SUMMER INTERNSHIP REPORT

ON

MULTI-MODEL REPRESENTATION LEARNING FOR CLINICAL DATA AND DDP OPTIMIZATION FOR OPTICAL FLOW

submitted in partial fulfillment of the requirements for
the completion of 7th semester of

B.Tech.

in

Computer Science and Engineering

By

Gokul Adethya T (106121045)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
TIRUCHIRAPPALLI – 620 015

NOVEMBER 2024

BONAFIDE CERTIFICATE

This is to certify that the report titled **MULTI-MODEL REPRESENTATION LEARNING FOR CLINICAL DATA AND DDP OPTIMIZATION FOR OPTICAL FLOW** is a bonafide record of the work done by

Gokul Adethya T (106121045)

in partial fulfillment of the requirements for the completion of seventh semester of **Bachelor of Technology** in **Computer Science and Engineering** of the **NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI,** during the year 2024.

DR. PUNIT RATHORE
Internship Guide / Mentor

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my guide and mentor, **Dr. Punit Rathore**, whose invaluable guidance, support, and encouragement made this work possible. Their insights and expertise have been instrumental throughout this research journey.

I also wish to extend my heartfelt thanks to **Dr. R. Bala Krishnan** for his mentorship during the summer internship program. I am deeply grateful to **Dr. S. Mary Saira Bhanu**, Head of the Department of Computer Science and Engineering, for her constant encouragement and for providing a conducive environment for learning and research. My heartfelt appreciation goes to **Dr. G. Aghila**, Director of National Institute of Technology, Trichy, for her leadership and vision, which have contributed immensely to the institution's growth and to my academic success.

Finally, I would like to extend my gratitude to the National Institute of Technology, Trichy, itself for offering me a stimulating academic environment, invaluable resources, and a vibrant community, all of which played a pivotal role in my personal and professional development. Thank you to everyone who has supported me in this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

1.1 Multi-Modal Representation Learning

Electronic Health Records (EHR) are comprehensive digital records that capture detailed information about a patient's health. These records include both structured data, such as demographics, vital signs, and lab test results, as well as unstructured data, like clinical notes and reports. EHR systems are widely used today for efficient and effective management of health records [1]. For instance, the U.S. healthcare system serves over 30 million patients annually, and the adoption of basic EHR systems by non-federal acute care hospitals significantly rose from 9.4% in 2008 to 83.8% in 2015 [2]. As of 2021, 78% of office-based physicians and 96% of non-federal acute care hospitals have adopted certified EHR systems [3]. Given this widespread usage, EHR databases now contain vast amounts of data, providing an invaluable resource for healthcare researchers to conduct data-driven studies aimed at improving patient outcomes [4].

Recent advancements in machine learning and deep learning have sparked interest in leveraging EHR data for healthcare applications [10–12]. These techniques show great potential for extracting valuable insights from EHRs, aiding in the accurate prediction of clinical outcomes like mortality [13] and readmission [13,14]. Predicting such outcomes can facilitate the early detection of patient deterioration [15], which supports more efficient nursing workflows. Many research studies have used deep learning to build predictive models based on EHR data, typically using vital signs, lab results, diagnostic histories, and medication information. However, unstructured data from sources such as clinical notes and radiology reports, available during patient admissions, could further enhance these models. As such, utilizing multimodal data from various sources during the prediction process can intuitively improve model performance.

In this study, we focus on integrating multimodal data—including clinical time series, chest X-ray radiographs (CXR), radiology notes, and ECG data—within a general fusion framework to enhance predictions of in-hospital mortality, extended length of stay, and 30-day readmission rates. We conduct ablation studies on various modality

combinations to investigate the impact of modality misalignment. The model is trained and tested using the extended MIMIC-MM dataset, which combines data from MIMIC-IV, MIMIC-CXR, MIMIC-IV-Note, and MIMIC-ECG datasets [5–8], while addressing missing modalities.

In summary, the key contributions of this work are:

- We propose a multimodal fusion framework that handles missing modalities, combining clinical time series (e.g., vital signs, lab results) with CXR images, radiology notes, and ECG data in EHR.
- We perform experiments on different modality combinations and use oversampling to study the effects of modality alignment, proposing solutions to address misalignment.

1.2 DDP Optical Flow Estimation

Optical flow estimation refers to the task of determining the apparent motion velocities of brightness patterns between two images, typically consecutive frames in a video, at the pixel level. This process is crucial in various computer vision applications, such as action recognition [29], video denoising [40], and frame interpolation [31]. As a fundamental vision task, optical flow estimation has been extensively studied over the past few decades, with two primary approaches emerging. Traditional methods, such as Lucas-Kanade [30] and Gunner-Farneback [34], formulate optical flow as an optimization problem between two images, producing a sparse or dense displacement map that aligns similar visual patterns.

In contrast, deep neural networks (DNNs) have recently demonstrated remarkable success in learning pixel-level tasks such as super-resolution [35], [36], semantic segmentation [15], image deblurring [41], image generation [22], and stylization [21]. DNNs have also been employed for optical flow estimation from consecutive frames [25], [28], [33], [47], achieving state-of-the-art performance on benchmark datasets like KITTI 2015 [16] and MPI Sintel [8]. However, a key challenge for DNN-based methods is the significant performance drop when models trained on one data distribution are applied to another. This distribution shift issue is particularly critical for DNN-based optical flow models during testing. Unlike humans, who can perceive

motion but struggle with accurately estimating motion vectors, capturing precise ground-truth optical flow in natural videos would require tracking 3D pixel trajectories, which is impractical for real-world scenarios.

This research aims to develop an optical flow model using Distributed Data Parallel (DDP) optimization across multi-GPU and multi-node setups. By leveraging Lightning-Fabric for distributed processing, the study explores advanced training techniques, including Fairscale's CPU offloading, mixed-precision training, and activation checkpointing, to maximize batch sizes and optimize large-scale optical flow models. Additionally, it investigates a test-time adaptation (TTA) algorithm to enhance model performance on unseen data by refining the optical flow map through iterative updates. The main objectives of the research are:

- To improve training efficiency for optical flow models through distributed training and memory optimization.
- To evaluate performance trends of optical flow estimation in multi-GPU environments.
- To create a test-time adaptation algorithm that enhances the model's generalization to new data by minimizing variance between original and augmented optical flow maps.

CHAPTER 2

2.1 Multi-Modal Representation Learning

Medical datasets comprise large collections of patient health records from hospitals, typically including various health-related aspects such as demographic data, lab tests, vital signs, medical images, diagnosis codes, notes, treatment and medication histories, and discharge summaries. Researchers in the medical field have increasingly applied machine learning techniques to various tasks, including medical predictive modeling, recommendations, disease diagnosis, and outcome prediction.

2.1.1 Research on EHR Time Series Variables:

Several studies have focused on leveraging time series data in electronic health records (EHR) for predictive modeling. RETAIN [10] used reversed time attention produced by RNNs to generate visit- and variable-level attention scores for embedding vectors of clinical time series. It incorporates diagnoses, medications, and procedures to form input vectors. Similarly, Dipole [16] employs a bidirectional RNN to combine multi-visit embeddings, while Med2Vec [11] learned visit-level representations from EHR by analyzing visit sequences and medical code co-occurrences. Med2Vec's representation was tested by predicting future medical codes and Clinical Risk Group (CRG) levels. BERT-based frameworks, such as Med-BERT [12], BEHRT [17], and G-BERT [18], have also been utilized for EHR feature extraction, particularly in diagnosis code and medication prediction tasks. G-BERT additionally incorporates the hierarchical structure of ICD-9 codes to improve embeddings. Ashfaq et al. [14] applied LSTM on learned EHR embeddings to predict 30-day readmission rates.

2.1.2 Research on Multimodal Data Input:

Medical datasets are multimodal, including data types like time-series variables (e.g., lab tests and vital signs), medical images, and unstructured text from clinical notes. Combining complementary information from different data sources holds great potential [19]. Zhang et al. [13] integrated time series data with unstructured clinical notes from MIMIC-III using LSTM and CNN for sequential feature extraction to

perform predictive modeling. Golovanevsky et al. [20] combined clinical test scores, genetic data (SNPs), and MRI images to diagnose Alzheimer's disease, employing cross-modal attention and self-attention modules to capture intra- and inter-modality correlations. Huang et al. [21] utilized Electronic Medical Records (EMR) and CT scan images to detect pulmonary embolism, with late fusion showing superior performance over other fusion methods. Yao et al. [22] concatenated selected clinical features with 3D CT image features from CNN for predicting pulmonary venous obstruction (PVO) and used a saliency map to identify the areas of focus in the model. Yan et al. [23] conducted breast cancer classification by combining pathological images and 29 selected features. They concatenated hidden states from CNN layers as image features, used a denoising autoencoder for EMR features, and combined both for classification. Nie et al. [24] integrated multi-channel medical images, demographic data, and tumor-related features to predict short overall survival (OS) time. Soenksen et al. [25] proposed an early fusion model that integrates tabular data, time series, text notes, and chest X-rays for diagnosing chest pathology, predicting length of stay, and forecasting 48-hour mortality. Many studies utilizing more than two modalities neglect robust cross-modal alignment techniques. When alignment is attempted, it often lacks thorough analysis of its impact. Most evaluations rely heavily on AUROC and AUPRC metrics, with scant attention to F1 scores, which commonly fall below 0.7, often within the 0.5 to 0.6 range, indicating subpar balanced precision and recall. ECG data is infrequently integrated with other modalities in existing research and has not been adequately tested for the tasks addressed in this study. This may stem from its limited suitability for certain tasks, as the observed low variance in ECG data could restrict its contribution to meaningful multimodal fusion. In our study we perform ablation to understand this further.

2.2. DDP Optical Flow Estimation

Optical flow prediction is a crucial task in computer vision that involves estimating the motion of objects between consecutive video frames at the pixel level. This technique enables applications such as motion tracking, object detection, and video stabilization. The KITTI dataset, one of the most widely used benchmarks for optical flow research, provides a diverse collection of real-world driving scenes captured from a moving vehicle. It includes ground truth optical flow data, allowing researchers to evaluate and compare the performance of various optical flow estimation algorithms. The dataset's

challenging conditions, such as occlusions and varying lighting, make it an essential resource for developing robust models that can generalize well to real-world scenarios.

2.2.1. Optical Flow Estimation

Traditional optical flow methods often adopt a variational framework, solving an energy minimization problem to align brightness patterns and enforce regularities in the flow field. Since the foundational work by Lucas et al. [30], this approach has seen significant success, further improved by techniques like coarse-to-fine refinement and descriptor matching [36], [37], [43], [44]. However, these methods typically target short-range motion and struggle with incomplete correspondences, making them less effective for large motions and real-world scenarios with occlusions.

Data-driven approaches to optical flow estimation have grown with the advent of deep learning. Dosovitskiy et al. [14] introduced FlowNet, the first DNN-based model capable of directly predicting dense optical flow maps from image pairs. FlowNet's success, achieved without complex optimization steps, spurred extensive research into DNN-based optical flow methods [25], [28], [33], [41], [47]. Recent models like RAFT [48] have surpassed traditional methods [29] on standard benchmarks [8], [16]. Despite varying architectures, these models are typically trained using supervised learning on large-scale synthetic datasets before being fine-tuned on smaller, real-world datasets. This reliance on synthetic data limits the practicality of these methods in real-world applications, where obtaining ground-truth optical flow for even a few videos is difficult. Furthermore, without fine-tuning on the target data distribution, these models are prone to distribution shift at test time.

Unsupervised learning methods for optical flow, such as SelFlow [39], SMURF [59], and MDFlow [62], have emerged as alternatives, requiring no annotated optical flow for training. However, they still assume that the training videos share a similar distribution with the test data, which may not hold in practice as test data distributions can evolve. In contrast, our method relies solely on test data information to adjust the model, making it more practical for real-world applications.

2.2.2 Test-Time Adaptation

Test-time adaptation (TTA) refers to techniques that adjust a model based solely on the unlabeled test sample at test time, enhancing the model's performance on out-of-distribution data. A crucial aspect of TTA is designing a self-supervised learning task based on the test sample. Shocher et al. [42] introduced a sample-specific super-resolution model, where the task was to upscale a downscaled version of the test image back to its original resolution. Similarly, Chi et al. [53] proposed an auxiliary task of reconstructing a blurry input image from deep features to adapt the model to each test sample. Other approaches, such as those by Sun et al. [46] and Wang et al. [54], focus on improving image classification robustness by predicting rotation angles or minimizing prediction entropy, respectively. In this study, we propose the first TTA framework specifically for optical flow estimation models, designing a self-supervised task based on representations from compressed video data.

CHAPTER 3

3.1 Multi-Modal Learning Framework

3.1.1 Dataset Description

In this study, we employed a multi-modal representation learning approach using clinical data sourced from various modalities:

1. Electronic Health Records (EHR)

1.1. Demographic Data: This includes tabular and categorical variables such as race and age. Notably, age was discretized into groups due to deidentification, which rendered numerical inaccuracies while maintaining relative correctness.

1.2. ICU Vitals: Comprising time-series signals alongside categorical time-series data (e.g., categorical events such as procedures), we selected 39 vitals from approximately 100 available.

2. Chest X-rays (CXR): A series of chest X-ray images were included, representing time-series data.

3. Electrocardiograms (ECG): This modality consisted of a series of 12-lead ECG signals, characterized as time-series of time-series data across 12 dimensions.

4. Clinical Notes: These encompass discharge summaries and radiology notes. The discharge notes, however, were excluded from analysis due to revealing sensitive information.

3.1.2. Dataset Preprocessing

3.1.2.1. Tasks

The dataset was utilized to address the following predictive tasks:

- Readmission Prediction: Assessing the likelihood of patient readmission within a month post-discharge.
- Mortality Prediction: Evaluating whether patients will survive or succumb to their conditions.
- Length of Stay Prediction: Determining whether patients stay for more than 3 or 7 days.

3.1.2.2. Setup

For this analysis, we concentrated on utilizing data from less than 24 hours and 48 hours post-admission:

- <24 hours of Patient Data: We focused on this shorter timeframe due to the limited number of samples, which may enhance the complexity of the task.
- <48 hours of Patient Data: This larger timeframe allows for more data availability.

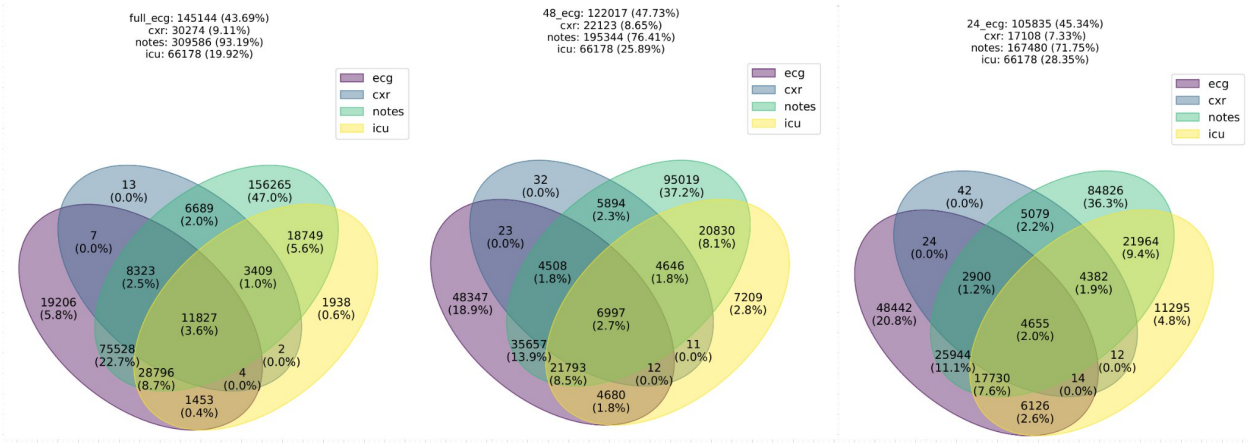

Figure 1: Relationships between modalities

Figure 1 illustrates the relationships between modalities linked to each other and their corresponding labels. The left panel represents the complete dataset, while the middle and right panels show samples collected within the first 24 and 48 hours post-admission, respectively.

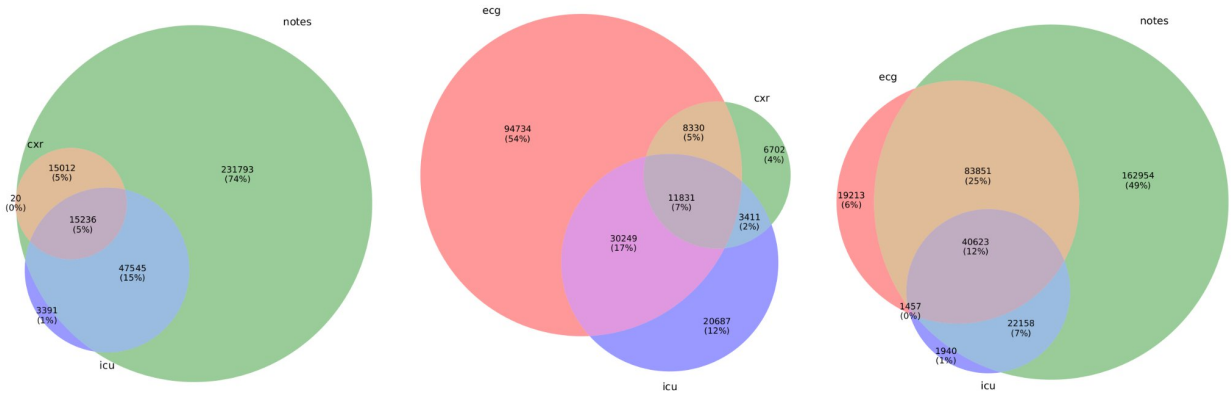

Figure 2 Relative composition of modalities using various combinations

Figure 2 presents the relative composition of modalities in various combinations: the left shows combinations of notes, CXR, and ICU vitals; the middle illustrates ECG, ICU, and CXR; while the right displays notes, ECG, and ICU.

3.1.2.3 Class Label Imbalance

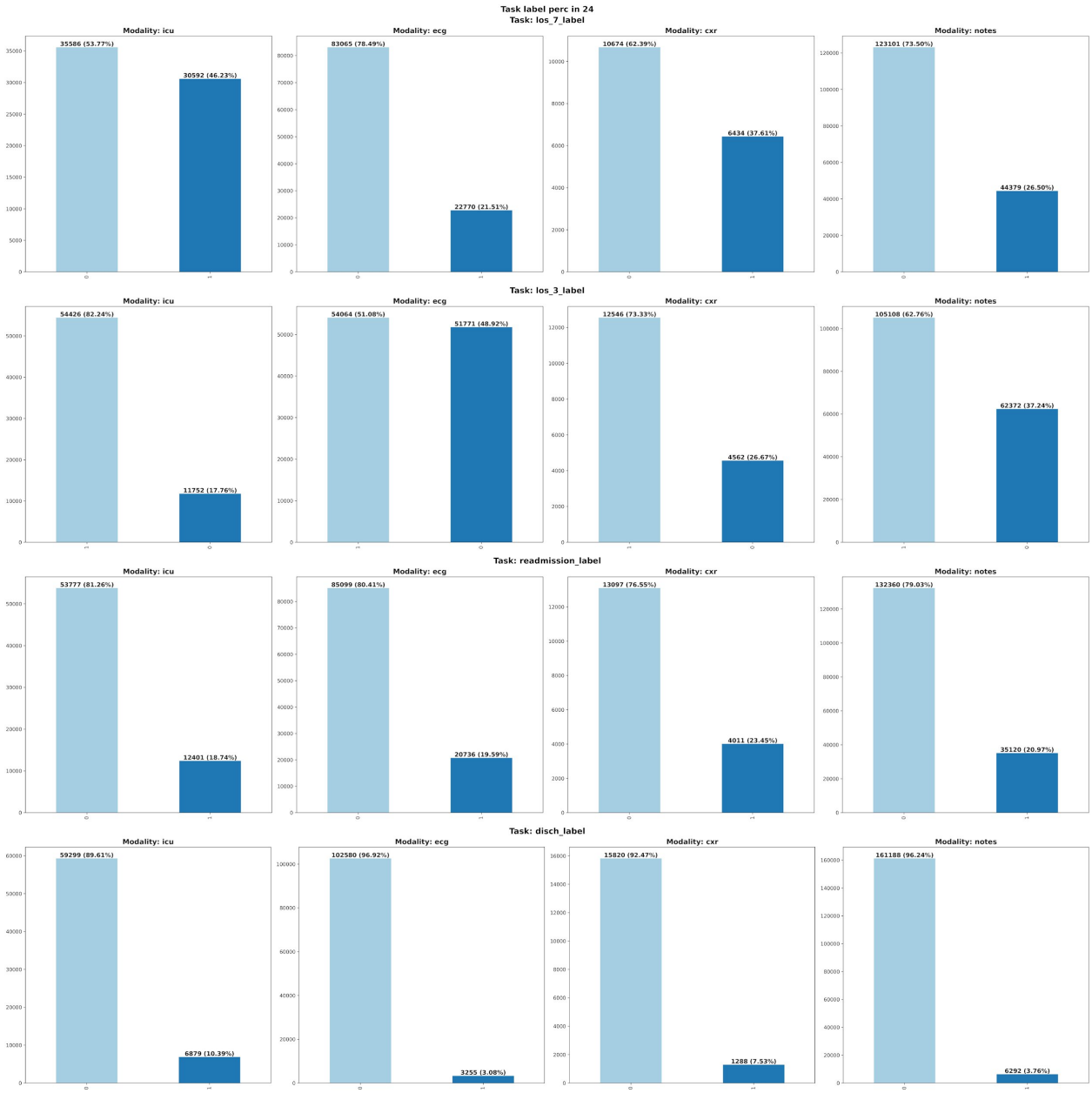

Figure 3: Class label composition for each modality for different tasks for <24hrs setup

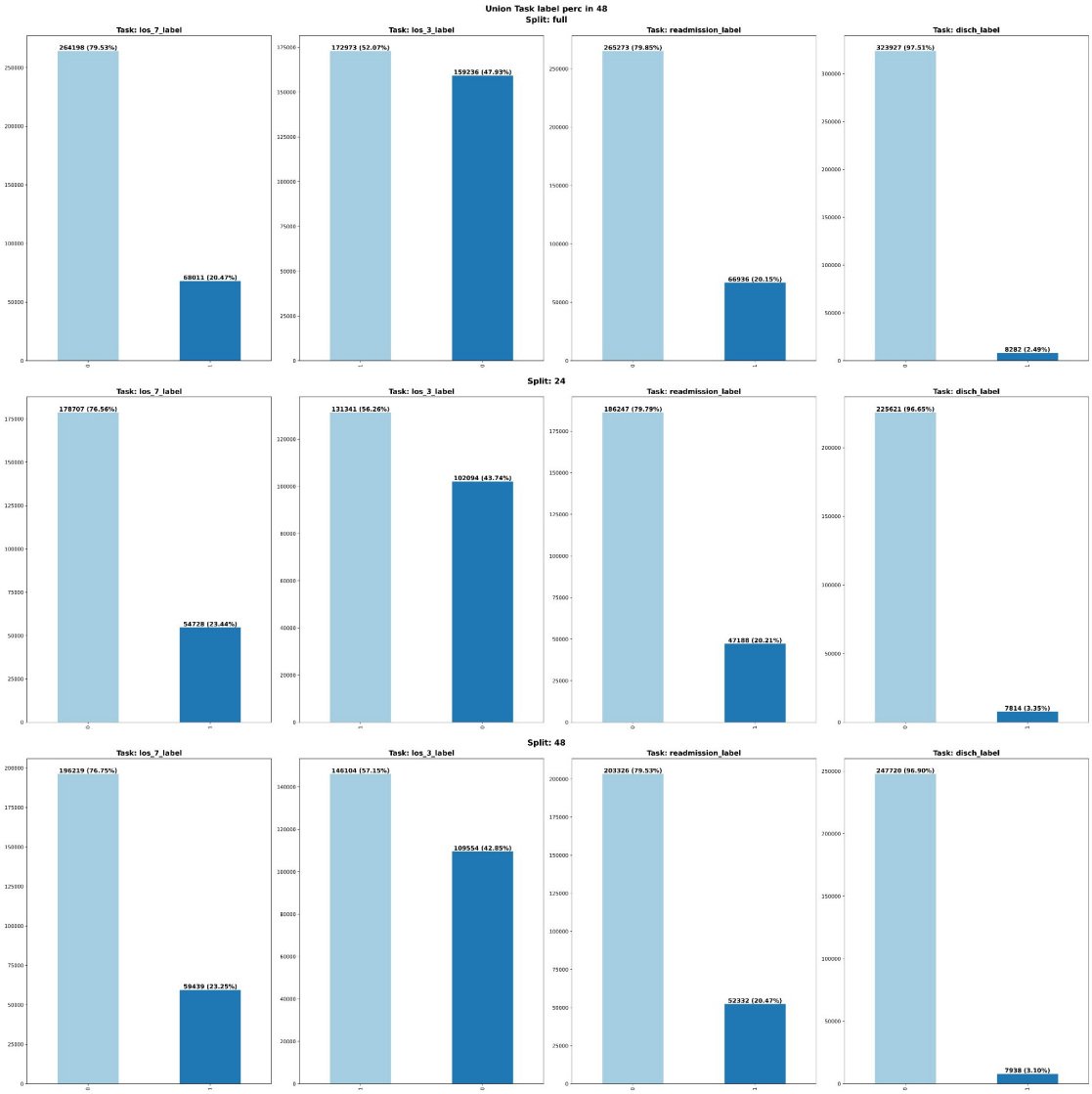

Figure 4: Union of modalities.

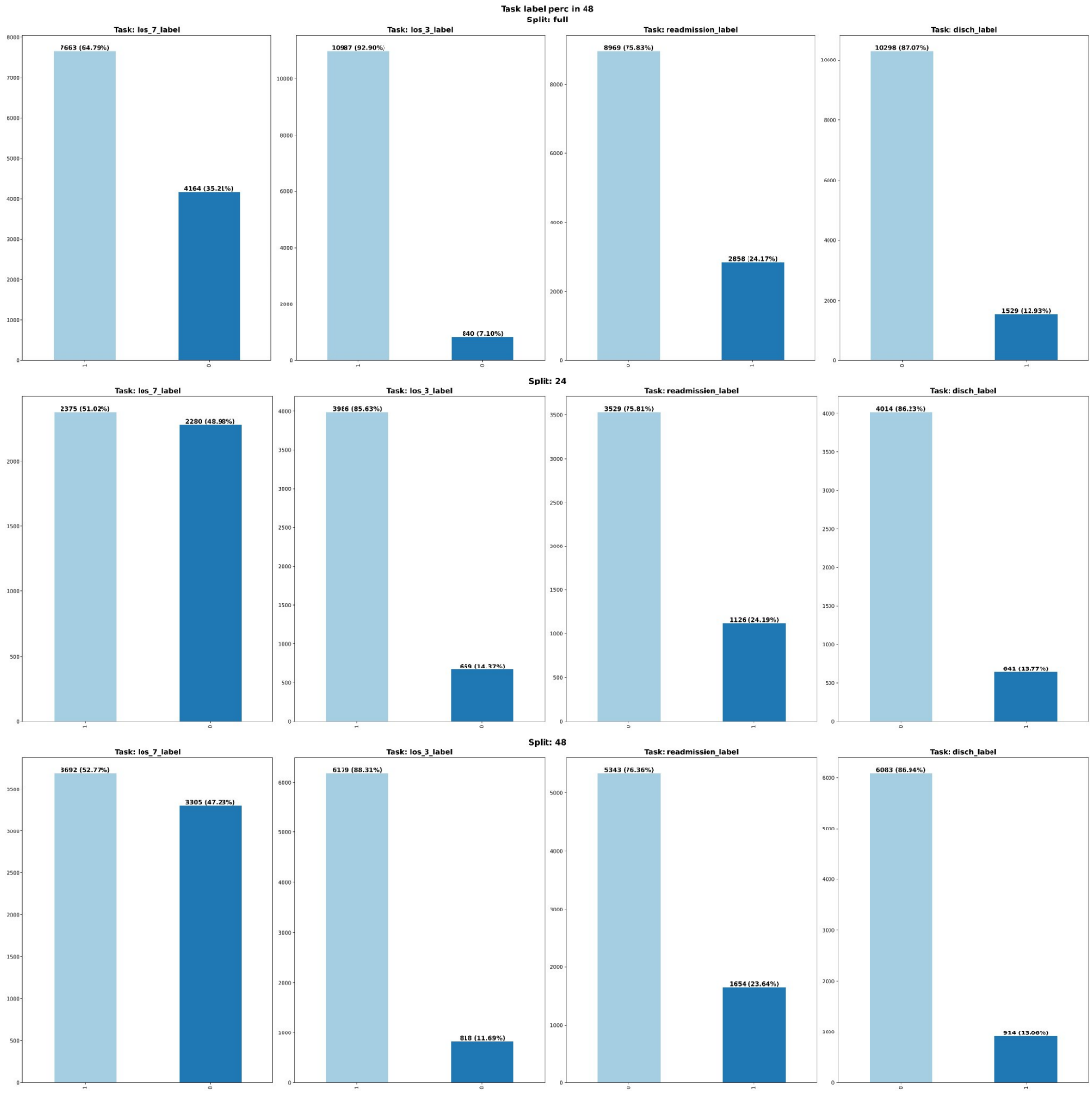

Figure 5: Interscetion of modalities.

Figure 4 illustrates the Class label composition for each modality for different tasks for <24hrs setup while the Figure 4 showcases the union of modalities, representing sampling strategies considering instances with missing modalities.

In contrast, Figure 5 illustrates the intersection of modalities, sampling only when all modalities are present.

3.1.2.4. Observations from Dataset Analysis

a. Class Imbalance Across Tasks and Modalities:

A substantial class imbalance was identified across nearly all tasks, modalities, and patient categories, especially for short-duration stays (less than 24 or 48 hours). This imbalance complicates model training and evaluation, risking biased performance metrics and inadequate generalization to unseen data.

b. Disproportionate Sample Distribution:

Each modality exhibited a skewed sample distribution relative to others, complicating multimodal learning where equal representation is preferred for effective information contribution.

c. Data Loss from Missing Modalities:

Excluding instances with missing modalities led to over 90% of available data being disregarded, significantly diminishing the dataset's size and potentially impairing model performance on multimodal data.

d. Random Sampling Misalignment:

Randomly sampled modalities resulted in disproportionate combinations, disrupting modality alignment and representation effects. Thus, effective sampling strategies are essential to accurately reflect the true relationships among modalities without introducing noise or bias.

3.1.3. Dataset Splitting/Balancing

1. Standardized Dataset Splitting:

A consistent split of the dataset into training, validation, and test sets was established, maintaining proportionality across tasks, modality combinations, and experimental setups for both supervised and self-supervised learning. Although the target split was 80%-10%-10%, the final distribution was 80.5% for training, 9.7% for testing, and 9.8% for validation. This approach ensures balanced class distributions for each task label and modality combination.

2. Balanced Sampling:

A balanced sampling strategy was adopted to ensure proportional representation of each modality combination, thereby enhancing the dataset's representativeness for each task and addressing class imbalance effectively.

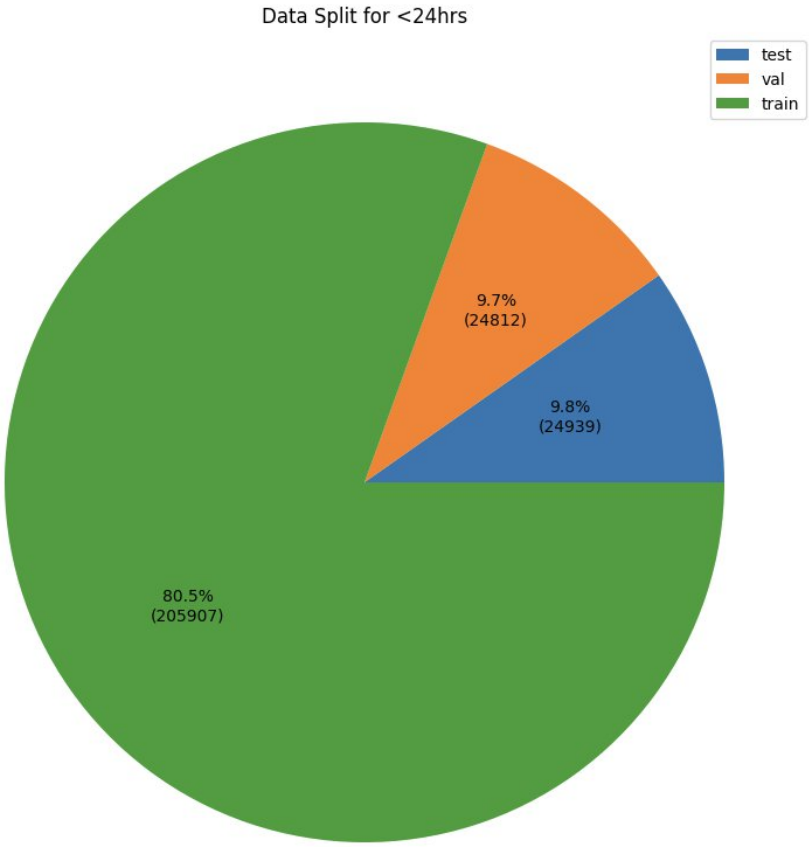

Figure 6: Train, Validation and Test split of the dataset.

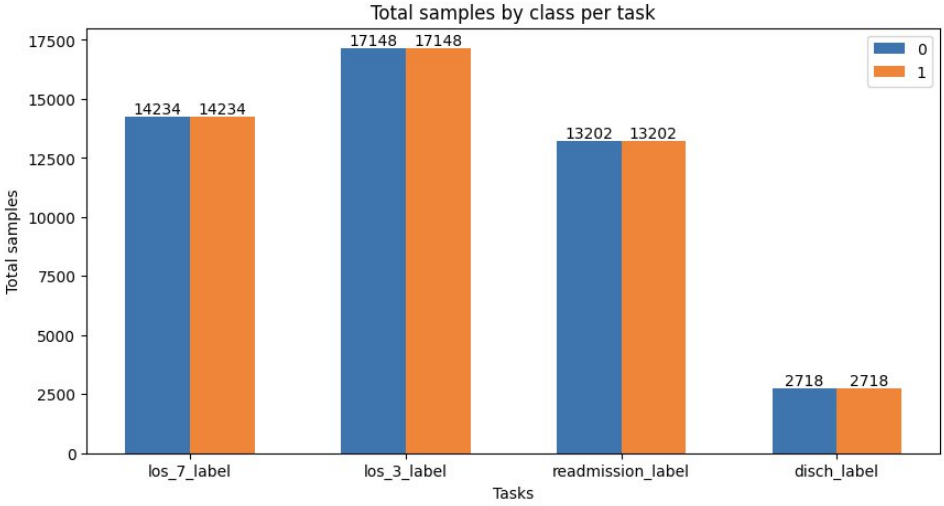

Figure 7: The class labels composition for each task in the validation set.

Figure 6 illustrates the training, validation, and test splits of the dataset, while Figure 7 provides a detailed composition of class labels for each task in the validation set. The mortality prediction task (denoted by the discharge label) reflects a skewed representation due to the rare occurrence of death events, maintaining a ratio

exceeding 10:1 for the minority class to prevent disruption to the training set.

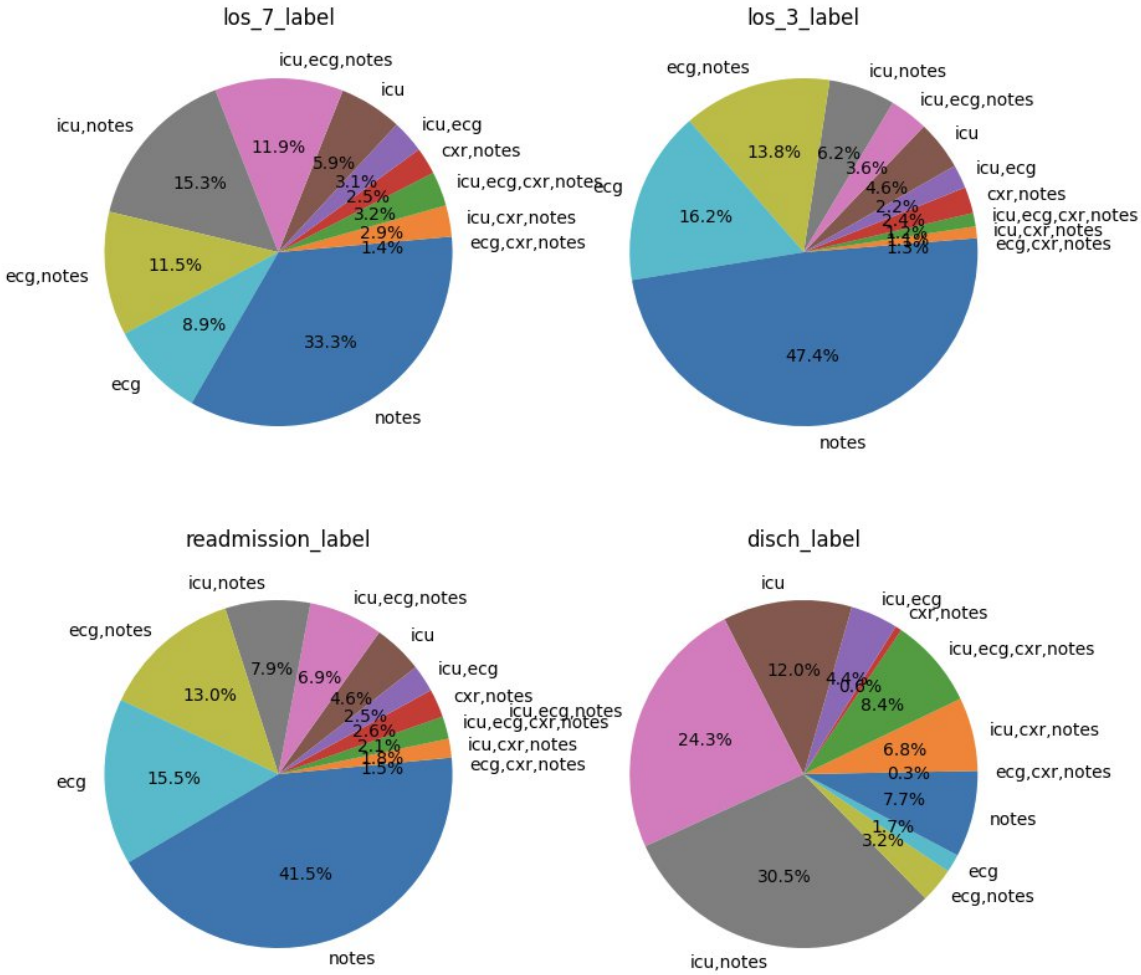

Figure 8: Percentage of different modality combinations with respect to different task labels.

Figure 8 displays the proportional representation of various modality combinations across different task labels, emphasizing that only combinations with a minimum of 50 samples were included to avoid outlier influence, while most combinations exceeded 100 samples. Notably, the "ECG, notes" combination does not overlap with "ECG," as these represent distinct intersections of modality combinations. Consequently, sampling within these intersections yields a balanced dataset when viewed holistically.

3.1.4. Data Description:

Admittime and dischtime were used to calculate the length of stay for each patient. The model's input was limited to the first 24 hours of monitored data, and this time

window was also used to generate the corresponding labels for predicting length of stay.

1. **Electronic Health Records (EHR):**

1.1. Demographic Data: The demographic data include the following categorical features:

anchor_age_num: This represents the de-identified age group of the patient (e.g., 0-9, 10-19, 91-100).

insurance_num: This encodes insurance types such as Medicaid, Medicare, and Other.

language_num: Binary feature indicating whether the patient speaks English or another language.

marital_status_num: Categorical feature representing marital status (Divorced, Married, Single, Widowed).

race_num: A feature that encodes the patient's race into one of 33 categories.

1.2. Time Series Data: Missing values in the time series were filled using the last available recorded value. If the missing entry was the first in the series, the mean value for that feature across the entire dataset was used. The time difference (delta time) was calculated using `charttime - admittime` to determine when the monitored value was recorded.

1.3. Chart Events (6 features): Heart Rate, Non-Invasive Blood Pressure (Systolic, Diastolic, and Mean), Respiratory Rate, O2 Saturation (Pulse Oximetry), Glasgow Coma Scale (GCS) - Verbal, Eye Opening, and Motor Response.

1.4. Lab Events (23 features): Glucose, Potassium, Sodium, Chloride, Creatinine, Urea Nitrogen, Bicarbonate, Anion Gap, Hemoglobin, Hematocrit, Magnesium, Platelet Count, Phosphate, White Blood Cells, Calcium (Total), MCH, Red Blood Cells, MCHC, MCV, RDW, Platelet Count, Neutrophils, Vancomycin.

1.5. Procedure Events (10 features): Foley Catheter, PICC Line, Intubation, Peritoneal Dialysis, Bronchoscopy, EEG, Dialysis (CRRT), Dialysis Catheter,

Chest Tube Removed, Hemodialysis.

2. Other Modalities:

2.1. CXR (Chest X-ray): Includes chest X-ray images and their corresponding chart times, which were converted to delta time.

2.2. Notes: Contains radiology reports (e.g., X-ray, CT, MRI) and the chart times of these reports, which were converted to delta time.

2.3. ECG: Includes 12-lead ECG data and corresponding chart times, which were also converted to delta time.

3.1.5. Model Architecture:

1. EHR Modality:

1.1. Modality Encoder:

1.1.1. Demographic Data: The demographic features (5 features) were encoded into embeddings of shape 38, resulting in a total demographic embedding *demo_embed* of size $5 \times 38 = 190$

1.1.2. Time Series Data: The time series data (39 features) included both chart events, lab events, and procedure events. The following two models were used to process these:

1.1.3. GRU-D: A standard GRU-D model that processes both the feature and time series data.

1.1.4. RNN with Time Series Data: The time stamps were passed through a Feed-Forward Network (FFN) to convert them from shape (N,) to (N, 39). The FFN output was then concatenated with the time series data (producing a shape of 78), and passed through the RNN. FFN layer consists of skip connection with sigmoid activation and linear layer as shown in Eq 1.

$$FFN(TS) = TS + Linear(Dropout(Sigmoid(Linear(TS)))) \quad (1)$$

1.1.5. RNN (2 Heads): The feature data was processed in an RNN with two layers, using two parallel heads. The first head processed the feature vector filled with mean or previous values X, while the second processed raw features with missing modalities filled as zero $X \times mask$ as shown in Eq 2.1-2.2. The outputs from both heads Z,Z' were concatenated and passed through a linear layer, producing an output

ts_embed of shape 578 as shown in Eq 2.3-2.4.

$$Z = RNN(X) \quad (2.1)$$

$$Z\prime = RNN(X \times mask) \quad (2.2)$$

$$If\ delta\ time\ TS\ is\ used: X = concat(X, FFN(TS)) \quad (2.3)$$

$$ts\_embed = Linear(ReLU(concat(Z, Z\prime))) \quad (2.4)$$

1.2. Time Series Encoder: A classic LSTM layer was applied to the (N, 578) vector, and the output from the final time step was used as the representation, resulting in an embedding of size 578 as shown in Eq 3.

$$ts\_embed = LSTM(ts\_embed) \quad (3)$$

1.3. Classifier: The time series embedding was concatenated with the demographic embedding, producing a vector of shape 768. This was passed through a linear layer to *output* logits for binary classification as shown in Eq 4.1-4.2:

$$embed = concat(ts\_embed, demo\_embed) \quad (4.1)$$

$$output = Linear(embed) \quad (4.2)$$

2. CXR, Notes, and ECG Modalities:

2.1. Modality Encoder:

2.1.1. CXR: A pretrained DenseNet model (torchxrayvision, densenet121-res224-mimic_ch) was used to generate a 1024-dimensional embedding for each chest X-ray.

2.1.2. Notes: Pretrained Clinical-T5-Base was used to generate embeddings for each radiology text, producing an embedding of shape 768.

2.1.3. ECG: A pretrained SEER ResNet model generated a 256-dimensional embedding for each 12-lead ECG recording. For each patient timestamp data X_i, the corresponding encoder generated a modality-specific embedding.

2.2. Intermediate Layer: For CXR and ECG, the embeddings were projected to 768 dimensions using a linear layer, while the notes embeddings remained at 768 due to the large size of the encoder.

$$M = Encoder(X_i) \quad (5.1)$$

$$M_i = Linear(X_i) \quad (5.2)$$

2.3. Timestamp Fusion: The delta time values were passed through a Feed-Forward Network (FFN) to convert their shape from $(1,)$ *to* $(768,)$, and the resulting

embedding was added to the intermediate layer's output M_i.

$$M_i = M_i + FFN(TS) \quad (5.3)$$

2.4. Time Series Encoder:

We incorporate a trainable padding token represented as a zero tensor and a classification token represented as a random tensor with a shape of 768. These are used to handle missing modalities, with the padding token embedding and classification token concatenated after the modality embedding or padding token embedding. We utilize a BERT transformer with 4 blocks and 4 attention heads to generate embeddings. The final embedding corresponding to the classification token is then used as shown in Eqn 6.1-6.2.

$$M = M + ([PAD], [PAD], \ldots n\ times) + [CLS] \quad (6.1)$$

Where $[PAD]$ is used to pad up to the maximum batch size or to handle missing data for this modality.

$$embed = BERT(M)[-1] \quad (6.2)$$

2.5. Classification Layer:

The generated embedding *embed* is passed through a linear layer to produce a logit output with shape 1. The classification output is:

$$output = Linear(embed) \quad (6.1)$$

3. Multi-Modality Fusion:

We utilize the embeddings before the classification stage for each modality and apply different fusion techniques prior to passing them to the classification layer.

3.1. Sum Fusion:

For sum fusion, the embeddings from each modality are summed:

$$embed = \sum embed_i \quad (7.1)$$

where i belongs to each modality

Afterward, the following transformation is applied:

$$output = Linear(ReLU(Linear(embed))) \quad (7.2)$$

3.2. Transformer Fusion:

For transformer-based fusion, we use a classification token $[CLS]$ of a random trainable tensor along with a trainable positional embedding, *pos_embed*, of size $(num\ of\ modalities, 768)$. The embedding is updated as shown in Eq 8.1-8.3 and is passed to BERT model to generate the final embeddings.

$$embed = embed + pos\_embed \quad (8.1)$$

$$embed = embed + [CLS] \quad (8.2)$$

$$embed = BERT(embed)[-1] \quad (8.3)$$

Classifier:

For the final classification step, the output is computed by passing the embedding through a linear layer as shown in Eq 8.4

$$output = Linear(embed) \quad (8.4)$$

3.1.6. Experimentation Setup:

1. Revamping the HADM Pipeline:

The HADM pipeline underwent significant modifications to accommodate missing modalities in the MIMIC IV dataset, particularly to support the inclusion of the ECG modality. The modifications were so extensive that a complete overhaul of the pipeline was necessary to ensure compatibility with the new data structure and to manage missing modalities efficiently.

2. Interlinked Data Loaders for Modalities:

Data loaders for each modality were designed to be queried in an interconnected manner, allowing for seamless multi-modal fusion. This setup ensures that any required modality can be accessed in conjunction with others, facilitating efficient

fusion during training and testing.

3. Training Configuration with PyTorch Lightning and DDP:

PyTorch Lightning was employed in combination with Distributed Data Parallel (DDP) to handle both single-modality and multi-modality training. The training setup is controlled through a single `.yaml` configuration file, ensuring uniformity and reproducibility across different experiments.

4. Monitoring with Weights & Biases (W&B):

The W&B platform was used to monitor several important aspects of the experiment:

- Binary Classification Metrics: Seven binary classification metrics were tracked across training, validation, and test sets to evaluate model performance.
- Single-Modality and Multi-Modality Performance: Metrics were used to separately evaluate single-modality samples (e.g., CXR-only or notes-only) and multi-modality samples (i.e., samples linked with other modalities).
- Metrics by Modality Combination: Metrics were also tracked for each combination of modalities, allowing a detailed view of performance across different modality configurations.

5. Embedding Visualization:

Embeddings from the training, validation, and test sets were sampled and visualized to examine how different training strategies, modalities, and modality combinations influence the learned representations.

6. Gradient Monitoring:

Gradients were tracked during training to monitor model optimization and to provide insights into learning dynamics.

7. Pipeline Structure for Modalities:

The pipeline for each modality is structured in the following way:

1. Modality Encoder: This component encodes individual samples for each modality.
2. Time Series Encoder: This encoder combines the embeddings of each sample into a unified representation for time-series data.

3. Classifier: A final binary classification head that generates the output for the model.

8. Class Imbalance Handling:

To handle class imbalance, the following strategies were used:

If the class distribution was less skewed than a 10:1 ratio, the minority class was oversampled to match the majority class. For more skewed distributions, the minority class was oversampled by 50% of the majority class, and the majority class was undersampled by 50%. Different random seeds were used in each epoch to maintain variability, while a pre-determined seed sequence ensured reproducibility and reduced overfitting on the oversampled minority classes.

3.2. Optical Flow Estimation

3.2.1. Optimization Technique

In this research, we focused on optimizing the existing **RAFT (Recurrent All-Pairs Field Transforms)** model for optical flow estimation using a multi-GPU and multi-node setup. The model was deployed in a distributed computing environment using **Distributed Data Parallel (DDP)** processing, facilitated by **Lightning-Fabric**. The primary aim of this setup was to efficiently scale the model's training to accommodate larger batch sizes without running into **out-of-memory (OOM)** errors, a common challenge in deep learning tasks with high computational demands like optical flow estimation. The RAFT model's architecture was not modified; instead, we concentrated on improving the scalability and memory handling to enhance computational efficiency.

To achieve these optimizations, a cluster of multiple GPUs and nodes was configured to maximize parallel processing capabilities. The use of **DDP** allowed us to distribute the workload across multiple GPUs and nodes, synchronizing their operations to ensure smooth and efficient model training. This setup provided the necessary computational resources to handle large-scale data while maintaining memory efficiency across devices.

Several key optimization techniques were employed to maximize the efficiency and scalability of the RAFT model in this distributed environment. First, **Fairscale's CPU offloading** was integrated to shift some memory loads from the GPU to the CPU, thereby freeing up GPU resources for more critical computations. This technique

helped mitigate memory bottlenecks, allowing for the handling of larger batch sizes. Next, we applied **mixed-precision training**, which combines 16-bit and 32-bit floating point operations. This significantly reduced memory consumption without compromising the accuracy of the optical flow estimations. Finally, we implemented **activation checkpointing** to further minimize memory usage. By only storing key intermediate results and recomputing others when necessary, we reduced the overall memory footprint during training, allowing for a higher batch size without OOM errors.

These optimizations, applied in tandem with a distributed multi-GPU, multi-node setup, enabled us to expand the computational capabilities of the RAFT model, ensuring more efficient training while preserving memory integrity.

3.2.2 Test-Time Adaptation Algorithm

We begin with an input image x, from which an initial optical flow map is generated:

$$opt = model(x) \tag{9.1}$$

Following this, the optical flow map is iteratively refined through a series of iii augmentation steps. Each step involves generating an augmented input $x\prime$ as follows:

$$x\prime = aug(opt, x) \tag{9.2}$$

The model then processes the augmented input $x\prime$ to produce an updated optical flow map:

$$opt\prime = model(x\prime) \tag{9.3}$$

The loss function is calculated as the variance between the original optical flow map opt and the augmented optical flow map $opt\prime$:

$$Loss = Variance(opt\prime, opt) \tag{9.4}$$

The primary goal of this algorithm is to reduce the variance between the optical flow map generated for the original input and the augmented images. The augmentation process generates random patches with pixel values constrained by the minimum and maximum values of the input image. These patches are then placed in random positions in both the original image x and its augmented counterpart $x\prime$. The optical flow map produced by the model during test-time adaptation guides the patch placement, and minimizing the variance between the original and augmented optical flow maps helps the model improve its predictions on unseen data.

CHAPTER 4

4.1. Metrics:

In this study, the performance of the model for the **mortality prediction task** within the MIMIC dataset is evaluated using three key metrics: F1 score, Area Under the Receiver Operating Characteristic Curve (AUROC), and End-Point Error (EPE). These metrics help provide a comprehensive understanding of the model's predictive accuracy, robustness, and effectiveness in handling both the binary classification task of mortality prediction and the continuous nature of optical flow estimation.

F1 Score

The F1 score is a metric that combines precision and recall into a single value, particularly useful for binary classification tasks like mortality prediction, where imbalanced data is often an issue. In the MIMIC dataset, where there are significantly fewer mortality cases compared to survivals, the F1 score helps assess how well the model balances false positives and false negatives. The formula for the F1 score is:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (10.1)$$

This metric gives equal importance to precision and recall, ensuring that both false positives and false negatives are minimized in predicting patient mortality.

AUROC Score

The AUROC (Area Under the Receiver Operating Characteristic Curve) is another critical metric used for the mortality prediction task. It assesses the model’s ability to distinguish between the two classes (mortality vs. survival) across different classification thresholds. The AUROC score reflects the probability that a randomly chosen positive instance (mortality) is ranked higher by the model than a randomly chosen negative instance (survival). A higher AUROC score indicates better discriminative performance, with a value of 1 representing perfect classification.

The AUROC score is calculated by plotting the true positive rate (recall) against the false positive rate (1-specificity) and measuring the area under this curve. The formula

for AUROC is:

$$AUROC = \int_0^1 TPR(t)dFPR(t) \quad (10.2)$$

This score provides insight into the model's overall performance across all classification thresholds, especially in imbalanced datasets where a high AUROC indicates strong classification power.

End-Point Error (EPE)

For the optical flow component of this study, we use the End-Point Error (EPE) to evaluate the precision of the model's flow predictions. EPE measures the Euclidean distance between the predicted flow vectors and the ground truth at each pixel, making it a standard metric in optical flow estimation. A lower EPE score indicates better alignment with the true motion.

The formula for EPE is:

$$EPE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(u_i - \hat{u}_i)^2 - (v_i - \hat{v}_i)^2} \quad (10.3)$$

Where:

- (u_i, v_i) represent the ground truth flow components in the x and y directions, respectively.
- $(\hat{u}_i, \hat{v}_i)$ are the predicted flow components.
- N is the number of pixels in the image.

EPE provides a direct quantitative measure of the accuracy of the optical flow model, with lower values indicating a closer match between the predicted and true optical flow.

4.2. Multi-Modal Learning Framework

1. EHR:

a. GRU-D vs RNN:

Table 1: GRU-D vs RNN F1-score comparison.

MODEL	TEST F1 SCORE
GRU-D	0.6625
RNN	0.71844

RNN demonstrates superior performance compared to GRU-D, both in terms of F1 score and training time. RNN achieved a higher F1 score (0.71844 vs. 0.6625) and was more efficient, taking almost 40% less time to train. This indicates that the RNN model is both faster and more effective for EHR data.

b. RNN vs RNN-TS:

Table 2: RNN vs RNN-TS F1-score comparison.

MODEL	TEST F1 SCORE
RNN	0.71844
RNN-TS	0.71549

RNN-TS (with time series data) performed almost similarly to the standard RNN (F1 score of 0.71549 vs. 0.71844). This suggests that incorporating time series data as positional embeddings does not significantly enhance the model's performance and may even slightly degrade it.

c. RNN Class Balance vs. Without Class Balance:

Table 3: RNN vs RNN-TS F1-score comparison.

MODEL	TEST F1 SCORE
RNN (NO CLASS BALANCE)	0.71288
RNN (WITH CLASS BALANCE)	0.71844

Class balancing provided a marginal improvement in performance, boosting the F1

score from 0.71288 to 0.71844. This suggests that while class balancing helps, its impact is not substantial for the EHR dataset.

d. Training Time:

Table 4: GRU-D vs RNN-TS training time comparison in seconds.

MODEL	TRAINING TIME (SECONDS)
GRU-D	135,230
RNN-TS	82,968

RNN-TS is significantly faster to train compared to GRU-D, indicating that the simpler RNN architecture is more efficient in this context, potentially due to the reduced complexity in handling time-series data compared to GRU-D.

2. **CXR:**

Table 5: Comparison of F1 scores for different configurations of class balance and positional embedding.

CONFIGURATION	TEST F1 SCORE
NO CLASS BALANCE AND POSITIONAL EMBEDDING	0.56609
WITH CLASS BALANCE AND POSITIONAL EMBEDDING	0.66634
WITH CLASS BALANCE BUT NO POSITIONAL EMBEDDING	0.62633

Class balancing improved the model's performance significantly, increasing the F1 score from 0.56609 to 0.66634. Additionally, adding positional embeddings further enhanced the performance (F1 score: 0.66634 vs. 0.62633), suggesting that positional embeddings in CXR data play a role in improving the model's capacity to capture temporal information.

3. Notes:

Table 6: Comparison of F1 scores for frozen and unfrozen configurations with/without class balance and positional embedding.

CONFIGURATION	TEST F1 SCORE
NO CLASS BALANCE (FROZEN)	0.12199
FROZEN (WITH CLASS BALANCE)	0.5004
FROZEN (NO POSITIONAL EMBEDDING)	0.24544
UNFROZEN (NO CLASS BALANCE)	0.66085
UNFROZEN (WITH CLASS BALANCE)	0.63192

Unfreezing the notes encoder and allowing the model to fine-tune the notes embeddings significantly improved performance, with the F1 score rising to 0.66085 (without class balance). Class balancing marginally reduced performance in the unfrozen model (0.63192). Frozen models performed poorly, highlighting the importance of fine-tuning. Additionally, positional embeddings play a significant role, as removing them drastically dropped the F1 score from 0.5004 to 0.24544.

4. ECG:

Table 7: Test F1 and AUROC scores comparison for models with and without class balance.

CONFIGURATION	TEST F1 SCORE	AUROC
WITHOUT CLASS BALANCE	0.0	0.5197
WITH CLASS BALANCE	0.6236	0.52089

Without class balancing, the ECG model failed to predict more than one class, resulting in an F1 score of 0. However, class balancing greatly improved the performance, yielding an F1 score of 0.6236. This demonstrates that class imbalance severely impacts the ECG modality, and balancing is crucial for achieving meaningful predictions. Additionally, this highlights that AUROC is not a reliable metric in skewed predictions, as the AUROC for the unbalanced ECG model was still comparable despite the F1 score being 0.

5. Total comparison:

Table 8: Test F1 and AUROC scores for different modalities.

Modality	Test F1 Score	AUROC Score
Notes	0.66085	0.62601
ECG	0.6236	0.52089
RNN-TS (EHR)	0.71549	0.77536
CXR	0.66634	0.52081

Among the single modalities, the EHR data using RNN-TS performed the best, with an F1 score of 0.71549 and AUROC of 0.77536. CXR also performed well, with an F1 score of 0.66634. Notes and ECG followed with slightly lower scores. The AUROC scores indicate that while F1 remains the primary measure of interest, AUROC is not as reliable in imbalanced datasets like ECG and CXR.

6. Multimodality:

Table 9: Performance comparison of F1 scores for different fusion methods.

Fusion Method	F1 Score
Sum Fusion of CXR + EHR	0.69789
Transformer Fusion of CXR + EHR	0.70182
Sum Fusion of CXR + EHR + Notes	0.45058
Transformer Fusion of CXR + EHR + Notes	0.66817
Sum Fusion of CXR + ECG + EHR + Notes	0.51967
Transformer Fusion of CXR + ECG + EHR + Notes	0.61993
Sum Fusion of CXR + EHR + Notes	0.45058
Transformer Fusion of CXR + EHR + Notes	0.66817

The transformer-based fusion consistently outperforms the sum fusion method across all modality combinations. For instance, **Transformer CXR + EHR** (F1 score: 0.70182) slightly outperforms **Sum CXR + EHR** (F1 score: 0.69789). Similarly, **Transformer CXR + ECG + EHR + Notes** (F1 score: 0.61993) shows significant improvement over the sum fusion of the same modalities (F1 score: 0.51967). This indicates that transformer-based fusion is more effective in learning from the complex

interactions between modalities.

Table 10: Comparison of F1 scores for multi-modality Transformer fusion results and single modality baselines.

Fusion/Modality	F1 Score
Multi Modality (Transformer Fusion)	
Transformer CXR + ECG + EHR + Notes	0.61993
Transformer CXR + Notes + ECG	0.55538
Transformer CXR + EHR + ECG	0.45439
Transformer CXR + EHR + Notes	0.66817
Transformer CXR + Notes	0.64509
Transformer Notes + ECG	0.58547
Single Modality Baseline	
EHR	0.71549
CXR	0.66634
Notes	0.66085
ECG	0.62360

Although multimodal fusion using transformers improves performance in some cases, adding more modalities does not always result in better outcomes. For example, **Transformer CXR + ECG + EHR + Notes** (F1 score: 0.61993) performed worse than EHR alone (F1 score: 0.71549). This suggests that poorly aligned modality combinations can degrade performance. The best-performing multimodal models should ideally surpass the highest-performing single-modality model. However, this is not consistently observed, indicating that proper modality alignment and fusion are critical for achieving optimal performance.

Transformer-based fusion provides better results compared to sum fusion in multimodal learning, especially when integrating complex and diverse modalities such as CXR, ECG, EHR, and Notes. However, simply combining modalities does not guarantee improved performance; well-aligned models must be carefully designed to achieve results that exceed the best-performing individual modalities. Ideally, the

performance of a multimodal model should be greater than the maximum F1 score of the individual modalities (e.g., EHR or CXR alone).

4.3. DDP Optical Flow Estimation

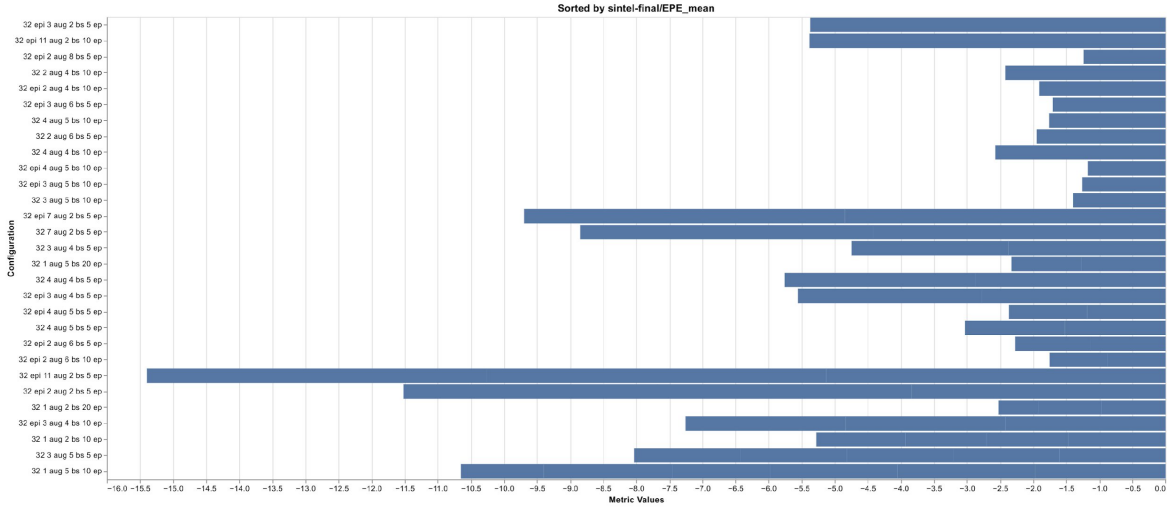

Figure 9: Percentage of absolute EPE score changes

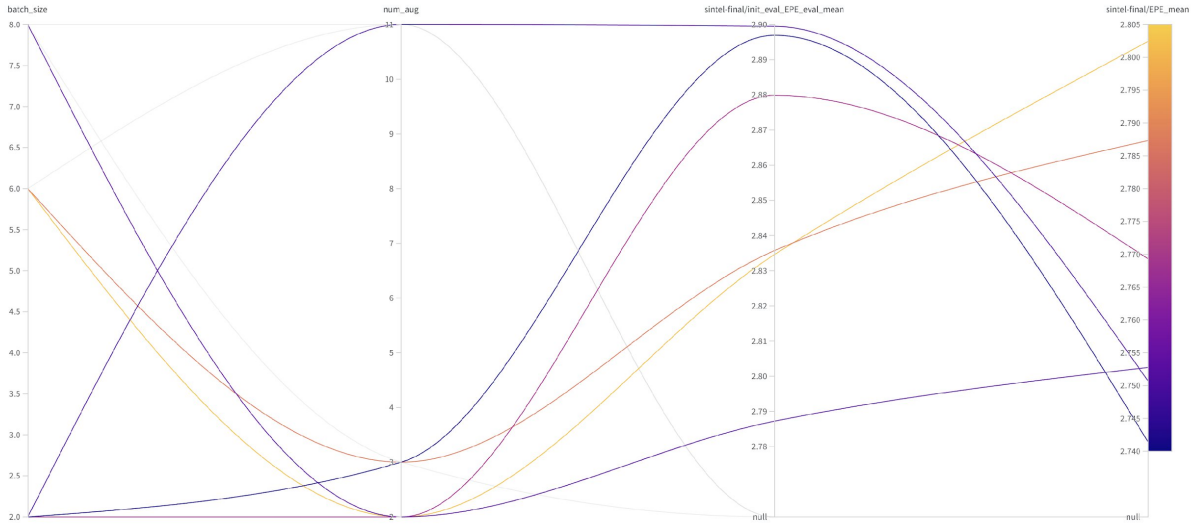

Figure 10: EPE scores for different configurations

Table 11: EPE scores for different configurations

BATCH SIZE	NUMBER OF AUGMENTATIONS	EPE VALUE
8	2	2.75277
2	11	2.75071
6	3	2.78732
2	3	2.74143
6	2	2.80244
2	2	2.76932

Figure 9 illustrates percentage of absolute EPE score decrease for different combinations of number of augmentation, number of epoch and batch size while the Figure 10 illustrates different combinations of number of augmentation, number of epoch and batch size. and its corresponding EPE scores. Table 11 illustrates the EPE scores for different batch size and number of augmentations using a system equipped with six GPUs. Across these different configurations, a reduction in EPE is observed, although the results do not exhibit a consistent trend. This suggests that while the applied augmentations and optimizations do contribute to minimizing EPE, further tuning of the batch size, number of GPUs, and augmentation strategies is required to achieve more definitive improvements.

For example, the configuration with a batch size of 2 and 11 augmentations resulted in the lowest EPE value of 2.75071. This configuration also illustrates the overall batch size calculation in distributed training, where the net batch size is determined by the following formula:

$$Net\ Batch\ Size = Batch\ Size \times (Num\ of\ Augmentations + 1) \times Number\ of\ GPUs$$

For example, the configuration of batch size 2 and 11 augmentations, with six GPUs, the net batch size would be:

$$2 \times (11 + 1) \times 6 = 1442$$

CHAPTER 5

5.1. Multi-Modal Learning Framework:

5.1.1. Conclusion

In conclusion, the analysis of multimodal datasets in this study underscores several key insights. The transformer-based fusion method proves superior to sum fusion in most cases, offering better performance in learning complex interactions between modalities like CXR, ECG, EHR, and Notes. However, combining modalities does not inherently improve performance. In some instances, poorly aligned combinations result in degraded outcomes compared to individual modalities, such as the EHR, which consistently performs well on its own.

The findings highlight the challenges posed by class imbalance, especially in modalities like ECG, where oversampling is essential to prevent skewed predictions. Additionally, the use of positional embeddings significantly enhances performance in CXR and Notes modalities, further emphasizing the importance of integrating temporal information in multimodal datasets.

Although multimodal learning holds promise, careful consideration must be given to modality alignment, dataset balance, and fusion techniques. Future research should aim to refine these aspects to ensure that multimodal models consistently surpass the performance of the best single-modality models.

5.1.2. Future Scope

1. **Bias Reduction (Downstream task only)**: Using placeholder embeddings of a missing modality as an **anchor** to alleviate model bias.
    a. Ideally, an unbiased model, when given no information, shouldn't be able to decide a class (i.e., 50-50 probability).
    b. Thus, we can try to use this as a data sample once in a while during training and study its effect on model performance.
2. Using one of the modality's encoders as a way to perform **positive pair mining** for inter-modal self-supervision and intra-modal self-supervision of other modalities. After a certain level of intra-modal training, we will be

introducing the inter-modal self-supervised learning phase. Usually, every sample aside from itself is considered a negative pair in both inter and intra-modal learning. This will, at some point, limit the model learning since other samples being -ve pairs are partially or sometimes completely wrong.

   a. Instead, we use a good-performing modality's encoder as a judge to perform the +ve pair mining by applying cosine similarity to that particular modality's representation. Since the threshold is unknown following can be done:
   b. After an epoch/regular period of time encode the entire dataset of the judging modality. Take the cosine similarity amongst every other pair and average them to get the **average threshold**.
   c. On the assumption that modality isn't reliable, a decay parameter can be set to slowly increase the max allowed positive pair generated by the modal, and **top k** can be chosen.
   d. Even after this, to reduce the detrimental effects of wrong positive pairs, we use the cosine similarity value found amongst the judging modality embeddings as a confidence score to weigh those terms in inter and intra-model loss.

3. Using **GradCAM** mask as a way to **weigh** the important intra-modal feature to boost inter-model representation learning:
   a. The whole idea of using inter-model alignment and intra-model alignment is based on the assumption that learned representations of their respective modalities are complemented by learned representations between modalities and vice-versa.
   b. But during cross-model alignment, if the other modality focuses on the unnecessary feature of the other modality, it might delay improvement/worsen the performance.
   c. Instead, we could apply GradCAM through the intra-modal loss to get the mask/weight representing the key impactful features during inter-modal loss.

Embeddings can also be viewed as PCA in 2D (meant for future scope to visualize alignment), the figures present PCA 2D visualizations of embedding outputs from a transformer-based model that integrates multiple medical modalities, including Chest X-rays

(CXR), ECG, EHR (Electronic Health Records), and clinical notes. Figure 11 shows the overall embedding output when fusing all four modalities together. Figures 12 to 15 individually depict the embeddings for each modality—Notes, Chest X-rays, EHR, and ECG, respectively—demonstrating the contribution of each data type to the model's fusion process. Lastly, Figure 16 provides PCA visualizations of projections from different training runs, highlighting their utility for phase-wise training analysis.

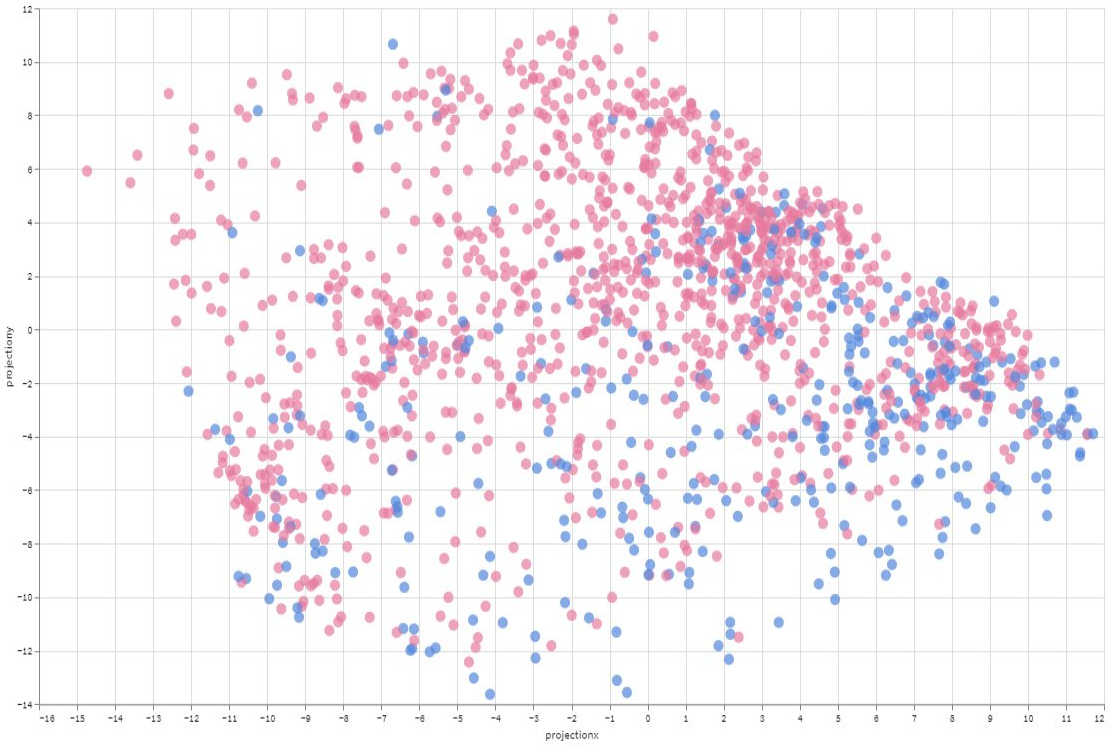

Figure 11: PCA 2D visualization of overall embedding.

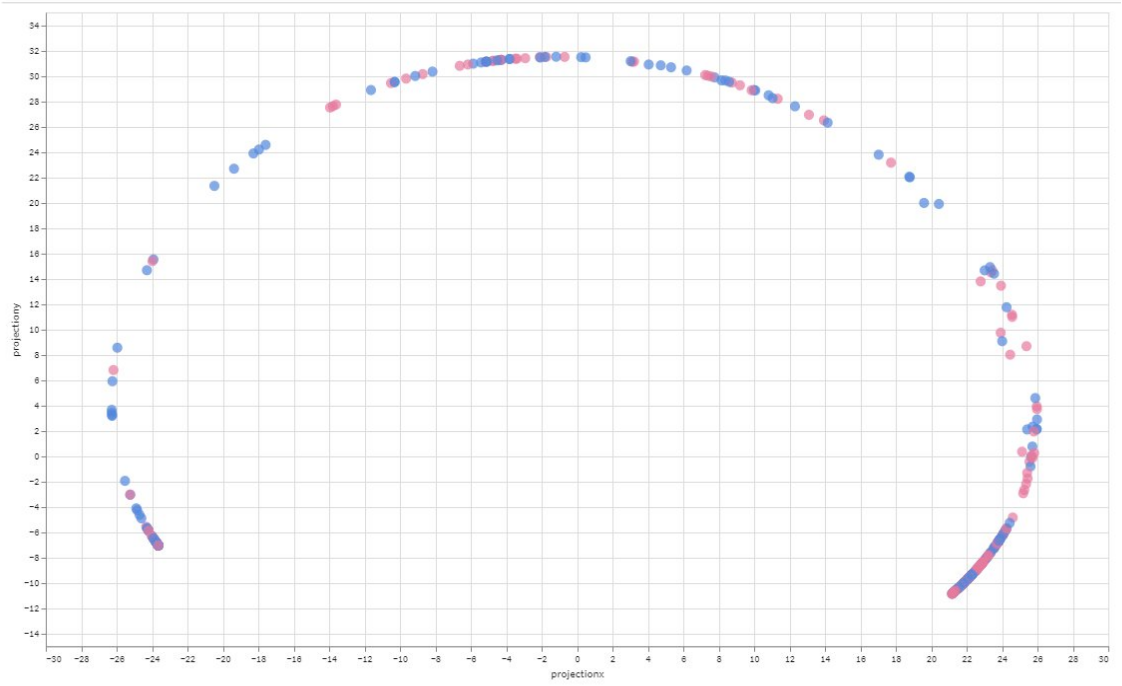

Figure 12: PCA 2D visualization of Notes embedding.

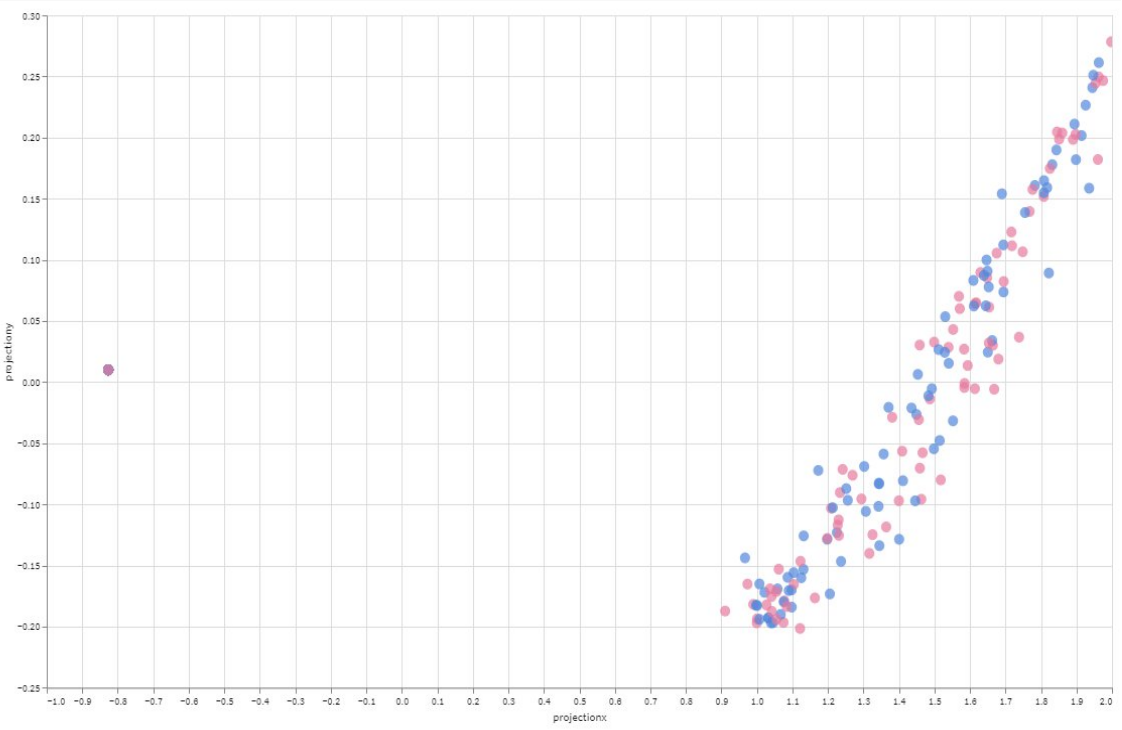

Figure 13: PCA 2D visualization of Chest X-Ray embedding.

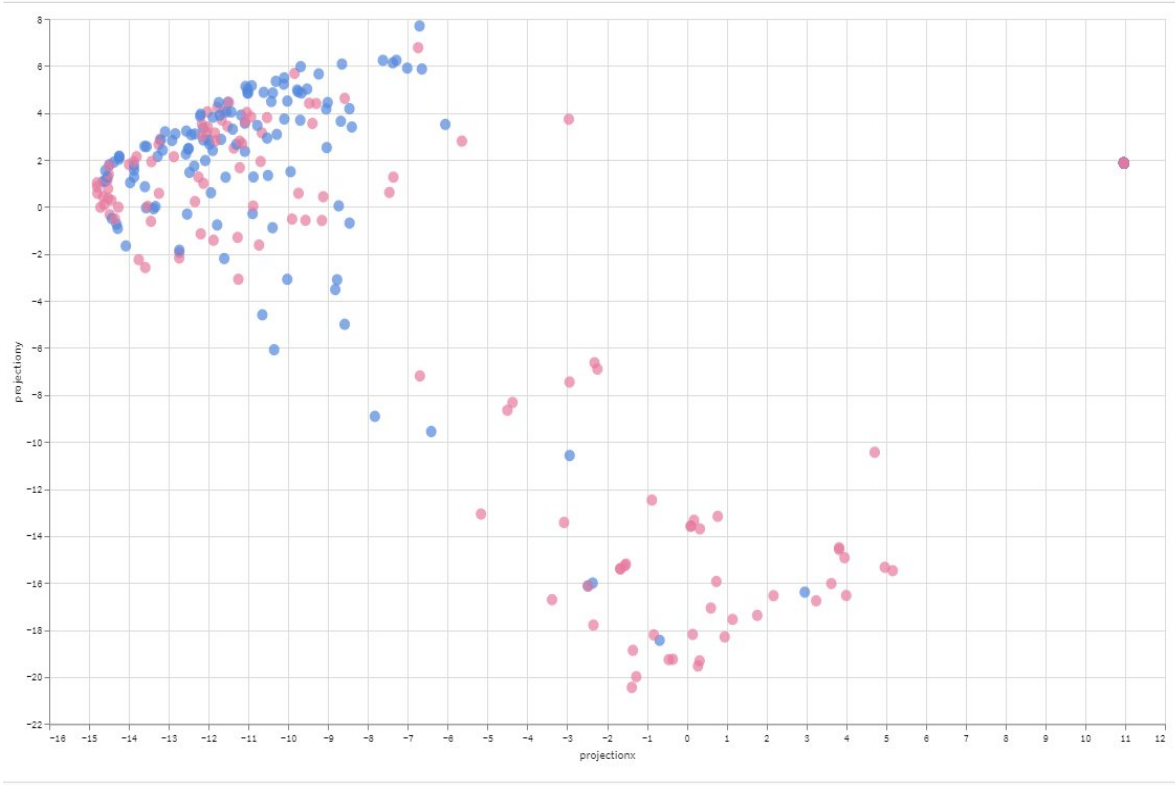

Figure 14: PCA 2D visualization of EHR embedding.

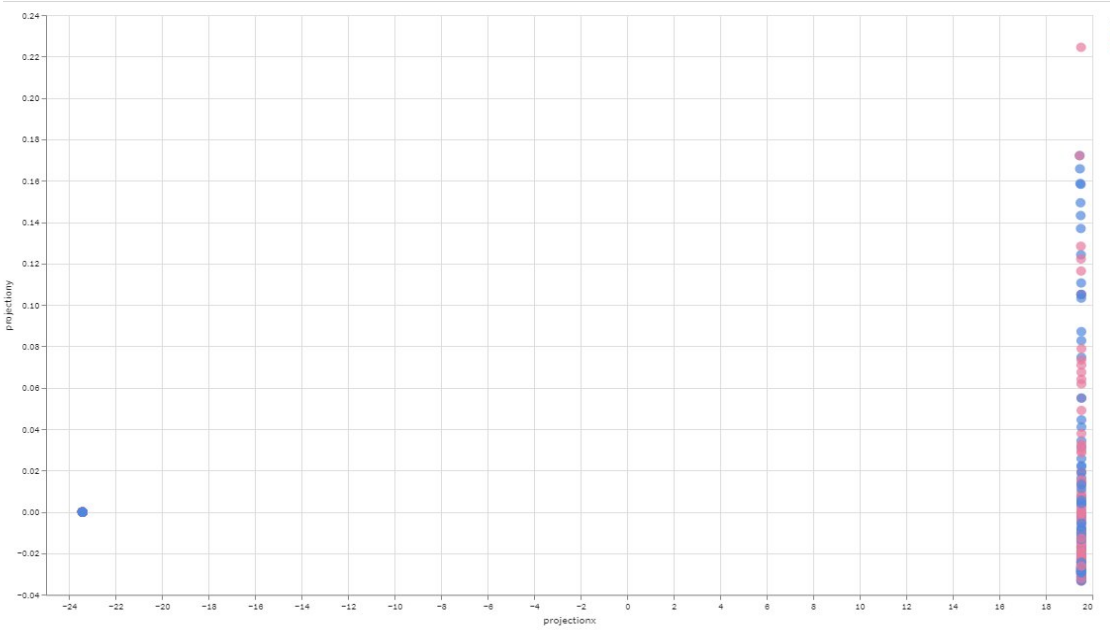

Figure 15: PCA 2D visualization of ECG embedding.

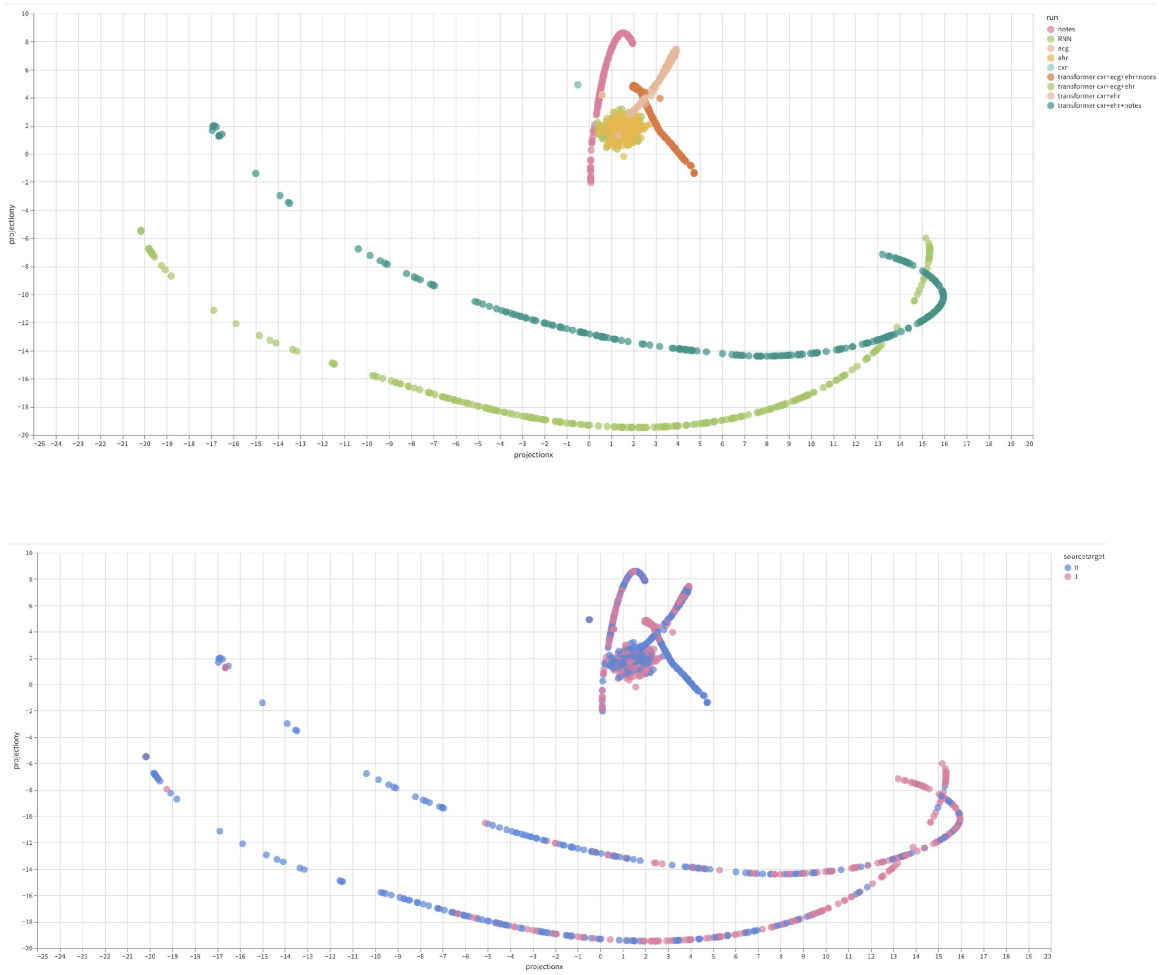

Figure 16: PCA 2D visualization of different runs.

5.2. DDP Optical Flow Estimation

5.2.2. Conclusion:

In conclusion, this research demonstrates the successful application of Distributed Data Parallel (DDP) processing using Lightning-Fabric for optimizing an optical flow model across multi-GPU and multi-node configurations. Key techniques such as CPU offloading, mixed-precision training, and activation checkpointing were employed to enhance memory efficiency and maximize batch size. The test-time adaptation algorithm introduced here, which iteratively refines optical flow maps through augmentations, contributes to reducing End-Point Error (EPE) between the predicted optical flow and the ground truth.

While empirical results show a general reduction in EPE with varying batch sizes and augmentations, the findings indicate that there is no clear, consistent trend across all configurations. This suggests the need for further experimentation, particularly with larger batch sizes, additional GPUs, and alternative augmentation strategies, to fully

optimize model performance.

5.2.2. Future Work:

While the current experiments demonstrate a reduction in EPE across different combinations of batch sizes and augmentations, there is no clear trend that applies universally to all configurations. This variability suggests that further experimentation with larger batch sizes and additional GPUs may be necessary to validate the observed outcomes and refine the adaptation algorithm.

Moving forward, we plan to explore new augmentation strategies that can potentially yield more consistent and significant improvements. By incorporating more diverse forms of image augmentation and optimizing model training further, we aim to enhance the robustness of the optical flow estimation process.

REFERENCES

[1] L.L. Weed, Medical records that guide and teach (concluded), Yearb. Med. Inform. 212 (1968) 1.

[2] J. Henry, Y. Pylypchuk, T. Searcy, V. Patel, et al., Adoption of electronic health record systems among US non-federal acute care hospitals: 2008–2015, ONC data brief. 35, 2008–2015 2016.

[3] ONC, National Trends in Hospital and Physician Adoption of Electronic Health Records |, HealthIT. Gov., 2023.

[4] T. Sarwar, S. Seifollahi, J. Chan, X. Zhang, V. Aksakalli, I. Hudson, K. Verspoor, L. Cavedon, The secondary use of electronic health records for data mining: data characteristics and challenges, ACM Comput. Surv. 55 (2022) 1–40.

[5] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L.A. Celi, R. Mark, MIMIC-IV (version 1.0), PhysioNet (2021), https://doi.org/10.13026/s6n6-xd98.

[6] A. Johnson, M. Lungren, Y. Peng, Z. Lu, R. Mark, S. Berkowitz, S. Horng, MIMIC-CXR-JPG - chest radiographs with structured labels (version 2.0.0), PhysioNet (2019), https://doi.org/10.13026/8360-t248.

Heliyon 10 (2024) e26772 15 Y. Wang, C. Yin and P. Zhang

[7] A.E. Johnson, T.J. Pollard, S. Berkowitz, N.R. Greenbaum, M.P. Lungren, C.Y. Deng, R.G. Mark, S. Horng, MIMIC-CXR: A large publicly available database of labeled chest radiographs, arXiv preprint arXiv:1901.07042, 2019 Jan 21.

[8] A. Johnson, T. Pollard, S. Horng, L.A. Celi, R. Mark, MIMIC-IV-Note: Deidentified free-text clinical notes (version 2.2), PhysioNet (2023), https://doi.org/10.13026/1n74-ne17.

[10] E. Choi, M.T. Bahadori, J. Sun, J. Kulas, A. Schuetz, W. Stewart, RETAIN: an interpretable predictive model for healthcare using reverse time attention mechanism, Adv. Neural Inf. Process. Syst. 29 (2016), Curran Associates, Inc. Available at https://proceedings.neurips.cc/paper/2016/hash/231141b34c82aa95e48810a9d1b33a79-Abstract.html.

[11] E. Choi, M.T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, J. Sun, Multi-layer representation learning for medical concepts, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD'16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 1495–1504.

[12] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, D. Zhi, Med-BERT: pretrained

contextualized embeddings on large-scale structured electronic health records for disease prediction, npj Digit. Med. 4 (2021) 1–13, https://doi.org/10.1038/s41746-021-00455-y.
[13] D. Zhang, C. Yin, J. Zeng, X. Yuan, P. Zhang, Combining structured and unstructured data for predictive models: a deep learning approach, BMC Med. Inform. Decis. Mak. 20 (2020) 280, https://doi.org/10.1186/s12911-020-01297-6.
[14] A. Ashfaq, A. Sant'Anna, M. Lingman, S. Nowaczyk, Readmission prediction using deep learning on electronic health records, J. Biomed. Inform. 97 (2019) 103256, https://doi.org/10.1016/j.jbi.2019.103256.
[15] M. Capan, P. Wu, M. Campbell, S. Mascioli, E.V. Jackson, Using electronic health records and nursing assessment to redesign clinical early recognition systems, Health Syst. 6 (2017) 112–121, https://doi.org/10.1057/hs.2015.19.
[16] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, J. Gao, Dipole: diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax NS Canada, ACM, 2017, pp. 1903–1911.
[17] Y. Li, S. Rao, J.R.A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, G. Salimi-Khorshidi, BEHRT: transformer for electronic health records, Sci. Rep. 10 (2020) 7155, https://doi.org/10.1038/s41598-020-62922-y.
[18] J. Shang, T. Ma, C. Xiao, J. Sun, Pre-training of graph augmented transformers for medication recommendation, in: S. Kraus (Ed.), Proceedings of the TwentyEighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, 2019, pp. 5953–5959, ijcai.org.
[19] A. Kline, H. Wang, Y. Li, S. Dennis, M. Hutch, Z. Xu, F. Wang, F. Cheng, Y. Luo, Multimodal machine learning in precision health: a scoping review, npj Digit. Med. 5 (2022) 171, https://doi.org/10.1038/s41746-022-00712-8.
[20] M. Golovanevsky, C. Eickhoff, R. Singh, Multimodal attention-based deep learning for Alzheimer's disease diagnosis, J. Am. Med. Inform. Assoc. 29 (2022) 2014–2022, https://doi.org/10.1093/jamia/ocac168.
[21] S.-C. Huang, A. Pareek, R. Zamanian, I. Banerjee, M.P. Lungren, Multimodal fusion with deep neural networks for leveraging CT imaging and electronic health record: a case-study in pulmonary embolism detection, Sci. Rep. 10 (2020) 22147, https://doi.org/10.1038/s41598-020-78888-w.
[22] Z. Yao, X. Hu, X. Liu, W. Xie, Y. Dong, H. Qiu, Z. Chen, Y. Shi, X. Xu, M.

Huang, J. Zhuang, A machine learning-based pulmonary venous obstruction prediction model using clinical data and CT image, Int. J. Comput. Assisted Radiol. Surg. 16 (2021) 609–617, https://doi.org/10.1007/s11548-021-02335-y.
[23] R. Yan, F. Zhang, X. Rao, Z. Lv, J. Li, L. Zhang, S. Liang, Y. Li, F. Ren, C. Zheng, J. Liang, Richer fusion network for breast cancer classification based on multimodal data, BMC Med. Inform. Decis. Mak. 21 (2021) 134, https://doi.org/10.1186/s12911-020-01340-6.
[24] D. Nie, J. Lu, H. Zhang, E. Adeli, J. Wang, Z. Yu, L. Liu, Q. Wang, J. Wu, D. Shen, Multi-channel 3D deep feature learning for survival time prediction of brain tumor patients using multi-modal neuroimages, Sci. Rep. 9 (2019) 1103, https://doi.org/10.1038/s41598-018-37387-9, Number: 1 Publisher: Nature Publishing Group.
[25] L.R. Soenksen, Y. Ma, C. Zeng, L. Boussioux, K. Villalobos Carballo, L. Na, H.M. Wiberg, M.L. Li, I. Fuentes, D. Bertsimas, Integrated multimodal artificial intelligence framework for healthcare applications, npj Digit. Med. 5 (2022) 1–10, Publisher: Nature Publishing Group.
[26] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017), Curran Associates, Inc. Available at https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.
[27] S.M. Lundberg, G. Erion, H. Chen, A. DeGrave, J.M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee, From local explanations to global understanding with explainable AI for trees, Nat. Mach. Intell. 2 (2020) 56–67, https://doi.org/10.1038/s42256-019-0138-9.
[28] D. Fryer, I. Strümke, H. Nguyen, Shapley values for feature selection: the good, the bad, and the axioms, IEEE Access 9 (2021) 144352–144360, https://doi.org/10.1109/ACCESS.2021.3119110, Conference Name: IEEE Access.
[29] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proc. 7th Int. Joint Conf. Artif. Intell. (IJCAI), Apr. 1981, pp. 674–679.
[30] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in Image Analysis, J. Bigun and T. Gustavsson, Eds. Berlin, Germany: Springer, 2003, pp. 363–370.
[31] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical flow

estimation based on a theory for warping," in Proc. Eur. Conf. Comput. Vis. Berlin, Germany: Springer, 2004, pp. 25–36.
[32] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in Proc. 12th Eur. Conf. Comput. Vis. (ECCV), Oct. 2012, pp. 611–625.
[33] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," Int. J. Comput. Vis. (IJCV), vol. 106, no. 2, pp. 115–137, Jan. 2014.
[34] A. Dosovitskiy et al., "FlowNet: Learning optical flow with convolutional networks," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 2758–2766.
[35] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3431–3440.
[36] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3061–3070.
[37] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1164–1172.
[38] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1933–1941.
[39] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2414–2423.
[40] Y. Kataoka, T. Matsubara, and K. Uehara, "Image generation using generative adversarial networks and attention mechanism," in Proc. IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci. (ICIS), Jun. 2016, pp. 1–6.
[41] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1647–1655.
[42] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 105–114.
[43] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual

networks for single image super-resolution," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jul. 2017, pp. 1132–1140.
[44] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2720–2729, doi: 10.1109/CVPR.2017.291.
[45] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8183–8192.
[46] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1701–1710.
[47] A. Shocher, N. Cohen, and M. Irani, "Zero-shot super-resolution using deep internal learning," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3118–3126.
[48] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8934–8943.
[49] P. Liu, M. Lyu, I. King, and J. Xu, "SelFlow: Self-supervised learning of optical flow," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4566–4575.
[50] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," Int. J. Comput. Vis. (IJCV), vol. 127, no. 8, pp. 1106–1125, Aug. 2019.
[51] G. Yang and D. Ramanan, "Volumetric correspondence networks for optical flow," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 32, 2019, pp. 1–12.
[52] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Testtime training with self-supervision for generalization under distribution shifts," in Proc. Int. Conf. Mach. Learn. (ICML), Nov. 2020, pp. 9229–9248.
[53] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in Proc. 30th Int. Joint Conf. Artif. Intell., Aug. 2021, pp. 402–419.
[54] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, and Y. Xu, "MaskFlownet: Asymmetric feature matching with learnable occlusion mask," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6277–6286.
[55] Z. Chi, Y. Wang, Y. Yu, and J. Tang, "Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning," in Proc. IEEE/CVF Conf. Comput. Vis.

Pattern Recognit. (CVPR), Jun. 2021, pp. 9133–9142.
[56] A. Stone, D. Maurer, A. Ayvaci, A. Angelova, and R. Jonschkowski, “SMURF: Self-teaching multi-frame unsupervised RAFT with full-image warping,” in Proc. IEEE/CVF Conf. Comput. Vis.Pattern Recognit. (CVPR), Jun. 2021, pp. 3886–3895, doi: 10.1109/CVPR46437.2021.00388.
[57] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, “Tent:Fully test-time adaptation by entropy minimization,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2021, pp. 1–15.
[58] L. Kong and J. Yang, “MDFlow: Unsupervised optical flow learning by reliable mutual knowledge distillation,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 2, pp. 677–688, Feb. 2023, doi: 10.1109/TCSVT.2022.3205375.