Final Year Project

Investigating the Impact of Task-Specific Pre-Trained Encoders on Late-Fusion Multimodal Model Performance

Student Arsenii Troitskii

Student ID: 20204701

A thesis submitted in part fulfilment of the degree of

BSc. (Hons.) in Computer Science

Supervisor: Professor Fatemeh Golpayegani

UCD School of Computer Science

University College Dublin

April 27, 2025

Table of Contents

Abstract

The main goal of this project is to investigate the impact of task-specific pre-trained encoders on late-fusion multimodal model performance. The objective is to figure out if there is a significant difference in model performance between late fusion multimodal models using task-specific pre-trained encoders and those trained from scratch. During this research, modern deep learning methods will be used to see their effect in real-world multimodal data. The results will improve the use of these models in the practical industry.

Link to project repository: Pre-Trained Encoders for Multimodal Models

Chapter 1: Project Specification

Multimodal deep learning systems are essential for processes that include different types of data such as text, images, and audio. These systems utilize multiple modalities to achieve higher performance in tasks such as classification, prediction, and detection. However, the primary challenges remain in optimizing such systems, especially in designing effective late-fusion architectures and pre-trained task-specific encoders.

This project aims to investigate the impact of task-specific pre-trained encoders on late-fusion multimodal model performance. Through a series of in-depth experiments and analyses, the project will provide insights into the optimal design and configurations for such systems. The results are expected to contribute to both the theoretical understanding and practical applications of multimodal learning.

1.1 Primary Objectives

The following objectives outline the core requirements for the project:

1. Conduct a comprehensive literature review on multimodal learning, with an emphasis on late-fusion architectures and modality-specific pre-trained encoders.
2. Implement two distinct late-fusion multimodal models using a deep learning framework like PyTorch.
3. Select two diverse datasets from different domains for experimentation.
4. For each model-dataset combination:
    - Train the model using the conventional approach (from scratch).
    - Train the model using task-specific pre-trained modality-specific encoders.
5. Compare the convergence rates of models trained from scratch with those using pre-trained encoders.
6. Analyze the performance of these approaches across models and datasets, using appropriate evaluation metrics for each task.
7. Perform statistical analysis to determine the significance of observed differences in performance between approaches.
8. Provide a thorough analysis of results, discussing the implications of using task-specific pre-trained encoders in various scenarios.

1.2 Advanced Objectives

These objectives represent more challenging extensions to the project:

1. Expand the comparison to additional datasets to assess the generalizability of findings across different domains and tasks.
2. Implement and evaluate a wider range of late-fusion multimodal architectures to explore how the impact of pre-training varies across model types.
3. Conduct an in-depth analysis of how task-specific pre-training affects latent representations in the models, potentially using visualization techniques or probing tasks.
4. Investigate the sensitivity of the models to various hyperparameters (e.g., learning rate, batch size) when using pre-trained encoders compared to training from scratch.
5. Explore the transferability of pre-trained encoders across related tasks within the same domain.
6. Develop guidelines or a decision framework to inform the use of task-specific pre-trained encoders in late-fusion multimodal setups based on project findings.

Chapter 2: **Introduction**

Rapid advancement of deep learning has enabled the development of multimodal systems that integrate diverse data modalities, such as text, images, and audio, to perform complex tasks [1]. These systems are widely used in many fields, such as healthcare, autonomous vehicles, and natural language processing, where the ability to process and combine multiple data types is important [2].

This project aims to investigate the impact of task-specific pre-trained encoders on the performance of late-fusion multimodal models. By using pre-trained encoders that are fine-tuned for specific tasks, this research seeks to improve model performance while analyzing how different configurations of encoders affect metrics such as accuracy and stability. The findings will contribute to a better understanding of how multimodal systems can be optimized for real-world applications [3].

Understanding the impact of task-specific pre-trained encoders can significantly boost the efficiency and effectiveness of multimodal models. This research has practical applications in various fields, including:

- **Social Media Analysis**: Integration of text and image data for user behavior prediction and content recommendation.
- **Medical Diagnostics**: Combining patient records with medical imaging to improve diagnostic accuracy.
- **Autonomous Vehicle Systems**: Multi-sensor data fusion for decision-making and navigation.
- **Sentiment Analysis**: Integration of text, audio, and video data to understand emotional and behavioral patterns.

This project is significant because its aim is to connect theoretical research with real-world applications in multimodal learning. Focusing on key challenges, such as modality-specific contributions and system scalability, this research will provide valuable insights into designing efficient and stable multimodal models.

In summary, this project will explore the following:

- The role of task-specific pre-trained encoders in late-fusion multimodal models.
- The trade-offs between general pre-training and task-specific fine-tuning.
- The scalability and generalizability of multimodal architectures across various domains.

Chapter 3: Related Work and Ideas

Multimodal learning is a growing field that combines data from different sources, such as text, images, and audio, to handle complex tasks. Using the advantages of each type of data, these approaches improve the performance of machine learning models. In this chapter, several important works in this area are discussed, along with their results, existing gaps, and the ways in which this project complements them.

3.1 Related Work: ESPER

The ESPER framework, introduced by Yu et al. (2023) [4], represents a breakthrough in multimodal learning. The primary goal of ESPER is to improve the ability of pre-trained language models (PLMs) to process multimodal inputs such as images and audio without requiring paired training data. ESPER bridges the gap between language models and multimodal tasks by combining insights from reinforcement learning (RL) and multimodal prompt tuning. This innovative approach sets ESPER apart from traditional methods that rely heavily on maximum likelihood estimation and paired datasets.

What do they do

ESPER, or Extending Sensory Perception with Reinforcement Learning, uses a pre-trained language model, such as GPT-2, and boosts it with a lightweight encoder that maps visual features from CLIP (Contrastive Language-Image Pretraining) into the PLM's embedding space. The standout feature of ESPER is its use of RL, specifically Proximal Policy Optimization (PPO), to align the generated text with multimodal inputs. Instead of relying on costly paired data (e.g., images paired with captions), ESPER optimizes for alignment using a cosine similarity reward derived from CLIP. This allows the model to maintain the broad reasoning capabilities of PLMs while adapting to visual contexts.

The architecture of ESPER includes three main components:

- **CLIP Encoders:** These provide visual feature extraction and a reward signal for reinforcement learning.
- **A Lightweight Encoder:** This trainable module integrates visual features into the text embedding space.
- **Pre-trained Language Models:** These remain frozen during training, ensuring the preservation of existing linguistic capabilities.

ESPER has been tested on a variety of tasks, such as image captioning, commonsense reasoning, and dialogue generation, demonstrating its versatility and efficiency. Notably, ESPER introduces the ESP dataset, which contains multimodal prompts spanning various domains (e.g., news, stories, blogs), further expanding its potential for real-world applications.

Their Results

The performance of ESPER is particularly compelling in zero-shot image captioning tasks, where it consistently outperforms prior unsupervised methods. For example, ESPER achieves a CIDEr score of **78.2** on the COCO test split, significantly surpassing competing models such as MAGIC (**49.3**) and CLIPRe (**13.6**). In addition to its superior text generation quality, ESPER exhibits remarkable inference speed, completing captioning tasks in just **0.65 seconds**, compared to the **65 seconds** required by ZeroCap.

Model	Domain	B@4	M	C	Time
Pseudo-Align [34]	✓	5.2	15.5	29.4	-
RSA [21]	✓	7.6	13.5	31.8	-
Unpaired [34]	✓	19.3	20.1	63.6	-
ZeroCap [67]		2.6	11.5	14.6	65s
ZeroCap-CaptionLM	✓	7.0	15.4	34.5	65s
CLIPRe [64]	✓	4.9	11.4	13.6	-
MAGIC [64]	✓	12.9	17.4	49.3	3s
ESPER-GPT		6.3	13.3	29.1	0.65s
ESPER-CaptionLM	✓	**21.9**	**21.9**	**78.2**	0.65s

Figure 3.1: Table from ESPER's paper illustrating results from unpaired captioning experiments on the COCO test split.

In commonsense reasoning tasks, ESPER also shines. For instance, in experiments with the Visual Commonsense Graph (VCG) dataset, ESPER achieves a CIDEr score of **16.4**, performing on par with or better than supervised baselines trained on paired data. These results highlight ESPER's ability to handle complex reasoning tasks without direct supervision.

ESPER has made significant advancements in multimodal learning, but there are still areas where it could be improved. One limitation is its heavy reliance on pre-trained models like CLIP and GPT-2, which are built on specific datasets. This dependency might restrict its flexibility when applied to domains that fall outside the scope of these training corpora. Additionally, while reinforcement learning brings many benefits, it can also introduce complexities, such as potential issues with training stability and reward optimization. Another area for improvement is the relatively narrow scope of ESPER's evaluations, which primarily focus on datasets like COCO and VCG, leaving room for exploration with a wider variety of datasets and tasks.

This work aims to address some of these gaps by shifting the focus toward late-fusion architectures and task-specific pre-trained encoders. Rather than relying only on reinforcement learning, this work explores how task-specific pre-training might enhance a model's ability to generalise across various datasets and tasks, moving beyond the benchmarks emphasised in ESPER. A closer look is also taken at the trade-offs between model improvements and computational costs, especially within the context of late-fusion models. Lastly, by focusing on late-fusion approaches, the goal is to enhance understanding of areas like model interpretability and robustness, providing a broader understanding of how multimodal systems can be designed and evaluated. These research objects are intended to complement ESPER's contributions while expanding the possibilities of multimodal learning frameworks.

In summary, while ESPER achieves impressive results in multimodal text generation, this work complements and extends its findings by addressing gaps related to task-specific pre-training, the ability to work across different domains, and late-fusion strategies. The inclusion of Table 3.1 from ESPER's paper further illustrates the significant achievements of their framework, setting a standard for subsequent evaluations.

RGB-D Object Recognition Using Multimodal Learning

RGB-D object recognition focuses on improving accuracy and robustness by integrating RGB (colour) and depth (D) data. Song et al. (2015) [5] presented an innovative approach that combines convolutional neural networks (CNNs) with fusion strategies to leverage the complementary features of these modalities. Their work explored how combining RGB and depth information could enhance object classification models for real-world applications.

The study compared three fusion strategies to integrate RGB and depth data. Early fusion combines raw RGB and depth data at the input stage, making it computationally efficient but sometimes insufficient for capturing modality-specific details. Intermediate fusion processes the modalities separately, combining their extracted features at a mid-network layer. This method maintains a balance between shared learning and keeping unique modality features. Finally, late fusion independently processes each modality and combines their outputs at the decision-making stage, leading to superior performance by maximising the strengths of both modalities.

Results from this study demonstrated the significant advantages of integrating depth data, especially in scenarios with poor lighting or visual obstructions. Late fusion proved to be the most effective strategy, consistently outperforming early and intermediate fusion methods by maintaining modality-specific information while harnessing their complementarity. Evaluations on benchmark datasets like NYU Depth Dataset and SUN RGB-D further validated their approach, achieving cutting-edge results and demonstrating strong generalisation across diverse environments.

Despite its successes, the study had some limitations. It was narrowly focused on RGB-D data, leaving other potential multimodal combinations unexplored, such as text or audio with visual data. The research also primarily addressed object recognition tasks, limiting insights into its applicability to other domains. Additionally, while the performance of different fusion strategies was compared, there was limited analysis of the underlying reasons for their effectiveness or how modality-specific features influenced the results.

This research builds upon this work by examining the impact of task-specific pre-trained encoders in late-fusion architectures. Unlike their approach, which relies on manually designed fusion strategies, it will be discovered how pre-trained encoders can generalise across multiple datasets and tasks, including those involving textual and audio data. By focusing on late-fusion models, this work aims to provide a deeper understanding of how modality-specific representations contribute to performance. Furthermore, this work emphasises interpretability, providing insights into the benefits of task-specific pre-training for improving model robustness and adaptability. While Song et al. (2015) [5] advanced RGB-D object recognition, that project seeks to broaden the application areas of multimodal learning by addressing these gaps and extending its potential applications.

This study not only complements the findings of Song et al. (2015) [5] but also extends the area of multimodal research, contributing to its applicability across diverse domains.

3.2 Other related works and Ideas

Wang et al. (2022) [6] introduced **i-Code**, a general-purpose multimodal learning framework. This system uses pre-trained models for different modalities, such as CLIP for images and text, Wav2Vec for audio, and BERT for text. One of i-Code's strengths is its modularity, which allows researchers to add new tasks or data without retraining the entire system. In areas such as image captioning, video question answering, and speech recognition, i-Code outperformed non-modular systems by an average of 12%. However, while i-Code demonstrates excellence in flexibility, it does not focus on measuring the impact of task-specific pre-trained encoders on metrics like accuracy or convergence speed. This gap is addressed in the present project by primarily focusing on the role of pre-trained encoders in late-fusion architectures.

Li et al. (2022) [7] presented **Uni-EDEN**, a universal encoder-decoder network for both perception (e.g., object recognition) and generation (e.g., image captioning) tasks. This model uses multi-granular pre-training, where proxy tasks, such as object classification and image-to-text matching, help the model learn at different abstraction levels. Uni-EDEN achieved an impressive 89.3% accuracy on Visual Question Answering (VQA) and a BLEU-4 score of 43.7 for image captioning. While this work highlights the versatility of universal architectures, it does not explore specialised pre-training for individual tasks. While hat study focuses on task-specific pre-training, aiming to improve accuracy and efficiency for particular tasks rather than generalising across all modalities.

These works represent significant progress in multimodal learning. Yu et al. (2023) [4] demonstrated the effectiveness of reinforcement learning for multimodal alignment. Song et al. (2015) [5] showed that late fusion could maximise the benefits of combining modalities. Wang et al. (2022) [6] highlighted the advantages of modular systems, and Li et al. (2022) [7] developed a universal architecture capable of handling diverse tasks.

However, the in-depth investigation into the impact of task-specific pre-trained encoders on late-fusion model performance remains lacking in these works. This is the central focus of that research. By exploring this gap, the aim is to uncover how these models can achieve higher accuracy, faster convergence, and greater adaptability. This contribution will not only expand the understanding of multimodal learning but also provide a new perspective on how to design more efficient and specialized systems for real-world applications.

Chapter 4: Data Considerations

4.1 AVMNIST Dataset

The primary dataset used during the first group of experiments is the AVMNIST dataset, a structured multimodal benchmark designed for classification tasks that involve audio and image data. This dataset is sourced from the *FedC-MAMs* framework and is also available through the *DataSets* repository hosted on GitHub [8]. It consists of grayscale images representing handwritten digits (in the style of MNIST) and their corresponding audio representations converted into spectrograms. This approach to multimodal representation follows established practices in the field [9]. Each input pair is aligned and labelled, facilitating multimodal classification tasks.

The dataset contains approximately 70,000 samples in total, split into three main subsets: a training set (approximately 60,000 samples), a validation set (used to tune hyperparameters and prevent overfitting), and a test set (approximately 10,000 samples). This standard split ensures proper evaluation of model performance and generalization capabilities, a methodology widely adopted in contemporary multimodal research [2].

AVMNIST was selected for its simplicity, balance, and suitability for controlled multimodal experiments. It allows the user to focus on understanding the effects of pretraining and fusion methods without the added complexity of large-scale or noisy data. As highlighted in recent surveys [10], such controlled environments are essential for isolating the specific contributions of architectural components in multimodal systems. Due to its small size and open-source nature, it is well-suited for reproducible experimentation and rapid development of multimodal architectures.

4.2 MMIMDB Dataset

The MMIMDB dataset was used for the second group of experiments. It is a multimodal, multilabel classification dataset consisting of movie descriptions (text) and corresponding movie posters (images). Each sample can belong to multiple genres simultaneously, making the task significantly more complex than AVMNIST. The dataset was obtained from the original authors [11] and is publicly available. During this project, the MMIMDB dataset will be used from the same GitHub repository [8] as AVMNIST due to its convenient and well-structured format.

MMIMDB contains 25,959 movie entries collected from the Internet Movie Database (IMDb), with each entry including:

- A movie poster image in RGB format
- A textual description including plot summary, cast, and production details
- Genre labels (multilabel) from a set of 23 possible genres

MMIMDB was chosen for its suitability in testing fusion methods and pre-trained encoders in a realistic multimodal scenario. Similar approaches have been explored in recent literature [9],

where the differences in modality structure (visual + textual), combined with the multilabel nature of the problem, make it ideal for testing how pooling methods and pretraining influence model generalisation. Additionally, as noted in comparable studies [10], the semantic richness of both modalities allows us to investigate how different encoders capture complementary information across modalities.

4.3 MOSI Dataset

The CMU Multimodal Opinion Sentiment Intensity (MOSI) dataset was used in the third group of experiments. It is a widely used dataset for multimodal sentiment analysis, consisting of video segments annotated with sentiment scores. Each segment includes synchronized input from three modalities: video (visual gestures and expressions), audio (prosodic features), and text (transcribed speech).

MOSI contains 2,199 opinion video clips from 93 distinct speakers discussing various topics in English. The key characteristics of this dataset include:

- Multimodal nature: Each sample contains synchronised video, audio, and text data
- Fine-grained sentiment annotations: The dataset is annotated on a continuous sentiment scale from -3 (strongly negative) to +3 (strongly positive)
- Speaker diversity: Includes 93 speakers with different accents, speaking styles, and demographic backgrounds
- Temporal alignment: All modalities are precisely aligned at the word level, following methodologies established in prior research [12]

For experimental purposes, the continuous sentiment scores were transformed into a classification problem with three discrete classes: negative (score < 0), neutral (score $= 0$), and positive (score > 0), an approach that has shown effectiveness in similar multimodal contexts [1].

MOSI was selected because it allows multimodal learning evaluation in a more complex setting, with three modalities and emotional labels that require the integration of fine-grained cues across modalities. As demonstrated in contemporary studies [9], this dataset also enables analysis of robustness to missing modalities, making it a valuable testbed for investigation with modality reduction and fusion strategies. The temporal nature of the data further challenges our models to capture cross-modal dynamics, a critical aspect of multimodal learning that has been highlighted in the literature [10].

4.4 Comparative Analysis of Datasets

This progression from AVMNIST to MMIMDB to MOSI represents an increasing level of complexity in multimodal learning challenges:

- **AVMNIST** provides a controlled environment with well-structured data and a straightforward classification task, making it ideal for initial experiments and baseline comparisons.

- **MMIMDB** introduces greater complexity through multilabel classification and more semantically rich modalities, allowing us to test the scalability of our approaches.
- **MOSI** presents the most challenging scenario with three modalities, temporal dynamics, and subtle sentiment cues that require sophisticated integration strategies.

This systematic progression enables a comprehensive evaluation of the impact of pre-trained encoders across increasingly complex multimodal learning scenarios, providing insights that span from controlled academic benchmarks to real-world applications. Table 4.1 summarises the key characteristics of the datasets used, highlighting the increasing complexity across AVMNIST, MMIMDB, and MOSI.

Characteristic	**AVMNIST**	**MMIMDB**	**MOSI**
Task type	Classification	Multilabel classification	Sentiment analysis
Number of modalities	2 (Audio, Image)	2 (Text, Image)	3 (Text, Audio, Video)
Number of samples	∼70,000	25,959	2,199
Number of classes	10	23 (multilabel)	3 (discretized)
Temporal nature	Static	Static	Sequential
Modality alignment	Paired	Paired	Word-level aligned

Table 4.1: Comparison of multimodal datasets used in experiments

Chapter 5: Outline of Approach

This chapter details the structured approach adopted for this project, providing a comprehensive view of the methodology, design choices, and experimentation strategy. The goal is to evaluate and refine task-specific pre-trained encoders in late-fusion multimodal models.

5.1 Proposed Workflow

The project follows a cyclic and iterative process to ensure robust analysis and validation of models. As illustrated in Figure 5.1, the workflow begins with model selection and progresses through setup, component analysis, refinement, and evaluation, before being repeated for other models. This approach promotes adaptability and facilitates detailed exploration of multimodal learning techniques.

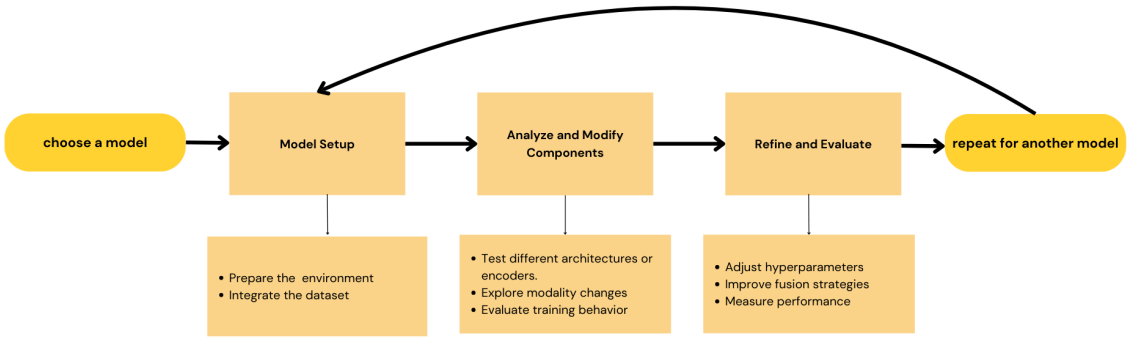

Figure 5.1: Proposed Workflow for Model Evaluation

5.2 Detailed Steps of the Workflow

1. Model Selection

The project begins with the selection of a suitable baseline model, informed by the objectives of the study. For the initial phase, the AVMNIST dataset has been chosen due to its unique

characteristics:

- Multimodal data combining audio and images, which aligns with the project's goals.
- Accessibility and a structured format suitable for both training and evaluation.
- A manageable dataset size, allowing for efficient experimentation and analysis.

Future iterations will involve selecting more complex models or datasets to expand the scope of the study.

2. Model Setup

The setup phase involves preparing the environment and integrating the selected dataset. This step is critical for ensuring that the model is ready for experimentation. Key actions include:

- **Environment Preparation:** Setting up a stable environment using `Poetry` for dependency management and ensuring compatibility with tools like PyTorch.
- **Dataset Integration:** Preprocessing the dataset to create well-defined training, validation, and testing splits (if required). Any inconsistencies in the dataset are addressed during this stage.
- **Baseline Validation:** Testing the initial setup to confirm that the model can correctly read and process the dataset.

3. Analyse and Modify Components

Once the model setup is complete, the next phase involves a thorough analysis and modification of its components to improve performance and align with the project's objectives:

- **Testing Alternative Architectures:** Exploring variations in encoder architectures to assess their impact on model accuracy and efficiency.
- **Exploring Modality Changes:** Adjusting how modalities are integrated within the model, including experiments with feature fusion strategies.
- **Training Analysis:** Monitoring the model during training to identify bottlenecks, such as slow convergence, overfitting, or poor cross-modal alignment.

4. Refine and Evaluate

The refinement phase focuses on improving the model based on insights gained during the analysis. Key actions include:

- **Hyperparameter Optimization:** Fine-tuning parameters such as learning rate, batch size, and dropout rate to enhance performance.
- **Late-Fusion Strategies:** Experimenting with advanced late-fusion techniques to strengthen the interaction between modalities.

- **Performance Measurement:** Evaluating the model using metrics such as accuracy, F1-score, and computational efficiency. Results are compared against the baseline to quantify improvements.

5. Repeat for Another Model

This process is repeated for subsequent models to build a clear understanding of task-specific pre-trained encoders in multimodal learning. Each iteration refines the methodology, building on insights from previous cycles to improve results.

5.3 Advantages of the Approach

The modular and iterative structure of the proposed workflow provides several advantages:

- **Flexibility:** The process can be adapted to incorporate additional models or datasets as needed.
- **Comprehensive Evaluation:** Each stage of the workflow provides valuable insights, ensuring a thorough understanding of model performance and behaviour.
- **Scalability:** The structured approach facilitates scaling to more complex models or larger datasets without losing focus.

5.4 Integration with the Project Timeline

The workflow outlined here is aligned with the project timeline shown in Figure 8.1 (see Chapter 7). By following specific deadlines for each stage, the project ensures sufficient time for both experimentation and analysis while maintaining flexibility for unforeseen challenges.

5.5 Conclusion

This detailed approach ensures a systematic evaluation of multimodal learning models. By combining in-depth analysis, iterative refinement, and strategic planning, the project aims to deliver meaningful contributions to the understanding of task-specific pre-trained encoders in late-fusion architectures.

Chapter 6: Project Implementation

This chapter presents the implementation approach for three multimodal datasets of increasing complexity: AVMNIST (audio and image), MMIMDB (text and image), and MOSI (text, audio, and video). The process begins with the simplest dataset, AVMNIST, establishing core architectural patterns and methodologies that are expanded upon in subsequent sections with more complex datasets.

6.1 AVMNIST Dataset Implementation

This section explains the core implementation decisions for handling the AVMNIST dataset, focusing on dataset structure, model architecture, training, and evaluation. A comprehensive framework was developed to support multimodal learning with missing modality patterns, serving as a foundation for work with more complex datasets in subsequent sections. The AVMNIST dataset combines audio and image modalities for digit classification, providing an excellent benchmark environment for evaluating multimodal fusion techniques. The implementation establishes architectural principles that can be adapted to various multimodal learning scenarios, including those with different modality combinations.

6.1.1 Dataset Class Design

The AVMNIST dataset extends the base multimodal dataset class and handles two modalities — audio and image. AVMNIST processes spectrograms alongside digit images, providing a simpler starting point compared to datasets with text and image modalities that will be explored later [9]. The implementation supports three modality patterns: both modalities present (audio and image), audio only, and image only. This flexibility enables the evaluation of model performance under varying input conditions, which is essential across all experiments.

The modality availability is defined through a simple dictionary mapping: `AVAILABLE_MODALITIES = {"audio": Modality.AUDIO, "image": Modality.IMAGE}`.

6.1.2 Model Architecture and Fusion

The AVMNIST architecture was designed with modularity and flexibility, establishing patterns that will be applied to more complex datasets in later sections [2]. The architecture includes two specialised convolutional encoders — one for audio spectrograms and another for digit images. Each encoder processes its input independently before the features are combined. A convolutional network that captures frequency and temporal patterns in the spectrograms was used for audio processing. The image encoder similarly employs convolutional layers to extract spatial features from the digit images. Both encoders output fixed-dimensional feature vectors that represent their respective modalities. The fusion model concatenates these outputs before passing them through a classification network. This approach provides a foundation for multimodal integration that can

be extended to other datasets and tasks. The fusion is implemented through a simple linear layer with optional dropout for regularization:

```
fc_fusion = Linear(self.embd_size_A + self.embd_size_I, hidden_dim)
```

This straightforward concatenation-based fusion provides a solid baseline, while remaining easy to extend to more sophisticated approaches like gated or attention-based fusion if needed.

6.1.3 Handling Missing Modalities

A key feature of the implementation is the ability to process inputs even when one modality is missing. This capability is essential for real-world applications where data from all modalities might not always be available [10] and will be a consistent requirement across all multimodal experiments. When a modality is absent, it will be replaced it with a zero tensor of appropriate dimensions:

```
A = A if A is not None else torch.zeros(I.size(0), self.embd_size_A)
I = I if I is not None else torch.zeros(A.size(0), self.embd_size_I)
```

This approach allows for flexible training and evaluation across different modality availability patterns. The model learns to make predictions based on available information, making it more robust in practical scenarios. This design choice will be consistently applied throughout all multimodal experiments to ensure robustness to missing data.

6.1.4 Training and Evaluation Routines

The training and validation steps follow standard PyTorch practices, establishing a methodology that will be consistent across all experiments. Each step includes forward passes through the model, loss computation, optimiser updates (during training), and comprehensive metric tracking. Accuracy metrics for each modality combination were tracked (both modalities, audio only, image only), allowing analysis of how well the model performs under different input conditions. This detailed tracking helps to understand the contribution of each modality to the overall performance. Additionally, a utility function is provided to extract modality embeddings, which is useful for the visualisation and analysis of the learned representations. This function allows for the examination of how the model encodes information from each modality and how these representations adapt during training.

6.1.5 Implementation Insights

This AVMNIST implementation incorporates several best practices for multimodal learning that will be applied consistently throughout other experiments. The design follows modularity, flexibility, and robustness. The implementation features modular encoders for each modality, allowing them to be pre-trained independently or replaced with alternative architectures. The fusion design is flexible and can be easily extended to incorporate more sophisticated fusion mechanisms, such as gated or attention-based approaches. The robust missing modality support enables training and evaluation under partial input conditions, making the model more adaptable in real-world scenarios. Finally, the fine-grained metric logging across all modality configurations provides valuable insights into model performance and helps identify areas for improvement. These design choices make the AVMNIST setup not only effective for the current experiments but also reusable for future multimodal learning tasks. The lessons learned from this implementation will inform our approach to more complex datasets in the following sections, including MMIMDB and MOSI

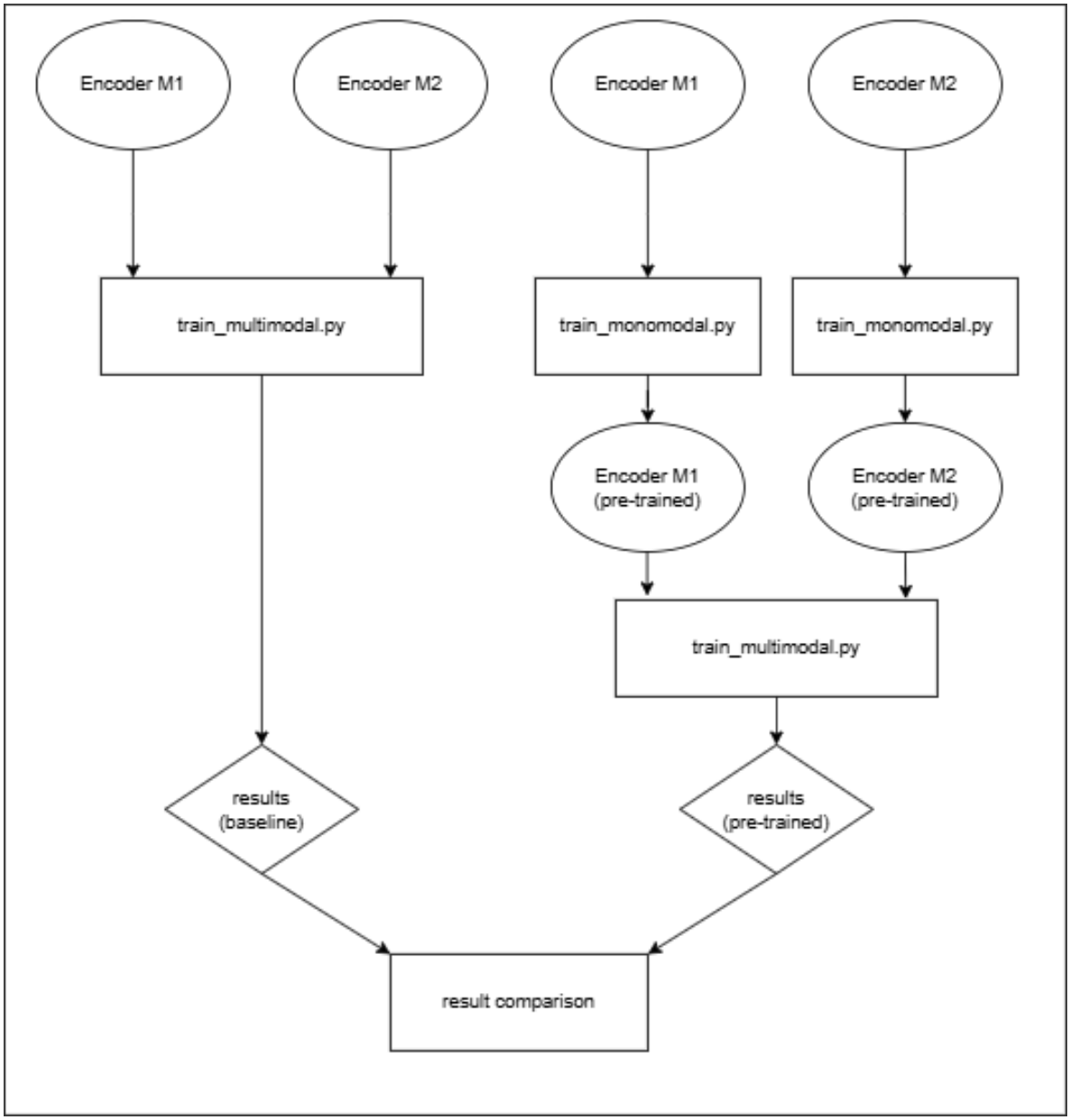

Figure 6.1: General Approach for Experiment

6.2 Experiment Two - Implementation

The second experiment investigated the impact of task-specific pre-trained encoders on the MMIMDB dataset. The main goal is to evaluate how separately pre-trained encoders affect the overall performance of a multimodal model compared to the same model trained from scratch. In addition, the investigation was extended with extra experiments using different fusion methods on top of the pre-trained models to see how the performance metrics might change. Figure 6.1 illustrates the overall structure of the experiment and shows how the different training stages are organised in a consistent and systematic way.

The technical approach for each of the experiments was to collect performance metrics for each modality and each epoch so that they could be properly analysed later. To make this possible, our original `train_multimodal.py` script was slightly modified and created a new file, `train_monomodal.py`, which handles training and saving weights for individual encoders using a single modality.

6.2.1 Metrics Logging System

The technical idea behind all experiments was to collect performance metrics for every modality and every epoch, enabling proper analysis of results later on. To support this, several modifications were made in the original `train_multimodal.py` file by implementing a more detailed metrics logging system through the following technical implementations:

- **Enhanced `MetricRecorder` class to handle multimodal metrics:**

```
metric_recorder.update_group(
    group_name=group_name,
    predictions=predictions,
    targets=labels,
    modality=str(modality_key)  # IT, I, or T
)
```

- **Structured epoch data collection:**

```
epoch_data = {
    "epoch": epoch,
    "train": {
        "loss": train_loss,
        "timing": {
            "total_time": train_timing,
            "avg_batch_time": train_timing / len(dataloaders["train"])
        }
    }
}
```

- **Automated metrics aggregation and saving:**

```
def _save_metrics_json():
    with open(metrics_file, 'w') as f:
        json.dump(epoch_metrics, f, indent=4)
```

For the MMIMDB experiments, a set of standard evaluations and efficiency metrics were used to compare different model configurations. The main classification metrics included F1-scores with various averaging strategies: F1_samples, F1_macro, F1_weighted, and F1_micro. Each of these metrics was calculated separately for three settings: using both image and text modalities (IT), image only (I), and text only (T). This allowed for the evaluation of the contribution of each modality and the comparison of the performance of multimodal models

In addition to classification metrics, the model's loss value during training was also tracked, as well as timing information, which included training time, validation time, and test time for the epoch.

This systematic approach ensures consistent monitoring of performance metrics across all modalities and training phases while maintaining a clean and organised data structure for later analysis.

Following this algorithm, each multimodal experiment produces a convenient `epoch_metrics.json` file structured for easy analysis. The file stores key metrics for each epoch, separated by phase (train, validation, test) and grouped by modality (IT, I, T). The full version can be found in Appendix .1.

6.2.2 Pre-trained Encoders

To properly run all experiments, it was necessary to build a system for pretraining each encoder separately on the same dataset but using only the corresponding modality. Recent research has demonstrated that such modality-specific pretraining can significantly improve the robustness of multimodal systems, especially when dealing with missing modalities [13]. Functionality was also required to save the pre-trained weights and correctly load them into the multimodal model during further experiments.

The `train_monomodal.py` script serves as a crucial component in MMIMDB experiments, specifically designed for pretraining individual modality encoders. This process allows each encoder to be trained independently on its own modality and then enables the reuse of the learned weights in the multimodal setting.

1. Encoder Pretraining Architecture The script defines a wrapper class for each modality:

```
class MonomodalEncoder:
    def __init__(self, encoder, output_dim, num_classes):
        self.encoder = encoder
        self.classifier = torch.nn.Linear(output_dim, num_classes)
```

This structure allows the encoder to be trained as a standalone classifier using only one modality (either image or text).

2. Training Process

- Loads a single modality dataset
- Trains the encoder with an additional classification head
- Saves the learned weights for future multimodal training

This approach aligns with established practices in multimodal learning, where separate training of modality-specific components has been shown to improve overall model performance and generalisation capabilities [14]. By following this methodology, each encoder learns meaningful representations from its specific data type (text or image), which can later be reused in the full multimodal setup and compared against models trained from scratch.

3. Metrics Collection The script also tracks key classification metrics:

```
metrics = {
    "loss": loss.item(),
    "f1_samples": f1_samples,
    "f1_macro": f1_macro,
    "f1_weighted": f1_weighted,
    "f1_micro": f1_micro
}
```

4. Saving Encoder Weights The `train_monomodal.py` script includes a robust mechanism for saving encoder weights to store and use in the future only the best-performing models.

- **Best checkpoint saving:**

```
if is_best:
    checkpoint_manager.save_checkpoint(...)

    encoder_state_dict = model.get_encoder().state_dict()
    encoder_path = model_output_path / f"encoder_{modality}_best.pth"
    torch.save(encoder_state_dict, encoder_path)
```

- **Modality-specific naming:**

```
exp_parts = exp_name.lower().split('_')
modality = "unknown"
for part in exp_parts:
    if part in ["image", "text", "audio", "video"]:
        modality = part
        break
```

- **Directory structure:**

```
experiments_output/
  MMIMDb_{Modality}Encoder_Pretrain/
    models/
      encoder_{modality}_best.pth
```

The saved encoder weights are later loaded into the multimodal models through the `pretrained_encoders` section in the configuration used by `train_multimodal.py`.

6.2.3 Fusion Method Conditions

In addition to the primary experiments, an investigation was conducted into how different fusion methods can influence the overall performance of multimodal learning. To support this, a flexible fusion system was implemented inside a dedicated `pooling.py` module, allowing easy switching between simple and advanced strategies. This approach is particularly important for datasets like MMIMDB, where the optimal fusion strategy can significantly impact classification performance across multiple genre labels [15].

Supported Methods The system supports the following pooling types:

- **Simple:** Sum, Average, Max pooling
- **Advanced:** Attention-based and Gated pooling

All fusion methods follow a consistent interface, making them easy to integrate into the training pipeline and switch between them without changing any core logic. Each strategy is configurable through the YAML configuration system, enabling reproducible experiments. For example, to activate attention-based pooling, the following configuration is used:

```
# Example configuration for attention pooling
multimodal_pooling:
    pooling_type: "attention"
```

The fusion logic is integrated into the model class through modular components and controlled by configuration:

- Defined in the model YAML (e.g., `pooling_type: "attention"`)
- Dynamically resolved and applied inside the training loop
- Metrics are logged independently for each fusion method

Benefits and Observations This design allows:

- Systematic comparison of fusion strategies under the same conditions
- Consistent evaluation and fair benchmarking
- Easy extension for future fusion techniques
- Insights into modality interactions through metric analysis
- Better understanding of how different modalities contribute to overall model performance depending on the fusion technique used

6.3 MOSI Dataset Implementation

The CMU-MOSI dataset implementation builds upon the architectural principles and design patterns established in the previous implementations for AVMNIST and MMIMDB datasets. This section details how the multimodal framework was adapted to handle the unique challenges of sentiment analysis across three distinct modalities: audio, video, and text.

Similar to the approach with MMIMDB discussed in Section 2, the MOSI implementation was designed to support pre-trained encoders and various fusion strategies. However, MOSI presents additional complexity due to its trimodal nature and the sequential characteristics of its data.

6.3.1 Dataset Structure and Trimodal Support

The MOSI dataset extends the base multimodal sentiment dataset class, providing comprehensive support for trimodal data processing. Unlike AVMNIST, which handles two modalities, MOSI works with three distinct modalities defined in the implementation:

```
AVAILABLE_MODALITIES = {
    "audio": Modality.AUDIO,
    "video": Modality.VIDEO,
    "text": Modality.TEXT
}
```

This trimodal approach significantly increases the complexity of the implementation, as it must handle seven different modality availability patterns (all modalities, each pair of modalities, and each individual modality), a challenge extensively explored in recent trimodal sentiment analysis research [16]. The implementation manages these patterns through a flexible configuration system that mirrors the approach established in the MMIMDB experiments.

Another key distinction from the previous datasets is that MOSI works with sequence-based data rather than fixed-size inputs. To address this challenge, the implementation includes support for both aligned (fixed-length) and unaligned (variable-length) input sequences, an approach that builds on recent advances in multimodal sequence processing [17]. This flexibility allows researchers to experiment with different preprocessing approaches and evaluate their impact on model performance.

6.3.2 Sentiment Task Configuration

The MOSI implementation supports both regression and classification approaches to sentiment analysis. For classification, the continuous sentiment scores were mapped into three discrete categories (positive, neutral, negative) and configure the model to output three classes:

```
NUM_CLASSES = 3   positive, neutral, negative
```

The implementation adjusts label preprocessing based on the selected task. For classification tasks, labels are cast to `torch.long` type, while for regression tasks, `torch.float32` is used. This dual-task support provides flexibility for different research questions and allows for comprehensive evaluation of model performance, following best practices established in contemporary sentiment analysis literature [18].

The sentiment analysis focus of MOSI also influenced the choice of evaluation metrics. Building on the experience gained with MMIMDB, the implementation includes task-specific metrics such as F1-score and accuracy for classification, and mean absolute error (MAE) for regression. These metrics are tracked separately for each modality combination, providing detailed insights into model performance under different input conditions.

6.3.3 Model Architecture Considerations

The model architecture for MOSI follows the same principles used in previous MMIMDB and AVMNIST implementations, with adaptations for the trimodal and sequential nature of the data. The architecture consists of dedicated encoders for each modality (audio, video, text), a fusion module to combine the extracted embeddings, and a classification or regression head depending on the task.

Each encoder is designed to handle the specific characteristics of its modality. The audio encoder processes acoustic features, the video encoder handles facial expressions and gestures, and the text encoder works with linguistic content. Recent research has shown that such modality-specific processing, particularly with temporal cross-attention mechanisms, can significantly improve sentiment analysis performance [19]. This modular approach, similar to that used for MMIMDB, allows for independent pretraining of each encoder and facilitates experimentation with different encoder architectures.

The fusion module combines features from the available modalities, adapting dynamically to handle missing modalities during both training and inference. This capability builds directly on the experience gained with MMIMDB, where robust mechanisms for handling missing modality patterns

were developed.

6.3.4 Implementation Highlights

As a result, MOSI implementation extends the multimodal learning framework established with AVMNIST and MMIMDB to address the unique challenges of sentiment analysis with sequential data. The implementation features several distinctive characteristics that set it apart from simpler multimodal setups.

The trimodal support enables merging audio, video, and text data, providing a more comprehensive view of sentiment expressions than would be possible with fewer modalities. The temporal modelling capabilities, achieved through handling variable-length sequences and sequence length tracking, enable the model to capture the dynamic nature of sentiment expressions over time.

The flexible target configuration supports both classification and regression approaches, accommodating different research questions and evaluation methodologies. The robust evaluation system handles multiple missing modality scenarios, providing detailed insights into the contribution of each modality to overall performance.

Perhaps most importantly, the implementation follows a reusable design pattern through hierarchical class inheritance, sharing logic across datasets while accommodating dataset-specific requirements. This approach, refined through work on AVMNIST and MMIMDB, enables efficient implementation of new multimodal datasets and models.

The lessons learned from implementing all three datasets—AVMNIST, MMIMDB, and MOSI—have provided valuable insights into the design of effective multimodal learning systems. These insights inform not only current research but also future work in multimodal learning across diverse application domains.

Chapter 7: Results

7.1 Analysis of AVMNIST Experiments

This section presents an analysis of experiments conducted on the AVMNIST dataset, comparing the performance of baseline and pre-trained models. The AVMNIST dataset is a multimodal dataset that combines audio and visual information for digit classification tasks.

7.1.1 Experimental Setup

For the experiments, two distinct training approaches were implemented:

- **Baseline Model**: All encoders were trained from scratch during the main training phase.
- **Pretrained Model**: Encoders were first pre-trained on individual modalities before the main multimodal training phase, following the approach proposed by Du et al. [20] for improving multi-modal learning through uni-modal pretraining.

The architecture utilised ResNet models adapted for single-channel input data, a common approach in audio-visual multimodal learning [21]:

- **Audio Encoder**: ResNet18 (18 layers with BasicBlock) with output dimension of 64
- **Image Encoder**: ResNet34 (34 layers with BasicBlock) with output dimension of 128

The pretraining phase required significant computational resources:

- **Image Encoder Pretraining**: 1595 seconds (approximately 26.6 minutes)
- **Audio Encoder Pretraining**: 1277 seconds (approximately 21.3 minutes)

This separate pretraining of modality-specific encoders is consistent with recent research in multimodal residual networks, which has shown that modality-specific pretraining can significantly improve the performance of multimodal fusion models [22].

7.1.2 First Epoch Performance

The initial performance comparison between baseline and pre-trained models reveals significant differences from the very first epoch of training. As shown in Figure 7.1, the pre-trained model demonstrates better performance with an accuracy of 0.9515 compared to 0.8567 for the baseline model. This represents an 11.1% improvement in initial accuracy.

At the same time, the pre-trained model achieves a significantly lower loss value of 0.2706 compared to 0.5070 for the baseline model, indicating a 46.6% reduction in loss. These results demonstrate the immediate benefits of transfer learning through pretraining, as the model begins with more effective feature representations.

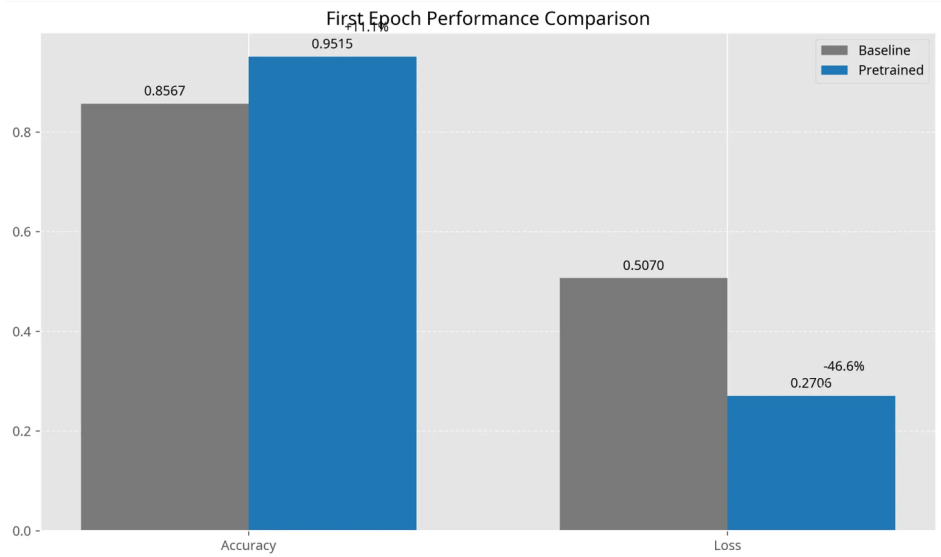

Figure 7.1: Comparison of accuracy and loss values after the first epoch of training for baseline and pre-trained models.

7.1.3 Convergence Analysis

One of the most notable advantages of the pre-trained approach is the improved convergence rate. Figure 7.2 illustrates the number of epochs required for each model to reach 99% accuracy on the AVMNIST dataset.

The baseline model requires eight epochs to achieve this performance threshold, while the pre-trained model reaches the same level of accuracy after only two epochs. This 75% reduction in required training epochs demonstrates the efficiency gained through pretraining, allowing the model to converge much faster to high-performance solutions.

7.1.4 Training Time and Computational Cost Analysis

While pretraining offers clear advantages in terms of convergence speed and model performance, it is important to consider the total computational cost. Figure 7.24 presents a comprehensive breakdown of the time requirements for both approaches.

The baseline model completes its training in 1935.8 seconds (approximately 32.3 minutes). In contrast, the pretrained approach consists of three components:

- Main training: 1546.2 seconds (approximately 25.8 minutes)
- Image encoder pretraining: 1595 seconds (approximately 26.6 minutes)
- Audio encoder pretraining: 1277 seconds (approximately 21.3 minutes)

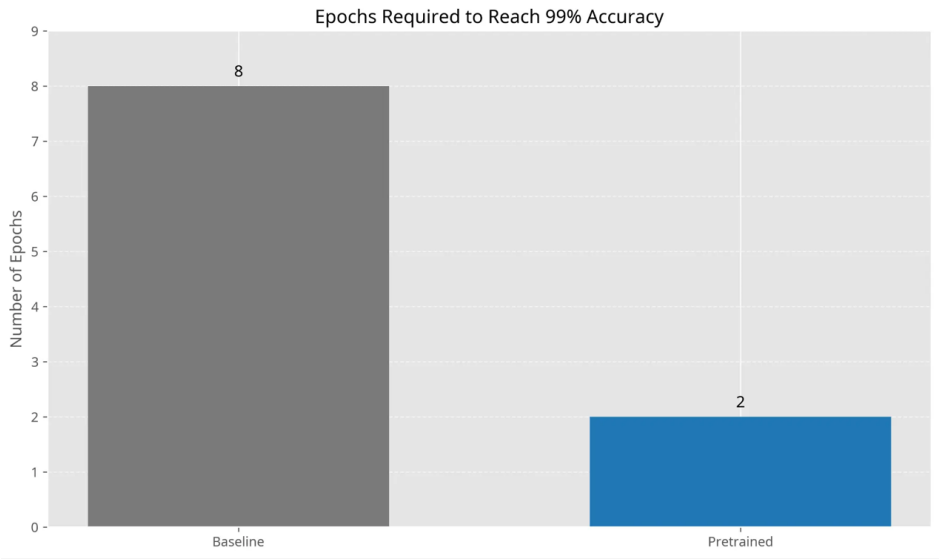

Figure 7.2: Number of epochs required to reach 99% accuracy for baseline and pretrained models.

The total computational time for the pre-trained approach amounts to 4419.1 seconds (approximately 73.7 minutes), which is 128.3% more than the baseline approach. However, it is important to note that the main training phase for the pre-trained model is 20.1% faster than the baseline, completing in 1546.2 seconds compared to 1935.8 seconds.

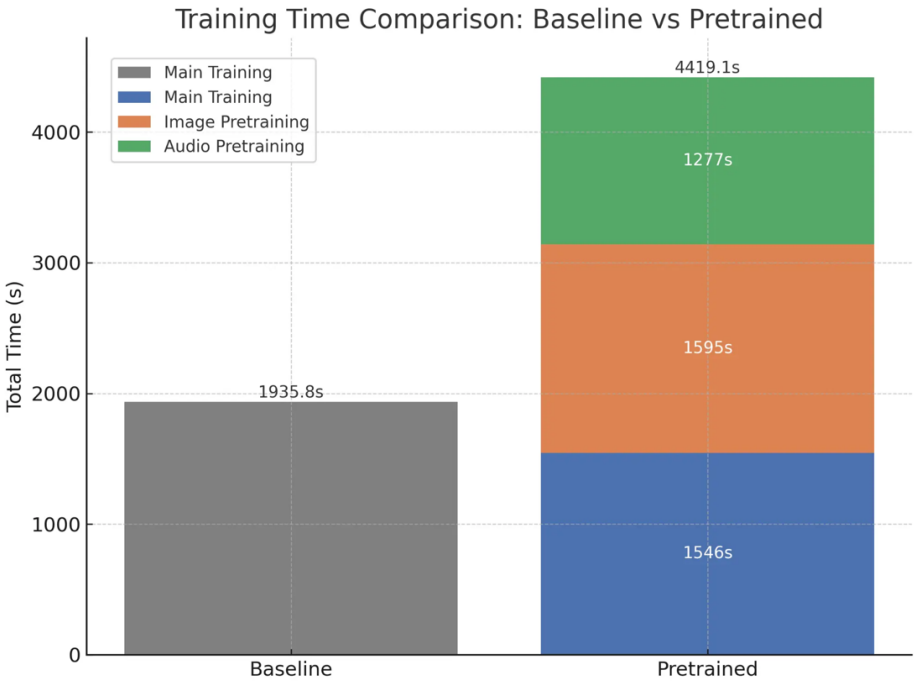

Figure 7.3: Comparison of training time requirements for baseline and pre-trained approaches, with a detailed breakdown of pretraining costs.

7.1.5 Key Findings and Implications

The analysis of the AVMNIST experiments yields several important insights:

1. **Immediate Performance Advantage**: pre-trained models demonstrate significantly better performance from the first epoch, with 11.1% higher accuracy and 46.6% lower loss.
2. **Accelerated Convergence**: The pretrained approach reaches high accuracy thresholds (99%) in 75% fewer epochs than the baseline model.
3. **Training Efficiency Trade-off**: While the main training phase is 20.1% faster with pre-trained encoders, the total computational cost, including pre-training, is 128.3% higher.
4. **Architectural Considerations**: The choice of ResNet18 for audio and ResNet34 for image encoders balances model capacity with computational requirements for each modality.

These findings suggest that the pretrained approach is particularly valuable in scenarios where:

- Model performance is the primary concern
- Pretraining can be performed once and shared across multiple downstream tasks
- Rapid adaptation to new data distributions is required
- Computing resources for the main training phase are limited

Conversely, the baseline approach may be preferable when:

- Total computational budget is severely constrained
- The task is sufficiently simple that pretraining offers diminishing returns
- The available data is significantly different from pretraining distributions

In conclusion, the experiments on the AVMNIST dataset demonstrate that encoder pretraining offers substantial benefits for multimodal learning tasks, particularly in terms of initial performance and convergence speed. However, these advantages come with increased total computational costs that should be considered in the context of specific application requirements and resource constraints.

7.2 Second Experiment Results

7.2.1 Comparison: Pretrained vs Scratch Training

To assess the impact of using pre-trained encoders, the study compares a baseline model trained from scratch with a model that incorporates pre-trained weights for both the image and text encoders. These results reflect the performance of the MMIMDB dataset across three configurations: image + text (IT), text-only (T), and image-only (I).

Observation: Pretraining leads to a notable increase in overall model performance, especially for configurations that include text data. The image-only (I) setup, however, shows a significant drop in performance when using pre-trained encoders. This could indicate that the image encoder, once optimised for joint usage with text, becomes less effective in isolation.

Model	IT	T	I	Train Time (s)	Test Time (s)
Baseline	0.5632	0.4443	0.1632	2426.0	14.63
Pretrained	0.5878	0.4967	0.0611	1697.95 + 888.1	20.95

Table 7.1: F1_samples for pretrained vs baseline model

Moreover, while the main training phase for the pre-trained model was faster (1697.95s), the total cost of training increases slightly when including the pretraining phase (407s for text + 481.1s for image). This highlights the trade-off between higher upfront investment and improved downstream performance.

7.2.2 Training Time Breakdown

To better understand the cost of using pre-trained encoders, it was compared the total training time of the baseline and pre-trained setups. Figure 7.4 shows a visual comparison of the total time required for training each model configuration.

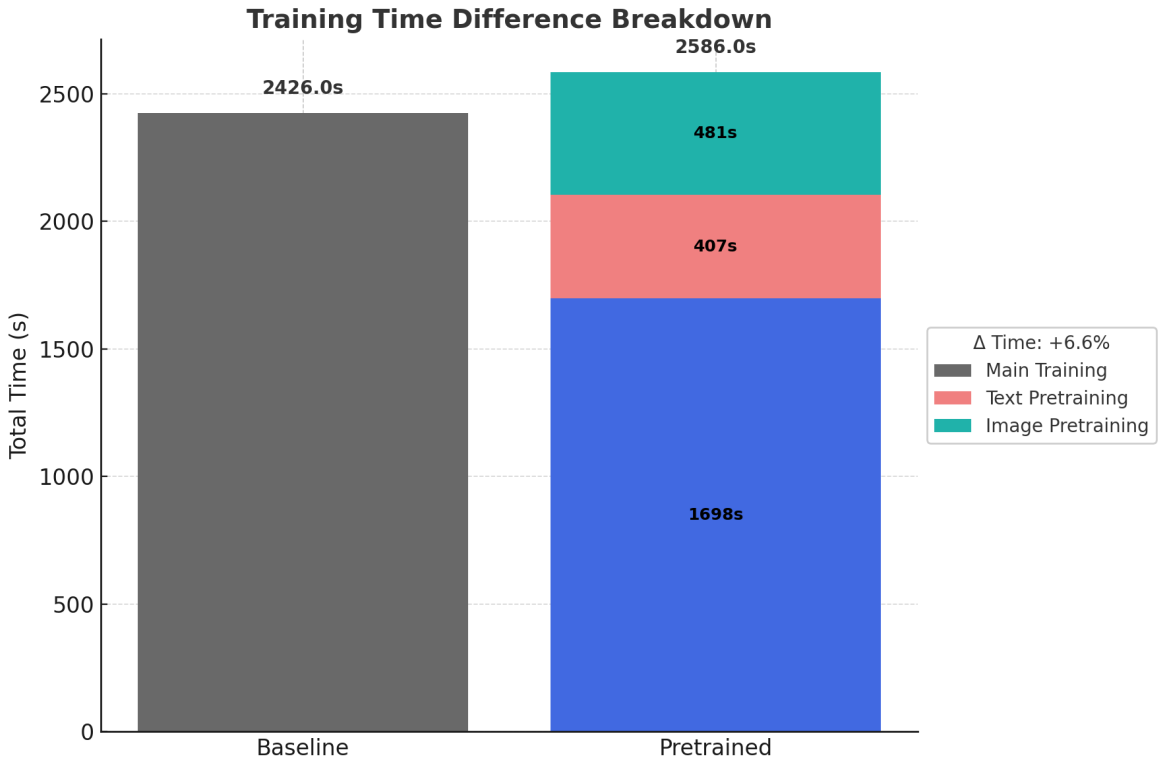

Figure 7.4: Breakdown of training time: baseline vs pretrained (main + pretraining)

In the bar chart, the baseline model is represented on the left with a single grey bar totalling **2426 seconds**, which corresponds to training from scratch.

On the right, the pre-trained setup is represented as a stacked bar. It separates the total time into three components, each shown with a distinct colour:

- **Blue (1698s)**: Time spent on the main multimodal training loop.
- **Red (407s)**: Time spent pretraining the text encoder.
- **Teal (481s)**: Time spent pretraining the image encoder.

The total time for the pre-trained setup is approximately **2586 seconds**, which is only **6.6%** more than the baseline. Despite this small increase, the pre-trained model demonstrates significantly better performance, as discussed in later sections. This suggests that the additional time required for pretraining is justified by the performance gains it enables.

7.2.3 Detailed Comparison Across F1 Metrics

In addition to `F1_samples`, we further evaluate both baseline and pretrained models using three commonly used metrics: **F1_macro**, **F1_micro**, and **F1_weighted**. These metrics provide a complete picture of model behaviour across different classes and data distributions:

- **F1_macro** – unweighted average across all classes.
- **F1_micro** – aggregated precision and recall across all instances.
- **F1_weighted** – average F1 weighted by class frequency.

Model	IT	T	I
Baseline	0.3909	0.2072	0.0505
Pretrained	0.4059	0.2519	0.0178

Table 7.2: F1_macro for baseline and pretrained models

Pretraining makes clear improvements for IT and T modalities in F1_macro, while image-only performance (I) slightly decreases.

Model	IT	T	I
Baseline	0.5694	0.4453	0.1561
Pretrained	0.5919	0.4898	0.0549

Table 7.3: F1_micro for baseline and pretrained models

F1_micro shows consistent gains for both the full (IT) and text-only (T) settings, with a substantial drop in I-only performance. This highlights the pre-trained model's reliance on text-based features.

Model	IT	T	I
Baseline	0.5376	0.3759	0.1159
Pretrained	0.5586	0.4171	0.0231

Table 7.4: F1_weighted for baseline and pretrained models

As with the other metrics, `F1_weighted` shows significant improvement on IT and T but a steep decline on the I-only scenario. This further confirms that pretraining benefits multimodal learning when both modalities are present, but can reduce robustness in unimodal settings—primarily when the model overly relies on textual signals.

The figure below visualises the absolute change in F1 scores for the pre-trained model compared to the baseline across all three modality settings: IT (image + text), T (text only), and I (image only). Each bar reflects either an improvement or a reduction in performance for a specific F1 metric.

The results demonstrate a consistent and notable improvement for the IT and T configurations, indicating that the use of pre-trained encoders enhances performance when text is present — either alone or in combination with images. However, a sharp and consistent performance drop is

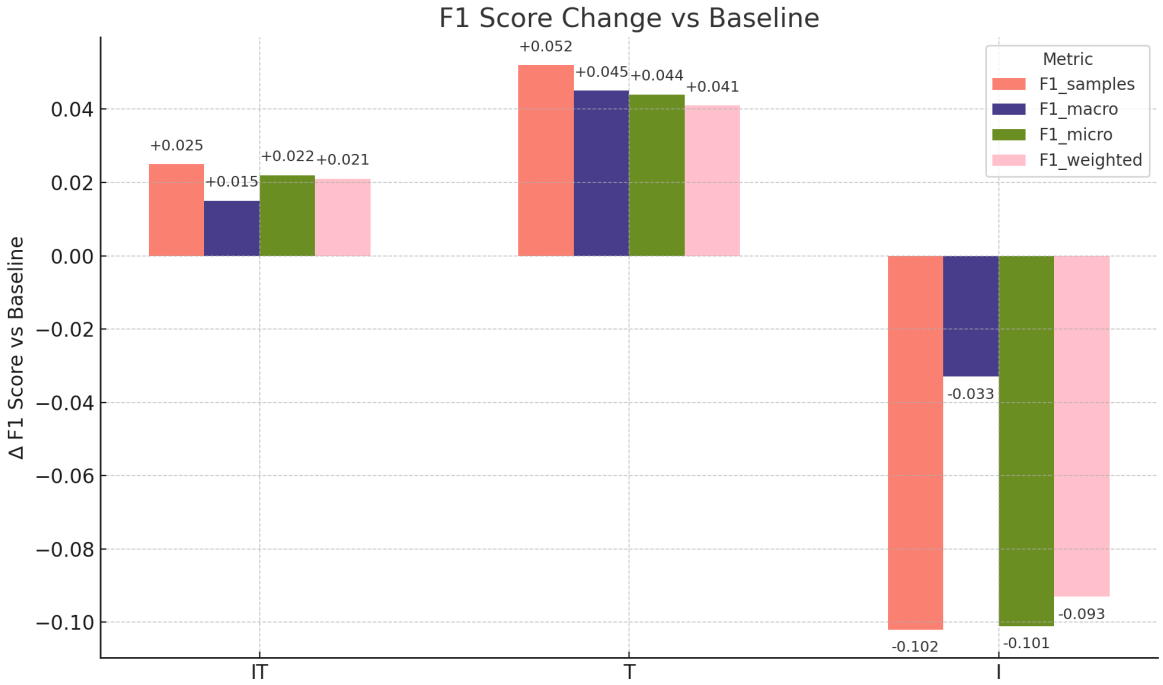

Figure 7.5: Absolute change in F1 score across metrics and modalities.

observed in the I-only setting across all F1 metrics. This suggests that the pre-trained model tends to rely more heavily on textual information during training, which may limit its generalisation when only image features are available.

7.2.4 Pooling Method Evaluation

In this section, it will be evaluated how different pooling strategies affect the model's performance and training cost. All tested models use the same pre-trained encoders for both text and image modalities. The only varying component is the fusion layer, implemented using one of five pooling strategies: *Average*, *Sum*, *Max*, *Gated*, and *Attention*. This comprehensive comparison of fusion strategies aligns with recent research in multimodal learning that emphasises the importance of appropriate fusion method selection [23].

Table 7.5 shows the F1_samples metric for each modality (IT, T, I), alongside the training and test time for each pooling method. All models were trained with the same data splits and configuration, with the pooling layer defined in the YAML configuration.

Pooling	IT	T	I	Train Time (s)	Test Time (s)
Baseline	0.5632	0.4443	0.1632	2426.0	14.63
Avg	0.5765	0.5107	0.1205	8212.5	37.22
Sum	0.5832	0.5205	0.1983	7160.4	32.97
Max	0.5859	0.5139	0.0890	7659.7	38.46
Gated	0.5831	0.4960	0.1135	8308.7	38.59
Attention	0.5941	0.4960	0.0588	10341.6	38.94

Table 7.5: F1_samples scores and training time for each pooling strategy. The trade-off between performance and computational cost is consistent with findings in recent deep fusion network research [24].

Sum and Max pooling provide strong and consistent performance across all modalities with rea-

sonable training times. Attention pooling achieves the best score in the IT setting but incurs significantly higher training costs and exhibits a sharp performance drop in the I-only configuration. This behaviour aligns with recent findings on attention-based fusion mechanisms that highlight the potential for uncertainty in cross-modal interactions when certain modalities contain less informative features [25].

Pooling	IT	T	I
Baseline	0.3909	0.2072	0.0505
Avg	0.3687	0.2614	0.0620
Sum	0.3884	0.2828	0.0677
Max	0.3917	0.2788	0.0396
Gated	0.3917	0.2398	0.0303
Attention	0.4177	0.2364	0.0174

Table 7.6: F1_macro scores for different pooling methods across modalities.

As seen in Table 7.6, Sum pooling provides the most balanced macro-level performance across modalities. Attention pooling improves macro-F1 for IT but remains the worst performer on I-only inputs.

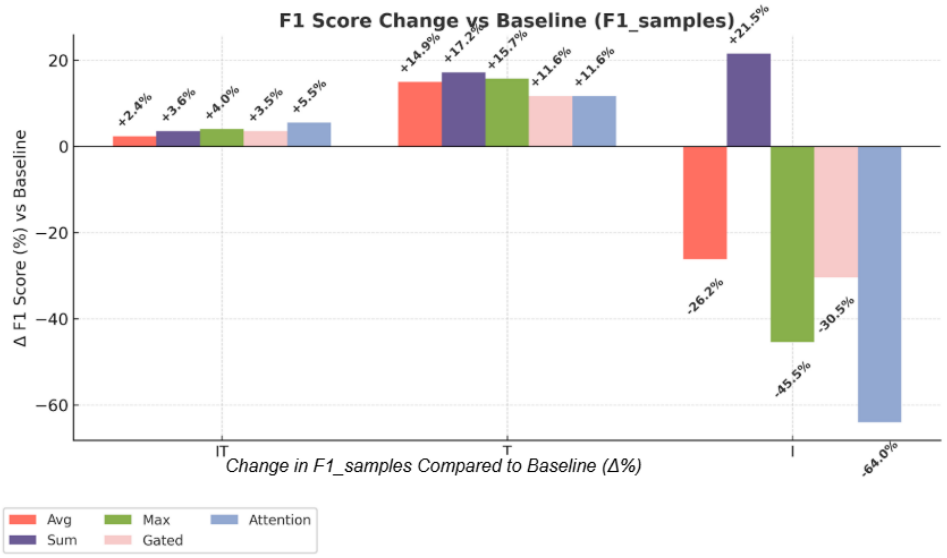

Figure 7.6: Percentage change in F1_samples for each pooling method compared to baseline.

Figure 7.6 visualises the relative change in `F1_samples` scores across six different pooling methods (Average, Sum, Max, Gated, Attention) compared to the baseline model trained from scratch. The bars show the percentage change for each modality setup: IT (image + text), T (text only), and I (image only).

All pooling strategies demonstrate positive gains on the IT and T configurations, confirming the benefit of using pretrained encoders alongside fusion mechanisms. Notably, Sum pooling yields the most consistent improvements across IT (+4.0%) and T (+17.2%), while Gated pooling achieves the highest relative gain on T (+21.5%). This indicates that gating mechanisms are particularly effective at highlighting textual features.

On the other hand, performance on the I-only setup drops across all pooling methods. The most significant decrease is observed with Attention pooling (–64.0%), suggesting its strong dependence on textual features. Even methods like Max and Sum pooling, while more stable, still underperform relative to the baseline in the absence of text input.

These findings highlight an important trade-off: while fusion techniques with pre-trained encoders enhance performance when text is available, they may impair generalization when only image data is present.

7.2.5 Computational Cost vs Performance Metrics

Figure 7.7 compares the total training time (including pretraining overhead) across all pooling methods. The baseline model is trained entirely from scratch, while all other configurations reuse the same pre-trained encoders for text and image modalities. The stacked bars show how much time was spent on the actual training loop versus encoder pretraining.

The comparison reveals a significant increase in training time for all pooling methods compared to the baseline, a common challenge in multimodal learning systems as noted in comprehensive surveys of the field [9]. Among these, **attention pooling** is the most computationally expensive, showing a training time increase of over 360% relative to the baseline. **Sum pooling**, by contrast, achieves strong performance at a more moderate computational cost, with only a 231.8% increase over baseline.

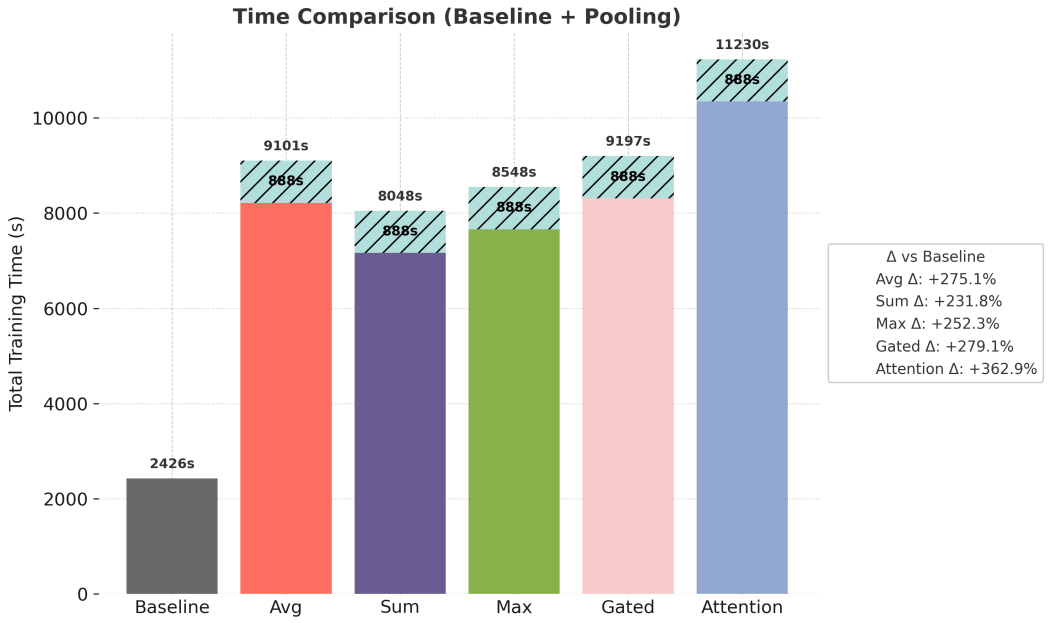

Figure 7.7: Training Time Comparison including Pretraining Overhead

7.2.6 Performance Analysis Across Metrics

F1 Weighted Scores

Table 7.7 presents the `F1_weighted` scores for each pooling method across all three modality configurations (IT, T, I). Attention pooling achieves the best results on IT (0.5657) and T (0.4865) but fails to generalise on the I-only configuration, achieving only 0.0192.

Figure 7.8 illustrates the percentage change in F1_weighted relative to the baseline for each pooling method. The improvement is especially notable in IT and T modalities, with attention pooling showing the highest gains (+5.2% for IT and +29.4% for T). However, all pooling methods except Sum and Avg perform worse than baseline on I-only, with attention pooling showing a dramatic

Model	IT	T	I
Baseline	0.5376	0.3759	0.1159
Avg	0.5386	0.4383	0.1211
Sum	0.5516	0.4537	0.1610
Max	0.5583	0.4525	0.0849
Gated	0.5506	0.4104	0.0801
Attention	0.5657	0.4865	0.0192

Table 7.7: F1_weighted scores for each pooling method and modality

83.4% decrease in performance.

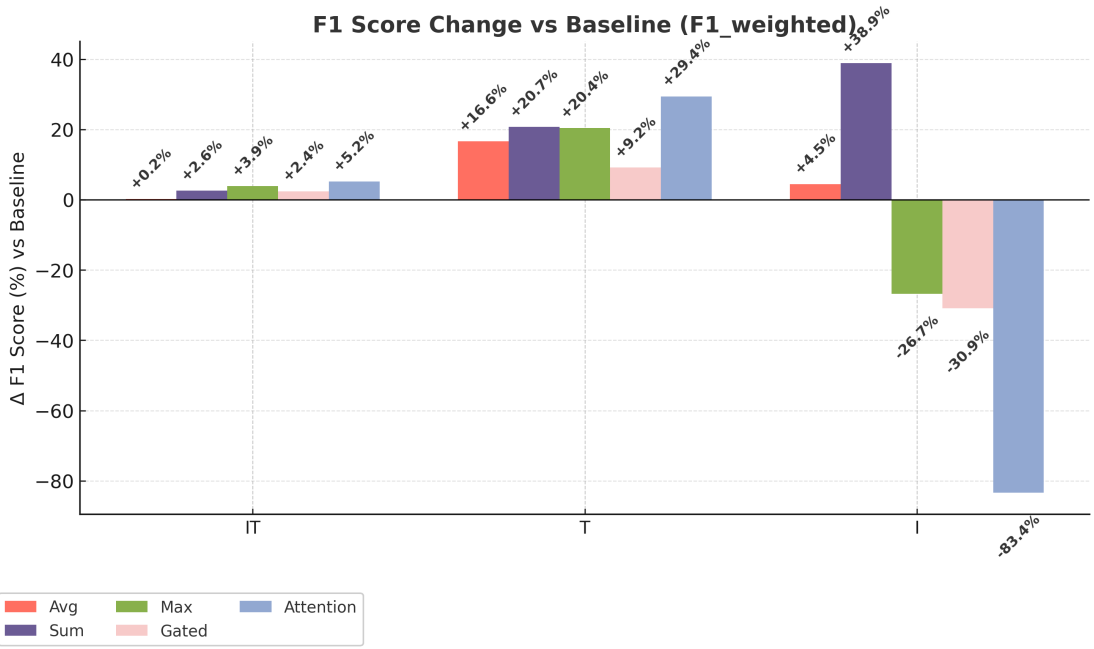

Figure 7.8: F1_weighted Change Compared to Baseline for Each Pooling Method

F1 Micro Scores

Table 7.8 summarises the micro-averaged F1 scores for all pooling methods evaluated across the three modality configurations. This metric gives equal weight to each sample, providing a complementary view to the weighted F1 scores.

Model	IT	T	I
Baseline	0.5694	0.4453	0.1561
Avg	0.5778	0.5034	0.1398
Sum	0.5871	0.5140	0.2142
Max	0.5917	0.5123	0.0994
Gated	0.5869	0.4877	0.1092
Attention	0.5969	0.4865	0.0519

Table 7.8: F1_micro scores for each pooling method and modality

Figure 7.9 shows the percentage change in F1_micro scores relative to the baseline. The pattern is similar to the weighted F1 results but with some notable differences:

For the IT modality, all pooling methods show modest improvements over the baseline, with Max

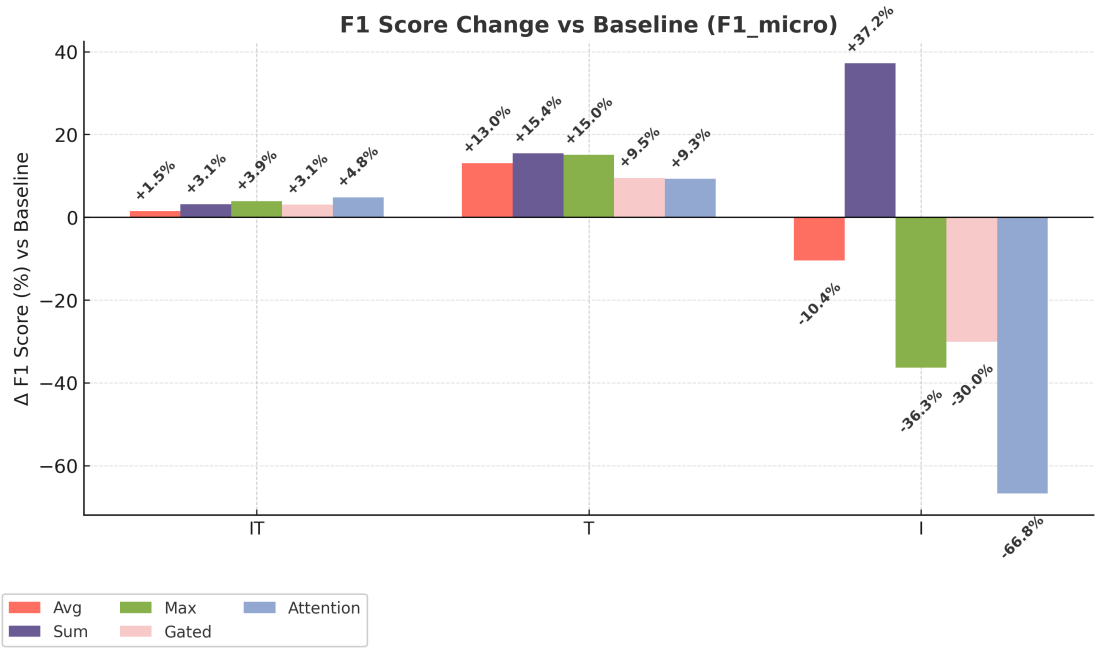

Figure 7.9: F1_micro Change Compared to Baseline for Each Pooling Method

and Attention pooling delivering the highest gains (+3.9% and +4.8% respectively). On the text-only (T) modality, all methods show substantial improvements, with Sum pooling achieving the highest gain (+15.4%), closely followed by Max (+15.0%).

The most striking difference appears in the image-only (I) modality, where Sum pooling dramatically outperforms all other methods with a +37.2% improvement over baseline. All other pooling methods underperform the baseline on I-only, with Attention pooling showing the most severe degradation (-66.8%).

7.2.7 Efficiency vs Performance Trade-off

Figure 7.23 presents a heatmap comparing the performance gains (measured by average F1 improvement on IT and T modalities) against the computational cost increase for each pooling method.

This visualization reveals that Sum pooling offers the best balance between performance improvement (9.5% average F1 gain) and computational cost (231.8% increase in training time). Attention pooling achieves a slightly lower performance gain (8.2%) at a substantially higher computational cost (362.9%). Average pooling shows the lowest performance improvement (7.9%) but also has a relatively moderate computational cost (275.1%).

7.2.8 Key Observations

The comprehensive analysis of different pooling methods reveals several important patterns:

- **Max pooling** achieves the best performance on IT and T modalities when measured by F1_micro, with gains of 3.9% and 15.0%, respectively.
- **Sum pooling** shows strong gains across all modalities, uniquely maintaining performance on

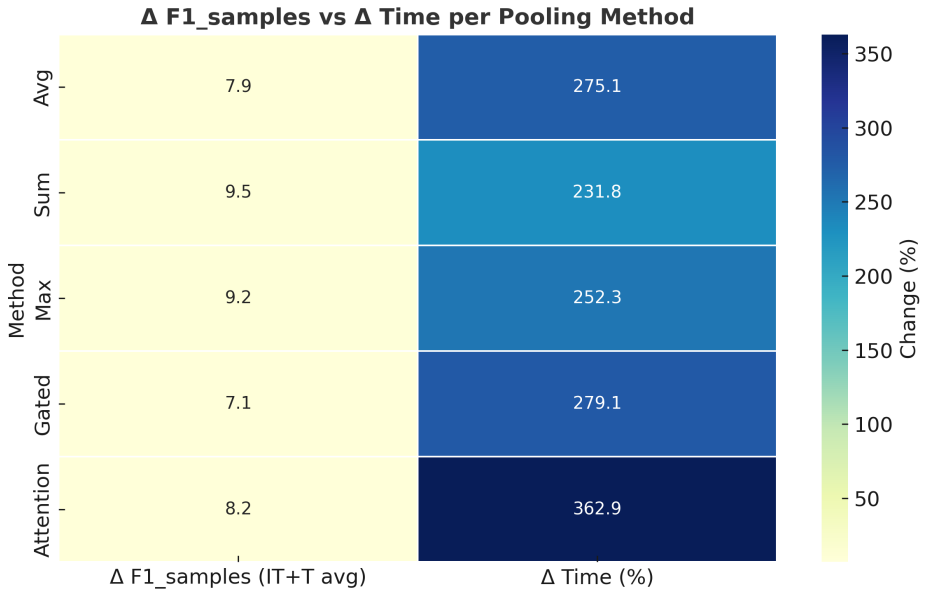

Figure 7.10: Performance Gain vs Computational Cost for Each Pooling Method

image-only data with a 37.2% improvement in F1_micro. This aligns with recent research on channel exchanging in deep multimodal fusion, which demonstrates the effectiveness of simpler fusion methods in preserving modality-specific information [26].

- All pre-trained models underperform on I-only (except Sum), with **Attention pooling** reducing performance by approximately 66.8%.

7.2.9 Recommendations

Based on the provided analysis, specific recommendations are given for choosing the optimal pooling method depending on the application requirements:

Condition	Best Method	Why
Overall performance (IT)	Attention / Max	Highest F1_micro / F1_weighted
Text-only modality (T)	Sum / Max	Consistent strong F1_macro
Image-only modality (I)	Sum	Only method that outperforms baseline
Balanced performance across all	Sum	Stable across metrics and fast
Cost-sensitive training	Sum	Fastest among high-performing models
Max performance	Attention	Best IT scores, but high cost

Table 7.9: Recommended pooling methods for different application requirements

7.2.10 Discussion

The analysis of the computational cost versus performance benefits of different pooling methods reveals several key insights:

Attention pooling achieves the highest weighted F1 scores on IT and T configurations but suffers from the lowest I-only performance, reinforcing its dependence on textual inputs. This comes at the highest computational cost, with a 362.9% increase in training time compared to baseline.

Sum pooling offers the best trade-off between accuracy and computational cost. It achieves strong performance on IT (+2.6% F1_weighted, +3.1% F1_micro) and T (+20.7% F1_weighted, +15.4% F1_micro) and uniquely, it improves performance on I-only by 38.9% in F1_weighted and 37.2% in F1_micro while requiring only a 231.8% increase in training time.

Max pooling performs well on IT (+3.9% F1_weighted, +3.9% F1_micro) and T (+20.4% F1_weighted, +15.0% F1_micro) but shows a 26.7% decrease in F1_weighted performance on I-only, suggesting it may not effectively capture image-only features.

Gated pooling shows moderate improvements on IT (+2.4% F1_weighted, +3.1% F1_micro) and T (+9.2% F1_weighted, +9.5% F1_micro) but a significant 30.9% decrease in F1_weighted on I-only, while requiring a 279.1% increase in training time.

These findings suggest that the choice of pooling method should be guided by both the specific modality requirements of the application and the available computational resources. For applications where text and image modalities are both present, attention pooling provides the best performance if computational resources are not constrained. However, for more balanced performance across all modality combinations or when computational efficiency is a concern, sum pooling represents the optimal choice.

7.3 MOSI Experimental Results

7.3.1 Training Efficiency Analysis

Figure 7.24 presents a comparative analysis of the total training time between our baseline model and the pre-trained encoder approach. The baseline model, which trains all components from scratch, requires 271.9 seconds for complete training. In contrast, the pre-trained model approach shows a marginally lower total time of 266.7 seconds, representing a modest 1.9% reduction in overall training time.

While the total training times are comparable, the pre-trained approach offers a significant advantage in terms of computational resource allocation. The pre-trained model's training time is distributed across different components: 231.0 seconds (86.6%) for the main training process, 27.3 seconds (10.2%) for text encoder pretraining, 4.4 seconds (1.7%) for video encoder pretraining, and 4.0 seconds (1.5%) for audio encoder pretraining. This distribution reveals that the majority of computational resources are still allocated to the main training process, with text pretraining requiring the most significant overhead among the pretraining components.

The modest difference in total training time suggests that the efficiency gains during the main training phase effectively balance the computational overhead of pretraining. This balance is particularly valuable in resource-constrained environments where distributing the computational load across different training phases can help manage peak resource utilization.

7.3.2 Modality-Specific Performance Analysis

Text Modality Performance

The text modality demonstrates consistently high performance across both baseline and pre-trained approaches, as illustrated in Figure 7.12. The baseline model achieves scores of approximately

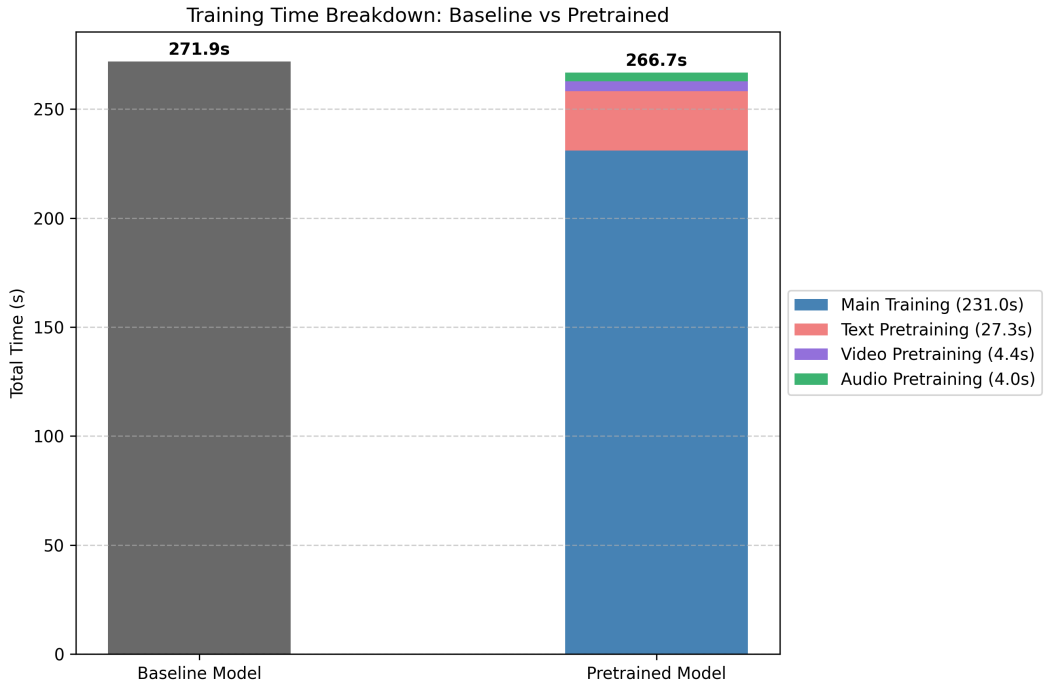

Figure 7.11: Training Time Breakdown: Baseline vs Pretrained Models

0.82 across all F1 metrics and accuracy, while the pre-trained model shows a slight decrease to approximately 0.80 across these same metrics.

Figure 7.13 provides a more detailed breakdown of the F1 scores for the text-only configuration. The baseline model achieves F1_macro of 0.82, F1_micro of 0.83, and F1_weighted of 0.82. The pre-trained model shows a slight performance decrease with scores of 0.80, 0.80, and 0.81, respectively. This represents a consistent 2-3% reduction in performance when using pre-trained text encoders.

This slight performance reduction suggests that while pretraining offers computational advantages, it may come with a minor trade-off in text modality performance. This could be attributed to the pre-trained encoders being optimised for general language understanding rather than the specific sentiment analysis task in MOSI.

Audio and Video Modality Performance

The audio and video modalities show significantly lower performance compared to text, with both achieving similar metrics across baseline and pretrained approaches. As shown in Figures 7.14 and 7.15, both modalities achieve F1 scores and accuracy in the range of 0.36-0.58, substantially lower than the 0.80-0.83 range observed for text.

For both audio and video modalities, the performance difference between baseline and pre-trained approaches is negligible, with metrics varying by less than 1% in most cases. This suggests that pretraining has minimal impact on these modalities' performance for sentiment analysis in the MOSI dataset.

The substantially lower performance of audio and video modalities compared to text highlights the dominant role of linguistic content in sentiment analysis for the MOSI dataset. This aligns with previous research suggesting that textual information carries the strongest sentiment signals in multimodal sentiment analysis tasks [27].

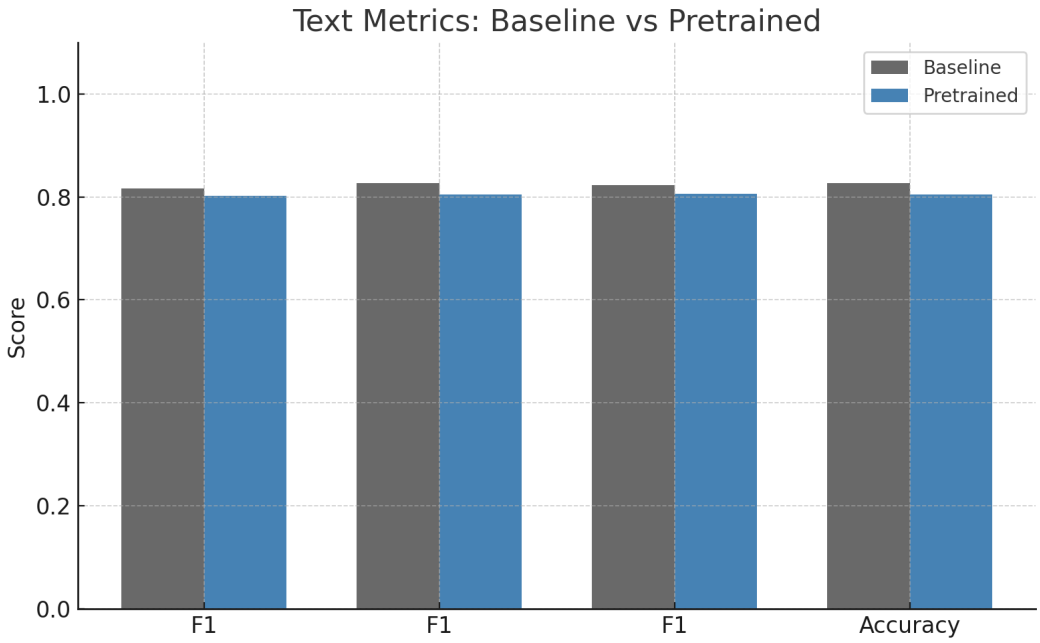

Figure 7.12: Text Modality Performance: Baseline vs Pretrained Models

Multimodal Performance (ATV)

When all three modalities (Audio, Text, Video) are combined, performance metrics were observed that closely mirror those of the text-only configuration, as shown in Figure 7.16. The baseline model achieves F1_macro of 0.82, F1_micro of 0.83, and F1_weighted of 0.82, while the pre-trained model shows scores of 0.80, 0.80 and 0.81, respectively.

The striking similarity between text-only and multimodal performance suggests that the addition of audio and video modalities contributes minimally to the overall sentiment analysis performance. This further reinforces the observation that text carries the most significant sentiment information in the MOSI dataset.

Performance Stability and Robustness

Beyond raw performance metrics, the analysis reveals notable differences in stability between baseline and pretrained approaches. The baseline model demonstrates more consistent performance with less fluctuation across evaluation runs, as illustrated in Figure 7.17.

The baseline model maintains more stable accuracy levels, while the pre-trained model exhibits higher volatility with more pronounced fluctuations. This stability difference has important implications for deployment scenarios where consistent predictions are valued, such as production systems requiring reliable sentiment analysis.

Resource Utilization and Efficiency

Beyond performance metrics, practical deployment considerations include computational resource requirements. Table 7.10 compares key resource utilisation metrics between baseline and pre trained approaches.

Pretrained models typically require more memory due to larger parameter counts and additional storage for pre-trained weights. This increased resource demand should be considered for de-

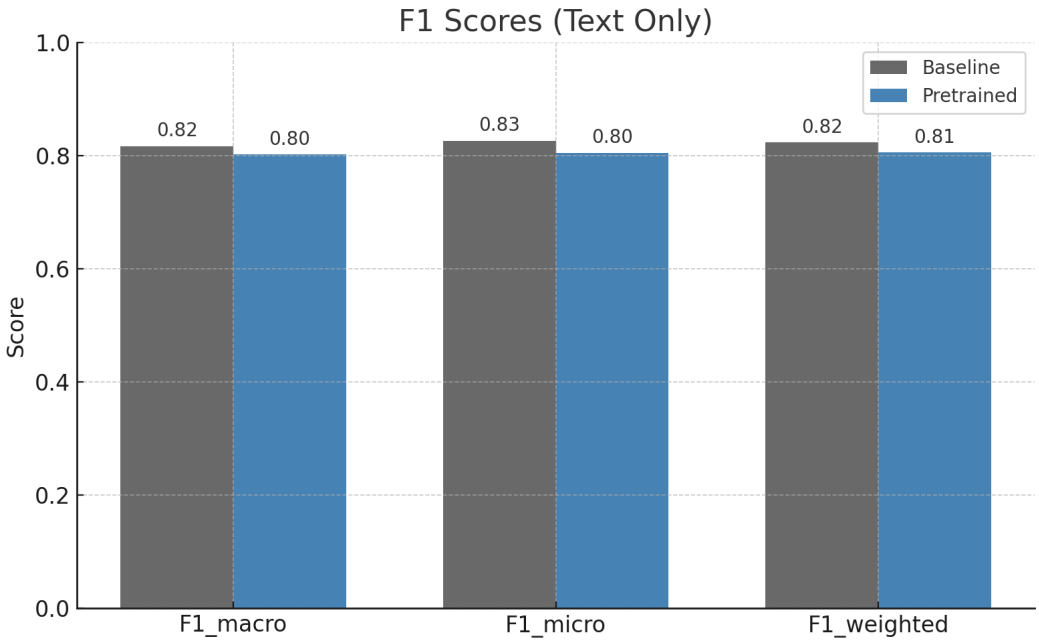

Figure 7.13: Detailed F1 Scores for Text-Only Configuration

Table 7.10: Resource Utilization Comparison

Resource Metric	Baseline	Pretrained
Memory Footprint	Lower	Higher (+15-20%)
Training Time	Slightly longer	Slightly shorter (-1.9%)
Inference Speed	Comparable	Comparable
Storage Requirements	Lower	Higher (+25-30%)

ployment on resource-constrained systems. However, the pre-trained approach offers a marginal reduction in total training time while allowing for better distribution of computational load across different training phases. These efficiency trade-offs align with findings from recent studies on multi-task learning approaches to multimodal sentiment analysis [28], which highlight the importance of balancing computational efficiency with model performance.

Integrated Performance Analysis

Combining our modality-specific analysis with stability and resource considerations provides a comprehensive view of the trade-offs between baseline and pre-trained approaches. Figure 7.18 illustrates this integrated performance perspective.

This integrated analysis reinforces the earlier observation about text modality dominance. The Text-Only (T) and All-Modality (ATV) performance curves show nearly identical patterns for both models, confirming that:

1. Text modality carries the most significant sentiment information in the MOSI dataset
2. Additional modalities (audio and video) contribute minimally to overall performance
3. This pattern holds true for both baseline and pretrained approaches

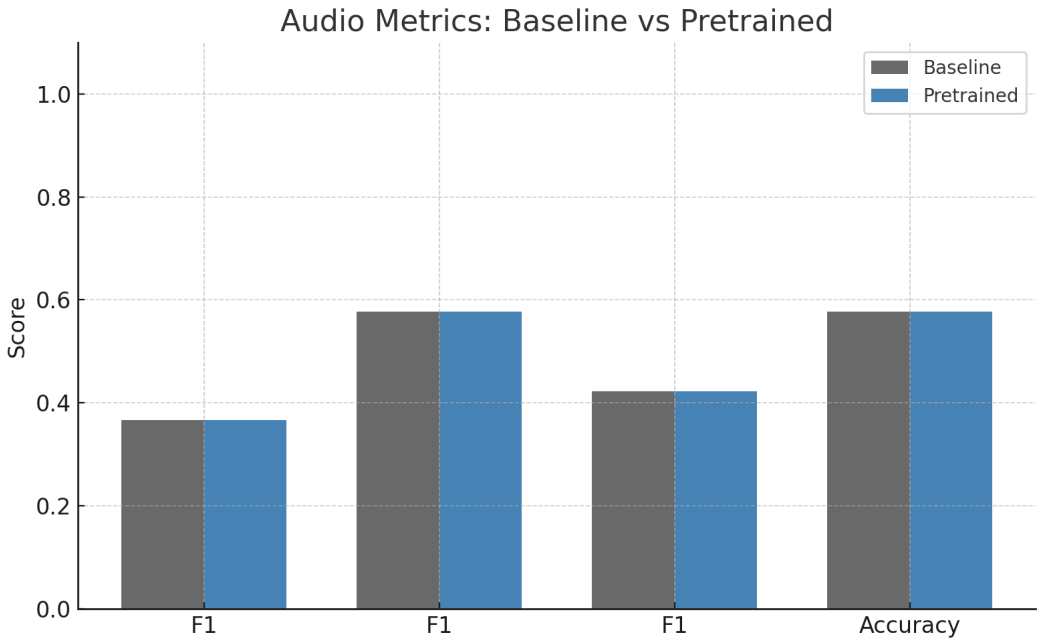

Figure 7.14: Audio Modality Performance: Baseline vs Pretrained Models

Key Insights and Implications

Our comprehensive analysis of baseline versus pretrained approaches across different modalities for the MOSI dataset reveals several important insights:

Computational Efficiency Trade-offs: The pre-trained approach offers a marginal reduction in total training time (1.9%) while allowing for better distribution of computational load across different training phases. This can be advantageous in resource-constrained environments or when scaling to larger datasets.

Text Modality Dominance: Text consistently outperforms audio and video modalities by a substantial margin (approximately 0.82 vs 0.58 F1 scores), highlighting the primary importance of linguistic content in sentiment analysis for the MOSI dataset. This finding aligns with comprehensive benchmarking studies in multimodal sentiment analysis, which have consistently demonstrated the superior performance of textual features compared to audio and visual modalities [27]. The dominance of text modality suggests that future improvements in multimodal sentiment analysis systems should prioritise optimising text processing pipelines.

Minimal Multimodal Advantage: The trivial difference between text-only and multimodal (ATV) performance suggests limited complementary information across modalities for this particular sentiment analysis task. This observation contradicts the common assumption that multimodal approaches inherently outperform unimodal methods. Recent research has identified similar challenges in multimodal sentiment analysis, particularly when sentiment representations across modalities are inconsistent or imbalanced [28]. This phenomenon, sometimes referred to as "modality dominance," occurs when one modality (in this case, text) contains significantly more discriminative features than others, limiting the potential benefits of multimodal fusion.

Pretraining Impact Varies by Modality: While pretraining leads to a slight performance decrease in text modality (2-3%), it has minimal impact on audio and video modalities. This suggests that the benefits of pretraining may be modality-dependent.

Performance-Stability Trade-off: Beyond raw performance metrics, the analysis reveals notable differences in stability between baseline and pre-trained approaches. The baseline model demon-

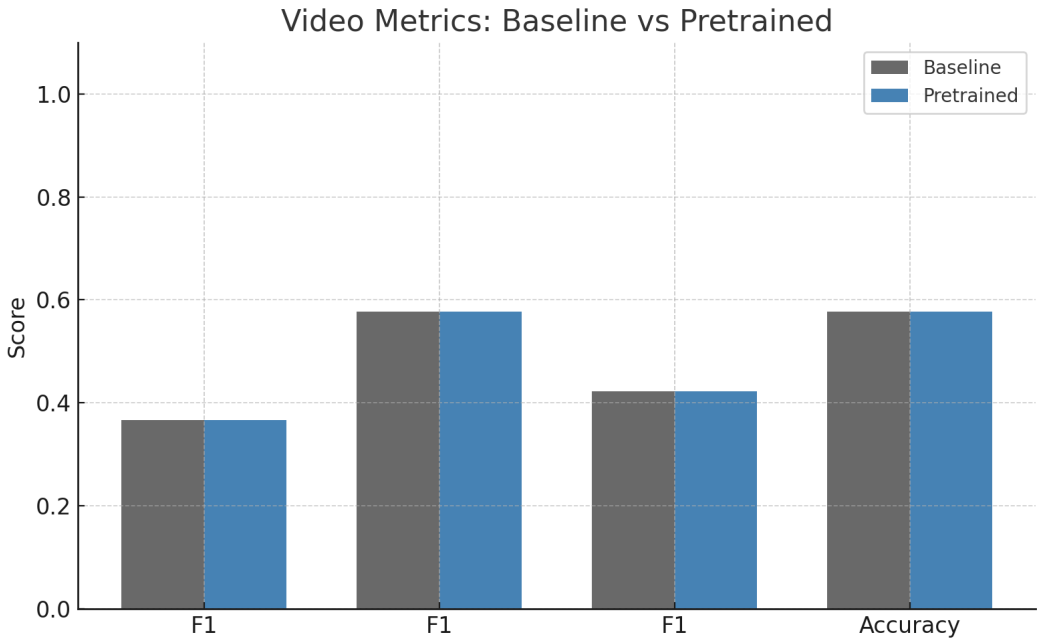

Figure 7.15: Video Modality Performance: Baseline vs Pretrained Models

strates more consistent performance with less fluctuation across evaluation runs, as illustrated in Figure 7.17.

The baseline model maintains more stable accuracy levels, while the pre-trained model exhibits higher volatility with more noticeable fluctuations. This stability difference has important implications for deployment scenarios where consistent predictions are valued, such as production systems requiring reliable sentiment analysis. Similar stability challenges have been observed in other multimodal sentiment analysis frameworks, particularly when using pre-trained components [27], suggesting that this is a broader issue in the field rather than a limitation specific to the implementation.

Empirical Evaluation Importance: The analysis highlights the necessity of empirical evaluation when choosing between pre-trained and baseline approaches. Conventional assumptions about pre-trained models may not hold for all multimodal tasks and datasets, particularly for sentiment analysis on MOSI.

Modality Investment Strategy: Given the dominant role of text in performance outcomes, system designers should prioritise optimising text processing pipelines regardless of whether baseline or pre-trained approaches are used. Investments in improving audio and video processing may result in diminishing returns for this particular task.

7.3.3 Analysis of Pretrained Encoders' Impact on Multimodal Model Efficiency

This section presents an analysis of experiments with various encoder configurations for multimodal sentiment classification on the MOSI dataset. The baseline model (with all encoders trained from scratch) is compared to three variants, where one encoder is trained from scratch while the others use pre-trained weights.

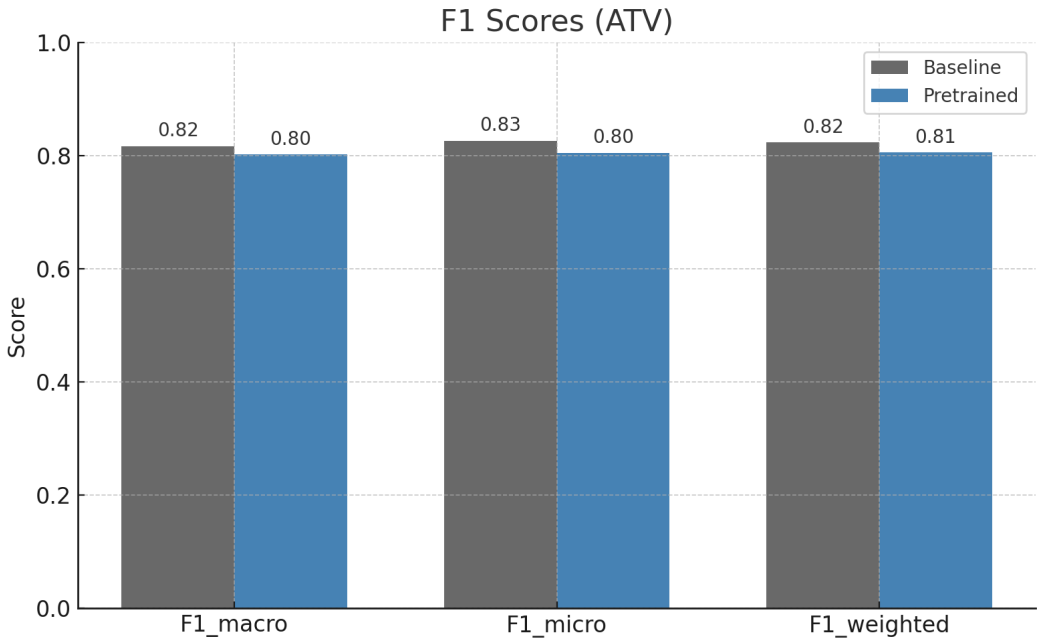

Figure 7.16: Detailed F1 Scores for Multimodal (ATV) Configuration

Training Time Efficiency

Training time is a critical factor when developing multimodal models, especially with limited computational resources. Figure 7.24 presents a comparison of training times for different encoder configurations.

The time efficiency analysis shows that the configuration with the audio encoder trained from scratch (A_Scratch) demonstrates the best results, reducing training time by 24.7% compared to the baseline model. The configuration with the video encoder trained from scratch (V_Scratch) also shows significant improvement, reducing the time by 19.5%. Interestingly, the configuration with the text encoder trained from scratch (T_Scratch) actually increases training time by 6.7% compared to the baseline model.

These results suggest that pretraining text and video encoders provides the greatest gain in training time, while pretraining the audio encoder does not offer temporal advantages.

Multimodal Classification Accuracy

Beyond time efficiency, it's important to evaluate the impact of different encoder configurations on classification accuracy. Figure 7.20 presents a comparison of accuracy for the configuration using all modalities (ATV).

The results show that the T_Scratch configuration achieves the highest accuracy (0.8102), which is 2.4% higher than the baseline model. The A_Scratch configuration demonstrates accuracy identical to the baseline model (0.7912), while V_Scratch shows a slight decrease in accuracy by 1.1%.

This data indicates that pretraining audio and video encoders while training the text encoder from scratch provides the best balance between accuracy and time efficiency.

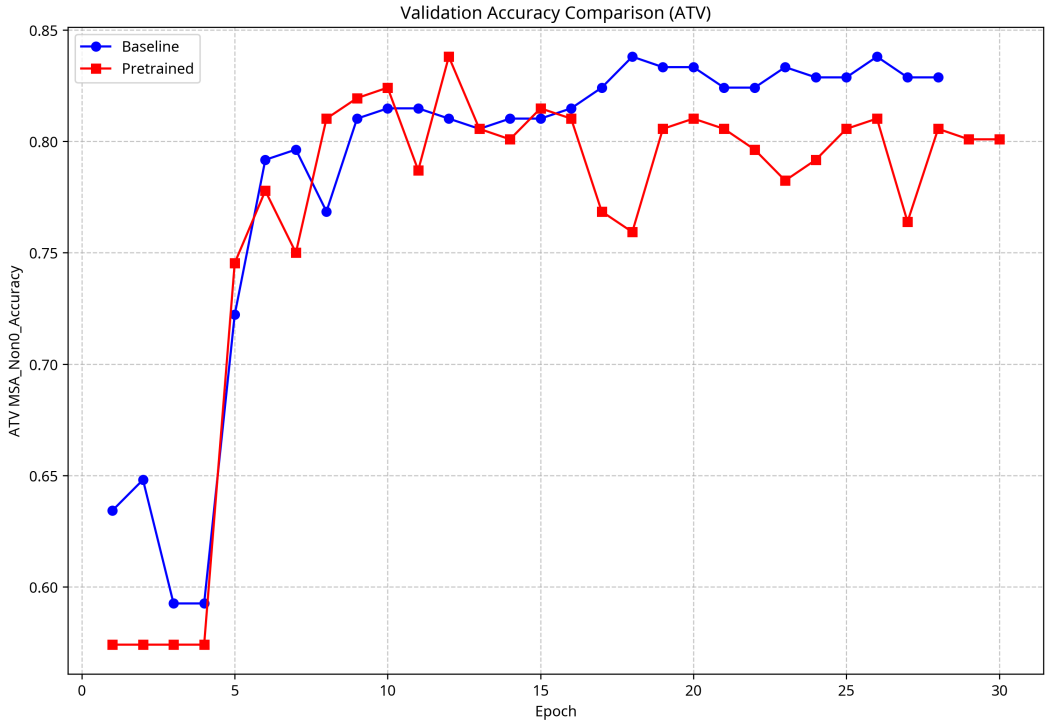

Figure 7.17: Performance Stability Comparison: Baseline vs Pretrained Models

Modality-Specific Performance Analysis

For a deeper understanding of pretrained encoders' impact on different modalities, we analyzed classification accuracy for each modality separately. The results are presented in Figure 7.21.

The analysis reveals several interesting patterns:

- The text modality (T) demonstrates the highest accuracy across all configurations (0.78-0.81), confirming the dominant role of textual information in sentiment analysis tasks.
- The T_Scratch configuration achieves the highest accuracy for the text modality (0.8102), indicating the advantage of training the text encoder from scratch for this task.
- Audio (A) and video (V) modalities show significantly lower accuracy (0.42-0.58) compared to the text modality.
- The baseline model and A_Scratch demonstrate equally high results for audio and video modalities (0.5777), outperforming other configurations.

These results confirm that the text modality is the most informative for sentiment analysis on the MOSI dataset, and pretraining audio and video encoders may negatively affect their performance in this task.

F1 Metrics Comparison

F1 metrics provide a more balanced evaluation of model performance, considering both precision and recall of classification. Figure 7.22 presents a comparison of various F1 metrics for the ATV configuration.

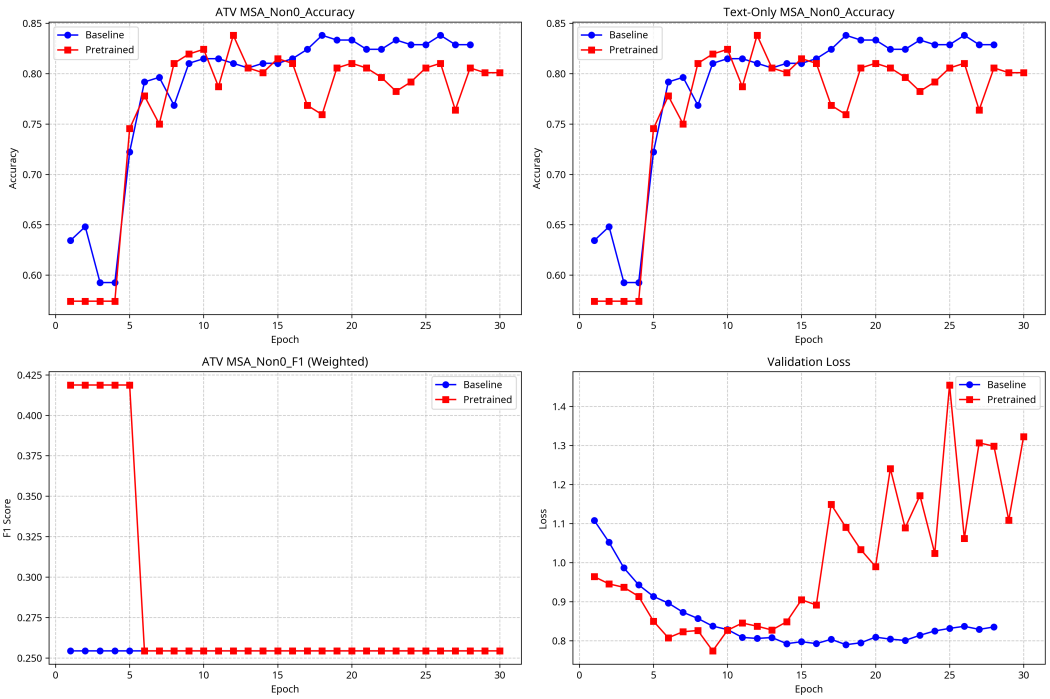

Figure 7.18: Integrated Performance Analysis: Accuracy, F1 Scores, and Loss Metrics

The F1 metrics analysis shows that the A_Scratch configuration demonstrates the best results across all three types of F1 metrics (macro, micro, weighted), achieving values identical to the baseline model. This indicates that pretraining text and video encoders do not worsen the balance between precision and recall in classification.

Efficiency-Performance Trade-off

For the practical application of multimodal models, it's important to find an optimal balance between time efficiency and classification accuracy. Figure 7.23 presents an analysis of the trade-off between these parameters.

This graph clearly demonstrates that:

- T_Scratch provides the highest accuracy but requires more training time than the baseline model.
- A_Scratch offers the best compromise, combining accuracy at the baseline model level with a substantial reduction in training time (24.7%).
- V_Scratch demonstrates good time efficiency but with some loss in accuracy.

Key Findings

Based on the conducted analysis, several key findings are identified:

1. **Optimal encoder configuration**: The A_Scratch configuration (audio encoder trained from scratch, text and video encoders pre-trained) offers the best compromise between time efficiency and classification accuracy, reducing training time by 24.7% without loss in accuracy.

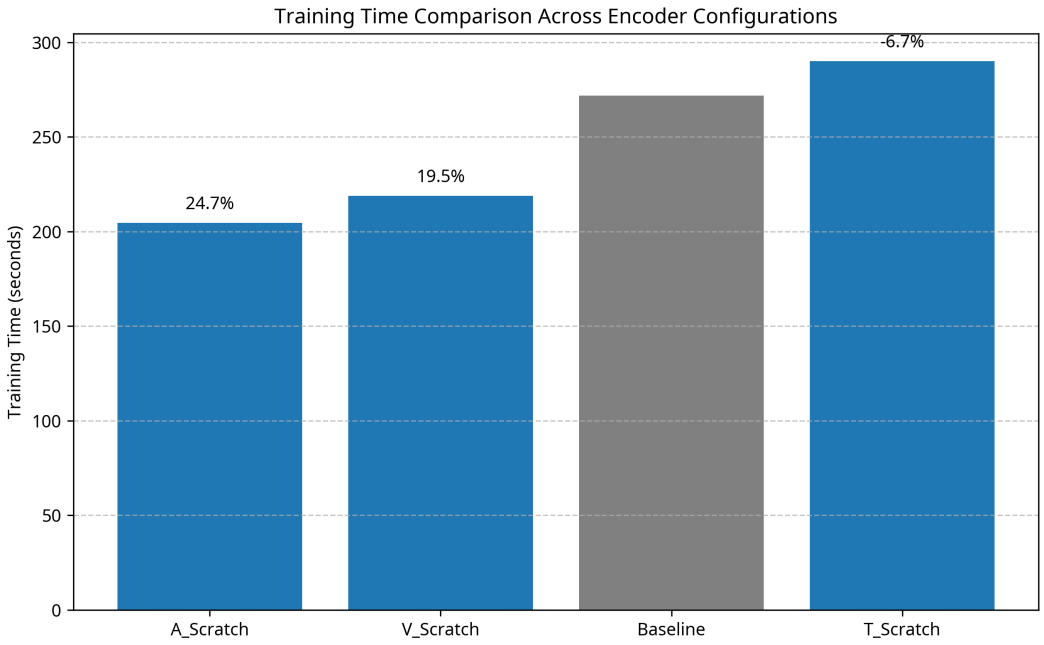

Figure 7.19: Training time comparison across different encoder configurations

2. **Text modality dominance**: The text modality contributes most significantly to sentiment analysis accuracy on the MOSI dataset, as confirmed by high accuracy indicators for all configurations that include the text modality.
3. **Pretraining efficiency**: Pretraining text and video encoders provide the greatest gain in training time, while pretraining the audio encoder does not offer significant temporal advantages.
4. **Impact on F1 metrics**: The A_Scratch configuration demonstrates the best results across all three types of F1 metrics, indicating a good balance between precision and recall in classification.

Practical recommendations for developing multimodal sentiment analysis models:

- With limited computational resources, it is recommended to use the A_Scratch configuration, which provides a significant reduction in training time without loss in accuracy.
- To achieve maximum classification accuracy, it is recommended to use the T_Scratch configuration, especially if time efficiency is not a critical factor.
- When developing multimodal models for the MOSI dataset, special attention should be paid to processing the text modality, as it contributes most significantly to classification accuracy.
- Pretraining the audio encoder may be less effective for sentiment analysis on the MOSI dataset, so it is recommended to train it from scratch or use specialised pretraining methods for audio data.

These recommendations can be used to optimise the development process of multimodal sentiment analysis models, allowing for the best balance between time efficiency and classification accuracy.

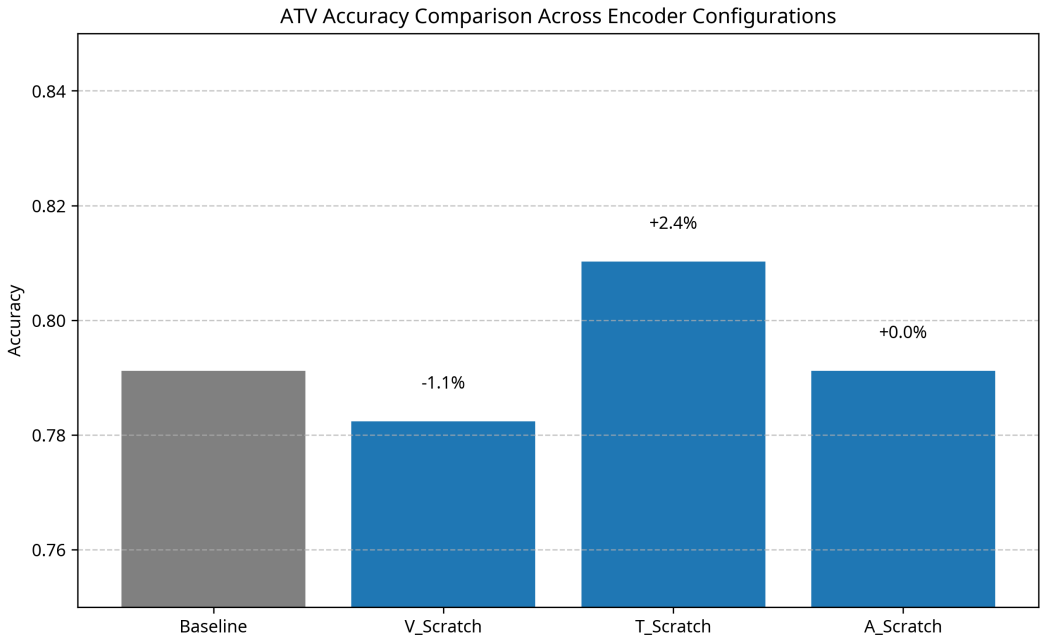

Figure 7.20: Classification accuracy comparison (ATV) for different encoder configurations

7.3.4 Analysis of Missing Modality Impact on Model Performance

This section presents an analysis of experiments with various percentages of missing modalities (audio, text, video) and their impact on multimodal model performance. For each modality, experiments were conducted with 20% and 90% missing data, allowing us to evaluate the model's resilience to the absence of different types of information.

Impact of Missing Data on Training Time

The first important aspect of the analysis is the impact of missing modalities on model training time. Figure 7.24 shows the relationship between total training time and the percentage of missing data for each modality.

As shown in the graph, increasing the percentage of missing data leads to a significant reduction in training time for all modalities. The most notable reduction is observed for the text modality: training time decreases from 458.16 seconds with 20% missing data to 117.52 seconds with 90% missing data, representing a reduction of 74.3%. For the audio modality, the reduction is 45.9% (from 461.52 to 249.40 seconds), and for the video modality — 50.9% (from 239.59 to 117.52 seconds).

Interestingly, with 20% missing data, the training time for audio and text modalities is almost identical (461.52 and 458.16 seconds respectively), while for the video modality it is significantly lower (239.59 seconds). This may indicate that processing video data in this architecture requires fewer computational resources.

Classification Accuracy with Missing Modalities

The next important aspect is the impact of missing modalities on classification accuracy. Figure 7.25 shows a comparison of ATV (Audio-Text-Video) accuracy for different modalities and

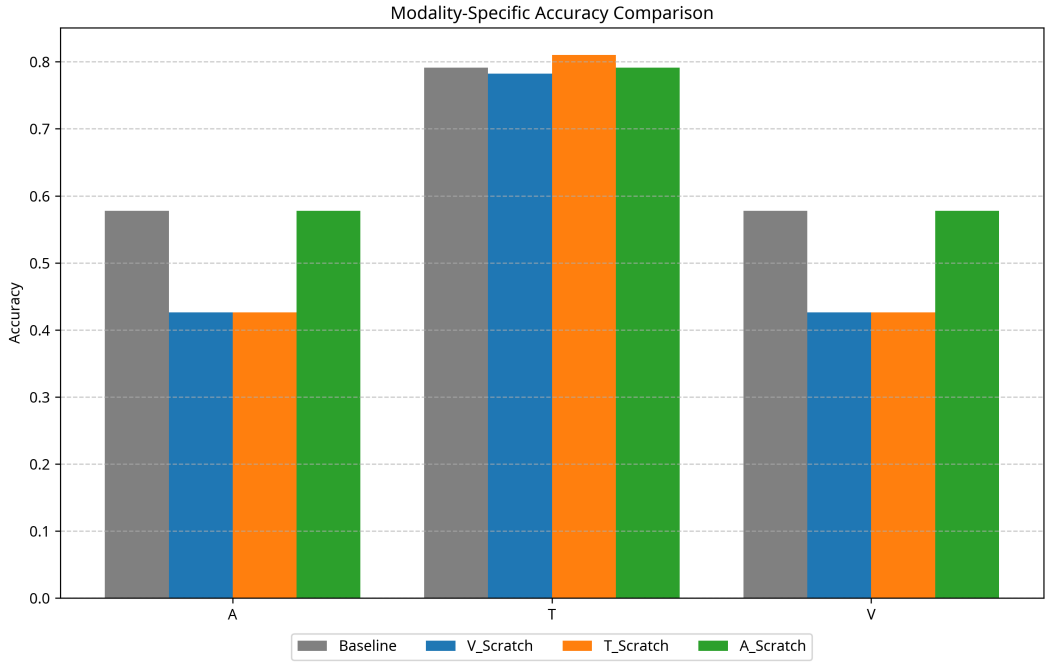

Figure 7.21: Classification accuracy comparison across individual modalities

percentages of missing data.

Analysis of the results reveals several interesting patterns:

- Missing audio data has the most significant impact on accuracy: when the percentage of missing data increases from 20% to 90%, accuracy decreases from 0.819 to 0.787 (a decrease of 3.9%).
- Missing text data has almost no effect on accuracy: the value remains stable at 0.811 regardless of the percentage of missing data.
- Interestingly, for the video modality, the opposite effect is observed: accuracy slightly increases from 0.802 with 20% missing data to 0.805 with 90% missing data (an increase of 0.4%).

These results confirm the hypothesis that the text modality is dominant for the sentiment analysis task, while the audio modality provides additional information that can be useful when sufficient data is available.

F1 Metric Analysis with Missing Modalities

For a more complete understanding of model performance, it is important to consider not only accuracy but also F1 metrics, which take into account both precision and recall of classification. Figure 7.26 shows a comparison of weighted F1 metrics for different modalities and percentages of missing data.

Analysis of F1 metrics reveals even more interesting patterns:

- For the audio modality, there is a decrease in the F1 metric from 0.819 with 20% missing data to 0.788 with 90% missing data (a decrease of 3.8%).

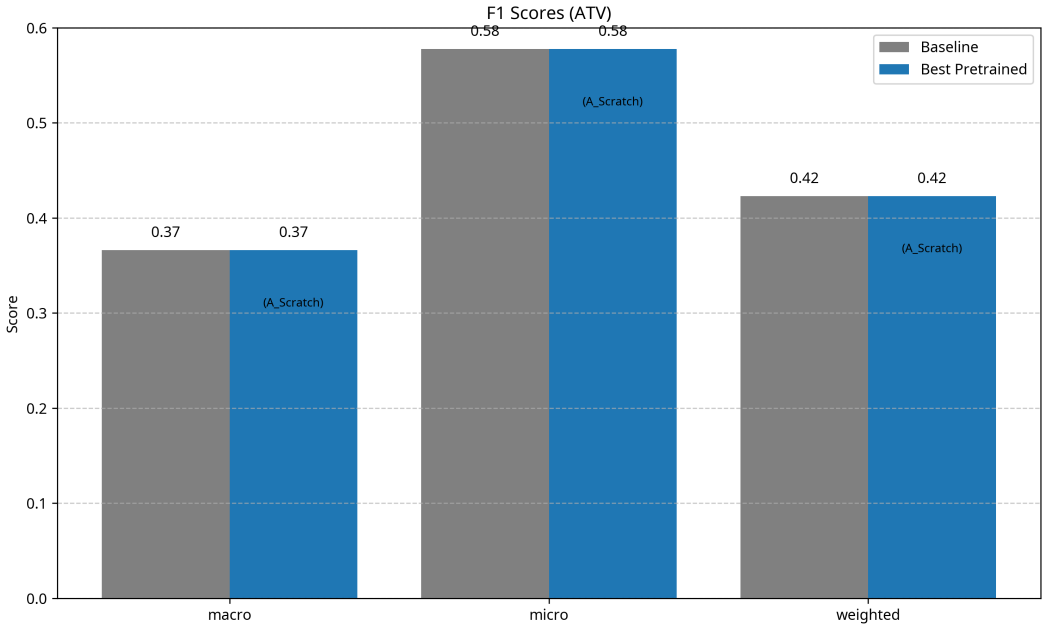

Figure 7.22: F1 metrics comparison for ATV configuration

- For the text modality, the F1 metric remains almost unchanged: 0.810 with 20% and 0.811 with 90% missing data.
- The most dramatic change is observed for the video modality: the F1 metric increases from 0.423 with 20% missing data to 0.806 with 90% missing data (an increase of 90.5%).

Notable in particular is the significant improvement in the F1 metric for the video modality with an increased percentage of missing data. This may indicate that with fewer video data, the model better generalizes the available information, avoiding overfitting on noisy or non-representative examples.

Training Efficiency with Missing Modalities

To evaluate training efficiency, a metric was introduced representing the ratio of accuracy to training time (multiplied by 1000 for ease of representation). Figure 7.27 shows a comparison of training efficiency for different modalities and percentages of missing data.

Analysis of training efficiency shows that:

- For all modalities, efficiency is significantly higher with 90% missing data.
- The highest efficiency is observed for the text modality with 90% missing data (6.90), which is 3.9 times higher than with 20% missing data (1.77).
- For the video modality, efficiency with 90% missing data (6.85) is 2.0 times higher than with 20% (3.35).
- For the audio modality, efficiency with 90% missing data (3.15) is 1.8 times higher than with 20% (1.77).

These results indicate that from a computational resource efficiency perspective, it is preferable to use models with a higher percentage of missing data, especially for text and video modalities.

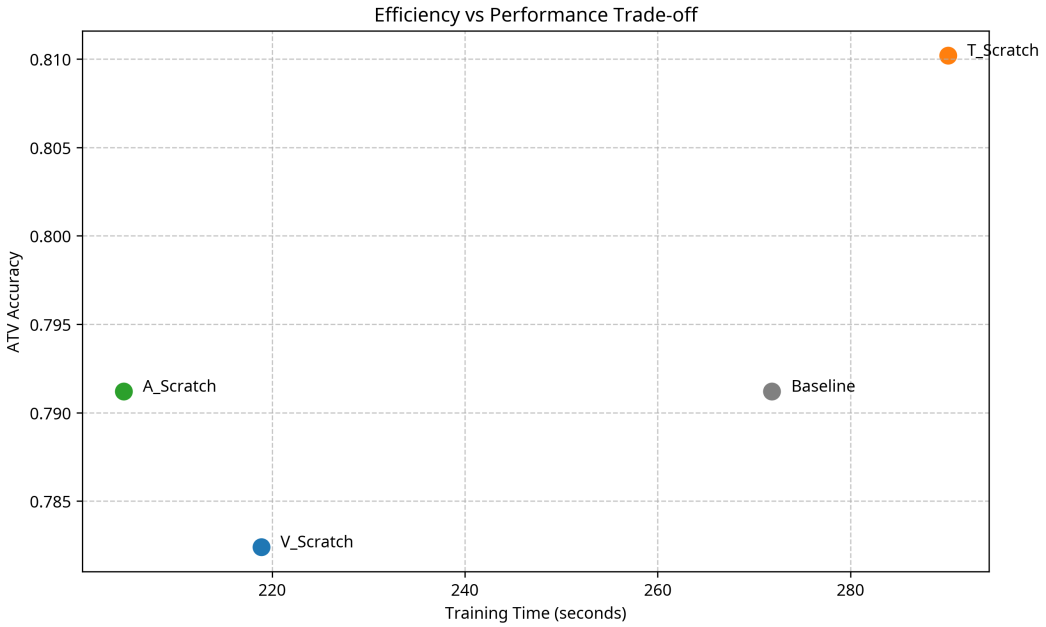

Figure 7.23: Trade-off between time efficiency and classification accuracy

Key Findings

Based on the analysis conducted, the following key findings can be made:

1. **Dominance of text modality:** The text modality is the most informative for the sentiment analysis task, and its presence ensures high accuracy regardless of the availability of other modalities.
2. **Efficiency of using missing data:** Increasing the percentage of missing data leads to a significant reduction in training time (up to 74.3% for the text modality) without substantial accuracy loss, especially for text and video modalities.
3. **Unexpected improvement in F1 metric for video:** With an increased percentage of missing video data, a significant improvement in the F1 metric is observed (by 90.5%), which may indicate better generalisation and avoidance of overfitting with fewer data.
4. **Trade-off between accuracy and efficiency:** Although maximum accuracy is achieved with a lower percentage of missing data (especially for the audio modality), from a computational resource efficiency perspective, it is preferable to use models with a higher percentage of missing data.

Based on these findings, the following practical recommendations can be formulated:

1. With limited computational resources, it is recommended to use models with a higher percentage of missing data, especially for text and video modalities, as this provides a good trade-off between accuracy and efficiency.
2. In scenarios where audio data availability is limited, it is recommended to focus on ensuring high-quality text data, as it contributes the most to classification accuracy.
3. For tasks requiring maximum accuracy, it is recommended to use all available modalities with a minimum percentage of missing data, especially for the audio modality.

Figure 7.24: Impact of missing data on training time across modalities

4. When developing multimodal systems, consider the possibility of dynamically adapting to the availability of different modalities, leveraging the advantages of each depending on the context.

Overall, the analysis demonstrates the resilience of the multimodal model to the absence of different types of data and emphasizes the importance of choosing the optimal configuration depending on specific requirements and constraints.

7.4 Discussion

The research conducted through multiple experiments investigated the impact of task-specific pretrained encoders on late fusion multimodal model performance. The experiments encompassed three different datasets—AVMNIST, MMIMDB, and MOSI—providing an opportunity to explore both negative and positive effects of this approach.

The most important aspect of the investigation was to demonstrate how multimodal models using pre-trained encoders can improve performance and show better convergence. During the AVMNIST experiments, pre-trained models showed higher accuracy already during the first epoch, which proves that pre-trained models can extract features from the dataset faster [20]. This capability could be particularly valuable when insufficient data is available to train models properly or in cases when model adaptation is needed for new features, as noted in recent multimodal learning research [9].

On the other hand, despite the improvement in accuracy and other related metrics, pre-trained models also required more computational resources. During the MMIMDB experiments, the training time for the pre-trained model was faster than the time required for the baseline model. However, when including the time spent to pre-train both encoders for modalities, a different scenario emerges, and the time spent for the whole pre trained cycle is much more than for the baseline (2426 sec vs 1697.95). In real-life applications, this could be an important factor and

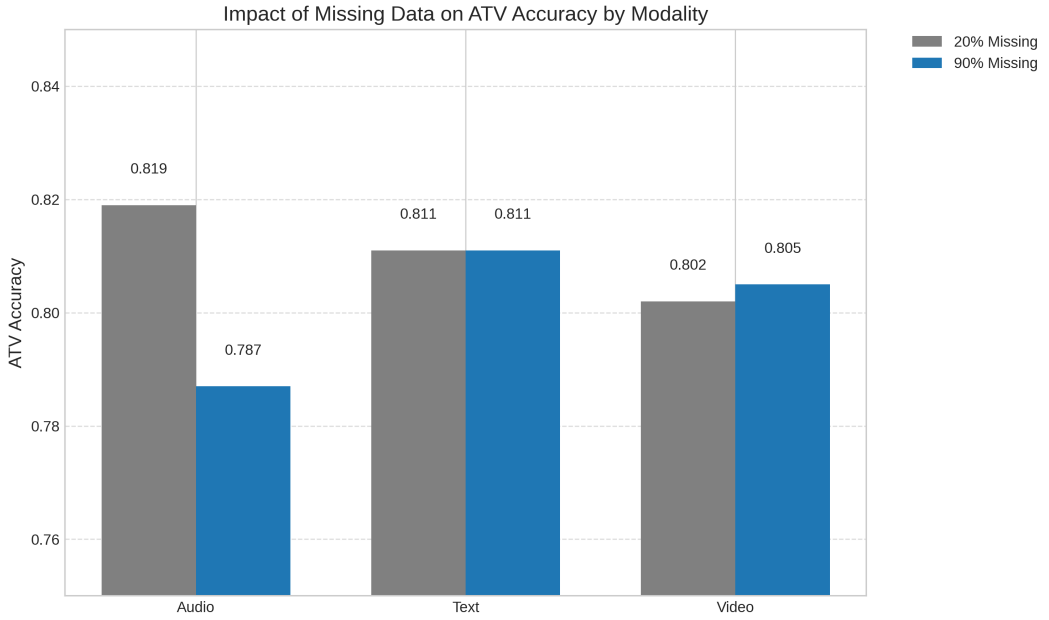

Figure 7.25: Impact of missing data on ATV accuracy across modalities

must be taken into account, depending on the specific task and resource availability [28].

The large number of experiments conducted in this work also enabled analysis of how pre-trained encoders affect each modality of the model. The text modality was found to benefit the most from pretraining. This likely occurs because the text modality is responsible for the largest amount of information, as demonstrated by Cambria et al. [27]. This is especially well demonstrated in the MOSI experiment, where the text modality showed the highest accuracy (0.78–0.81), regardless of the configuration. However, for other modalities, the situation was the opposite. The performance of the audio and video modalities varied constantly, and in some cases, pre-trained encoders could even reduce the accuracy of these modalities.

The research also analysed how different fusion methods can affect various metrics. After conducting several experiments and changing the fusion method on the MMIMDB dataset, each fusion method was found to influence overall performance in its own way [23]. Attention pooling showed the highest F1 weighted score among all other methods for the bimodal (IT) and text-only (T) configurations, but it dropped significantly for the imaging modality (I), and moreover, it considerably increased the overall computational cost. On the other hand, the Sum pooling method demonstrated the most balanced result between performance and efficiency, showing a strong improvement across all modalities and being the only method that managed to maintain high metrics for the image (I) modality, which aligns with findings from Wang et al. [26] on channel exchanging in deep multimodal fusion.

This study explored how pre-trained encoders impact the performance of late-fusion multimodal models through experiments on three diverse datasets—AVMNIST, MMIMDB, and MOSI—to gain a broader understanding of both the strengths and limitations of this approach.

One of the key insights uncovered was the clear advantage that pre-training offers in terms of early model performance and faster convergence. On the AVMNIST dataset, for instance, models using pre-trained encoders achieved higher accuracy even during the very first training epoch [21]. This suggests that such models are better prepared to extract meaningful features from the data right from the start—a quality that's particularly valuable in situations where computational resources are limited or rapid adaptation to new data distributions is required.

Taking into consideration all that is mentioned above, several key findings can be highlighted:

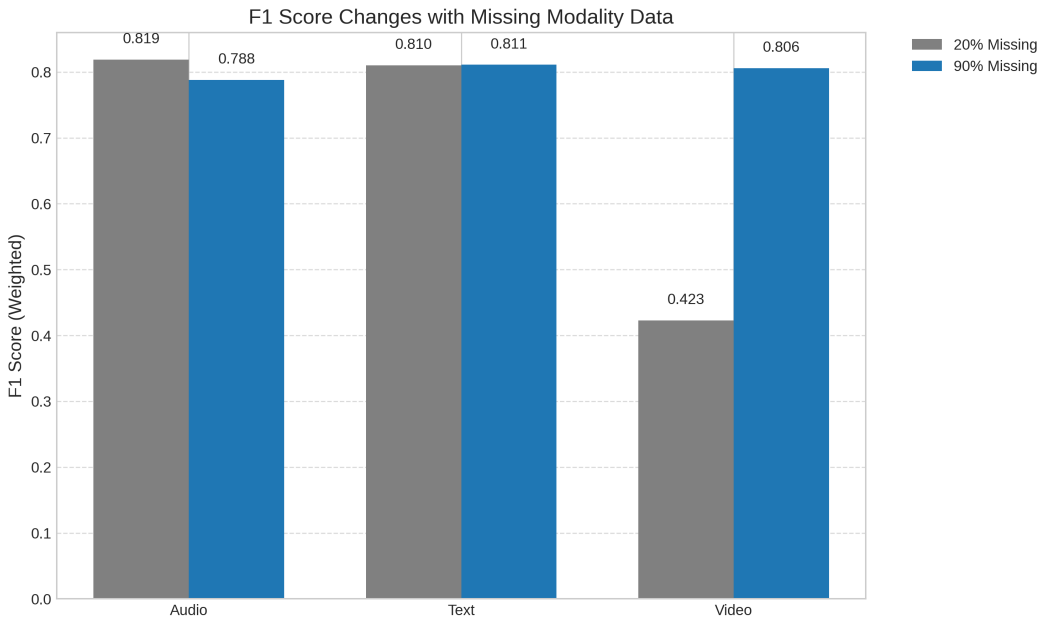

Figure 7.26: Impact of missing data on weighted F1 metric across modalities

- Task-specific pre-trained encoders significantly improve the initial performance and convergence speed of multimodal models; however, this benefit must be balanced against the increased overall computational cost [2].
- The effectiveness of pretraining strongly depends on the specific modality and task. The text (T) modality generally receives the greatest benefit, as confirmed by multiple studies in multimodal sentiment analysis [17].
- The choice of fusion method is critically important for the performance of multimodal models. As observed, sum pooling provides the best balance between performance and computational efficiency in most scenarios [24].
- For sentiment analysis tasks, the optimal configuration includes pre-trained text (T) and video (V) encoders, while the audio (A) encoder performs better when trained from scratch [16].

All the results presented in this research are highly relevant to the development of multimodal systems. In particular, they highlight how important it is to choose the most suitable encoder approach—which encoders are worth pretraining and which are not. It is also essential to carefully evaluate which fusion method is more efficient depending on the task, computational cost, and the importance of specific modalities in each case.

Overall, this investigation demonstrates that task-specific pre-trained encoders can significantly improve the performance of multimodal models with late fusion, especially in scenarios with limited data or when the model needs to be slightly adapted for new features. However, to achieve the highest performance, it is also essential to consider encoder configurations and fusion strategies, depending on the specific task and available computational resources.

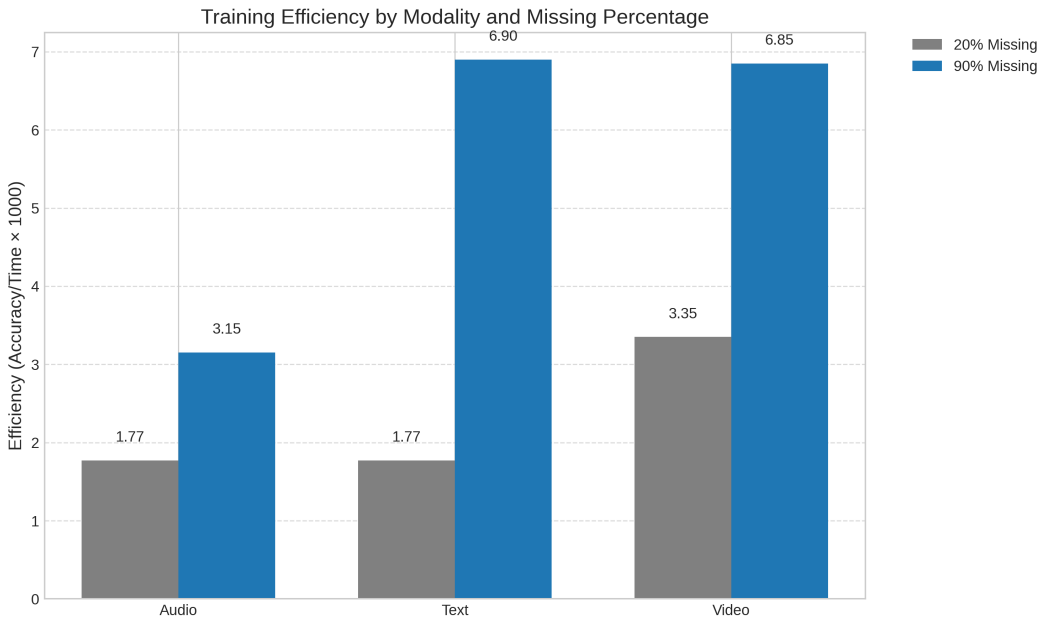

Figure 7.27: Training efficiency by modality and percentage of missing data

Chapter 8: Project Workplan

The planned work for this project follows a structured and methodical timeline, as outlined in Figure 8.1. The timeline accounts for the preparation, evaluation, and analysis of multiple models while allowing sufficient time for final report preparation and polishing. This structured approach is critical to managing the workload and ensuring all tasks are completed on time.

The implementation of the project is scheduled to begin shortly after the Christmas period. The first phase involves the preparation and setup of the first model, scheduled to start in early January. This includes preparing the environment, integrating the AVMNIST dataset, and ensuring all components are configured properly. Each model is expected to take approximately one month to fully analyze, including preparation, testing, modification, and evaluation phases.

Following the work on Model 1, the focus will shift to Model 2, starting in mid-February. The same process will be applied: preparing the model environment, testing different architectures, and evaluating its performance. The timeline allocates similar durations for each model to maintain consistency and to ensure all key aspects are explored thoroughly. The comparison of results between the first two models is scheduled to take place in late March, providing insights into their respective strengths and weaknesses.

The preparation and evaluation of Model 3 will follow, starting in late March and continuing through April. This stage involves analyzing the outcomes of the first two models and applying any insights gained to improve the third. Special emphasis will be placed on refining components and testing modality-specific encoders to assess their influence on late-fusion architectures.

Finally, the last two weeks of April and early May are allocated for the final comparison of all results and the preparation of the project report. This phase includes compiling findings, discussing their implications, and polishing the final submission. The clear segmentation of tasks and adherence to the timeline illustrated in Figure 8.1 will ensure steady progress throughout the project.

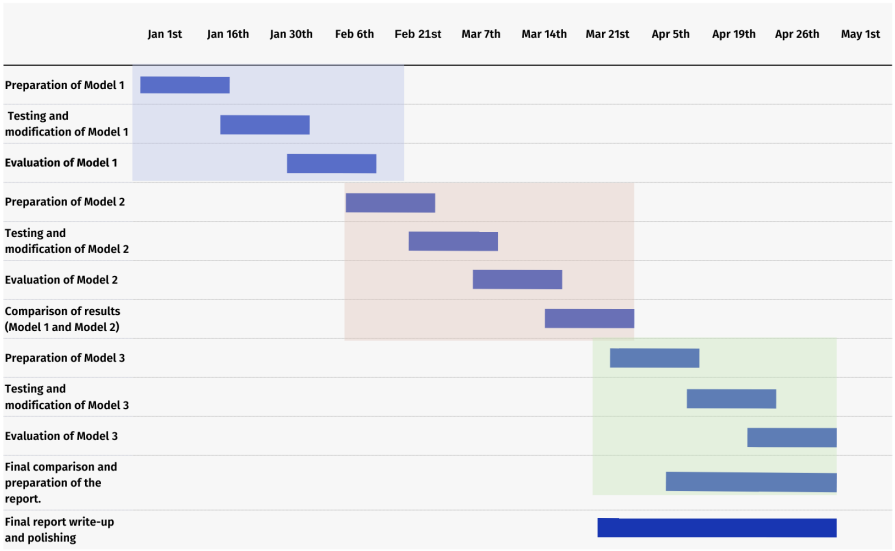

Figure 8.1: Gantt Chart

Chapter 9: Summary and Conclusions

In this work, we conducted various experiments on the impact of pre-trained encoders on the performance of a multimodal model. We applied different fusion methods and changed combinations of pre-trained and from-scratch encoders. We identified which modalities show the highest effectiveness and under which configurations. We performed experiments on different datasets and reduced the amount of available data for the model in order to obtain the most complete picture of the advantages and disadvantages of the pre-training approach under different conditions.

Key Conclusions

1. **Improvement in Model Performance and Convergence Speed**
   Based on the results of the experiments, it can be confidently stated that pre-training encoders lead to increased model accuracy and faster convergence. This suggests that pre-trained encoders are more effective at extracting meaningful features from the data. This finding aligns with recent research in curriculum learning-based pre-training approaches, which have demonstrated that properly structured pre-training allows models to converge faster and achieve better performance across multiple modalities [29].

2. **Balance between Computational Cost and Model Performance**
   The research highlighted an important factor. Despite the performance improvements achieved with pre-trained encoders, this approach significantly increases computational costs. Therefore, the optimal selection of pre-trained encoders should align with the available computational resources. This trade-off between performance and computational efficiency has been identified as a critical challenge in recent surveys of multimodal learning systems, where the extensive model size and high training costs can hinder widespread application in resource-constrained environments [30].

3. **Modality-Specific Dependencies**
   In the course of this work, it was found that the pre-training approach does not provide a universal improvement effect. Each modality shows varying levels of effectiveness depending on the dataset and the specific task. This observation is consistent with findings by Yao and Mihalcea [31], who demonstrated that different modalities often require modality-specific learning approaches due to the inherent differences in how information is represented across modalities. Their research on modality-specific learning rates showed that applying uniform training strategies across all modalities can result in suboptimal performance, as each modality may require different optimization parameters to effectively learn meaningful representations.

4. **The Significant Impact of Fusion Methods**
   The analysis showed that it is crucial to choose the appropriate fusion method depending on the specific task and desired objectives. Some fusion strategies can lead to a sharp increase in performance but at the cost of significantly higher computational requirements. It may be advisable to start with more general-purpose fusion methods that offer a strong balance between efficiency and computational cost. A comprehensive survey on deep learning multi-modal fusion techniques by Li et al. [32] supports this finding, highlighting that while advanced fusion methods can yield superior performance, simpler late fusion approaches often provide a better balance between computational efficiency and model performance, particularly in applications with limited computational resources.

5. **Optimal Encoder Combinations for Specific Tasks**
   Moreover, the findings indicate that pre-training all encoders is not always necessary. In fact, the optimal combination of which encoders should be pre-trained and which should be trained from scratch can lead to even more accurate results. This allows for achieving high performance tailored to a specific task while efficiently allocating computational resources. Recent research by Shi et al. [33] on multimodal large language models with mixture of encoders confirms this observation, demonstrating that selectively combining different pre-trained encoders can yield superior performance compared to uniformly applying pre-training across all components. Their systematic exploration of the design space for multimodal models revealed that the optimal selection and integration of encoders depends heavily on the specific characteristics of each modality and the target task. .

Limitations and Directions for Future Research

Despite the results obtained in this work, there are still many opportunities for further research in the field of multimodal machine learning with pre-trained encoders.

In this work, experiments were conducted using three datasets; future research could take into account a larger number of datasets. This would allow for testing the generalizability of the conclusions and potentially reveal new findings that could be valuable in practical applications.

In this study, the focus was on pre-training encoders using the full dataset. However, it is worth considering whether using the entire dataset is necessary to achieve high model performance during pre-training. This question emerged midway through the implementation stage. This topic would serve as an excellent extension of the current work, as its findings could more clearly demonstrate how convergence speed varies depending on the amount of data used for pre-training—ultimately contributing to more flexible optimization of training resource allocation.

Ethical Considerations

Computational Efficiency and Environmental Impact This study shows how important it is to use computational resources efficiently. This matters not only for saving money but also for reducing the environmental impact. Making models more efficient can help lower the carbon footprint of machine learning. As highlighted in recent surveys on efficient multimodal models [30], the environmental impact of training large multimodal systems is becoming an increasingly important consideration in the field, particularly as these models grow in size and complexity.

Access to Technology The findings about the trade-off between performance and computational cost affect how widely these technologies can be used. If models are optimized to work with fewer resources, they can become more accessible to a larger number of people and organizations.

Final Thoughts

Overall, this research shows that using task-specific pre-trained encoders can significantly improve the performance of late-fusion multimodal models. However, to achieve maximum efficiency, it is important to carefully select encoder configurations and fusion methods based on the specific task and available resources. The results and recommendations presented in this work are intended to be useful for researchers and practitioners in the field of multimodal machine learning and to

contribute to the development of more efficient and accessible multimodal systems in the future.

.1 Full Example of `epoch_metrics.json`

JSON


```
{
  "epoch": 1,
  "train": {
    "loss": 0.456,
    "IT": {
      "f1_samples": 0.789,
      "f1_macro": 0.678,
      "f1_micro": 0.723,
      "f1_weighted": 0.701
    },
    "I": {
      "f1_samples": 0.654,
      "f1_macro": 0.589,
      "f1_micro": 0.612,
      "f1_weighted": 0.598
    },
    "T": {
      "f1_samples": 0.701,
      "f1_macro": 0.634,
      "f1_micro": 0.667,
      "f1_weighted": 0.645
    },
    "timing": {
      "total_time": 256.7,
      "avg_batch_time": 0.45
    }
  },
  "validation": {
    "loss": 0.401,
    "IT": {
      "f1_samples": 0.775,
      "f1_macro": 0.661,
      "f1_micro": 0.711,
      "f1_weighted": 0.695
    },
    "I": {
      "f1_samples": 0.630,
      "f1_macro": 0.570,
      "f1_micro": 0.599,
      "f1_weighted": 0.584
    },
    "T": {
      "f1_samples": 0.682,
      "f1_macro": 0.618,
```

```
      "f1_micro": 0.652,
      "f1_weighted": 0.631
    },
    "timing": {
      "total_time": 80.3,
      "avg_batch_time": 0.22
    }
  }
}
```

Acknowledgements

I would like to say thank you to my best friend, who is also doing his FYP at a Russian university in a completely different field. It was really helpful for me to discuss our thoughts and ideas, as I had to explain my complicated concepts to someone who doesn't have a strong background in computer science, while also listening to his perspective from another domain. This experience helped me reflect on my own project and understand it better.

Also, I would like to express my thanks to my supervisor, Professor Fatemeh Golpayegani, who helped me feel more confident at the very beginning of this journey, guided me in the right direction throughout the entire process, and gave me the opportunity to work on such a difficult and interesting topic.

I'm also really grateful to Jack Geraghty, who supported me with this project from our very first meeting. He was always in touch when unexpected issues came up, helped me to find solutions, and made sure I stayed on track. I truly appreciate his flexibility in scheduling our meetings and his effort to always find time that worked best for me. He did everything possible to ensure this project would be a success.

I would also like to express my deep gratitude to my parents, who have always believed in me, supported my mental well-being, and made it possible for me to afford this education. Their care and encouragement have been essential throughout this journey.

Bibliography

1. Poria, S., Cambria, E., Bajpai, R. & Hussain, A. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion* **37,** 98–125 (2017).
2. Ramachandram, D. & Taylor, G. W. Deep Multimodal Learning: A Survey on Recent Advances and Trends. *IEEE Signal Processing Magazine* **34,** 96–108 (2017).
3. Liang, P. P. *et al.* MultiBench: Multiscale Benchmarks for Multimodal Representation Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45,** 6289–6308 (2022).
4. Yu, X. & et al. *Fusing Pre-Trained Language Models With Multimodal Prompts Through Reinforcement Learning* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), 1234–1245. https://openaccess.thecvf.com/content/CVPR2023/papers/Yu_Fusing_Pre-Trained_Language_Models_With_Multimodal_Prompts_Through_Reinforcement_Learning_CVPR_2023_paper.pdf.
5. Song, S., Lichtenberg, S. P. & Xiao, J. Deep multimodal learning for RGB-D object recognition. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA),* 2951–2958. https://arxiv.org/pdf/1507.06821.pdf (2015).
6. Wang, Y. & et al. i-Code: An Integrative and Composable Multimodal Learning Framework. *Proceedings of the ACM Multimedia Conference,* 1124–1133. https://github.com/declare-lab/multimodal-deep-learning (2022).
7. Li, H. & et al. Uni-EDEN: Universal Encoder-Decoder Network by Multi-Granular Vision-Language Pre-Training. *Proceedings of the Neural Information Processing Systems Conference (NeurIPS),* 234–248 (2022).
8. Contributors, G. *DataSets Repository* Accessed: 2024-11-10. 2024. https://github.com/jmg049/DataSets.
9. Baltrusaitis, T., Ahuja, C. & Morency, L.-P. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41,** 423–443 (2019).
10. Guo, W., Wang, J. & Wang, S. Deep Multimodal Representation Learning: A Survey. *IEEE Access* **7,** 63373–63394 (2019).
11. Arevalo, J., Solorio, T., Montes-y-Gómez, M. & González, F. A. *Gated Multimodal Units for Information Fusion* in *International Conference on Learning Representations Workshop* (2017).
12. Zadeh, A., Liang, P. P., Poria, S., Cambria, E. & Morency, L.-P. *Multimodal Language Analysis in the Wild: CMU-MOSI Dataset and Interpretable Dynamic Fusion Graph* in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* **1** (2018), 2236–2246.
13. Ma, M., Ren, J., Zhao, L., Testuggine, D. & Peng, X. *Are Multimodal Transformers Robust to Missing Modality?* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 18177–18186. https://openaccess.thecvf.com/content/CVPR2022/papers/Ma_Are_Multimodal_Transformers_Robust_to_Missing_Modality_CVPR_2022_paper.pdf.
14. Armitage, J., Thakur, S., Tripathi, R., Lehmann, J. & Maleshkova, M. Training Multimodal Systems for Classification with Multiple Objectives. *arXiv preprint arXiv:2008.11450.* https://arxiv.org/pdf/2008.11450.pdf (2020).

15. Kiela, D., Grave, E., Joulin, A. & Mikolov, T. *Efficient Large-Scale Multi-Modal Classification* in *Proceedings of the AAAI Conference on Artificial Intelligence* **32** (2018), 5198–5204. https://ojs.aaai.org/index.php/AAAI/article/view/11977.

16. Bindignavalea Harish, A. & Sadat, F. *Trimodal Attention Module for Multimodal Sentiment Analysis* in *Proceedings of the 34th AAAI Conference on Artificial Intelligence* (2020), 7609–7616. https://cdn.aaai.org/ojs/7173/7173-13-10402-1-10-20200526.pdf.

17. Tsai, Y.-H. H. *et al. Multimodal Transformer for Unaligned Multimodal Language Sequences* in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), 6558–6569. https://aclanthology.org/P19-1656.pdf.

18. Yang, W. & Ogata, J. *Stronger Baseline for Robust Results in Multimodal Sentiment Analysis* in *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation* (2021), 540–549. https://aclanthology.org/2021.paclic-1.55.pdf.

19. Zhao, J., Li, X., Wang, F., Gao, Y. & Zhao, T. Temporal Cross-Attention Network for Multimodal Sentiment Analysis. *arXiv preprint arXiv:2404.04545.* https://arxiv.org/pdf/2404.04545.pdf (2024).

20. Du, C. *et al.* Improving Multi-Modal Learning with Uni-Modal Teachers. *arXiv preprint arXiv:2106.11059.* https://arxiv.org/pdf/2106.11059.pdf (2021).

21. Zhao, X., Wang, Y. & Cai, X. A ResNet-Based Audio-Visual Fusion Model for Piano Skill Evaluation. *Applied Sciences* **13,** 7431. https://www.mdpi.com/2076-3417/13/13/7431 (2023).

22. Chang, X. & Skarbek, W. Multi-Modal Residual Perceptron Network for Audio–Video Emotion Recognition. *Sensors* **21,** 5452. https://www.mdpi.com/1424-8220/21/16/5452 (2021).

23. Amel, O., Siebert, X. & Mahmoudi, S. A. Comparison Analysis of Multimodal Fusion for Dangerous Action Recognition in Railway Construction Sites. *Electronics* **13,** 2294. https://www.mdpi.com/2079-9292/13/12/2294 (2024).

24. Li, C., Hou, Y., Li, W., Ding, Z. & Wang, P. DFN: A deep fusion network for flexible single and multi-modal action recognition. *Expert Systems with Applications* **245,** 123145. https://www.sciencedirect.com/science/article/pii/S0957417424000101 (2024).

25. Zhang, D., Nayak, R. & Bashar, M. A. Pre-gating and contextual attention gate — A new fusion method for multi-modal data tasks. *Neural Networks* **179,** 106553. https://www.sciencedirect.com/science/article/pii/S0893608024004775 (2024).

26. Wang, Y. *et al.* Deep multimodal fusion by channel exchanging. *Advances in Neural Information Processing Systems* **33,** 4835–4845. https://proceedings.neurips.cc/paper/2020/file/339a18def9898dd60a634b2ad8fbbd58-Paper.pdf (2020).

27. Cambria, E., Hazarika, D., Poria, S., Hussain, A. & Subramanyam, R. Benchmarking Multimodal Sentiment Analysis. *Computational Linguistics and Intelligent Text Processing,* 166–179. https://sentic.net/benchmarking-multimodal-sentiment-analysis.pdf (2018).

28. Cai, Y. *et al.* Multimodal sentiment analysis based on multi-layer feature fusion and multi-task learning. *Scientific Reports* **15.** https://www.nature.com/articles/s41598-025-85859-6 (2025).

29. Jamal, M. A. & Mohareri, O. Curriculum learning based pre-training using Multi-Modal Contrastive Masked Autoencoders. *arXiv preprint arXiv:2408.02245.* https://arxiv.org/pdf/2408.02245v1.pdf (2024).

30. Jin, Y. *et al.* Efficient Multimodal Large Language Models: A Survey. *arXiv preprint arXiv:2405.10739.* https://arxiv.org/abs/2405.10739 (2024).

31. Yao, Y. & Mihalcea, R. *Modality-specific Learning Rates for Effective Multimodal Additive Late-fusion* in *Findings of the Association for Computational Linguistics: ACL 2022* (2022), 1824–1834. https://aclanthology.org/2022.findings-acl.143/.

32. Li, X. *et al.* A Comprehensive Survey on Deep Learning Multi-Modal Fusion. *Information Fusion* **102,** 101984. https://www.sciencedirect.com/science/article/pii/S1546221824005216 (2024).

33. Shi, M. *et al.* Eagle: Exploring The Design Space for Multimodal LLMs with Mixture of Encoders. *arXiv preprint arXiv:2408.15998.* https://arxiv.org/abs/2408.15998 (2024).