Iris species.

These findings were reflected in the cluster metrics which are shown in Figure 3.7. CAT produces a summary with multiple different clustering metrics to give a comprehensive overview of clusters. The moderate DBI and imperfect purity score were consistent with the overlap observed between Versicolor and Virginica in Figure 3.5.



Metric	Value
ARI	0.64
DBI	0.83
Purity	0.85

Figure 3.7: Cluster metric summary produced by CAT. Results were for agreement between K-Means clustering with 3 groupings and ground truth species label.

To demonstrate what perfect separation looks like within CAT, we binarized the Iris dataset into two classes. Iris Setosa as one class and Iris Versicolor and Iris Virginica combined as the other. The PCA plots are shown in Figure 3.8

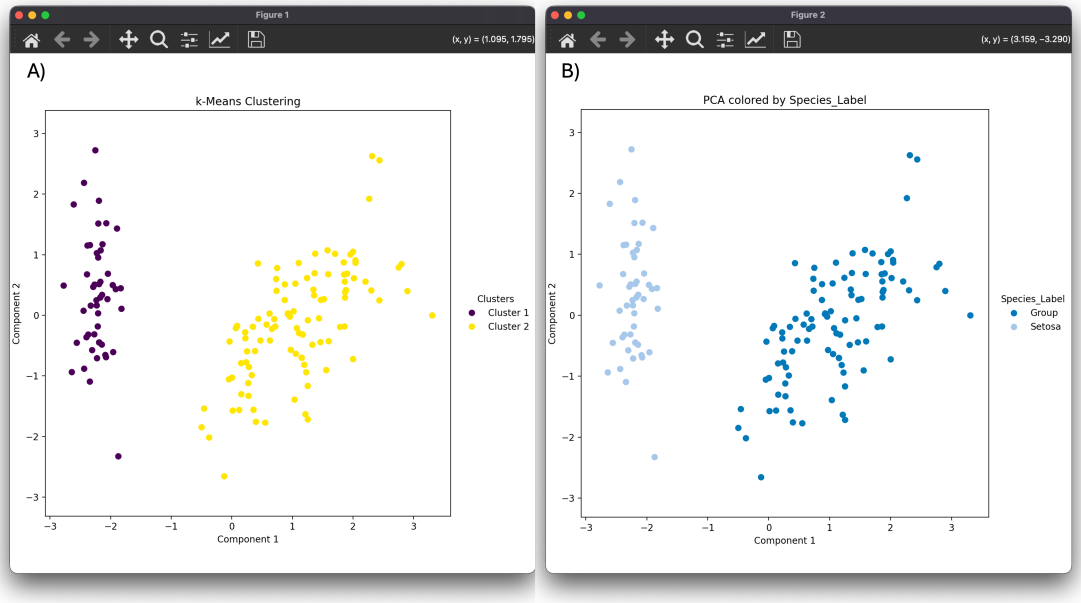
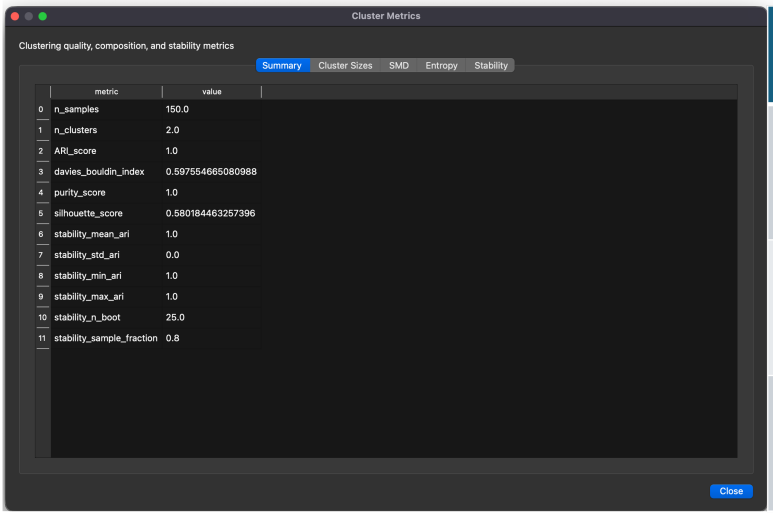


Figure 3.8: PCA plots of Iris features reduced to two dimensions. A) Colors plots by K-Means cluster (2 clusters). B) Colors plots by Iris Setosa or the grouped classification (both Iris Versicolor and Viriginica)

The K-Means algorithm clustering result closely corresponds to the binarized ground truth. Figure 3.9 shows the ARI and Purity Score at 1. This indicates there was perfect clustering between the predicted labels from K-Means and the binarized ground truth. The DBI was 0.60, which is expected since the intra-cluster spread between the two clusters is relatively large as noted on the second principal component. Notably, the DBI improved compared to the three-cluster solution, where the proximity between Clusters 1 and 3 in the reduced-dimensional space (Figure 3.5A) inflated the index by penalizing the overlap between Iris Versicolor and Iris Virginica.



Metric	Value
ARI	1.0
DBI	0.60
Purity	1.0

Figure 3.9: Cluster metric summary produced by CAT. Results are for agreement between K-Means clustering with 2 groupings and binarized species label.

TBI Dataset Example

The Iris dataset is a small but defined set whose features distinguish the different species well. As such, to evaluate CAT's capability to translate to clinical data, we evaluated the TBI dataset described in Section 3.2. Unlike the features in the Iris dataset, the embeddings extracted from models are a vector representation of different HCT scans.

Since mortality is a commonly used clinical variable, to begin, K-Means was first applied

with the number of clusters set to 2, matching the binary mortality label. For visualization, embeddings were reduced down to 2 dimensions using UMAP shown below in Figure 3.10.

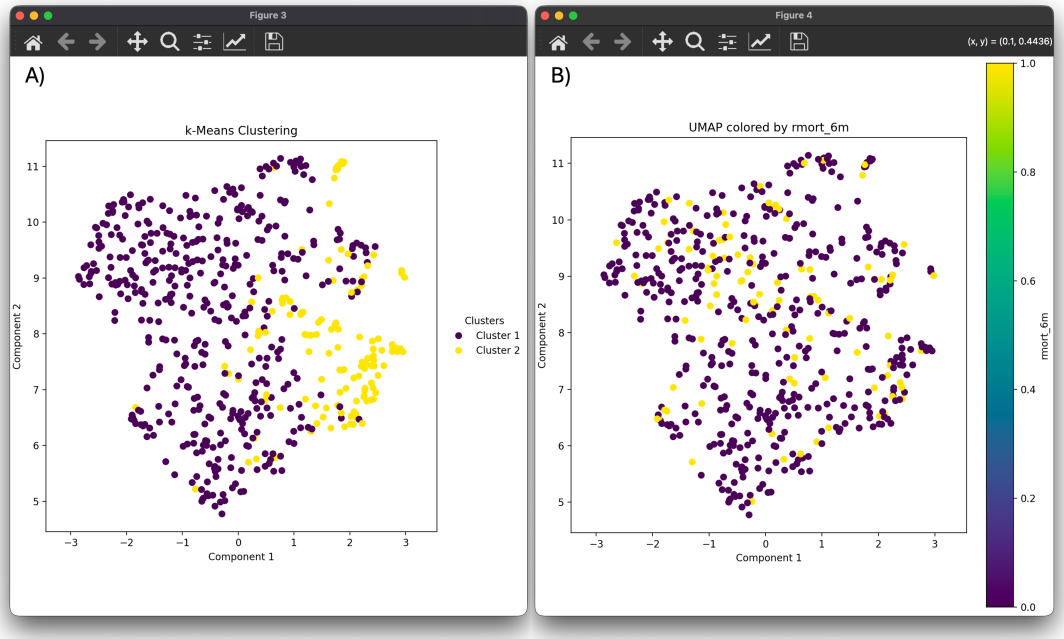
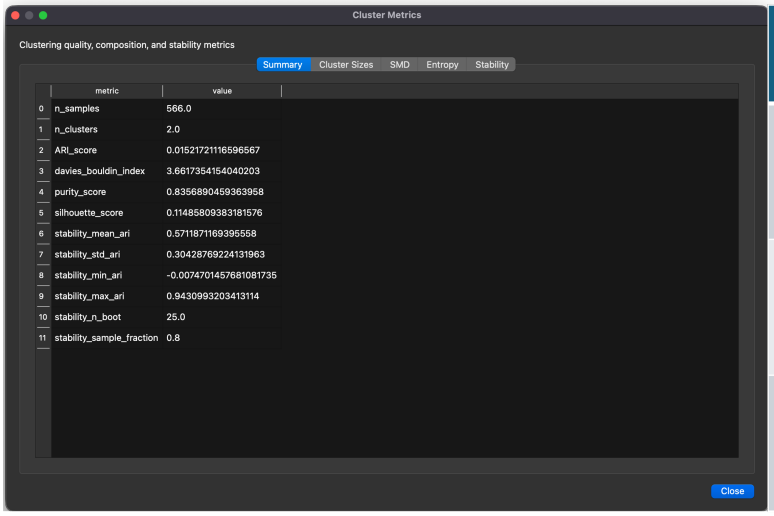


Figure 3.10: UMAP plots of TBI embeddings reduced to two dimensions. A) Colors plots by K-Means cluster (2 clusters). B) Colors plots by mortality at 6 months

In Figure 3.10A, UMAP is displayed with points colored by K-Means clusters. Figure 3.10B displays the same UMAP but with points colored by mortality at 6 months (1 deceased and 0 survived). Calculations of the metrics are shown in Figure 3.11



Metric	Value
ARI	0.02
DBI	3.7
Purity	0.84

Figure 3.11: Cluster metric summary produced by CAT. Results are for agreement between K-Means clustering with 2 groupings and TBI patient mortality at 6 months.

The metrics with respect to the K-Means clustering and mortality resulted in an ARI of 0.02, a DBI of 3.7, and a Purity Score of 0.84. The low ARI and higher DBI indicates poor agreement between the mortality truth and K-Means clustering, with clustering from K-Means containing large intra-cluster spread and relatively close centroid distances. Although Purity Score is high, the Cluster 1 from the clustering algorithm overlaps greatly with the survival labeled points, inflating the metric.

By making use of the color-coding functionality of CAT, it's easy to switch between labels to explore groupings present in the data. Coloring points by scanner manufacturer showed visible groupings as shown in Figure 3.12B. To compliment this, clustering was redone using K-Means setting the number of clusters to 3 (to match the number of vendors in the dataset). Corresponding cluster metrics are shown in Figure 3.13.

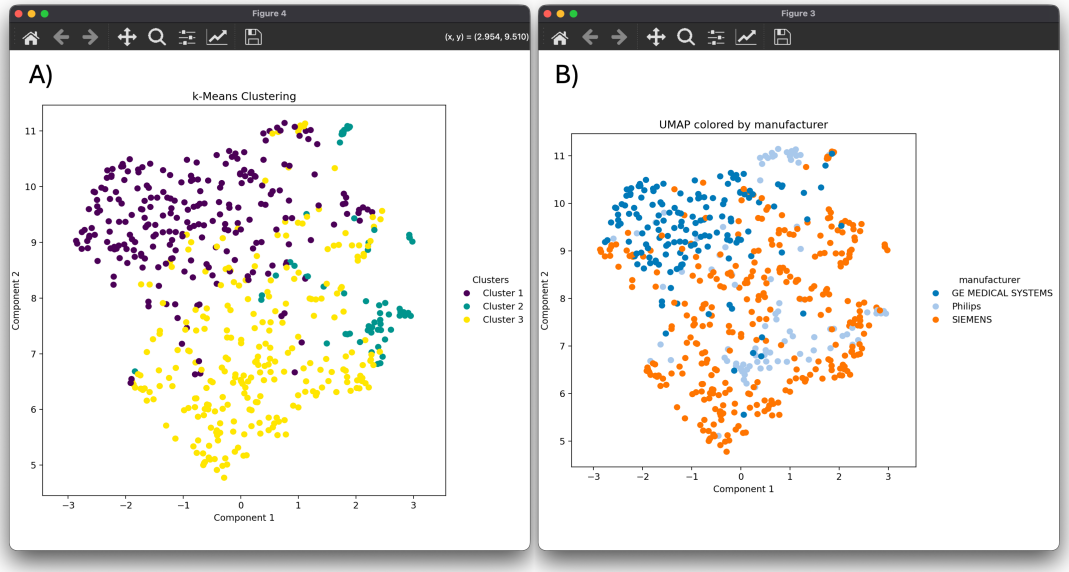
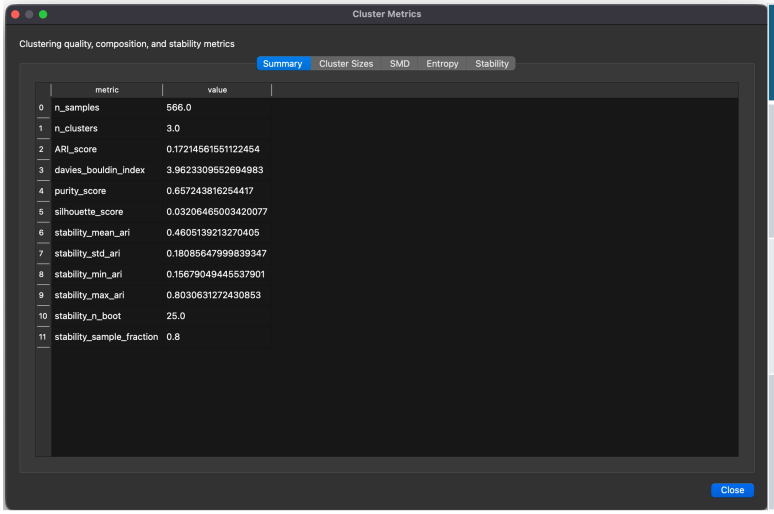


Figure 3.12: UMAP plots of TBI embeddings reduced to two dimensions. A) Colors plots by K-Means cluster (3 clusters). B) Colors plots by scanner vendor.



Metric	Value
ARI	0.17
DBI	4.0
Purity	0.66

Figure 3.13: Cluster metric summary produced by CAT. Results are for agreement between K-Means clustering with 3 groupings and scanner vendor.

Comparing the K-Means cluster assignments (Figure 3.12A) against the manufacturer ground truth labels (Figure 3.12B) reveals moderate visual correspondence, with regional tendencies suggesting that manufacturer identity captures some of the variance driving clus-

ter formation, though substantial overlap remains throughout the UMAP space. The metrics with respect to the K-Means clustering and scanner manufacturer resulted in an ARI of 0.17, a Purity score of 0.66, and a DBI of 4.0. Stand alone, these values indicate poor alignment, but when compared to mortality, that had an ARI of 0.02, the alignment is greater than random. Using the built-in cluster composition function, we can further evaluate how the clusters relate to the manufacturer labels (Figure 3.14).

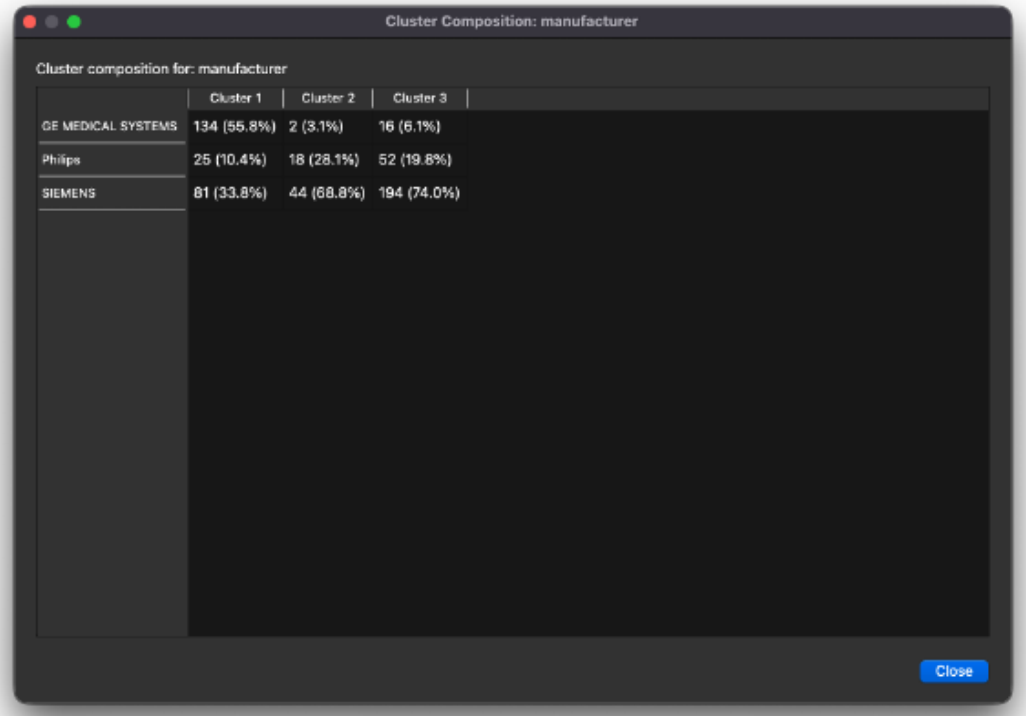


Figure 3.14: Cluster Composition from CAT. Shows how each cluster produced by clustering algorithm is comprised of reference label (scanner manufacturer).

Cluster composition shows how much of each cluster is comprised of the different labels. Note that Cluster 1 is mostly comprised of GE (134 cases) and Cluster 3 mostly comprised of SIEMENS (194 cases). This compositional breakdown provides a level of interpretive detail that the summary metrics alone cannot convey. While an ARI of 0.17 suggests only moderate overall agreement with manufacturer labels, the cluster composition reveals that

this moderate score is driven by strong manufacturer-specific separation in two of the three clusters, partially offset by the heterogeneity in Cluster 2. This illustrates how CAT's combination of summary metrics and compositional breakdown allows users to move beyond a single aggregate score and identify which specific subgroups are driving or obscuring cluster structure.

Outputs of CAT were verified using the same input data into a Jupyter notebook and evaluating each metric separately. UMAP plots and K-Means clustering were also evaluated by using the same seed input into the backend parameters window.

3.5 Discussion

CAT was developed to as a tool to streamline cluster analysis for medical imaging AI embeddings. By centralizing dimensionality reduction visualization, multiple clustering algorithms, quantitative evaluation metrics, and cluster composition breakdowns within a single interface, CAT enables researchers to move beyond single or limited metric are reported in the literature. As demonstrated across the use cases presented, there were several instances where one metric alone would have provided a misleading representation of clustering behavior. The examples showed the value of evaluating clusters using multiple metrics and visualizations rather than relying on a single measurement.

The Iris dataset use case established a foundation for interpreting CAT's outputs under controlled conditions where ground truth labels are known and cluster structure is well characterized. The three-cluster case produced metric scores with an ARI of 0.64, DBI of 0.83, and Purity of 0.85. These metrics illustrated the visually apparent but imperfect cluster separation, specifically at the boundary between Iris Versicolor and Iris Virginica. The binarized case then demonstrated what perfect cluster separation looks like within the tool, with ARI and Purity reaching 1.0 while the DBI reflected the expected intra-cluster spread of the merged class. Together these two cases illustrate how the same tool outputs

can be interpreted differently depending on the problem, and how the combination of visual and quantitative outputs provides a more complete picture than either alone.

The TBI use case introduced a more realistic and complex scenario, where the relationship between cluster structure and reference labels is not apparent. This demonstrated the tool's ability to explore data through clustering and dimensionality reduction, rapidly evaluating the same clustering solution against multiple reference variables. This revealed that scanner manufacturer was more of a source of variance compared 6-month mortality. This was further validated when the cluster composition output revealed that this moderate manufacturer alignment was driven primarily by strong separation between GE Medical Systems and Siemens scans, a finding that the summary metrics alone would not have surfaced. This has direct practical implications, as unrecognized manufacturer-driven clustering in medical imaging datasets can impact downstream analyses. Identification of non-clinically relevant factors that influence the data motivates the potential use of harmonization techniques to mitigate their effects prior to model training.

A current limitation of CAT is the absence of statistical analysis. Reported values correspond to a single seed rather than an average over multiple iterations. Chapter 4 shows statistics for clustering metrics to be incorporated into CAT. Additionally, CAT is currently a Python based tool and not an executable, meaning users will have to be familiar with Python environments and package management to be able to run the software. Future iterations of CAT will also implement interactive plotting, where users would be able to obtain metadata about the individual datapoints. This functionality will allow for a more complete and in-depth analysis of their dataset.

3.6 Conclusion

We present CAT, a tool that is meant to streamline and optimize cluster analysis of features and embeddings of medical datasets. CAT is designed to run various clustering algorithms,

dimensionality reduction techniques, and employ cluster metrics. These core functionalities enable both qualitative and quantitative evaluations of clusters. Through use cases with the Iris dataset and TBI dataset, we demonstrated that combining multiple evaluation metrics with interactive dimensionality reduction visualizations and cluster composition outputs allows for a more complete and reliable characterization of clusters within the dataset. Future work will focus on expanding supported algorithms and metrics, incorporating parameter selection guidance, and introducing robust statistics to further quantify clustering metrics.

CHAPTER 4

EVALUATIONS OF EMBEDDINGS FROM DEEP LEARNING MODELS AND SUBSEQUENT TASK-ORIENTED FINETUNING

4.1 Introduction

Deployed AI models often fail to achieve reported performance when tested on external groups due to the heterogeneity of various attributes present across multiple institutions [131]. Due to the variety of different scanner manufacturers, acquisition parameters, patient populations, and clinical protocols, AI models may erroneously rely on spurious correlations rather than meaningful clinical patterns found in the imaging data. When deep learning models convert imaging data into vector-level representations, known as embeddings, these spurious correlations are also encoded, ultimately influencing downstream tasks. When these encoded signals are correlated with a clinical label, deep learning networks are prone to exploiting this signal rather than use the underlying clinically relevant information. This phenomenon is known as shortcut learning, which can degrade model performance across different populations leading to missed diagnosis, poor segmentations, and incorrect predictions [132, 133]. Several studies have shown the presence of shortcuts in different imaging modalities. For example, one group developed a high performing model to predict COVID-19 severity from chest radiographs [134]. Even though the model achieved high performance metrics in the task of distinguishing between COVID-19 positive and negative (AUC > 0.99), it was observed, through saliency map analysis, that the model was making predictions based on radiographic markers rather than characteristics of the lungs. Another group found high performance of hip radiographs in relation to scanner manufacturers with a reported AUC of 1.0 [135, 136]. Through cluster analysis, they discovered that most positive hip fractures came from one scanner model, indicating the classifier was achieving high classification per-

formance due to the learned shortcut instead of the clinical signals.

Evaluation of these shortcuts is paramount to ensure that deployed algorithms can encode clinically relevant signals effectively, thus making them more generalizable. Steinmann et al. categorize shortcut detection strategies including model utility, perturbation, model explanations, and casual analysis [46]. Model utility approaches evaluate potentially learned shortcuts by assessing model performance under conditions that would break a shortcut. One commonly used method for assessing shortcuts is out-of-distribution (OOD) evaluation as it evaluates model performance using testing data outside the training data (i.e., another site dataset) [137]. A drop in performance on the OOD data could suggest that the model is relying on spurious signals instead of clinically relevant information.

In contrast, model explanation approaches evaluate potentially learned shortcuts by evaluating encoded information in the model. One method is linear probing which assesses the embeddings directly to determine if learned shortcuts can be predicted [138, 139].If shortcut variable can be predicted with high certainty from embeddings, this suggests that the embeddings have encoded spurious information. For visual analysis, dimensionality reduction techniques such as t-SNE, UMAP, or PCA can be applied to these embeddings to reveal how each case relates to one another, further validating learned shortcuts [123–125].

Despite the use of these OOD evaluations, linear probing, and visual-based approaches, several limitations remain. OOD requires access to well curated external test sets, which are often unavailable at the scale needed for clinical tasks [46]. Furthermore, this evaluation only reveals the existence of a shortcut when performance drops, offering no insight into what the learned shortcut is [46]. Linear probing is useful at identifying that an embedding has encoded a confounder. While it shows that the confounder is present in the data, it does not confirm that the model uses this information in its final predictions [140]. Visualization based approaches are qualitative in nature, thus providing only suggestive interpretation and lacking quantitative interpretability [140].

To address these limitations, this study evaluated the extracted embeddings from three separate deep learning architectures on two clinical datasets. The extracted embeddings were analyzed using both quantitative and qualitative methods including dimensionality reduction and clustering metrics to determine the extent of scanner manufacturer bias. Furthermore, all pretrained architectures underwent task-oriented finetuning to assess embedding behavior in relation to the model architectures pretraining regime.

4.2 Methods

4.2.1 Datasets

This study used two different databases to evaluate embeddings: 1) HCTs from the University of Chicago Covid-19 Data Mart and 2) HCTs from a np-TBI dataset from a randomized control trial, The Prehospital Tranexamic Acid for TBI Trial described in Section 3.2.2 hereto referred as the TXA TBI Dataset [82]. To sequentially finetune the different models, SinoCT was used [141]. Details of the datasets are shown in Table 4.1

Table 4.1: Breakdown of the different HCTs datasets used in the study. These HCTs came from (i) the UChicago Datamart, (ii) a multi-institutional clinical trial dataset, or (iii) the labeled SinoCT dataset of normal and abnormal HCT scans.

Dataset	**Number of Cases**	**Purpose**	**Number of Sites**	**Number of Vendors**
UChicago Datamart	485	Embedding Evaluation	1	2
TXA TBI Dataset	966	Embedding Evaluation	8	10
SinoCT	9,778	Finetune Models	1	1

Post Viral Infection HCTs from the University of Chicago Data Mart

The University of Chicago COVID-19 Data Mart is a collection of de-identified patients who have had a COVID-19 lab result. Images from patients who tested positive for COVID-19 and had a HCT scan were chosen, resulting in HCT images from a total number of 485 patients. In the dataset, there are HCTs primarily from 2 vendors. Specifically, the COVID-

19 dataset will be used in this study to evaluate embeddings extracted from deep learning models. Table 4.2 shows the demographics distribution of the COVID-19 dataset. DICOM files were converted to and stored in NIFTI format for subsequent input to pretrained models.

Table 4.2: UChicago COVID-19 Data Mart Dataset Patient Characteristics

Variable	**Counts**
Age (yrs), mean ± SD	65.0 ± 18.2
Sex at Birth	Male (243) Female (232)
Scanner Manufacturer	Philips (131) GE (344)
Site	UChicago (475)

TXA TBI Dataset

The TXA TBI Dataset, as described in Section 2.2.1, is a larger data set from a clinical trial evaluating the efficacy of TXA [82]. Demographics of the TXA TBI Dataset are shown in Table 4.3. Since the entire dataset consists of images from 8 different scanners and 10 different acquisition sites. To isolate the effects of site and vendor in this study while reducing the potential of confounding factors, the dataset was reduced to two sites corresponding to the most prevalent scanner manufacturers. Furthermore, HCTs from the vendor Neurologica were removed from the dataset since there was only 13 such scans. Figure 4.1 shows the breakdown of exclusion for this study. DICOM files were converted to and stored in NIFTI format.

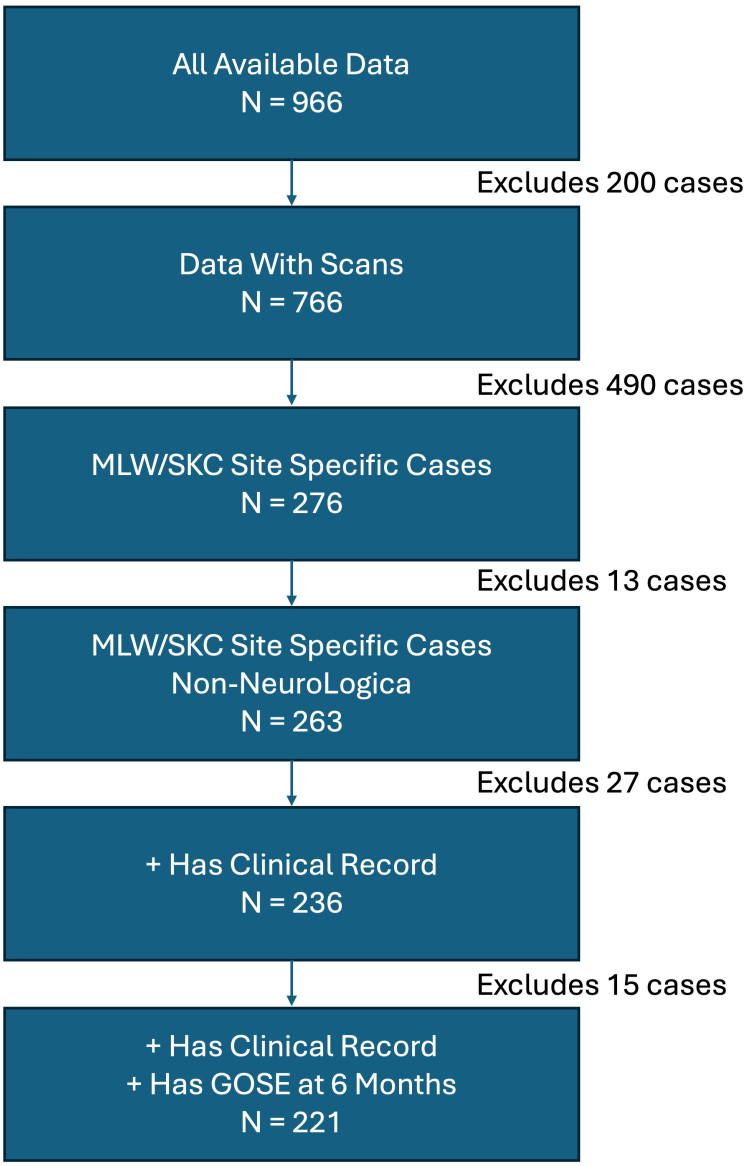


Figure 4.1: TXA TBI Dataset exclusion criteria for embedding analysis. Out of all the available data with Scans (766), the two sites with the two most prevalent scanner vendors were chosen. Total number of scans was further reduced if scans did not have clinical data available.

Table 4.3: TXA TBI Dataset Patient Characteristics

Variables	**Counts**
Age (yrs), mean ± SD	42.6 ± 17.9
Sex at Birth	Male (186) Female (50) Missing (27)
Scanner Manufacturer	SIEMENS (161) GE (102)
Site	SKC (173) MLW (90)
GOSE at 6 Months	1 (death) (34)
	2 (2)
	3 (23)
	4 (27)
	5 (5)
	6 (19)
	7 (32)
	8 (good recovery) (79)
	Missing (42)

SinoCT

The SinoCT dataset consists of 9,778 cases with both HCT scans and sinograms. Each case was labeled as either abnormal or normal. HCTs were obtained at Stanford University with all scans obtained from a GE scanner. Further details of the SinoCT dataset can be found in its original publication [141]. To remain consistent with finetuning procedures used by the authors of CT-FM, DICOMs were converted to NRRD file format [110].

4.2.2 Deep Learning Architectures and Embedding Extraction

ResNet50

For this study, a 2D ResNet50 was used which was pretrained on ImageNet, a dataset of 1.4 million natural images used to classify 1,000 categories [142]. Further details of ResNet50 are described in Section 1.3.2.

HCT inputs to the pretrained ResNet50 architecture were resampled to 2.0 mm x 2.0 mm x 5.0 mm spacing using bilinear interpolation. Volumes were then padded and center-cropped, resulting in a 128 x 128 x 48 scan with a single channel. To incorporate multiple imaging windows different HU ranges were used. The windows were blood (0–80 HU), subdural (-125–175 HU), stroke (28–36 HU), and bone (-900–2100 HU). Each window was normalized respectively. The 4 windowed images were then concatenated along the z-axis. Furthermore, since the ImageNet pretrained ResNet50 expects a 3-channel RGB input, the grayscale inputs to the ResNet50 architecture replicated the slices across the 3 channels prior to processing. Preprocessing steps are shown in Figure 4.2A.

To obtain embeddings, 2D slices were input into the model, using the output of the penultimate layer prior to the classification head. This resulted in a 2048-dimension slice-level embedding. This process was repeated for each 2D slice in the input data, resulting in 192 slice-level embeddings (48 for each of the four windowed images). Lastly, a scan-level embedding was obtained by mean-pooling across the slice-level embeddings, resulting in a final 2048-dimension embedding.

RAD-DINO

To evaluate how model pretraining influences embedding structure depending on pretraining domain (i.e. pretrained on natural images compared to pretraining on medical images), RAD-DINO was used. This 2D model was pretrained using about 882,000 chest radiographs [40]. Details on model architecture and pretraining are discussed in Section 1.3.3.

Volumes were resampled to 2.0 mm x 2.0 mm x 5.0 mm spacing using bilinear interpolation. RAD-DINO was originally pretrained on chest radiographs resized to 224x224 pixels. Thus, HCTs processed by RAD-DINO were padded and center-cropped to 224 x 224 x 32 for consistency [40]. The reduced depth of 32 slices (relative to the 48 used for ResNet50) was necessary to accommodate GPU memory constraints introduced by the larger 224 $\times$

224 spatial resolutions. A single blood window (0-80 HU) was applied and then normalized. This process is shown in Figure 4.2B.

RAD-DINO is a vision transformer that outputs a 768-dimension embedding which represents the 2D slice. Like ResNet50, the 2D slices were processed individually to obtain slice-level embeddings. The final stack was mean-pooled across the slice-level representations to obtain a final scan-level embedding.

CT-FM

To assess the dataset's CT imaging domain, CT-FM was used in this study. CT-FM is an architecture that was pretrained on 148,000 CT scans (including head, chest, abdomen, and extremities) [110]. Details of pretraining and model architecture are further discussed in Section 1.3.4.

As shown in Figure 4.2A, HCT input volumes for CT-FM followed the same preprocessing pipeline as ResNet50. While CT-FM was pretrained on fewer images (counting each scan as a single input), it trades training size for a 3D architecture, enabling axial data to be encoded. CT-FM has a SegResNet encoder that outputs a final 512-dimension embedding from the adaptive average pooling across the 3D feature maps.

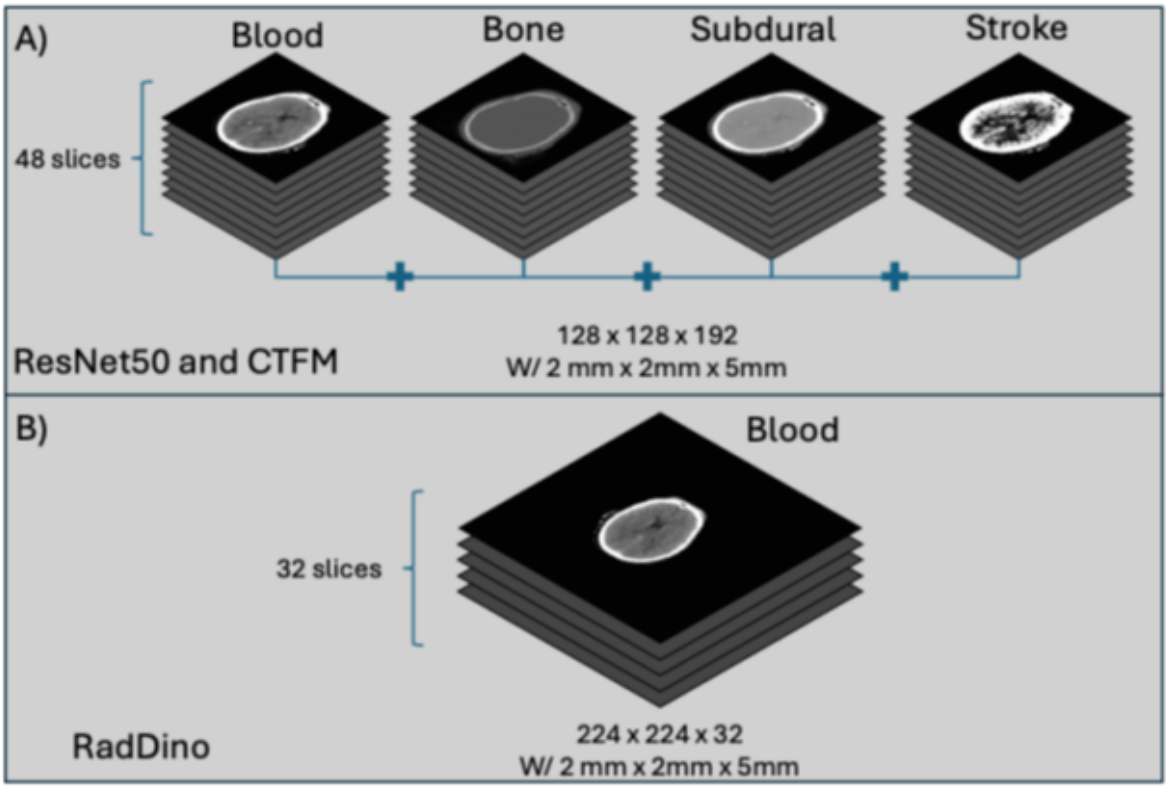


Figure 4.2: Preprocessing of HCTs for the different architectures in this study. A) Shows the different HU windows applied to the HCT scan. After windowing, each stack of the same brain scan gets concatenated to be processed by ResNet50 or CT-FM. B) Single blood window is applied with a higher resolution but a decrease in depth for the RAD-DINO inputs.

4.2.3 Dimensionality Reduction

As discussed in Section 3.3.2, extracted embeddings are high-dimensional vectors. Therefore, dimensionality reduction techniques are needed to visualize how HCTs relate to one another. Prior to reducing the dimensionality of the embeddings, the embeddings were first standardized to ensure that no single dimension dominates. When dimensions contain large numerical ranges are present it can disproportionately influence distance-based metrics such as UMAP, regardless of underlying context. As shown in Figure 4.3, scan-level embeddings were reduced to 2D using UMAP. To qualitatively assess potential underlying patterns, UMAP plots were colored by differing variables such as clinical variables and physical acquisition parameters.

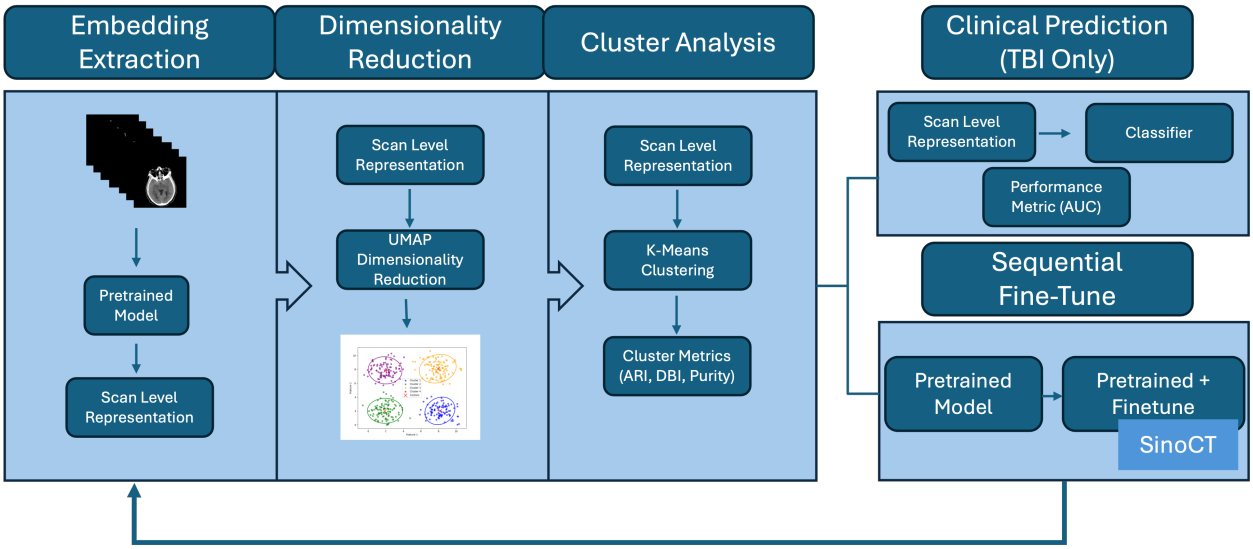


Figure 4.3: Flow chart of methods for embedding evaluation. First HCT embeddings are extracted from deep learning architectures to obtain scan-level representations. Qualitative and quantitative analysis is done on these extracted embeddings to determine the how cases relate to one another. For the TXA TBI Dataset, embeddings are used to make the clinical prediction of GOSE ≤ 3 at 6 months. Processes can be redone following sequential finetuning of pretrained deep learning architectures.

Primarily for our study, scanner manufacturer was the variable of interest. UMAP was selected over other dimensionality reduction techniques, such as PCA and t-SNE. In contrast to PCA, UMAP captures nonlinear relationships in the data while preserving both local and global structure of the data [125]. Compared to t-SNE, UMAP is computationally efficient, making it ideal for fast visualization of larger embedding sizes in this study. UMAP was fit separately per architecture.

UMAP was implemented using the umap-learn Python package (version 0.5.12) [125]. Embeddings were reduced to two dimensions for each model using UMAP, with number of neighbors = 15, minimum distance = 0.1, Euclidean distance, and a fixed random seed (42) to ensure reproducibility.

4.2.4 Cluster Analysis

Dimensionality reduction techniques described in the previous section are limited for cluster analysis because human evaluation is limited to what the eye can perceive in 2D or 3D space. Section 3.3.1 described several clustering algorithms used to evaluate groupings and patterns encoded in embeddings; "seeing" beyond 2D or 3D. Figure 4.3 shows how the extracted embeddings were clustered using K-Means clustering [119, 120]. K-Means was used over other methods like FCM since the primary ground truth label being evaluated (scanner manufacturer) was a categorical label, with every scan originating from a single vendor. Furthermore, external evaluation methods such as the ARI and Purity Score require hard assignments to compare ground truth labels. In contrast, DBI doesn't require ground truth labels, relying instead on the geometric properties of the clusters themselves. Scan-level embeddings were reduced to two dimensions using PCA since both K-Means and DBI rely on Euclidean distances [119, 130, 143]. As the number of dimensions of the embeddings grows (embeddings from extracted architectures have dimensions of 512, 768, or 2048), the distance measurement becomes less meaningful due to the curse of dimensionality [144]. Following dimensionality reduction, reduced embeddings were clustered using K-Means with the number of clusters equal to the number of unique labels in the ground truth. For example, since the ground truth label being explored is scanner manufacturer, which contains two unique values for the COVID-19 dataset, then the number of clusters from K-Means would be set to two. Clustering was performed using the K-Means algorithm, with k-means++ initialization, as implemented in scikit-learn (version 1.7.2) [145]. Similarly, PCA dimensionality reduction was also done using scikit-learn.

To assess how well K-Means clusters aligned with truth labels, both the ARI and Purity Score were used (see Section 3.3.4). The DBI was used to evaluate the intra-cluster spread and distance from each cluster centroid of the reduced embeddings [130]. The ARI is an external clustering metric used to measure the similarity between the truth label and the K-

Means clustering by evaluating all pairs of cases [127]. The Purity Score is another external evaluation metric that assesses how much a cluster contains a single class [128]. The ARI and DBI were calculated using the scikit-learn (version 1.7.2) Python package [145].Purity Score was calculated following the formulation of Manning et al. [128]. All these cluster metrics are discussed in further detail in Section 3.3.4.

4.2.5 Subsequent Use in Predictive Stage

Next the goal was to investigate how embeddings with potential short cuts might affect a classification task. Thus, to assess how embeddings extracted from pretrained models could classify a clinically relevant task, a simple Support Vector Machine (SVM) classifier was used on the TBI dataset. The clinical variable chosen was the Glasgow Outcome Scale Extended (GOSE) at six months. Briefly, the clinician-reported GOSE score measures a patient's disabilities and recovery following a TBI incident [146]. The scale ranges from 1 to 8, with 1 indicating a deceased patient and 8 indicating good patient recovery. Scale details can be found in Table 4.4.

Table 4.4: GOSE scores and respective outcomes

GOSE Score	**Outcome**
1	Deceased
2	Vegetative State
3	Lower Severe Disability
4	Upper Severe Disability
5	Lower Moderate Disability
6	Upper Moderate Disability
7	Lower Good Recovery
8	Upper Good Recovery

There is no standard cut-point to the GOSE to describe the "favorable" and "unfavorable" outcomes [147]. For this study, the "unfavorable" outcomes were defined as GOSE ≤ 3, while "favorable" outcomes were defined as GOSE > 3. This threshold was initially intended to

isolate severe patients (GOSE 1-2) but given the limited number of cases in this range, GOSE = 3 was included in the "unfavorable" category to provide sufficient cases for model training.

Extracted embeddings were first reduced to the corresponding number of principal components that capture 95% of the variance. The reduced embeddings served as the input into the classifier, with the ground truth being the binarized GOSE ≤ 3 at 6 months. Data were split into 80%/20% for training and testing respectively. The SVM classifier was trained with an RBF kernel, class weights adjusted to account for label imbalance, and probability calibration via 5-fold cross-validation (Platt scaling, non-ensemble) to produce reliable probability estimates. Model performance was evaluated using the receiver operating characteristic (ROC), where the area under the curve (AUC) was the performance metric in the task of distinguishing between GOSE ≤ 3 and GOSE > 3 at 6 months.

4.2.6 Role of SinoCT in Finetuning Each Pretrained Architecture

To evaluate how underlying embedding patterns shift across these models given a task-oriented goal, each model was sequentially finetuned. The SinoCT dataset is a large set of HCT scans, labeled either abnormal or normal [141]. The SinoCT dataset was divided into 80%/10%/10% (training, validation, and test) to match the author's evaluation setup for CT-FM [110]. To get models to converge, each model required different training and classification head hyperparameters. Training parameters for each model are found in Table 4.5.

All architectures used in this study had a different pretraining imaging modality, the role of SinoCT was to finetune the architectures using the specific modality and anatomy of the COVID-19 Data Mart and TXA TBI datasets. The curators of the SinoCT dataset labeled the HCT scans of patients with either a normal or abnormal label [141]. By attaching a classification head and finetuning the parameters of each architecture, the model learns to process the HCT inputs for a specific task, rather than relying on patterns learned from the

Table 4.5: Training Parameters

Parameter	**ResNet50**	**RAD-DINO**	**CT-FM**
Backbone	2D CNN	2D ViT	3D CNN
Intensity Windows	4	1	4
Input Size	128x128x48	224x224x32	128x128x48
Learning Rate	1e-4	1e-2	1e-4
Scheduler	Warmup-Cosine	Cosine Annealing	Warmup-Cosine
Max Epochs	40	40	20
Loss Function	BCE with Logits Loss	BCE with Logits Loss	BCE with Logits Loss
GPUs	4 (DDP)	4 (DDP)	1

specific pretraining (e.g., natural images, or a different imaging modality entirely).

Both ResNet50 and CT-FM were pretrained using the Lighter framework, which was built on Project Lighter and MONAI [148, 149]. This framework was used to replicate the results presented by the CT-FM authors [110]. RAD-DINO was finetuned using a custom loop using PyTorch (version 2.5.1). Every architecture used Binary Cross-Entropy with Logits Loss. The classification head for all models follows a two-layer multilayer perceptron (MLP) layout. Details are shown in Table 4.6.

Table 4.6: Classification head hyperparameters for each model.

Parameter	ResNet50	RAD-DINO	CT-FM
Hidden Dimension	1024	384	256
Activation	ReLU	ReLU	ReLU
Dropout	0.5	0.3	0.3
Output	Binary Logit	Binary Logit	Binary Logit

Final weights were selected based on the best validation AUROC, and training ended early if there was no improvement for 10 epochs. The final weights were used to evaluate the held-out test set. Following sequential finetuning of each model, the embeddings from both the COVID and TBI datasets were extracted. Dimensionality reduction, cluster metrics, and clinical predictions using these embeddings were analyzed.

4.2.7 Performance Evaluation and Statistical Analyses

For each of the cluster metrics, confidence intervals were calculated using 1,000 bootstrap iterations [150]. In each iteration, the embeddings were sampled with replacement. Each cluster metric was then recomputed for that respective sample. The resulting distribution of estimates was used to calculate the upper and lower bounds of the 95% confidence interval (CI).

To determine statistical significance when comparing cluster metrics between pretrained and sequentially finetuned architecture embeddings, the cluster metric difference in cluster metrics was calculated. If the 95% CI of the difference excludes 0, then the difference was considered to be statistically significant [151, 152].

To analyze performance metrics for disease classification from the extracted embeddings, CIs were constructed using 1,000 bootstrap iterations [150]. In each iteration, the test set probabilities and ground truth labels were resampled with replacement such that the resulting indices had the same length as the original test set size. AUC values were then calculated for each iteration, and the resulting distribution was used to estimate the 95% CI upper and lower bounds. Similar to the cluster metrics, the AUC difference was calculated between the pretrained and sequentially finetuned models. To account for multiple comparisons, Bonferroni correction was applied (p-value $\times$ number of comparisons), with a corrected p-value < 0.05 considered statistically significant.

4.3 Results

The extracted embeddings from each of the three architectures were qualitatively and quantitatively assessed using dimensionality reduction and cluster metrics. In addition, the qualitative and quantitative results for both the pretrained and sequentially finetuned architectures were generated to assess the effect of task-oriented finetuning on the extracted embeddings.

4.3.1 Dimensionality Reduction

University of Chicago COVID-19 Data Mart

To qualitatively assess the effects of sequential finetuning on embeddings extracted from deep learning models, the embeddings were reduced to 2 dimensions using UMAP. Figure 4.4 shows the embedding patterns in this reduced space, with each data point (individual patient scan) colored by scanner manufacturer for the COVID dataset. Visual inspection suggests that the embeddings from the pretrained models are highly clustered by vendors, but these groupings are reduced following the task-oriented finetuning with SinoCT. The embeddings extracted from sequentially finetuned ResNet50 and RAD-DINO show vendors highly intermixed, while CT-FM still shows grouping, though less separated.

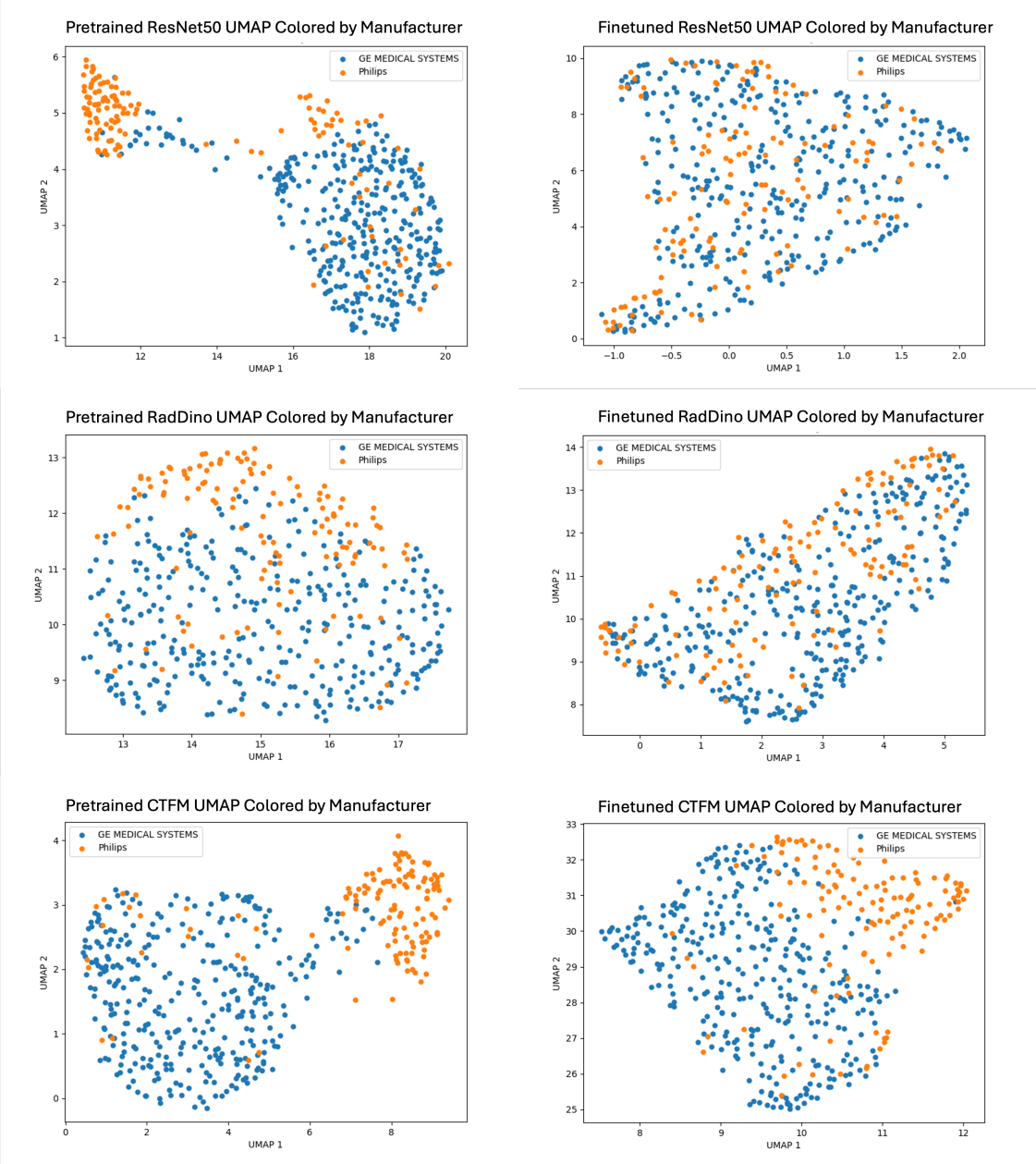


Figure 4.4: 2D UMAPs of the extracted embeddings from the COVID-19 dataset using pretrained (left) and sequentially finetuned (right) models. From Top down the models are ResNet50, RAD-DINO, and CT-FM. Cases (data points) are colored by scanner manufacturer. UMAP 1 and UMAP 2 represent the reduced dimensions of each embedding produced by UMAP dimensionality reduction.

TXA TBI Dataset

The resulting UMAPs from the TBI dataset are shown in Figure 4.5 below, with data points colored by scanner manufacturer. Visually, all the pretrained architecture embeddings show vendor grouping. While clusters do not appear compact, they are well separated. Following sequential finetuning of the model with SinoCTs, the vendor separation appears to collapse for ResNet50. Cluster shapes tied to the vendor tend to spread out and collapse; the relative patterns remain visible even after sequential finetuning of RAD-DINO and CT-FM.

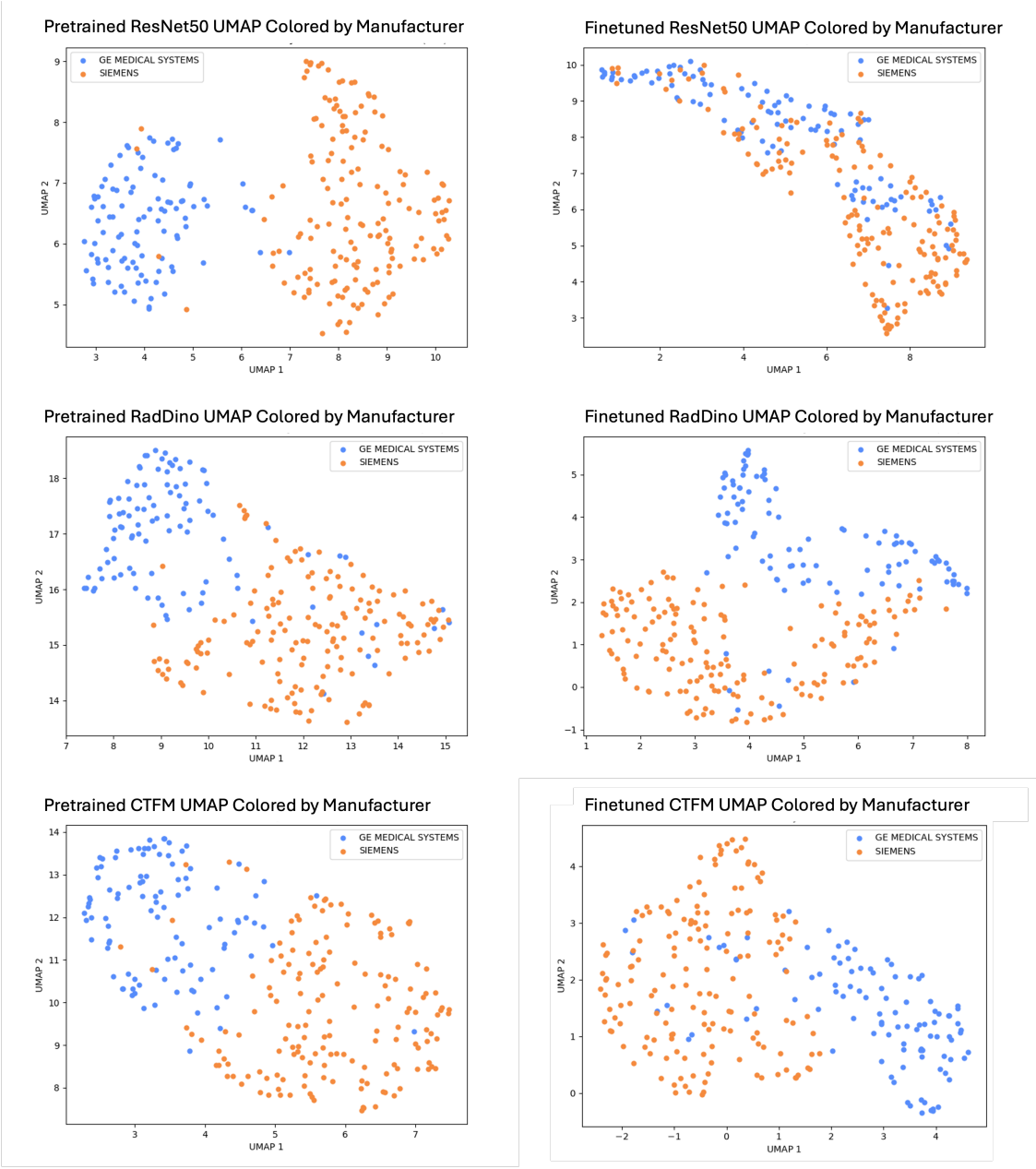


Figure 4.5: 2D UMAPs of the extracted embeddings from the COVID dataset using pretrained (left) and sequentially finetuned (right) models. From Top down the models are ResNet50, RAD-DINO, and CT-FM. Cases (data points) are colored by scanner manufacturer.UMAP 1 and UMAP 2 represent the reduced dimensions of each embedding produced by UMAP dimensionality reduction.

Figure 4.6 recolors the same plots in Figure 4.5 using acquisition sites. Same pattern clustering is visible as when data points were colored by manufacturer label.

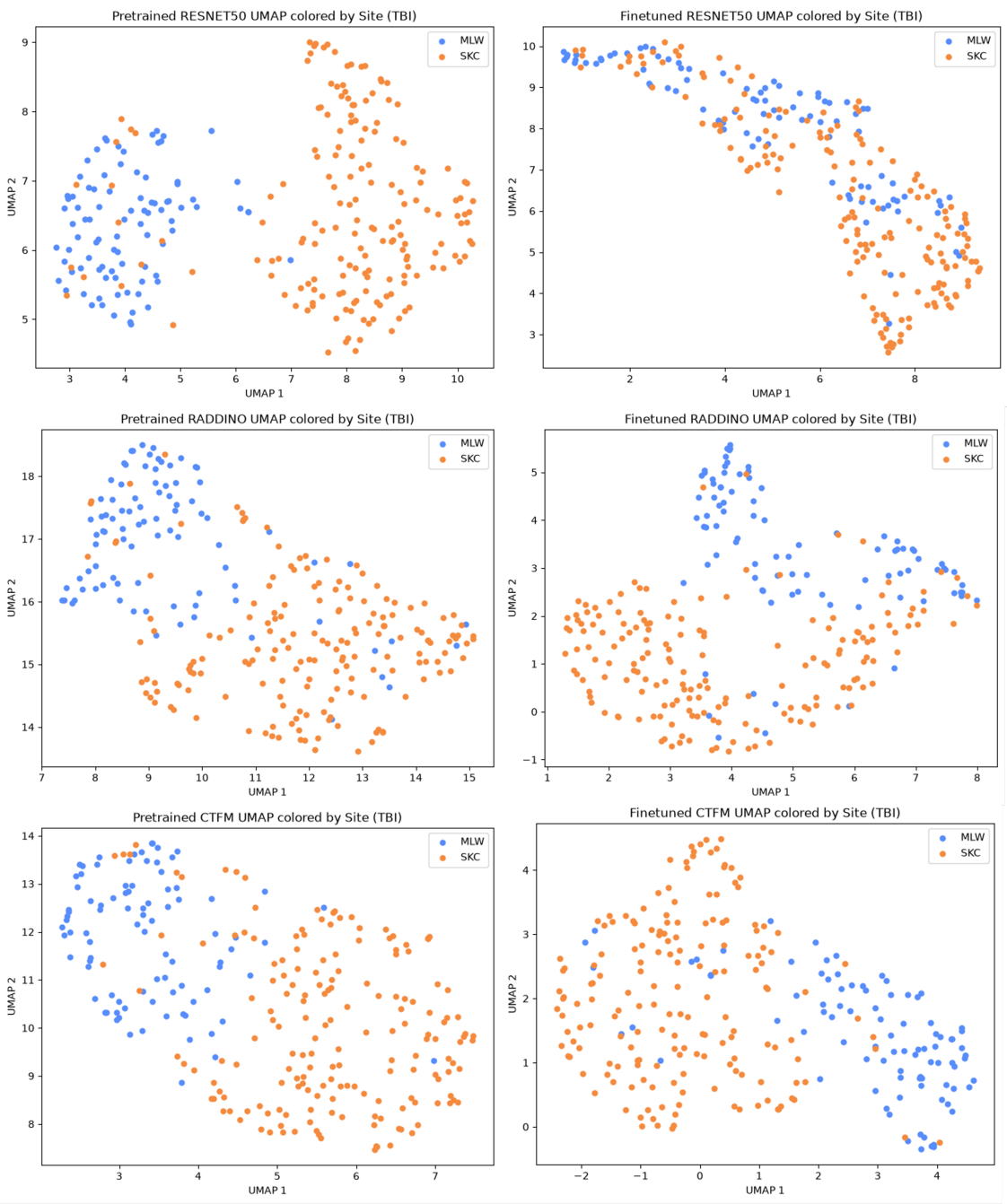


Figure 4.6: 2D UMAPs of the extracted embeddings from the TBI dataset using pretrained (left) and sequentially finetuned (right) architectures. From Top down the architectures are ResNet50, RAD-DINO, and CT-FM. Cases (data points) are colored by acquisition site. UMAP 1 and UMAP 2 represent the reduced dimensions of each embedding produced by UMAP dimensionality reduction.

4.3.2 Cluster Analysis

University of Chicago COVID-19 Data Mart

Table 4.7 shows the ARI, DBI, and Purity Scores for the COVID dataset. For all architectures, finetuning resulted in a decrease in the ARI and Purity Score while increasing the DBI; that is the finetuning demonstrated reduction in "clustering" by the shortcut of the vendor. The desired direction for all three metrics, and all differences were statistically significant (all Δ 95% CIs excluded zero). The largest changes occurred for ResNet50 and CT-FM. ResNet50's finetuned ARI dropped to near zero (0.0078), and CT-FM's dropped substantially as well (0.028), both consistent with the correspondingly large increases in DBI and decreases in Purity Score for these two models. RAD-DINO showed the smallest difference in ARI, though it was still statistically significant.

Table 4.7: Cluster Metrics, including the ARI, DBI, and Purity Score, for the different experimental arms and architectures for the COVID dataset. Arrows indicate desired change for metrics evaluating extracted embeddings from pretrained to sequentially finetuned. Bold values indicate statistically significant differences (95% CI excludes zero).

Experiment	**ARI↓**	**ΔARI**	**DBI↑**	**ΔDBI**	**Purity Score↓**	**ΔPurity Score**
ResNet50 - Pretrained	0.13 [0.092, 0.18]	**0.13 [0.081, 0.17]**	1.2 [1.1, 1.5]	**-11 [-21, -3.8]**	0.88 [0.85, 0.91]	**0.15 [0.11, 0.20]**
ResNet50 - Finetuned	0.0078 [-0.0017, 0.020]		12 [5.0, 22]		0.73 [0.69, 0.76]	
RAD-DINO - Pretrained	0.044 [0.024, 0.067]	**0.023 [4.6e-5, 0.048]**	2.4 [2.0, 3.0]	**-1.6 [-2.7, -0.78]**	0.78 [0.74, 0.81]	**0.043 [0.0063, 0.080]**
RAD-DINO - Finetuned	0.021 [0.0053, 0.042]		4.0 [3.1, 5.2]		0.73 [0.70, 0.77]	
CT-FM - Pretrained	0.16 [0.12, 0.20]	**0.13 [0.094, 0.17]**	1.2 [1.1, 1.4]	**-1.4 [-2.2, -0.84]**	0.91 [0.88, 0.94]	**0.16 [0.11, 0.21]**
CT-FM - Finetuned	0.028 [0.0099, 0.052]		2.6 [2.0, 3.4]		0.75 [0.71, 0.79]	

TXA TBI Dataset

Table 4.8 shows the cluster metrics for the TBI dataset. In contrast to the COVID datasets, where all models shared the same pattern after finetuning, ResNet50 was the only model

that produced statistically significant changes across all three metrics, each in the desired direction (all Δ 95% CIs excluded zero). For RAD-DINO and CT-FM, the only metric that reached statistical significance was the ΔDBI. For CT-FM, the change in DBI was in the direction opposite to that desired. DBI decreased following finetuning, indicating that embeddings clustered by scanner manufacturers became more compact and better separated. These findings are further summarized in Figure 4.7.

Table 4.8: Cluster Metrics, including the ARI, DBI, and Purity Score, for the different experimental arms and architectures for the TXA TBI dataset. Arrows indicate desired change for metrics evaluating extracted embeddings from pretrained to sequentially finetuned. Bold values indicate statistically significant differences (95% CI excludes zero).

Experiment	**ARI↓**	**ΔARI**	**DBI↑**	**ΔDBI**	**Purity Score↓**	**ΔPurity Score**
ResNet50 - Pretrained	0.20 [0.14, 0.26]	**0.14 [0.075, 0.21]**	1.2 [0.98, 1.6]	**-0.55 [-1.3, -0.065]**	0.91 [0.85, 0.95]	**0.19 [0.097, 0.26]**
ResNet50 - Finetuned	0.062 [0.034, 0.096]		1.8 [1.3, 2.5]		0.72 [0.65, 0.78]	
RAD-DINO - Pretrained	0.11 [0.072, 0.15]	-0.042 [-0.10, 0.013]	1.4 [1.2, 1.8]	**-0.42 [-0.63, -0.14]**	0.81 [0.76, 0.86]	-0.038 [-0.095, 0.019]
RAD-DINO - Finetuned	0.15 [0.11, 0.21]		1.8 [1.6, 2.2]		0.85 [0.80, 0.90]	
CT-FM - Pretrained	0.10 [0.020, 0.21]	-0.037 [-0.13, 0.069]	2.4 [1.5, 4.5]	**1.2 [0.25, 3.2]**	0.79 [0.67, 0.91]	-0.043 [-0.17, 0.084]
CT-FM - Finetuned	0.14 [0.099, 0.19]		1.2 [0.99, 1.5]		0.84 [0.78, 0.89]	

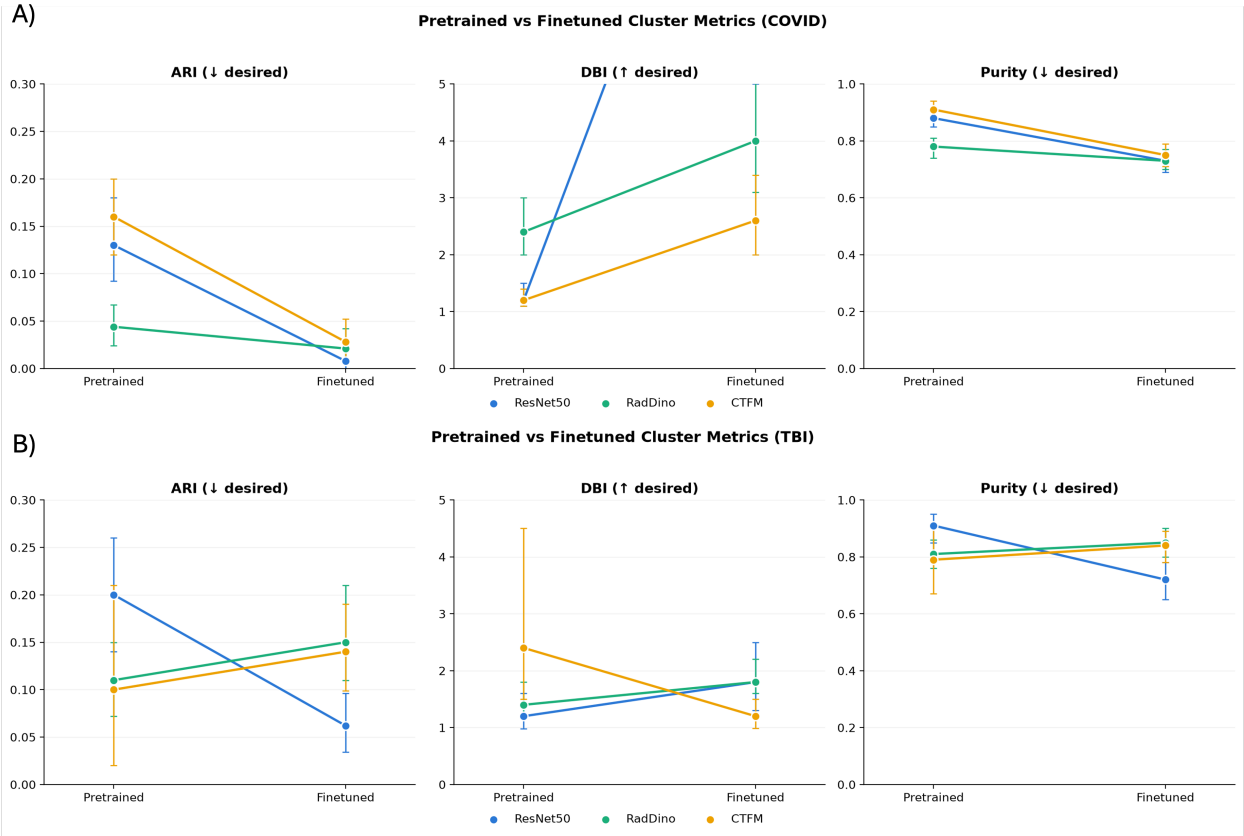


Figure 4.7: Paired slope plots indicating the metric changes following sequential finetuning of the models. Architectures are ResNet50 (blue), RAD-DINO (green), and CT-FM (orange). A) shows the cluster metrics for the COVID-19 dataset and B) shows cluster metrics for the TXA TBI dataset. Data points are shown with the corresponding 95% CI. Arrows indicate the desired trajectory for each metric, indicating reduction in clustering following sequential finetuning.

4.3.3 TBI GOSE at 6 Month Prediction

Table 4.9 shows, for the TBI dataset, the AUCs across the different models for the task of binary classification of GOSE less than or equal to 3.

RAD-DINO showed the best prediction performance with both pretrained and sequentially finetuned architectures having 0.79 [0.72, 0.85] and 0.80 [0.72, 0.86]. ResNet50 and CT-FM resulted in a slight increase in classification performance following sequential finetuning, although they failed to show statistical significance. Table 4.10 compares the ΔAUC

Table 4.9: Model performance comparison (AUC with 95% CI) for RAD-DINO, ResNet50, and CT-FM on the TXA TBI dataset. AUCs reported for embeddings from pretrained models and sequentially finetuned models.

Model	**AUC (95% CI)**	**ΔAUC (95% CI)**
ResNet50 - Pretrained	0.70 [0.62, 0.78]	-0.032 [-0.11, 0.038]
ResNet50 - Finetuned	0.73 [0.65, 0.81]	
RAD-DINO - Pretrained	0.79 [0.72, 0.85]	-0.0092 [-0.064, 0.042]
RAD-DINO - Finetuned	0.80 [0.72, 0.86]	
CT-FM - Pretrained	0.56 [0.48, 0.64]	-0.043 [-0.127, 0.043]
CT-FM -Finetuned	0.60 [0.52, 0.68]	

Table 4.10: Pairwise comparisons of AUC across architectures. Bold values indicate statistically significant differences (Bonferroni adjusted $p < 0.05$)

Model	**ΔAUC (95% CI)**	**Adjusted P**
ResNet50 v RAD-DINO	-0.065495 (-0.125, -0.001)	0.132
ResNet50 v CT-FM	**0.130362 (0.037, 0.233)**	0.024
RAD-DINO v CT-FM	**0.195857 (0.099, 0.295)**	<0.001

values across the sequentially finetuned models. ResNet50 and RAD-DINO had significant increases in performance compared to CT-FM (based on adjusted Bonferroni P value).

Figure 4.8 shows the prediction probabilities from the logistic regression classifier compared to the true GOSE value at 6 months for each architecture. Prediction probabilities are colored by outcome (red GOSE ≤ 3, green GOSE > 3). Kendall's τ was used to assess the correlation between predicted probability and GOSE score for each experiment arm [143].

Both ResNet50 and RAD-DINO show moderate negative correlations for both pretrained and sequentially finetuned models. CT-FM correlations were the worst with τ = -0.168 and τ = -0.123 for pretrained and finetuned architectures, respectively. All embedding

representations showed a significant negative relationship between predicted probability and 6-month GOSE, with RAD-DINO embeddings yielding the strongest association (τ = -0.37 pretrained, -0.35 finetuned) and CT-FM the weakest (τ = -0.17, -0.12).

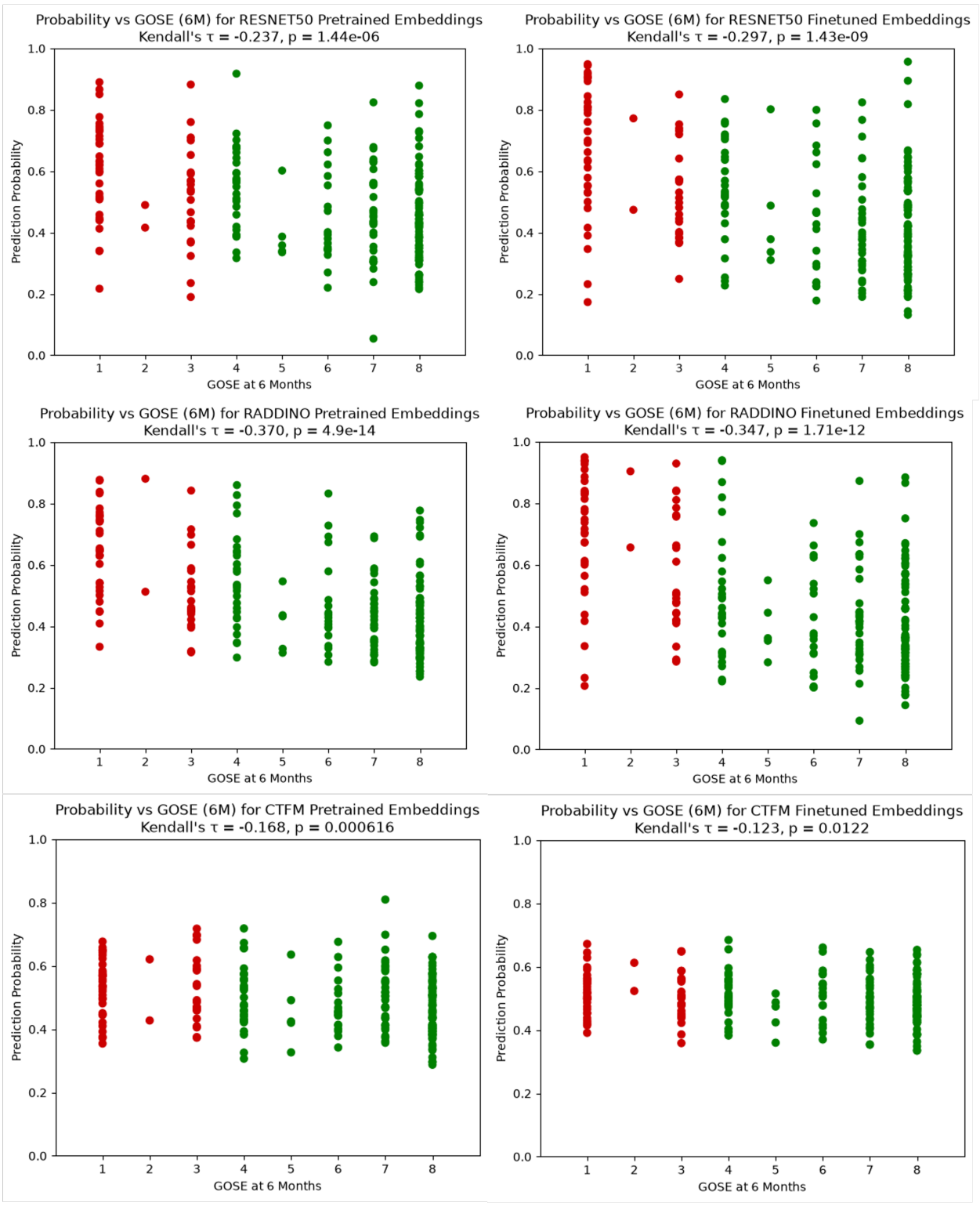


Figure 4.8: Logistic regression model prediction probabilities plotted against GOSE. Prediction probabilities are colored by outcome (red GOSE ≤ 3, green GOSE > 3). Kendall's τ for each respective test is shown above plot

4.4 Discussion and Conclusion

The study results show how embedding patterns from pretrained architectures, and the effect of the presence of shortcuts due to different clinical sites and different CT vendors, change after task-oriented finetuning. We examined HCTs, from a UChicago COVID-19 data mart, and a multi-institutional TBI dataset, using dimensionality reduction, cluster analysis, and clinical predictions with an SVM. Qualitative analysis of the dimensionality-reduced embeddings showed distinct clustering by scanner manufacturer in both datasets. Figures 4.4 and 4.5 show that embedding from pretrained architectures clustered more tightly and separately by vendor, especially for ResNet50 and CT-FM. Task-oriented sequential finetuning changed the embedding structure.

Following finetuning with SinoCT, UMAP plots of ResNet50 and RAD-DINO COVID HCT embeddings, color-coded by manufacturer, indicate that manufacturer was no longer the factor driving the clustering. CT-FM still showed vendor-level groupings, although the clusters were less compact and closer together than those from the pretrained model. This suggests that finetuning reduced the manufacturer signal but did not fully eliminate it, and that it was less effective for CT-FM than for ResNet50 or RAD-DINO.

After finetuning with SinoCT, all pretrained models from the TBI dataset showed a pattern similar to that of the COVID dataset, with clear separation when UMAPs were colored by vendor. After finetuning and assessing the reduced embeddings, cases still clustered by vendor for RAD-DINO and CT-FM. Although clusters remained separated by vendor, they appeared less compact and less distinct than their pretrained counterparts. ResNet50 showed a more intermixed relationship between scanner manufacturers, though vendor groupings could still be discerned. In Figure 4.6, recoloring cases by site shows that one site predominantly used one vendor and the other site the other. This indicates that site information may have been more prevalent in the embeddings than scanner manufacturer, because site and vendor were closely confounded in this dataset.

Cluster behavior in the extracted embeddings was further evaluated using cluster metrics. ARI, DBI, and Purity Score were chosen because they capture both external and internal characteristics of clusters. Tables 4.7 and 4.8 report the calculated cluster metrics with 95% confidence intervals for the COVID and TBI HCT datasets, respectively. For all models, task-oriented finetuning reduced ARI and Purity Scores while increasing DBI relative to scanner manufacturer labels. Lower ARI and Purity Scores, together with higher DBI, indicate less clustering by vendor. For the COVID HCT dataset, each sequentially finetuned model produced embeddings with near-zero ARI, indicating that the relationship between ground truth and K-Means clustering is approximately due to random chance. ResNet50 showed the largest reduction in clustering, with all metric changes statistically significant. This result aligns with the UMAPs, in which both ResNet50 and CT-FM showed reduced clustering by vendor. RAD-DINO, although visually showing loose clustering by vendor, the reported an ARI of 0.044, which contradicts this pattern. This was most likely due to vendor separation in the extracted embeddings from the pretrained model not being strong enough, leading to K-Means clustering along another dominant axes rather than along the vendor boundary. The DBI reflects the reduction in clustering: 2.4 [2.0, 3.0] for pretrained and 4.0 [3.1, 5.2] for sequentially finetuned, with higher DBI indicating less compact clusters. This further supports the value of using multiple metrics to quantify clusters.

The contradiction between visual and quantitative assessment was most evident in the TBI dataset, where the data loosely clustered by scanner manufacturer in both pretrained and sequentially finetuned models. This is reflected in the lower ARI values around 0.1, indicating loose clustering. ARI differences failed to reach statistical significance for RAD-DINO and CT-FM, with ResNet50 reaching near zero. This contradiction may be because site and manufacturer were confounded, where K-Means may have clustered cases along site-related or acquisition parameters axes that correlate with vendor identity. This would produce cluster assignments that differ from vendor labels, thereby reducing ARI. However,

this would require additional analysis to verify (e.g., checking cluster assignment against site labels directly).

The extracted embeddings from both pretrained and sequentially finetuned models were used as inputs to an SVM to classify GOSE scores as negative or positive (GOSE $\leq$3). Table 4.9 reports the AUC values with their 95% CI. Both ResNet50 and CT-FM improved in AUC when using embeddings from the sequentially finetuned architectures, suggesting slight gains in classification performance. However, this difference failed to reach statistical significance. RAD-DINO achieved the highest performance, with AUCs of 0.79 [0.72, 0.85] and 0.8 [0.72, 0.86] for both the pretrained and finetuned arms, respectively. Analysis of prediction probabilities and GOSE scores showed a moderately strong negative correlation for the finetuned ResNet50 embeddings and a moderate-to-strong negative correlation for both RAD-DINO models, as measured by Kendall's τ. Prediction probabilities and GOSE showed a weaker correlation for embeddings for both CT-FM arms. These findings were consistent with the AUC results: RAD-DINO produced the highest AUCs and Kendall's τ, while CT-FM produced the lowest values. It's worth noting that about 16% of GOSE scores in the dataset are missing. In multi-center trial cohorts missing outcome data often are Missing Not at Random, where loss to follow-up tends to skew data towards patients with worse outcomes [153]. Because these patients were dropped, results may bias findings towards more favorable outcomes.

In summary, we evaluated embeddings extracted from three pretrained architectures using HCTs from two separate datasets investigate potential presence shortcuts in extracted embeddings. We also performed sequential finetuning of these architectures using a third dataset of HCTs labeled as abnormal or normal to assess the effect of finetuning on extracted embeddings. Qualitative dimensionality reduction and cluster analysis showed that the extracted embeddings group by scanner manufacturer, with metrics and visual assessment agreeing. For the finetuned architectures, quantitative clustering performance decreased,

while qualitative disagreement remained for the TBI dataset. In the classification arm of this study, embeddings from RAD-DINO achieved the highest discriminative performance for predicting ≤ 3, with the highest AUC and the strongest correlation between predicted probability and outcome severity, whereas CT-FM embeddings showed comparatively weaker performance in both classification and correlation analyses. Together, these findings indicate how quantitative and qualitative assessment of embeddings can complement each other. The results also indicate that vendor-based clustering decreases following task-oriented finetuning of these models. Finetuning further suggests improved classification for predicting GOSE $\leq$ 3 at 6 months, although this difference failed to reach statistical significance.

CHAPTER 5

SUMMARY AND FUTURE DIRECTIONS

This dissertation contributed to medical AI by developing machine learning methods for preparing and evaluating embeddings from HCT scans of patients with TBI and COVID. The machine learning methods developed included segmentation of HCT scans from patients with TBI, a tool to analyze embedding clusters, and an investigation of shortcut influences in embeddings across different models. This work has direct implications for various AI applications, including improvements in model performance, evaluation of the direct effects of shortcuts in HCTs, and lays a foundation for the development of more robust and generalizable models.

Both imaging and clinical information are used to assess patients with TBI and to inform treatment plans. Given the complexity of TBI, the limitations of current assessment tools such as the GCS, and the heterogeneity of injury presentations, clinicians can often struggle to determine the most appropriate course of treatment. Thus, high-performing and reliable machine learning tools can help physicians and treatment teams improve diagnosis, predict patient outcomes, and guide clinical decisions, thereby enhancing patient quality of life.

In Chapter 2, we developed a CNN-based deep learning segmentation tool to accurately delineate the brain/skull interface in HCT scans from patients with TBI. This chapter addressed the preprocessing step of skull segmentation that supports many downstream applications, such as lesion detection, volumetric analysis, and classification. However, current methods are often trained or developed using healthy or non-traumatic populations. Traditional approaches, such as intensity-based models, rely on gray-scale features and patterns to segment the brain. Thus, translation of these models may fail to generalize to the complex patterns present in HCT scans, such as midline shifts, skull fractures, edema, and hemorrhages. In this study, we developed a 2.5D cross-validation ensemble-based segmentation model using a U-Net backbone. Rather than implementing a traditional 2D input, the 2.5D

approach leverages axial information superior and inferior to the slice of interest while using 2D convolutions. Furthermore, the ensemble approach uses 8 models from cross-validation and produces 8 outputs, which are then aggregated into a final segmentation map. This approach was tested using different training schemes, including training on np-TBI cases only, sequential finetuning with p-TBI, and a combined approach using both np-TBI and p-TBI training sets. All models demonstrated strong performance on the held-out np-TBI testing set, using DSC and HD as performance metrics. The model trained on the np-TBI and the further finetuned model saw a noticeable drop in performance when evaluating on the held-out p-TBI dataset, indicating struggles when more severe artifacts and irregularities are present in the HCT scans. In contrast, the model trained with both np-TBI and p-TBI images demonstrated strong performance when segmenting these more complex cases.

Future work should include evaluation of model performance on an out-of-distribution dataset to assess segmentation efficacy. As discussed in Chapter 4, manufacturer and acquisition-site parameters may affect tasks such as segmentation. Although this method performed well at producing segmentation masks for the brain, it was limited to a specific brain window, which could lead to poor segmentation for images outside this intensity range. Introducing different windows and levels during training could improve segmentation and overall robustness. The TBI datasets used were also limited to blunt trauma (np-TBI) and penetrating trauma (p-TBI), thus excluding a subpopulation of TBI that deals with shockwave injury resulting primarily from blasts. By incorporating HCT scans from this population, the model could learn underlying patterns specific to this injury, making the model much more generalizable.

Chapter 3 developed a cluster analysis tool (CAT) to analyze underlying patterns present in either extracted embeddings or raw features. Unsupervised clustering methods are crucial for determining how different cases and images relate to one another, with cases sharing similar features clustering together. This technique is especially useful for identifying patient

or disease subgroups that can meaningfully impact medical AI. The primary method of evaluating embeddings is largely qualitative. Embeddings are reduced to 2 or 3 dimensions, enabling people to view the clusters. However, embeddings are high dimensional vectors which can contain clinically relevant information that 2 or 3 dimensions may fail to capture. Thus, by expanding cluster analysis to quantitative metrics, we can further evaluate grouping in the underlying data not visible to the naked eye. Furthermore, clinical data rarely clusters in a neat and compact way. Each quantitative metric measures something different, and there is considerable variety amongst them. Thus, there is a lack of consensus on how to evaluate clusters using both quantitative and qualitative methods. To address this, CAT was developed to enable researchers to effectively analyze extracted embeddings and raw features by using both dimensionality reduction and clustering metrics. By comparing the Iris dataset and TXA TBI dataset, we showed that clinical data clustering metrics differ from those of well-clustered data, and that using alternative metrics can help redefine cluster analysis.

Although CAT functionalities cover the fundamentals of cluster analysis, future work will focus on introducing statistical measures for clustering metrics. With the ability to bootstrap metrics, users will be able to draw statistical conclusions within the tool. Moreover, the tool could benefit from more quality-of-life features, such as interactive plots that provide users with information about the specific cases within a grouping. Features like this will facilitate analysis by allowing users to reference clustering groups to specific cases. Lastly, to provide explainability for cluster analysis, we aim to add a descriptive page that will guide users through quantitative and qualitative approaches for evaluating and interpreting clustering results, helping them select methods best suited to their needs.

Chapter 4 evaluated, with the aid of CAT, the extracted embeddings from three deep learning models with different pretraining regimes: ResNet50 (pretrained on ImageNet), RAD-DINO (pretrained on chest radiographs), and CT-FM (pretrained on CT scans). Deep learning models are capable of various tasks such as classification, segmentation, and predic-

tions. Recently, foundation models for medical AI have shown promise in medical imaging, using a single model to produce embeddings for various downstream tasks. Unfortunately, models can encode unwanted information, such as manufacturer and site-specific details, that can influence these downstream tasks. Models in this study were used to extract features from HCT scans from two different databases: the University of Chicago COVID-19 Data Mart and an TXA TBI dataset. To assess the effects on embeddings following task-specific finetuning, a third dataset of HCTs with cases labeled normal or abnormal was used. The extracted embeddings from the COVID dataset showed cluster separation by HCT scanner manufacturer. These clusters were further validated using quantitative cluster metrics, which agreed with qualitative cluster analysis. The sequentially finetuned models demonstrated that the resulting embeddings exhibited a statistically significant reduction in clustering by vendor across all models. When models were used to extract embeddings from the TBI dataset, a similar vendor-based clustering pattern was observed. However, the embeddings from the sequentially finetuned models were still separated by vendor, i.e., showed the shortcuts due to different vendors. Further observations revealed that both the scanner manufacturer and the acquisition site were confounding factors (i.e., shortcuts), suggesting that clustering may be driven by site-specific parameters not present in the single-site University of Chicago Data Mart. To determine whether sequential finetuning could improve classification on the TBI dataset, an SVM was trained using 5-fold cross-validation to classify binarized GOSE at 6 months (GOSE ≤ 3 at 6 months). Although ResNet50 and RAD-DINO showed decent classification performance, the difference between the pretrained and sequentially finetuned models failed to reach statistical significance.

Future work for this chapter should include further investigation into cluster separation by site/vendor to determine the cause of the shortcut in the TBI dataset. By determining the cause, different harmonization techniques, such as COMBAT or similar harmonization methods used in other imaging domains, could also be applied prior to embedding extraction

to assess whether this reduces vendor-based clustering [154, 155]. The TBI dataset is from a clinical drug study, testing the effects of TXA. Further investigation will be needed to determine whether these factors influenced the classification predictions, given that some patients received placebo doses. All these potential future directions could provide further explanation of how these embeddings change and whether shortcut influences negatively affect downstream tasks.

In conclusion, this work demonstrated how different AI methods can be applied for data preparation and evaluating embeddings from different models in the domain of HCT scans of patients with TBI and COVID-19. Chapter 2 demonstrated how deep learning can be applied to address the limitations of brain extraction from HCT scans from different subgroups of TBI patients. In Chapter 3, we introduced CAT, a cluster analysis software program designed to provide a more comprehensive approach to cluster analysis through both qualitative and quantitative methods. Chapter 4 evaluated vendor-based shortcuts present in HCT-extracted embeddings from three different models. Furthermore, this chapter demonstrated how sequential finetuning reshaped embeddings and revealed the effects on downstream classification tasks. This work collectively highlights the role of AI in medical imaging and how these upstream tasks are expected to be crucial for assisting the development of clinical tools that are reliable and generalizable.

REFERENCES

[1] U. F. O. Themes, "Imaging the Head and Brain," July 2019.

[2] J. F. Kolen and S. C. Kremer, *A Field Guide to Dynamical Recurrent Networks*. John Wiley & Sons, Jan. 2001. Google-Books-ID: NWOcMVA64aAC.

[3] G. B. Goh, N. O. Hodas, and A. Vishnu, "Deep learning for computational chemistry," *Journal of Computational Chemistry*, vol. 38, no. 16, pp. 1291–1307, 2017. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.24764.

[4] B. Hanin, "Which Neural Net Architectures Give Rise to Exploding and Vanishing Gradients?," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018.

[5] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, "Artificial intelligence in healthcare: past, present and future," *Stroke and Vascular Neurology*, vol. 2, Dec. 2017.

[6] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," *Nature Medicine*, vol. 25, pp. 44–56, Jan. 2019.

[7] R. Hirani, K. Noruzi, H. Khuram, A. S. Hussaini, E. I. Aifuwa, K. E. Ely, J. M. Lewis, A. E. Gabr, A. Smiley, R. K. Tiwari, and M. Etienne, "Artificial Intelligence and Healthcare: A Journey through History, Present Innovations, and Future Possibilities," *Life*, vol. 14, p. 557, Apr. 2024.

[8] J. Bajwa, U. Munir, A. Nori, and B. Williams, "Artificial intelligence in healthcare: transforming the practice of medicine," *Future Healthcare Journal*, vol. 8, pp. e188–e194, July 2021.

[9] G. Edison, "Transforming Medical Decision-Making: A Comprehensive Review of AI's Impact on Diagnostics And Treatment," *BULLET : Jurnal Multidisiplin Ilmu*, vol. 2, pp. 1121–1133, July 2023.

[10] Y. A. Fahim, I. W. Hasani, S. Kabba, and W. M. Ragab, "Artificial intelligence in healthcare and medicine: clinical applications, therapeutic advances, and future perspectives," *European Journal of Medical Research*, vol. 30, p. 848, Sept. 2025.

[11] S. A. Alowais, S. S. Alghamdi, N. Alsuhebany, T. Alqahtani, A. I. Alshaya, S. N. Almohareb, A. Aldairem, M. Alrashed, K. Bin Saleh, H. A. Badreldin, M. S. Al Yami, S. Al Harbi, and A. M. Albekairy, "Revolutionizing healthcare: the role of artificial intelligence in clinical practice," *BMC Medical Education*, vol. 23, p. 689, Sept. 2023.

[12] F. Pesapane, G. Gnocchi, C. Quarrella, A. Sorce, L. Nicosia, L. Mariano, A. C. Bozzini, I. Marinucci, F. Priolo, F. Abbate, G. Carrafiello, and E. Cassano, "Errors in Radiology: A Standard Review," *Journal of Clinical Medicine*, vol. 13, p. 4306, July 2024.

[13] S. Katal, B. York, and A. Gholamrezanezhad, "AI in radiology: From promise to practice - A guide to effective integration," *European Journal of Radiology*, vol. 181, p. 111798, Dec. 2024.

[14] Y. Li, X. Yi, J. Fu, Y. Yang, C. Duan, and J. Wang, "Reducing misdiagnosis in AI-driven medical diagnostics: a multidimensional framework for technical, ethical, and policy solutions," *Frontiers in Medicine*, vol. 12, p. 1594450, Oct. 2025.

[15] T. M. Buzug, "Computed Tomography," in *Springer Handbook of Medical Technology* (R. Kramme, K.-P. Hoffmann, and R. S. Pozos, eds.), pp. 311–342, Berlin, Heidelberg: Springer, 2011.

[16] R. A. Schulz, J. A. Stein, and N. J. Pelc, "How CT happened: the early development of medical computed tomography," *Journal of Medical Imaging*, vol. 8, p. 052110, Sept. 2021.

[17] M. C. Florkow, K. Willemsen, V. V. Mascarenhas, E. H. Oei, M. van Stralen, and P. R. Seevinck, "Magnetic Resonance Imaging Versus Computed Tomography for Three-Dimensional Bone Imaging of Musculoskeletal Pathologies: A Review," *Journal of Magnetic Resonance Imaging*, vol. 56, pp. 11–34, July 2022.

[18] E. J. van Beek and E. A. Hoffman, "Functional Imaging: CT and MRI," *Clinics in chest medicine*, vol. 29, pp. 195–vii, Mar. 2008.

[19] R. Lamba, J. P. McGahan, M. T. Corwin, C.-S. Li, T. Tran, J. A. Seibert, and J. M. Boone, "CT Hounsfield Numbers of Soft Tissues on Unenhanced Abdominal CT Scans: Variability Between Two Different Manufacturers' MDCT Scanners," *American Journal of Roentgenology*, vol. 203, pp. 1013–1020, Nov. 2014.

[20] E. S. Pretorius, "Introduction to Ultrasound, CT, and MRI," in *Radiology Secrets Plus*, pp. 13–18, Elsevier, 2011.

[21] S. Kato, T. Nakamoto, and M. Iizuka, "Early Diagnosis and Estimation of Pulmonary Congestion and Edema in Patients With Left-Sided Heart Diseases From Histogram of Pulmonary CT Number," *Chest*, vol. 109, pp. 1439–1445, June 1996.

[22] M. L. Giger, H. Chan, and J. Boone, "Anniversary Paper: History and status of CAD and quantitative image analysis: The role of *Medical Physics* and AAPM," *Medical Physics*, vol. 35, pp. 5799–5820, Dec. 2008.

[23] B. Sahiner, A. Pezeshk, L. M. Hadjiiski, X. Wang, K. Drukker, K. H. Cha, R. M. Summers, and M. L. Giger, "Deep learning in medical imaging and radiation therapy," *Medical Physics*, vol. 46, Jan. 2019.

[24] R. Najjar, "Redefining Radiology: A Review of Artificial Intelligence Integration in Medical Imaging," *Diagnostics*, vol. 13, p. 2760, Aug. 2023.

[25] N. Antropova, B. Q. Huynh, and M. L. Giger, "A deep feature fusion methodology for breast cancer diagnosis demonstrated on three imaging modality datasets," *Medical Physics*, vol. 44, pp. 5162–5171, Oct. 2017.

[26] W. Khan, S. Leem, K. B. See, J. K. Wong, S. Zhang, and R. Fang, "A Comprehensive Survey of Foundation Models in Medicine," *IEEE Reviews in Biomedical Engineering*, vol. 19, pp. 283–304, 2026.

[27] W. Zhang, J. Tanida, K. Itoh, and Y. Ichioka, "Shift-invariant pattern recognition neural network and its optical architecture," in *Proceedings of annual conference of the Japan Society of Applied Physics*, vol. 564, p. 734, Montreal, CA, 1988.

[28] W. Zhang, K. Doi, M. L. Giger, Y. Wu, R. M. Nishikawa, and R. A. Schmidt, "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network," *Medical Physics*, vol. 21, pp. 517–524, Apr. 1994.

[29] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, pp. 611–629, June 2018.

[30] S. Aburass, O. Dorgham, J. Al Shaqsi, M. Abu Rumman, and O. Al-Kadi, "Vision Transformers in Medical Imaging: a Comprehensive Review of Advancements and Applications Across Multiple Diseases," *Journal of Imaging Informatics in Medicine*, vol. 38, pp. 3928–3971, Dec. 2025.

[31] K. Arik, "From CNNs to Vision Transformers: A Survey on Addressing Domain Shift and Adaptation Concepts in Medical Image Analysis," *Archives of Computational Methods in Engineering*, June 2026.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Aug. 2023. arXiv:1706.03762 [cs.CL].

[33] S. Cuenat and R. Couturier, "Convolutional Neural Network (CNN) vs Vision Transformer (ViT) for Digital Holography," in *2022 2nd International Conference on Computer, Control and Robotics (ICCCR)*, pp. 235–240, Mar. 2022.

[34] I. Goodfellow, A. Courville, and Y. Bengio, *Deep learning*. Adaptive computation and machine learning, Cambridge, Massachusetts: The MIT Press, 2016.

[35] P. Cunningham, M. Cord, and S. J. Delany, "Supervised Learning," in *Machine Learning Techniques for Multimedia* (M. Cord and P. Cunningham, eds.), pp. 21–49, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. Series Title: Cognitive Technologies.

[36] C. F. G. D. Santos and J. P. Papa, "Avoiding Overfitting: A Survey on Regularization Methods for Convolutional Neural Networks," *ACM Computing Surveys (CSUR)*, vol. 54, pp. 213:1–213:25, Sept. 2022.

[37] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, pp. 24–29, Jan. 2019.

[38] V. Rani, S. T. Nabi, M. Kumar, A. Mittal, and K. Kumar, "Self-supervised Learning: A Succinct Review," *Archives of Computational Methods in Engineering*, vol. 30, pp. 2761–2775, May 2023.

[39] S. Pai, I. Hadzic, D. Bontempi, K. Bressem, B. H. Kann, A. Fedorov, R. H. Mak, and H. J. W. L. Aerts, "Vision Foundation Models for Computed Tomography," Jan. 2025.

[40] F. Pérez-García, H. Sharma, S. Bond-Taylor, K. Bouzid, V. Salvatelli, M. Ilse, S. Bannur, D. C. Castro, A. Schwaighofer, M. P. Lungren, M. T. Wetscherek, N. Codella, S. L. Hyland, J. Alvarez-Valle, and O. Oktay, "Exploring scalable medical image encoders beyond text supervision," *Nature Machine Intelligence*, vol. 7, pp. 119–130, Jan. 2025.

[41] B. Dai, X. Shen, and J. Wang, "Embedding Learning," *Journal of the American Statistical Association*, vol. 117, no. 537, pp. 307–319, 2022.

[42] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. v. Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the Opportunities and Risks of Foundation Models," July 2022. arXiv:2108.07258 [cs.LG].

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015. arXiv:1512.03385 [cs.CV].

[44] A. Mansour, J. D. Fuhrman, F. E. Ammar, A. Loggini, J. Davis, C. Lazaridis, C. Kramer, F. D. Goldenberg, and M. L. Giger, "Machine Learning for Early Detection of Hypoxic-Ischemic Brain Injury After Cardiac Arrest," *Neurocritical Care*, vol. 36, no. 3, pp. 974–982, 2022.

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," June 2021. arXiv:2010.11929 [cs.CV].

[46] D. Steinmann, F. Divo, M. Kraus, A. Wüst, L. Struppek, F. Friedrich, and K. Kersting, "Navigating Shortcuts, Spurious Correlations, and Confounders: From Origins via Detection to Mitigation," Dec. 2024. arXiv:2412.05152 [cs.LG].

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," May 2019. arXiv:1810.04805 [cs.CL].

[48] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020.

[49] M. Moor, O. Banerjee, Z. S. H. Abad, H. M. Krumholz, J. Leskovec, E. J. Topol, and P. Rajpurkar, "Foundation models for generalist medical artificial intelligence," *Nature*, vol. 616, pp. 259–265, Apr. 2023.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015. arXiv:1505.04597 [cs.CV].

[51] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," Mar. 2015. arXiv:1411.4038 [cs.CV].

[52] "ILSVRC2015 Results."

[53] O. O. Adekunle, J. D. Akinyemi, K. T. Ladoja, and O. F. W. Onifade, "Lung Cancer Classification from CT Images Using ResNet," Oct. 2025. arXiv:2510.16310 [eess.IV] version: 1.

[54] P. V. Ch and P. Aruchamy, "Bayesian-tuned ResNet-50 framework for enhanced detection of large-cell lung carcinoma," *Biomedical Signal Processing and Control*, vol. 123, p. 110476, Sept. 2026.

[55] J. M. Amine and M. Mourad, "Toward accurate Alzheimer's detection: transfer learning with ResNet50 for MRI-based diagnosis," *Frontiers in Neuroscience*, vol. 19, p. 1664418, 2025.

[56] N. Sebastian and B. Ankayarkanni, "Enhanced ResNet-50 with Multi-Feature Fusion for Robust Detection of Pneumonia in Chest X-Ray Images," *Diagnostics*, vol. 15, p. 2041, Aug. 2025.

[57] A. Rath, B. S. P. Mishra, and D. K. Bagal, "ResNet50-based Deep Learning model for accurate brain tumor detection in MRI scans," *Next Research*, vol. 2, p. 100104, Mar. 2025.

[58] B. Goswami, A. B. Somaraj, P. Chakrabarti, R. Gudi, and N. Punjabi, "Classifier Enhanced Deep Learning Model for Erythroblast Differentiation with Limited Data," in *Pattern Recognition* (A. Antonacopoulos, S. Chaudhuri, R. Chellappa, C.-L. Liu, S. Bhattacharya, and U. Pal, eds.), (Cham), pp. 18–30, Springer Nature Switzerland, 2025.

[59] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," Feb. 2021. arXiv:2103.00020 [cs.CV].

[60] K. Desai and J. Johnson, "VirTex: Learning Visual Representations from Textual Annotations," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Nashville, TN, USA), pp. 11157–11168, IEEE, June 2021.

[61] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning Robust Visual Features without Supervision," Feb. 2024. arXiv:2304.07193 [cs.CV].

[62] M. Hoq, L. Tarbox, D. Johann, L. Larson-Prior, and F. Prior, "Harnessing Native-Resolution 2D Embeddings for Lung Cancer Classification: A Feasibility Study with the RAD-DINO Self-supervised Foundation Model," *Journal of Imaging Informatics in Medicine*, Dec. 2025.

[63] H. Farquhar, "Foundation Model Robustness to Technical Acquisition Parameters in Chest X-Ray AI: A Multi-Architecture Comparative Study with External Validation," Jan. 2026. ISSN: 3067-2007 Pages: 2026.01.25.26344809.

[64] A. Ziller, A. C. Erdur, M. Trigui, A. Güvenir, T. T. Mueller, P. Müller, F. Jungmann, J. Brandt, J. Peeken, R. Braren, D. Rueckert, and G. Kaissis, "Interpretable 2D Vision Models for 3D Medical Images," Dec. 2023. arXiv:2307.06614 [eess.IV].

[65] X. Luo, H.-H. S. Wang, and M. L. Li, "Towards Interpretable, Sequential Multiple Instance Learning: An Application to Clinical Imaging," *AMIA Annual Symposium Proceedings*, vol. 2025, pp. 804–813, Feb. 2026.

[66] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," July 2020. arXiv:2002.05709 [cs.LG].

[67] F. Fadzil, A. K. C. Mei, A. Mohd Khairy, R. Kumar, and A. N. Mohd Azli, "Value of Repeat CT Brain in Mild Traumatic Brain Injury Patients with High Risk of Intracerebral Hemorrhage Progression," *International Journal of Environmental Research and Public Health*, vol. 19, p. 14311, Jan. 2022.

[68] M. Lousick, S. Edwards, G. Keijzers, and R. A. F. Pellatt, "Utility of computed tomography brain scans in intubated patients with overdose," *Emergency Medicine Australasia*, vol. 37, no. 1, p. e14510, 2025. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1742-6723.14510.

[69] C. Brossard, B. Lemasson, A. Attyé, J.-A. de Busschère, J.-F. Payen, E. L. Barbier, J. Grèze, and P. Bouzat, "Contribution of CT-Scan Analysis by Artificial Intelligence to the Clinical Care of TBI Patients," *Frontiers in Neurology*, vol. 12, June 2021.

[70] J. Kleesiek, G. Urban, A. Hubert, D. Schwarz, K. Maier-Hein, M. Bendszus, and A. Biller, "Deep MRI brain extraction: A 3D convolutional neural network for skull stripping," *NeuroImage*, vol. 129, pp. 460–469, Apr. 2016.

[71] N. Stocchetti, M. Carbonara, G. Citerio, A. Ercole, M. B. Skrifvars, P. Smielewski, T. Zoerle, and D. K. Menon, "Severe traumatic brain injury: targeted management in the intensive care unit," *The Lancet. Neurology*, vol. 16, pp. 452–464, June 2017.

[72] H. Schneiderman, "Glasgow Coma Creep: Problems of Recognition and Communication," *The American Journal of Medicine*, vol. 128, pp. 1270–1271, Dec. 2015.

[73] G. Teasdale, A. Maas, F. Lecky, G. Manley, N. Stocchetti, and G. Murray, "The Glasgow Coma Scale at 40 years: standing the test of time," *The Lancet. Neurology*, vol. 13, pp. 844–854, Aug. 2014.

[74] "Traumatic Brain Injury (TBI) | National Institute of Neurological Disorders and Stroke."

[75] "New Framework for Classifying Traumatic Brain Injury (TBI) Could Transform Patient Care | National Institute of Neurological Disorders and Stroke," Sept. 2025.

[76] "Human Brain Mapping | Neuroimaging Journal | Wiley Online Library."

[77] J. Muschelli, "Recommendations for Processing Head CT Data," *Frontiers in Neuroinformatics*, vol. 13, p. 61, Sept. 2019.

[78] J. Muschelli, N. L. Ullman, W. A. Mould, P. Vespa, D. F. Hanley, and C. M. Crainiceanu, "Validated automatic brain extraction of head CT images," *NeuroImage*, vol. 114, pp. 379–385, July 2015.

[79] J. I. Glaser, A. S. Benjamin, R. Farhoodi, and K. P. Kording, "The roles of supervised machine learning in systems neuroscience," *Progress in Neurobiology*, vol. 175, pp. 126–137, Apr. 2019.

[80] S. N. J. Eali, D. Bhattacharyya, T. R. Nakka, and S.-P. Hong, "A Novel Approach in Bio-Medical Image Segmentation for Analyzing Brain Cancer Images with U-NET Semantic Segmentation and TPLD Models Using SVM," *Traitement du Signal*, vol. 39, pp. 419–430, Apr. 2022.

[81] M. Nouman, M. Mabrok, and E. A. Rashed, "Neuro-TransUNet: Segmentation of stroke lesion in MRI using transformers," June 2024. arXiv:2406.06017 [eess].

[82] S. E. Rowell, E. N. Meier, B. McKnight, D. Kannas, S. May, K. Sheehan, E. M. Bulger, A. H. Idris, J. Christenson, L. J. Morrison, R. J. Frascone, P. L. Bosarge, M. R. Colella, J. Johannigman, B. A. Cotton, J. Callum, J. McMullan, D. J. Dries, B. Tibbs, N. J. Richmond, M. L. Weisfeldt, J. M. Tallon, J. S. Garrett, M. D. Zielinski, T. P. Aufderheide, R. R. Gandhi, R. Schlamp, B. R. H. Robinson, J. Jui, L. Klein, S. Rizoli, M. Gamber, M. Fleming, J. Hwang, L. E. Vincent, C. Williams, A. Hendrickson, R. Simonson, P. Klotz, G. Sopko, W. Witham, M. Ferrara, and M. A. Schreiber, "Effect of Out-of-Hospital Tranexamic Acid vs Placebo on 6-Month Functional Neurologic Outcomes in Patients With Moderate or Severe Traumatic Brain Injury," *JAMA*, vol. 324, pp. 961–974, Sept. 2020.

[83] H. Huang, M. Xin, X. Wu, J. Liu, W. Zhang, K. Yang, and J. Zhang, "The efficacy of tranexamic acid treatment with different time and doses for traumatic brain injury: a systematic review and meta-analysis," *Thrombosis Journal*, vol. 20, p. 79, Dec. 2022.

[84] S. Yutthakasemsunt, W. Kittiwatanagul, P. Piyavechvirat, B. Thinkamrop, N. Phuenpathom, and P. Lumbiganon, "Tranexamic acid for patients with traumatic brain injury: a randomized, double-blinded, placebo-controlled trial," *BMC Emergency Medicine*, vol. 13, p. 20, Nov. 2013.

[85] S. Zehtabchi, S. G. Abdel Baki, L. Falzon, and D. K. Nishijima, "Tranexamic acid for traumatic brain injury: a systematic review and meta-analysis," *The American Journal of Emergency Medicine*, vol. 32, pp. 1503–1509, Dec. 2014.

[86] Z. Xue, S. Antani, L. R. Long, D. Demner-Fushman, and G. R. Thoma, "Window Classification of Brain CT Images in Biomedical Articles," *AMIA Annual Symposium Proceedings*, vol. 2012, pp. 1023–1029, Nov. 2012.

[87] N. Baughan, H. M. Whitney, K. Drukker, B. Sahiner, T. Hu, H. J. G. Kim, M. F. McNitt-Gray, K. J. Myers, and M. L. Giger, "Sequestration of imaging studies in MIDRC: stratified sampling to balance demographic characteristics of patients in a multi-institutional data commons," *Journal of Medical Imaging*, vol. 10, p. 064501, Nov. 2023.

[88] H. Zhang, A. M. Valcarcel, R. Bakshi, R. Chu, F. Bagnato, R. T. Shinohara, K. Hett, and I. Oguz, "Multiple Sclerosis Lesion Segmentation with Tiramisu and 2.5D Stacked Slices," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019* (D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, eds.), (Cham), pp. 338–346, Springer International Publishing, 2019.

[89] L. Han, Y. Chen, J. Li, B. Zhong, Y. Lei, and M. Sun, "Liver segmentation with 2.5D perpendicular UNets," *Computers & Electrical Engineering*, vol. 91, p. 107118, May 2021.

[90] T. Sørensen, *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content: And Its Application to Analyses of the Vegetation on Danish Commons*. Munksgaard in Komm., 1948. Google-Books-ID: rpS8GAAACAAJ.

[91] A. A. Taha and A. Hanbury, "An Efficient Algorithm for Calculating the Exact Hausdorff Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 2153–2163, Nov. 2015.

[92] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York: Chapman and Hall/CRC, May 1994.

[93] Student, "The Probable Error of a Mean," *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908.

[94] "Levene, H. (1960) Robust Tests for Equality of Variances. In Olkin, I., Ed., Contributions to Probability and Statistics, Stanford University Press, Palo Alto, 278-292. - References - Scientific Research Publishing."

[95] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, pp. 44–53, Jan. 2018.

[96] M. Romero, Y. Interian, T. Solberg, and G. Valdes, "Targeted transfer learning to improve performance in small medical physics datasets," *Medical Physics*, vol. 47, pp. 6246–6256, Dec. 2020.

[97] Z. N. K. Swati, Q. Zhao, M. Kabir, F. Ali, Z. Ali, S. Ahmed, and J. Lu, "Brain tumor classification for MR images using transfer learning and fine-tuning," *Computerized Medical Imaging and Graphics*, vol. 75, pp. 34–46, July 2019.

[98] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling Laws for Neural Language Models," Jan. 2020. arXiv:2001.08361 [cs].

[99] P. Furtado, "Loss, post-processing and standard architecture improvements of liver deep learning segmentation from Computed Tomography and magnetic resonance," *Informatics in Medicine Unlocked*, vol. 24, p. 100585, Jan. 2021.

[100] J. Zubin, "A technique for measuring like-mindedness," *The Journal of Abnormal and Social Psychology*, vol. 33, no. 4, pp. 508–516, 1938.

[101] "Anthropology Publications," Dec. 2020.

[102] G. McLachlan, "Cluster analysis and related techniques in medical research," *Statistical Methods in Medical Research*, vol. 1, pp. 27–48, Mar. 1992.

[103] K. Poulakis and E. Westman, "Clustering and disease subtyping in Neuroscience, toward better methodological adaptations," *Frontiers in Computational Neuroscience*, vol. 17, Oct. 2023.

[104] A. Dahl and N. Zaitlen, "Genetic Influences on Disease Subtypes," *Annual Review of Genomics and Human Genetics*, vol. 21, pp. 413–435, Aug. 2020.

[105] E. Parimbelli, S. Marini, L. Sacchi, and R. Bellazzi, "Patient similarity for precision medicine: A systematic review," *Journal of Biomedical Informatics*, vol. 83, pp. 87–96, July 2018.

[106] T. J. Loftus, B. Shickel, J. A. Balch, P. J. Tighe, K. L. Abbott, B. Fazzone, E. M. Anderson, J. Rozowsky, T. Ozrazgat-Baslanti, Y. Ren, S. A. Berceli, W. R. Hogan, P. A. Efron, J. R. Moorman, P. Rashidi, G. R. Upchurch, and A. Bihorac, "Phenotype clustering in health care: A narrative review for clinicians," *Frontiers in Artificial Intelligence*, vol. 5, p. 842306, Aug. 2022.

[107] M. C. V. Guilamet, M. Bernauer, S. T. Micek, and M. H. Kollef, "Cluster analysis to define distinct clinical phenotypes among septic patients with bloodstream infections," *Medicine*, vol. 98, p. e15276, Apr. 2019.

[108] N. Mars, A. Kerola, M. Kauppi, M. Pirinen, O. Elonheimo, and T. Sokka-Isler, "Cluster analysis identifies unmet healthcare needs among patients with rheumatoid arthritis," *Scandinavian Journal of Rheumatology*, vol. 51, pp. 355–362, Sept. 2022. _eprint: https://doi.org/10.1080/03009742.2021.1944306.

[109] H. Muhammad, C. S. Sigel, G. Campanella, T. Boerner, L. M. Pak, S. Büttner, J. N. M. IJzermans, B. G. Koerkamp, M. Doukas, W. R. Jarnagin, A. L. Simpson, and T. J. Fuchs, "Unsupervised Subtyping of Cholangiocarcinoma Using a Deep Clustering Convolutional Autoencoder," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019* (D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, eds.), vol. 11764, pp. 604–612, Cham: Springer International Publishing, 2019. Series Title: Lecture Notes in Computer Science.

[110] S. Pai, I. Hadzic, A. Fedorov, R. H. Mak, and H. J. Aerts, "Foundation model embeddings for quantitative tumor imaging biomarkers," May 2025.

[111] T. H. Julian, H. Dou, J. Duan, J. Huang, E. Yoo, D. J. Green, A. Strange, E. Alhathli, M. Sperrin, P. A. Keane, E. Y. Chew, B. Keavney, T. W. Fitzgerald, J. Cooper-Knock, E. Birney, A. F. Frangi, and P. I. Sergouniotis, "Multi-omic analysis of deep learning-derived phenotypes links ophthalmic imaging to cardiovascular and neurological traits," *Nature Cardiovascular Research*, vol. 5, pp. 541–554, June 2026.

[112] Y. Feng, M. Kim, X. Yao, K. Liu, Q. Long, and L. Shen, "Deep multiview learning to identify imaging-driven subtypes in mild cognitive impairment," *BMC Bioinformatics*, vol. 23, p. 402, Sept. 2022.

[113] E. Vorontsov, A. Bozkurt, A. Casson, G. Shaikovski, M. Zelechowski, K. Severson, E. Zimmermann, J. Hall, N. Tenenholtz, N. Fusi, E. Yang, P. Mathieu, A. van Eck, D. Lee, J. Viret, E. Robert, Y. K. Wang, J. D. Kunz, M. C. H. Lee, J. H. Bernhard, R. A. Godrich, G. Oakley, E. Millar, M. Hanna, H. Wen, J. A. Retamero, W. A. Moye, R. Yousfi, C. Kanan, D. S. Klimstra, B. Rothrock, S. Liu, and T. J. Fuchs, "A foundation model for clinical-grade computational pathology and rare cancers detection," *Nature Medicine*, vol. 30, pp. 2924–2935, Oct. 2024.

[114] H. Wahab, I. Mehmood, and H. Ugail, "CapCLIP: A Vision-Language Representation Alignment Approach for Wireless Capsule Endoscopy Analysis," May 2026. arXiv:2605.08493 [cs.CV].

[115] A. Mahdizadeh, P. A. Moghadam, X. He, S. Mirabbasi, P. Nasiopoulos, and L. Sigal, "SCALE-VLP: Soft-Weighted Contrastive Volumetric Vision-Language Pre-training with Spatial-Knowledge Semantics," Nov. 2025. arXiv:2511.02996 [cs.CV].

[116] T. Ding, S. J. Wagner, A. H. Song, R. J. Chen, M. Y. Lu, A. Zhang, A. J. Vaidya, G. Jaume, M. Shaban, A. Kim, D. F. K. Williamson, H. Robertson, B. Chen, C. Almagro-Pérez, P. Doucet, S. Sahai, C. Chen, C. S. Chen, D. Komura, A. Kawabe, M. Ochi, S. Sato, T. Yokose, Y. Miyagi, S. Ishikawa, G. Gerber, T. Peng, L. P. Le, and F. Mahmood, "A multimodal whole-slide foundation model for pathology," *Nature Medicine*, vol. 31, pp. 3749–3761, Nov. 2025.

[117] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-1809.1936.tb02137.x.

[118] A. Unwin and K. Kleinman, "The Iris Data Set: In Search of the Source of Virginica," *Significance*, vol. 18, pp. 26–29, Dec. 2021.

[119] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5.1, pp. 281–298, University of California Press, Jan. 1967.

[120] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, Mar. 1982.

[121] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Boston, MA: Springer US, 1981.

[122] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, (Portland, Oregon), pp. 226–231, AAAI Press, Aug. 1996.

[123] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, pp. 559–572, Nov. 1901. _eprint: https://doi.org/10.1080/14786440109462720.

[124] L. Maaten and G. E. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, 2008.

[125] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," Sept. 2020. arXiv:1802.03426 [stat.ML].

[126] J. Pagès, "Analyse factorielle de données mixtes," *Revue de Statistique Appliquée*, vol. 52, no. 4, pp. 93–111, 2004.

[127] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, Dec. 1985.

[128] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," July 2008. ISBN: 9780511809071.

[129] W. M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association*, vol. 66, pp. 846–850, Dec. 1971.

[130] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, pp. 224–227, Apr. 1979.

[131] A. C. Yu, B. Mohajer, and J. Eng, "External Validation of Deep Learning Algorithms for Radiologic Diagnosis: A Systematic Review," *Radiology: Artificial Intelligence*, vol. 4, p. e210064, May 2022.

[132] S. Müller and P. Berens, "Mitigating Shortcut Learning via Feature Disentanglement in Medical Imaging: A Benchmark Study," Apr. 2026. arXiv:2602.18502 [cs.CV].

[133] C. Ong Ly, B. Unnikrishnan, T. Tadic, T. Patel, J. Duhamel, S. Kandel, Y. Moayedi, M. Brudno, A. Hope, H. Ross, and C. McIntosh, "Shortcut learning in medical AI hinders generalization: method for estimating AI model generalization without external data," *npj Digital Medicine*, vol. 7, p. 124, May 2024.

[134] A. J. DeGrave, J. D. Janizek, and S.-I. Lee, "AI for radiographic COVID-19 detection selects shortcuts over signal," *Nature Machine Intelligence*, vol. 3, pp. 610–619, July 2021.

[135] B. G. Hill, F. L. Koback, and P. L. Schilling, "The risk of shortcutting in deep learning algorithms for medical imaging research," *Scientific Reports*, vol. 14, p. 29224, Nov. 2024.

[136] M. A. Badgeley, J. R. Zech, L. Oakden-Rayner, B. S. Glicksberg, M. Liu, W. Gale, M. V. McConnell, B. Percha, T. M. Snyder, and J. T. Dudley, "Deep learning predicts hip fracture using confounding patient and healthcare variables," *npj Digital Medicine*, vol. 2, p. 31, Apr. 2019.

[137] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, pp. 665–673, Nov. 2020.

[138] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," Nov. 2018. arXiv:1610.01644 [stat.ML].

[139] S. Cajas, A. Marzullo, S. Kapadia, F. Santos, F. O. Osorio, Q. Kong, A. Quarta, P.-C. Kuo, M. Patel, R. I. R. Sillery, and L. A. Celi, "Shortkit-ML: A Unified Multi-Perspective Framework for Detecting Shortcut Learning in Medical Imaging Embeddings," Apr. 2026. ISSN: 3067-2007 Pages: 2026.04.29.26352053.

[140] B. Glocker, C. Jones, M. Roschewitz, and S. Winzeck, "Risk of Bias in Chest Radiography Deep Learning Foundation Models," *Radiology: Artificial Intelligence*, vol. 5, p. e230060, Sept. 2023.

[141] S. M. Hooper, J. A. Dunnmon, M. P. Lungren, D. Mastrodicasa, D. L. Rubin, C. Ré, A. Wang, and B. N. Patel, "Impact of Upstream Medical Image Processing on Downstream Performance of a Head CT Triage Neural Network," *Radiology: Artificial Intelligence*, vol. 3, p. e200229, July 2021.

[142] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009. ISSN: 1063-6919.

[143] M. G. KENDALL, "A NEW MEASURE OF RANK CORRELATION," *Biometrika*, vol. 30, pp. 81–93, June 1938.

[144] R. Bellman, *Dynamic Programming*. Princeton University Press, Aug. 2021.

[145] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[146] L. Wilson, K. Boase, L. D. Nelson, N. R. Temkin, J. T. Giacino, A. J. Markowitz, A. Maas, D. K. Menon, G. Teasdale, and G. T. Manley, "A Manual for the Glasgow Outcome Scale-Extended Interview," *Journal of Neurotrauma*, vol. 38, pp. 2435–2446, Sept. 2021.

[147] D. A. Zuckerman, J. T. Giacino, and Y. G. Bodien, "Traumatic Brain Injury: What Is a Favorable Outcome?," *Journal of Neurotrauma*, vol. 39, pp. 1010–1012, July 2022.

[148] I. Hadzic, S. Pai, K. Bressem, B. Foldyna, and H. J. Aerts, "Lighter: Configuration-Driven Deep Learning," *Journal of Open Source Software*, vol. 10, p. 8101, July 2025.

[149] M. J. Cardoso, W. Li, R. Brown, N. Ma, E. Kerfoot, Y. Wang, B. Murrey, A. Myronenko, C. Zhao, D. Yang, V. Nath, Y. He, Z. Xu, A. Hatamizadeh, A. Myronenko, W. Zhu, Y. Liu, M. Zheng, Y. Tang, I. Yang, M. Zephyr, B. Hashemian, S. Alle, M. Z. Darestani, C. Budd, M. Modat, T. Vercauteren, G. Wang, Y. Li, Y. Hu, Y. Fu, B. Gorman, H. Johnson, B. Genereaux, B. S. Erdal, V. Gupta, A. Diaz-Pinto, A. Dourson, L. Maier-Hein, P. F. Jaeger, M. Baumgartner, J. Kalpathy-Cramer, M. Flores, J. Kirby, L. A. D. Cooper, H. R. Roth, D. Xu, D. Bericat, R. Floca, S. K. Zhou, H. Shuaib, K. Farahani, K. H. Maier-Hein, S. Aylward, P. Dogra, S. Ourselin, and A. Feng, "MONAI: An open-source framework for deep learning in healthcare," Nov. 2022. arXiv:2211.02701 [cs.LG].

[150] B. Efron, "Bootstrap Methods: Another Look at the Jackknife," *The Annals of Statistics*, vol. 7, pp. 1–26, Jan. 1979.

[151] M. J. Gardner and D. G. Altman, "Confidence intervals rather than P values: estimation rather than hypothesis testing.," *British Medical Journal (Clinical research ed.)*, vol. 292, pp. 746–750, Mar. 1986.

[152] S. H. Tan and S. B. Tan, "The Correct Interpretation of Confidence Intervals," *Proceedings of Singapore Healthcare*, vol. 19, pp. 276–278, Sept. 2010.

[153] A. Hazewinkel, J. Bowden, K. H. Wade, T. Palmer, N. J. Wiles, and K. Tilling, "Sensitivity to missing not at random dropout in clinical trials: Use and interpretation of the trimmed means estimator," *Statistics in Medicine*, vol. 41, pp. 1462–1481, Apr. 2022.

[154] F. Orlhac, J. J. Eertink, A.-S. Cottereau, J. M. Zijlstra, C. Thieblemont, M. Meignan, R. Boellaard, and I. Buvat, "A Guide to ComBat Harmonization of Imaging Biomarkers in Multicenter Studies," *Journal of Nuclear Medicine*, vol. 63, pp. 172–179, Feb. 2022.

[155] W. E. Johnson, C. Li, and A. Rabinovic, "Adjusting batch effects in microarray expression data using empirical Bayes methods," *Biostatistics*, vol. 8, pp. 118–127, Jan. 2007.