
PhysioBind: Towards aligning Motion, IMU, EMG and Text

By

Md Shadman Aadeeb (0122320005)

Submitted in partial fulfilment of the requirements
of the degree of Master of Science in Computer Science and Engineering

May 11, 2026



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNITED INTERNATIONAL UNIVERSITY

Approval Certificate

This thesis, titled ‘**PhysioBind: Towards aligning Motion, IMU, EMG and Text**’, submitted by **Md Shadman Aadeeb**, Student ID: **0122320005**, has been accepted as **Satisfactory** in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on ___/___/2026.

Board of Examiners

1.

Prof. Dr. Swakkhar Shatabda
Professor
BRAC University

Supervisor

2.

Dr. Riasat Azim
Assistant Professor
United International University

Co-Supervisor

3.

Prof. Dr. Khondaker Abdullah Al Mamun
Director IRIIC and MSCSE
United International University

Head Examiner

4.

Prof. Dr. Muhammad Nomani Kabir
Professor and UG Program Coordinator, Dept. of CSE
United International University

Examiner-I

5.

Dr. Ohidujjaman
Assistant Professor
United International University

Examiner-II

6.

Dr. Suman Ahmmmed
Associate Professor
Head Dept. of CSE and Director, CDIP
United International University

Ex-Officio

Declaration

This is to certify that the work entitled ‘**PhysioBind: Towards aligning Motion, IMU, EMG and Text**’ is the outcome of the research carried out by me under the supervision of **Prof. Dr. Swakkhar Shatabda, Professor and Dr. Riasat Azim, Assistant Professor**’.

Md Shadman Aadeeb
MSCSE Program
Student ID: 0122320005
Dept. of Computer Science and Engineering
United International University
Dhaka, Bangladesh

In my capacity as supervisor of the candidate’s thesis, I certify that the above statements are true to the best of my knowledge.

Prof. Dr. Swakkhar Shatabda
Professor
Dept. of Computer Science and Engineering
BRAC University

Abstract

Multimodal alignment and multimodal generation are two important technologies that are widely used in the field of Artificial Intelligence (AI). Multimodal alignment allows AI models to identify similarities between two different modalities of data by transforming them into a shared representation with the same dimensional structure. Multimodal generation, on the other hand, enables an AI model to generate one type of data from another using this common representation. Together, these technologies provide a useful way to compare information from different modalities and to transfer or generate data from one modality to another.

In this research, multimodal alignment and multimodal generation were performed across four different modalities: three-dimensional keypoint sequences, inertial measurement unit signals, electromyography signals, and textual data. For this purpose, four encoders and four decoders were developed so that each modality could be mapped into a shared embedding space and then reconstructed or generated across modalities. To overcome the lack of fully paired datasets, an unpaired cross-modal training method was used. This approach led to the development of an Explainable AI-based system that can provide meaningful insights into a person’s movement and motion by connecting information from multiple sources.

The results showed that the proposed models performed well, especially in preserving motion-related information across the different modalities. With the help of the developed encoders and decoders, the models were able to match or surpass several previous benchmarks. The text generation model achieved a BERTScore of 40.57, which was higher than previous models. For EMG-to-pose generation, the 3D keypoint position RMSE was 0.0873, which is very close to the performance reported by the authors of the dataset paper. Similarly, for IMU-to-pose generation, the average rotation error was approximately 17.6589° , also close to the dataset authors’ result. Further analysis also revealed interesting patterns in how movement-related information is carried across different modalities. In addition, embedding arithmetic showed that two embeddings could be combined to produce mixed results, suggesting that the shared embedding space learned meaningful relationships between modalities. Overall, these findings show that the proposed multimodal framework can effectively capture, transfer, combine, and explain motion information across heterogeneous data sources.

Acknowledgements

First and foremost, I express my deepest gratitude to the Almighty Allah for granting me the strength, patience, and perseverance to complete this work.

This work would not have been possible without the support, encouragement, and guidance of many amazing people who stood by me throughout this journey. I am sincerely grateful to everyone who contributed to this work in one way or another.

I would like to express my deepest appreciation to my supervisors, **Professor Dr. Swakkhar Shatabda** and **Dr. Riasat Azim**, maestros in the fields of Bioinformatics and Artificial Intelligence, for their unwavering guidance, constant support, and exceptional patience throughout this research. They believed in me, stood by me during difficult phases, and were patient with me at every stage of this work. Their profound expertise, insightful feedback, and continuous encouragement helped me shape my ideas, overcome challenges, and grow as a researcher.

I am deeply grateful to **Professor Dr. Khondaker Abdullah-Al-Mamun** for his Research Methodology and Technical Writing course. His teachings guided me at every step of the research process, and throughout this work, I often recalled his lessons and found them immensely useful in designing, conducting, and clearly presenting my research.

I would like to sincerely thank **Professor Dr. Salekul Islam**, **Professor Dr. A. K. M. Muzahidul Islam**, **Professor Dr. Mohammad Nurul Huda**, and **Professor Dr. Dewan Md. Farid** for their excellent technical courses. The knowledge, insights, and skills I gained from these courses greatly enriched my learning and contributed meaningfully to my academic development.

I would like to extend my sincere appreciation to **Mr. Akib Zaman**, my university senior and former colleague, and a brilliant PhD student at MIT, for encouraging and advising me to enroll in this program. He generously shared his insights and experiences, which helped me make informed decisions throughout the process.

My heartfelt thanks go to my wife, **Dr. Nanziba Rahman Suzana**, for her unwavering support, patience, and encouragement throughout this journey. Her understanding and belief in me were a constant source of strength during challenging times.

Finally, I would like to thank my parents and my younger brother for their unconditional love, encouragement, and emotional support. Their faith in me has always been my greatest motivation.

Publication List

Journal Articles from This Thesis

1. **Aadeeb, Md. S.**, Azim, R., and Shatabda, S. “*PhysioBind: Multimodal Alignment and Generation of Motion, IMU, EMG, and Text with Explainable AI Support.*” [Submitted to **Neurocomputing** (Elsevier)]. (**Impact Factor: 6.0; SJR Quartile: Q1**).

Other Publications During the Master’s Program

1. Alam, M. R., Saha, S., Bostami, M. B., Islam, M. S., **Aadeeb, M. S.**, and Islam, A. M., “*A Survey on IoT-Driven Smart Parking Management Systems: Approaches, Limitations, and Future Research Agenda,*” **IEEE Access**, vol. 11, pp. 119523–119543, Oct. 2023. (**Impact Factor: 3.9; SJR Quartile: Q1**).
2. Islam, A. M., Rabbani, R. B., **Aadeeb, M. S.**, Zareei, M., Mansoor, N., and Zubaer, M. M., “*Simulation Tools for Cognitive Radio Networks: A Survey,*” **IEEE Access**, Feb. 2025. (**Impact Factor: 3.9; SJR Quartile: Q1**).
3. **Aadeeb, M. S.** and Farid, D. M., “*Advancing Generative AI for Visualizing Drug Addiction: Comparative Analysis and Dataset Innovation,*” in **Proceedings of the 27th International Conference on Computer and Information Technology (ICCIT 2024)**, IEEE, Dec. 2024, pp. 3505–3510.

Table of Contents

Table of Contents	v
List of Figures	vi
List of Tables	ix
List of Algorithms	x
1 Introduction	1
1.1 Research Overview and Context	1
1.2 Motivation and Research Significance	3
1.3 Research Objectives and Goals	3
1.4 Methodological Approach	4
1.5 Research Outcome	5
1.6 Structure and Organization of the Thesis	5
2 Background	6
2.1 Preliminaries	6
2.1.1 Multimodal data representation and embedding	6
2.1.2 Multimodal alignment and multimodal generation	8
2.1.3 Encoders and Decoders	10
2.1.4 Unpaired Cross-Modal Learning	11
2.1.5 Explainable AI	12
2.2 Modalities and Description	13
2.2.1 3D Keypoint Sequence or Pose Sequence	13
2.2.2 Electromyographic (EMG) Signals	14
2.2.3 Inertial Measurement Unit (IMU) Signals	16
2.2.4 Text Describing Motion	17
2.3 Literature Review	18
2.4 Gap Analysis	20
2.5 Summary	23

3	Methodology	24
3.1	Dataset	24
3.1.1	Dataset Collection and Description	24
3.1.2	Data Preprocessing	26
3.2	Model Architectures	29
3.2.1	Pose Encoder	29
3.2.2	Pose Decoder	30
3.2.3	IMU Encoder	31
3.2.4	IMU Decoder	33
3.2.5	EMG Encoder	34
3.2.6	EMG Decoder	35
3.2.7	Text Decoder	36
3.2.8	Text Encoder	38
3.3	Training Strategy	41
3.4	Explainability Analysis of Learned Embeddings	47
3.5	Graphical User Interface Development	49
3.6	Embedding Arithmetic Module	50
4	Results and Evaluation	52
4.1	Sequential Multimodal Sensor Representation and Reconstruction	52
4.1.1	Pose $\rightarrow$ Latent Embedding $\rightarrow$ Pose Reconstruction	52
4.1.2	EMG $\rightarrow$ Latent Embedding $\rightarrow$ EMG Reconstruction	54
4.1.3	IMU $\rightarrow$ Latent Embedding $\rightarrow$ IMU Reconstruction	55
4.1.4	Bidirectional Cyclic Cross-Modal Reconstruction (Pose and EMG)	56
4.1.5	Bidirectional Cyclic Cross-Modal Reconstruction (Pose and IMU)	59
4.1.6	Bidirectional Cyclic Cross-Modal Reconstruction (EMG and IMU)	62
4.2	Text Encoder and Decoder Evaluation	65
4.2.1	Text Decoder Evaluation	65
4.2.2	Text Encoder Evaluation	66
4.3	Benchmark Comparison with State-of-the-Art Methods	67
4.3.1	Comparative Performance on the Pose–Text HumanML3D Dataset	67
4.3.2	Comparative Performance on the Muscle in Action Dataset	67
4.3.3	Comparative Performance on the DIP-IMU Dataset	68
5	Conclusion	69
5.1	Summary	69
5.2	Limitations	69
5.3	Future Work	70
	References	74

List of Figures

1.1	Illustration of multimodal alignment for $n = 4$ modalities, where modality-specific encoders map data into a shared embedding space.	1
1.2	Example of multimodal generation, where an embedding produced by an encoder is decoded into other modalities.	2
1.3	Flow chart illustrating the methodological steps followed in this research. .	4
2.1	From the event of a person running, six different modalities of data (RGB image, thermal image, EMG signal, IMU signals, 3D keypoint sequence, and text) are collected using six different sensors or data collection tools (RGB camera, thermal camera, EMG sensor, IMU sensor, VIBE model, and pen and paper).	7
2.2	Six different modalities of data are converted into embeddings within a shared embedding space. Since the modalities correspond to the same event, their embeddings are located close to one another in the shared embedding space.	8
2.3	Embeddings from similar events aligned closed to one another while the embeddings from different events are positioned distantly	9
2.4	Comparison of EMG signal reconstruction from different modalities. EMG signals decoded from audio embeddings (left) show poor correspondence with the ground-truth EMG, whereas EMG signals decoded from IMU embeddings (right) closely resemble the actual EMG signals.	10
2.5	3D keypoint sequence represented using the SMPL format.	13
2.6	Pipeline illustrating the use of the VIBE model to estimate SMPL parameters from RGB video, followed by SMPL-based 3D keypoint sequence generation.	14
2.7	Electromyographic signals recorded from eight different muscle groups. . . .	15
2.8	Example setup illustrating sEMG sensor placement and EMG signal acquisition for analyzing exercise form [1].	15
2.9	Use of IMU sensors to analyze gait patterns [2].	16
2.10	Heatmap visualization of IMU data collected from 17 sensors.	17
2.11	Example of a motion corresponding to the textual description: “A person raises their hands above their head and then lowers them.”	18

3.1	Architecture of the Pose Encoder used to map 3D keypoint sequences into a shared embedding space.	29
3.2	Architecture of the Pose Decoder, which reconstructs pose parameters from the shared embedding space.	30
3.3	Transformer-based IMU Encoder architecture used to encode temporally dependent IMU signals into the shared embedding space.	32
3.4	Architecture of the IMU Decoder used to reconstruct IMU signals from the shared embedding space.	33
3.5	Architecture of the EMG Encoder. The network maps EMG signal sequences into the shared embedding space.	34
3.6	Architecture of the EMG Decoder. The network reconstructs EMG signal sequences from the shared embedding space.	35
3.7	Transformer-based text decoder architecture. The decoder predicts the next token conditioned on previously generated text (target) and a memory sequence obtained from the shared embedding space.	36
3.8	Text encoder pipeline composed of pretrained components. The input text is mapped to a fixed-dimensional semantic embedding using a CLIP text encoder, followed by a text-to-motion transformer that generates discrete motion tokens. These tokens are decoded into 3D joint trajectories using a VQ-VAE and subsequently converted to SMPL pose parameters via regression before being projected into the shared embedding space.	39
3.9	Occlusion-based explainability results for a jumping jack motion. Normalized importance scores indicate the relative contribution of individual EMG channels (top) and pose joints (bottom) to the learned embedding. Higher values correspond to greater sensitivity of the embedding to occlusion, highlighting key muscles and joints involved in the movement.	48
3.10	Graphical user interface developed for interacting with the encoders, decoders, and latent embedding space.	49
3.11	Example of embedding arithmetic. A pose generated from an IMU embedding (head bowing) and a pose generated from an EMG embedding (foot movement) are combined through embedding addition. The resulting pose exhibits both motions simultaneously.	50
4.1	2D UMAP visualization of pose and reconstructed pose	53
4.2	Qualitative comparison of the original (top) and reconstructed (bottom) 3D pose sequences across frames.	53
4.3	2D UMAP visualization of EMG and reconstructed EMG embeddings.	54
4.4	Qualitative comparison of original (left) and reconstructed (right) EMG signals across eight muscle channels.	55
4.5	2D UMAP visualization of IMU and reconstructed IMU embeddings.	55

4.6	Qualitative comparison of original (left) and reconstructed (right) IMU signals.	56
4.7	Embedding-space analysis for the Pose $\rightarrow$ EMG $\rightarrow$ Pose cycle.	57
4.8	Qualitative visualization of the Pose $\rightarrow$ EMG $\rightarrow$ Pose cycle. The reconstructed pose closely matches the original, with minor deviations highlighted.	57
4.9	Embedding-space analysis for the EMG $\rightarrow$ Pose $\rightarrow$ EMG cycle.	58
4.10	Qualitative visualization of the EMG $\rightarrow$ Pose $\rightarrow$ EMG cycle. Motion structure is largely preserved, with minor deviations highlighted.	59
4.11	Embedding-space analysis for the Pose $\rightarrow$ IMU $\rightarrow$ Pose cycle.	60
4.12	Qualitative visualization of the Pose $\rightarrow$ IMU $\rightarrow$ Pose cycle. The reconstructed pose preserves the original motion pattern, with minor deviations highlighted.	60
4.13	Embedding-space analysis for the IMU $\rightarrow$ Pose $\rightarrow$ IMU cycle.	61
4.14	Qualitative visualization of the IMU $\rightarrow$ Pose $\rightarrow$ IMU cycle. Motion is largely preserved, with visible deviations highlighted.	62
4.15	Embedding-space analysis for the EMG $\rightarrow$ IMU $\rightarrow$ EMG cycle.	63
4.16	Qualitative visualization of the EMG $\rightarrow$ IMU $\rightarrow$ EMG cycle. Motion integrity is largely preserved, with minor deviations highlighted.	63
4.17	Embedding-space analysis for the IMU $\rightarrow$ EMG $\rightarrow$ IMU cycle.	64
4.18	Qualitative visualization of the IMU $\rightarrow$ EMG $\rightarrow$ IMU cycle. Arm motion is preserved, while head and shoulder motion is reduced due to modality limitations.	65
4.19	Qualitative example of text generation from a pose input. The pose is encoded into the latent space and decoded into a natural language description.	66
4.20	Qualitative evaluation of the text encoder. Two pose sequences generated from embeddings produced by the text encoder using the phrases “ <i>a person is kneeling down</i> ” and “ <i>a person is kicking a ball</i> ”. Although the intended actions are recognizable, the motions exhibit irregular limb movements that significantly distort the overall motion.	67

List of Tables

2.1	Comparison of multimodal capabilities across related works	22
3.1	Summary of datasets used in this study.	26
4.1	Interpretation of negative log-likelihood (NLL) loss relative to vocabulary size ($ \mathcal{V} = 9,568$, $\log(\mathcal{V}) \approx 9.17$). Lower values indicate better language modeling performance.	66
4.2	Comparative performance on the Pose-Text HumanML3D dataset. Higher is better for all metrics.	67
4.3	Comparative performance on the Muscle in Action dataset using 3D key-point position RMSE (lower is better).	68
4.4	Comparative performance on the DIP-IMU dataset using average rotation error (lower is better).	68

List of Algorithms

3.1	Temporal Scaling and Noise Augmentation	27
3.2	3D Keypoint to SMPL Pose Regression	40
3.3	Training Procedure for Pose Encoder–Decoder	41
3.4	Training Procedure for IMU / EMG Encoder with Frozen Pose Networks . .	43
3.5	Training Procedure for IMU / EMG Decoder with Frozen Encoders	45
3.6	Training Procedure for Text Decoder	46
3.7	Occlusion-Based XAI for Pose and EMG Embeddings	47

Chapter 1

Introduction

This chapter provides an overview of the core ideas that are incorporated into this research. It introduces multimodal alignment and generation, and explains why combining 3D motion, IMU, EMG, and textual data is important in the context of physical activity analysis. The motivation behind the work, along with the main research objectives, is then discussed. Finally, the chapter outlines the overall methodological approach and highlights the key outcomes of the study.

1.1 Research Overview and Context

In order to enable AI models to work with different modalities of data, multimodal alignment is carried out. Multimodal alignment is a process in which data from different modalities are transformed into a common representation of the same dimensionality, referred to as embeddings, within a shared space. In Fig. 1.1, a diagram illustrating the concept of multimodal alignment is shown for $n = 4$ modalities.

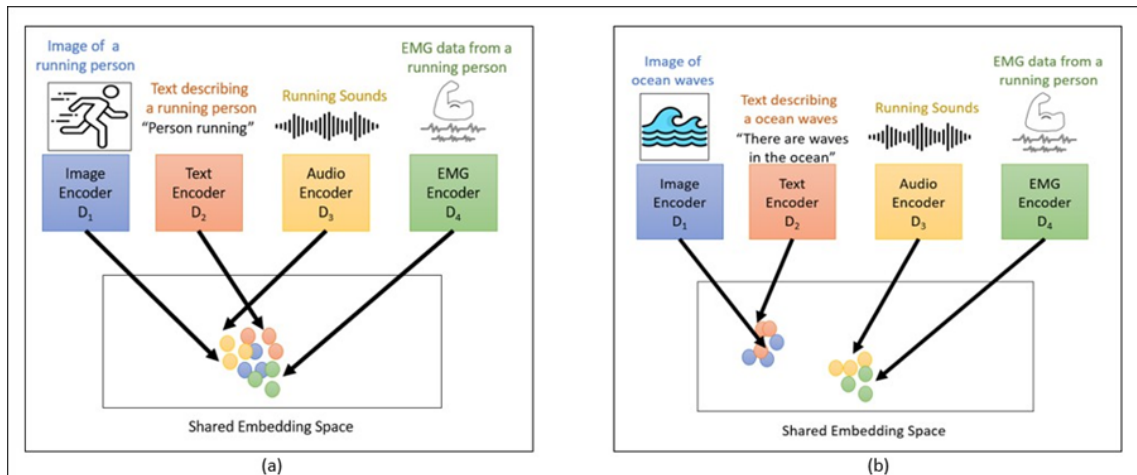


Figure 1.1: Illustration of multimodal alignment for $n = 4$ modalities, where modality-specific encoders map data into a shared embedding space.

In the case of multimodal alignment, n encoders can be used for n data modalities.

Using an encoder E_i , the data D_i can be transformed into an N -dimensional embedding M_i . The embeddings M_1 to M_n are then mapped into a shared embedding space. If the data D_j and D_k from two different modalities represent the same event, their corresponding embeddings M_j and M_k will be located close to one another in the shared embedding space.

An example of this can be considered where D_j represents an image of a person running and D_k represents the textual description “A person is running.” Similarly, D_j may denote the audio associated with running, while D_k denotes the corresponding Electromyography (EMG) signals generated during the same activity. If D_j and D_k do not represent the same event, their corresponding embeddings M_j and M_k will be located far apart in the shared embedding space.

Multimodal generation refers to the process of converting data from one modality into another. If the embeddings in the shared embedding space are able to sufficiently capture the features of an event during encoding, a decoder can then be used to transform a given embedding into data of a different modality. In Fig. 1.2, an example of multimodal generation is illustrated, where an embedding generated by an encoder of one modality is transformed into data belonging to other modalities.

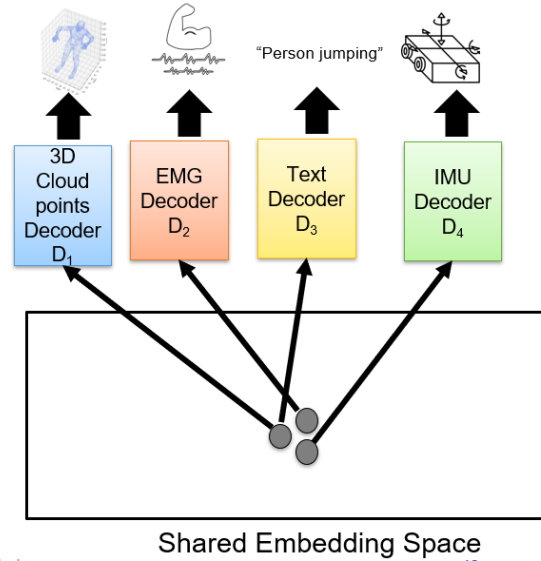


Figure 1.2: Example of multimodal generation, where an embedding produced by an encoder is decoded into other modalities.

In recent years, with the increase in the usage of AI models, several researchers have attempted to develop models for multimodal alignment and generation [3, 4, 5, 6, 7, 8, 9, 10]. With the passage of time, an increasing number of modalities have been incorporated in these research works. However, there are several useful modalities for which multimodal alignment and multimodal generation has not been done yet all together [11, 12, 13, 14]. The data modalities of 3D keypoint motion, Inertial Measurement Unit values, Electromyographic signals and textual description can disseminate useful information regarding physical activities specially during different exercises. However, all of these modalities

of data are not available together in most of the state of the art fitness facilities. Developing techniques to align these modalities as well as cross generation of these modalities can result in several useful applications which has been discussed further in detail in the next section.

1.2 Motivation and Research Significance

In gymnasiums, fitness centers and rehabilitation facilities, physical motion related data are very much crucial for analysis and inference. Based on a review of several scholarly works, it has been found that storing motion data in the form of 3D key points can help in better data analysis specially for examining the condition of the joint angles [6, 14]. EMG signals are another important data in this regard which conveys information regarding the activation of different muscles [12, 13]. Moreover, in order to analyze the speed of movement and orientation of different limbs during movement, IMU data has been used in a number of research works [11, 15, 16]. And lastly, textual description of motion has always been used as the primary tool to preserve motion information [17, 18].

However, most of the state of the art gymnasiums, fitness centers and rehabilitation facilities do not have the means to collect all 4 of these data together [19, 20, 21]. Multimodal alignment and multimodal generation can thus play a key role in solving this problem.

By introducing multimodal alignment, an examiner will be able to group similar data from different modalities. For example, by using textual description embeddings, they can query the storage for EMG, 3D Motion Sequence and IMU values that are relevant to it. Moreover, motion similarity between a fitness instructor’s movement and a trainee athlete’s movement can be compared based on the difference between their data embeddings. Furthermore, two or more motion data from different modalities can be combined together by using embedding arithmetic.

Using the process of multimodal generation, a fitness center can easily attain the data from other modalities without having the requisite instruments. For example, by collecting the EMG data using EMG sensors, 3D motion data, IMU data and text descriptions can be generated altogether.

Thus, based on the above discussion, it can be clearly concluded that multimodal alignment and multimodal generation using the 4 modalities of data including 3D motion data, EMG, IMU and text can play an important role in the sports and rehabilitation industries.

1.3 Research Objectives and Goals

This research focuses on learning a unified representation of human motion by jointly modeling multiple complementary modalities, including IMU signals, EMG signals, 3D keypoints sequence or pose sequence, and natural language descriptions. The goal is

to enable effective cross-modal generation, interpretation, and analysis within a shared embedding space. The specific objectives of this research are outlined below:

- Performing multimodal alignment by mapping all four modalities (IMU signals, EMG signals, 3D keypoint sequences and text) into a shared embedding space.
- Developing methods to attain one modality from another and thus performing multimodal generation.
- Implementing explainable AI (XAI) techniques to present the rationale behind the generated data.
- Performing validation and comparing with state of the art research works in terms of relevant metrics.

1.4 Methodological Approach

The methodology of this research has been carefully designed considering the unpaired nature of the datasets and the data scarcity. The first step involved the collection of the datasets, followed by dataset preprocessing and preparation. The next phase involved the development of the first pair of encoder and decoder which was the 3D motion (or pose) sequence encoder and decoder respectively. Afterwards, using the 3D motion sequence encoders as the base, using the unpaired training strategy, each of the other pairs of encoders and decoders were developed for EMG, IMU as well as text. Once the models were successfully developed, they were rigorously validated. Moreover, the developed encoders and decoders were used to perform several downstream regression tasks to assess their performance by comparing with previous state of the art research works. Lastly, an XAI method was implemented considering the joint angles and muscle activation for each of the modalities. In Fig. 1.3, a flow chart is presented showing the different steps of the methodological approach.

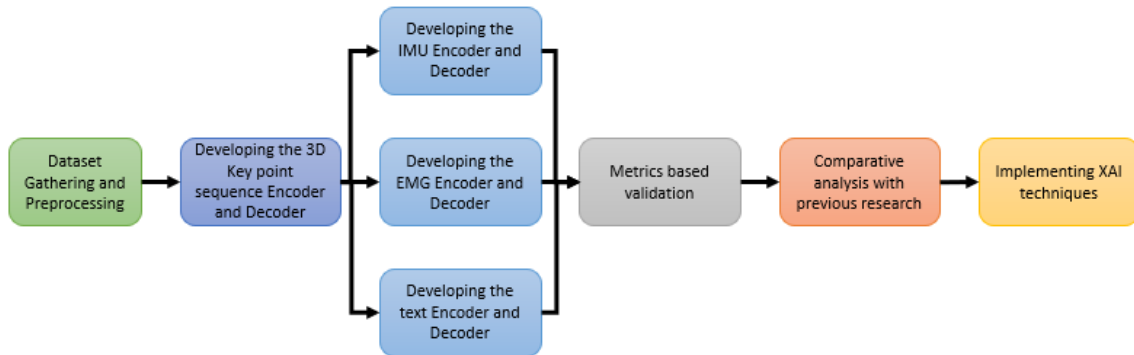


Figure 1.3: Flow chart illustrating the methodological steps followed in this research.

1.5 Research Outcome

This research work managed to perform the multimodal alignment and multimodal generation using 4 new modalities altogether. This is a novel task which further adds to the ongoing research trend. The key contributions of this research work which has been attained as outcomes are listed below:

- Development of a shared embedding space for four modalities—3D key point sequences, IMU, EMG, and text—using an unpaired dataset.
- Cross-modal generation enabled through modality-specific decoders built upon the shared embedding space.
- Development of explainable artificial intelligence (XAI) techniques for analyzing joint movement and muscle activation across all four modalities.
- Achievement of benchmark-level performance, matching or outperforming several state-of-the-art methods on downstream regression tasks.
- Development of a prototype interface supporting multimodal alignment, multimodal generation, embedding arithmetic, and XAI-based analysis.

1.6 Structure and Organization of the Thesis

The rest of this book is organized as follows: chapter 2 presents the background of this research, including the literature review and research gap. chapter 3 describes the methodology adopted in this work. chapter 4 presents the results obtained from validation and comparison using different evaluation metrics. Finally, chapter 5 concludes the book by discussing the findings of this research and its limitations.

Chapter 2

Background

In this chapter, at first the foundational concepts behind multimodal alignment, multimodal generation and the four modalities that are being aligned are discussed. Afterwards the findings from the literature review are presented shedding light on the existing research gap.

2.1 Preliminaries

In this section, the background concepts and relevant terminologies required for understanding this research are discussed. It includes a discussion on multimodal data representation, encoders and decoders, multimodal alignment and generation, unpaired cross-modal learning and explainable AI.

2.1.1 Multimodal data representation and embedding

Multimodal data refers to information originating from heterogeneous sources. These data differ from one another in terms of dimensionality, structure, units, sampling rates, and other characteristics. However, despite these differences, they can convey information about the same underlying event. An example of this is shown in Fig. 2.1, where six different modalities of data are collected from the common event of a person running using six different sensors or data collection tools.

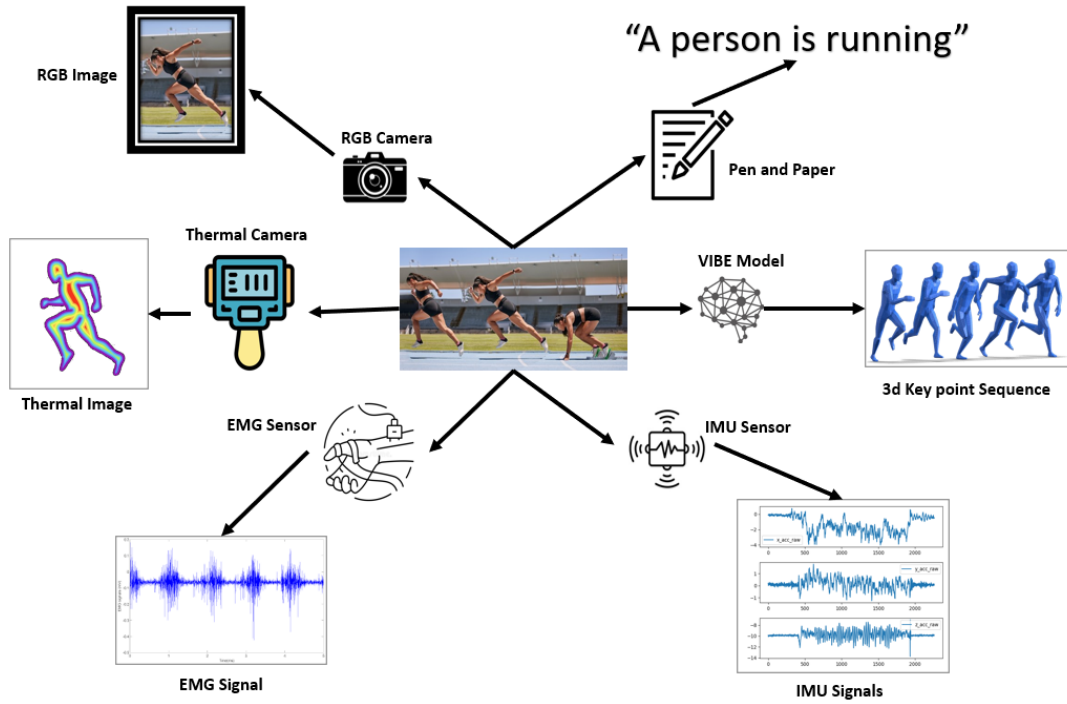


Figure 2.1: From the event of a person running, six different modalities of data (RGB image, thermal image, EMG signal, IMU signals, 3D keypoint sequence, and text) are collected using six different sensors or data collection tools (RGB camera, thermal camera, EMG sensor, IMU sensor, VIBE model, and pen and paper).

Due to the fundamental differences between these modalities, a direct comparison between them is not possible. However, if the data from each modality are transformed into embeddings—numerical representations with the same structure and dimensionality—direct comparison becomes feasible. In Fig. 2.2, the embeddings generated from the modalities shown in Fig. 2.1 are illustrated. Since all modalities correspond to the same event, their respective embeddings are located close to one another within the shared embedding space.

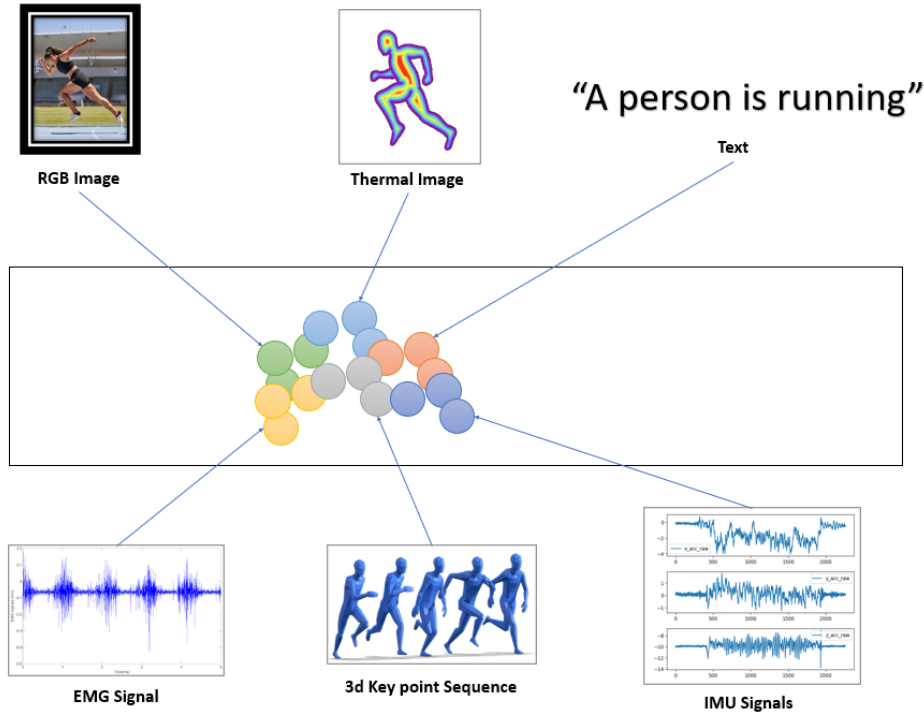


Figure 2.2: Six different modalities of data are converted into embeddings within a shared embedding space. Since the modalities correspond to the same event, their embeddings are located close to one another in the shared embedding space.

2.1.2 Multimodal alignment and multimodal generation

Multimodal alignment refers to the process of mapping data from different modalities into a shared embedding space. When embeddings are derived from the same event, they are expected to lie close to one another in this space, whereas embeddings originating from different events should be farther apart. An example of multimodal alignment is illustrated in Figure 2.3, where embeddings are generated from an audio recording of wing-flapping sounds and from multiple RGB images of animals and birds. Owing to their semantic similarity, the embeddings corresponding to the audio recording and images of flying birds are positioned close to each other in the embedding space, as shown in the figure.

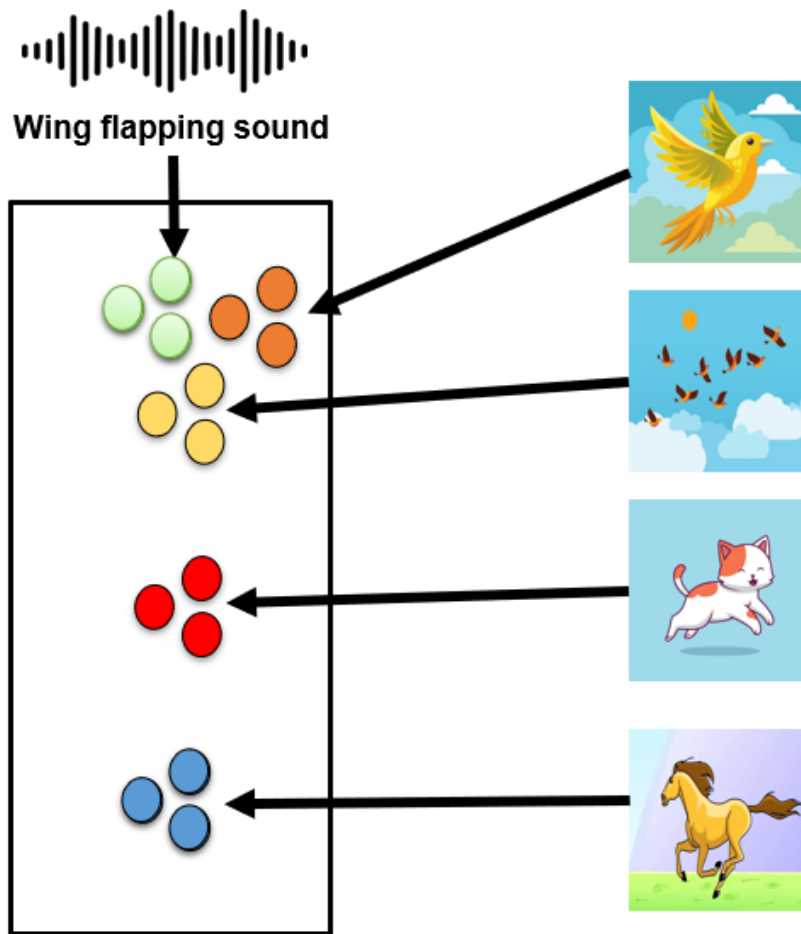


Figure 2.3: Embeddings from similar events aligned closed to one another while the embeddings from different events are positioned distantly

In contrast, multimodal generation involves producing data in one modality from embeddings obtained from another modality. In this process, data from a source modality is first encoded into the shared embedding space, after which a decoder associated with a target modality is used to reconstruct data in that modality. Successful multimodal generation requires that the involved modalities capture semantic information at a comparable level of detail. For instance, given only the audio recording of an exercise, it may be difficult to accurately reconstruct the corresponding electromyographic signals produced during the same activity. However, from the embeddings of an IMU signal, the original EMG values can be easily reconstructed. Figure 2.4 illustrates examples of multimodal generation in cases where the regenerated data closely matches the original data, as well as cases where accurate regeneration is not possible.

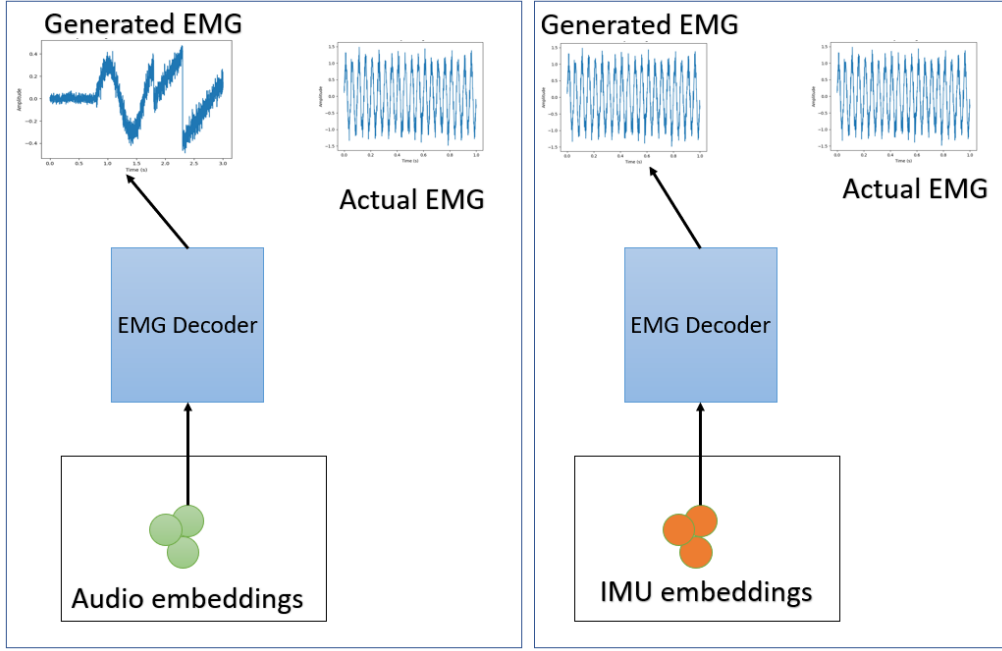


Figure 2.4: Comparison of EMG signal reconstruction from different modalities. EMG signals decoded from audio embeddings (left) show poor correspondence with the ground-truth EMG, whereas EMG signals decoded from IMU embeddings (right) closely resemble the actual EMG signals.

2.1.3 Encoders and Decoders

An encoder is a function that maps raw input data from a given modality into a shared embedding space. Encoders are modality-specific and are designed according to the characteristics of the corresponding data. A decoder, in contrast, reconstructs or generates raw modality data from embeddings in the shared space. The quality of the shared embedding space directly influences the reconstruction accuracy of the decoders.

Let $\mathcal{X}_i$ denote the data space of the i -th modality, where $i \in \{1, 2, \dots, M\}$ and M is the total number of modalities. The encoder for modality i is defined as

$$E_i : \mathcal{X}_i \rightarrow \mathcal{Z}. \quad (2.1)$$

Given an input sample $x_i \in \mathcal{X}_i$, the corresponding embedding is

$$z_i = E_i(x_i), \quad z_i \in \mathcal{Z} \subset \mathbb{R}^N. \quad (2.2)$$

Each encoder is tailored to the structure of its input modality, while the latent space $\mathcal{Z}$ is shared across all modalities.

Similarly, the decoder for modality j is defined as

$$G_j : \mathcal{Z} \rightarrow \mathcal{X}_j. \quad (2.3)$$

Given an embedding $z \in \mathcal{Z}$, the reconstructed or generated output is

$$\hat{x}_j = G_j(z). \quad (2.4)$$

For cross-modal generation from modality i to modality j , the input data is first encoded using E_i and then decoded using G_j , yielding

$$\hat{x}_j = G_j(E_i(x_i)). \quad (2.5)$$

2.1.4 Unpaired Cross-Modal Learning

Paired multimodal datasets are often difficult and costly to collect in real-world settings. Unpaired cross-modal learning addresses this limitation by allowing different modalities to be learned independently, provided that a common reference modality is available across all datasets. In this work, the reference modality serves as an anchor that enables alignment of otherwise unpaired modalities within a shared latent embedding space.

Let $\mathcal{X}_r$ denote the data space of the reference modality and $\mathcal{X}_i$ denote the data space of the i -th non-reference modality, where $i \in \{1, 2, 3, \dots\}$. Each dataset consists of paired samples (x_r, x_i) describing the same underlying event, where the reference signal $x_r \in \mathcal{X}_r$ is present in all datasets, while $x_i \in \mathcal{X}_i$ corresponds to a modality-specific observation. Direct pairing between non-reference modalities is not assumed.

A reference encoder–decoder pair (E_r, D_r) is first trained using samples from the reference modality. Given a reference input x_r , the encoder produces a latent embedding

$$z_r = E_r(x_r), \quad (2.6)$$

which is decoded back into the reference space as

$$\hat{x}_r = D_r(z_r). \quad (2.7)$$

The reference encoder–decoder is trained to minimize a reconstruction loss

$$\mathcal{L}_{\text{ref}} = L_r(x_r, \hat{x}_r), \quad (2.8)$$

ensuring that the latent space preserves the structure of the reference modality. After training, both E_r and D_r are frozen, thereby defining a stable shared embedding space $\mathcal{Z}$.

For each non-reference modality i , an encoder E_i is trained using dataset-specific paired samples (x_r, x_i) . The modality- i input is mapped into the shared latent space as

$$z_i = E_i(x_i). \quad (2.9)$$

To align modality i with the reference modality, the latent embeddings produced by

E_i are encouraged to match the corresponding reference embeddings:

$$\mathcal{L}_{\text{align}}^{(i)} = \|z_i - z_r\|_2. \quad (2.10)$$

In addition to latent alignment, cross-modal reconstruction through the reference modality is enforced by decoding z_i using the frozen reference decoder:

$$\hat{x}_i^{(r)} = D_r(z_i), \quad (2.11)$$

which is compared against the reference signal using a reconstruction loss

$$\mathcal{L}_{\text{recon}}^{(i,r)} = L_r(x_r, \hat{x}_i^{(r)}). \quad (2.12)$$

The encoder E_i is trained by minimizing the combined objective

$$\mathcal{L}_{E_i} = \mathcal{L}_{\text{align}}^{(i)} + \mathcal{L}_{\text{recon}}^{(i,r)}. \quad (2.13)$$

Once the encoder E_i is trained, it is frozen and used to train a modality-specific decoder D_i . Given an input sample x_i , the frozen encoder produces a latent embedding

$$z_i = E_i(x_i), \quad (2.14)$$

which is then decoded back into the modality- i space as

$$\hat{x}_i = D_i(z_i). \quad (2.15)$$

The decoder D_i is trained to minimize a reconstruction loss

$$\mathcal{L}_{D_i} = L_i(x_i, \hat{x}_i), \quad (2.16)$$

ensuring accurate reconstruction of modality- i signals from the shared latent representation. By freezing E_i during this stage, the decoder learns to operate consistently within the established latent space without altering the previously learned alignment.

Through this two-stage training process, all modalities are consistently embedded into the shared latent space and can be decoded both through the reference modality and back into their original domains, enabling unpaired yet coherent cross-modal generation.

2.1.5 Explainable AI

Explainable Artificial Intelligence (XAI) aims to make the predictions of machine learning models more interpretable by identifying which input features influence the model output. Deep neural networks are often considered black-box models, as they provide limited insight into how individual features contribute to their predictions.

Given a trained model $f(\cdot)$ and an input sample $x \in \mathbb{R}^N$, an XAI method assigns an

importance score to each input feature,

$$\phi(x_i), \quad i = 1, \dots, N, \quad (2.17)$$

where $\phi(x_i)$ reflects the influence of the i -th feature on the model output.

In this work, these importance scores are used to analyze joint angle engagement and muscle activation across different modalities of activity-based data. Details of the XAI procedure are provided in the methodology section.

2.2 Modalities and Description

In this section the four modalities used in this research are discussed including 3D keypoint sequence or pose sequence, EMG, IMU and text description.

2.2.1 3D Keypoint Sequence or Pose Sequence

To represent and analyze the motion of a person performing an exercise, a sequence of 3D keypoints or pose can be used. In recent years, one of the most widely adopted representations for 3D human body modeling is the Skinned Multi-Person Linear (SMPL) format [22].

The SMPL model represents the human body using a mesh consisting of 6,890 three-dimensional vertices. The SMPL representation is parameterized by two sets of variables, denoted as β and θ . The shape parameter β controls the body shape of the individual, while the pose parameter θ represents the relative rotations of 24 body joints, expressed in a 3D axis-angle format. Given a template mesh and the parameters β and θ , a pre-trained SMPL model can generate a posed 3D human mesh. A temporal sequence of pose parameters can therefore be used to model human motion over time. An example of a 3D keypoint sequence represented in the SMPL format is shown in Figure 2.5.

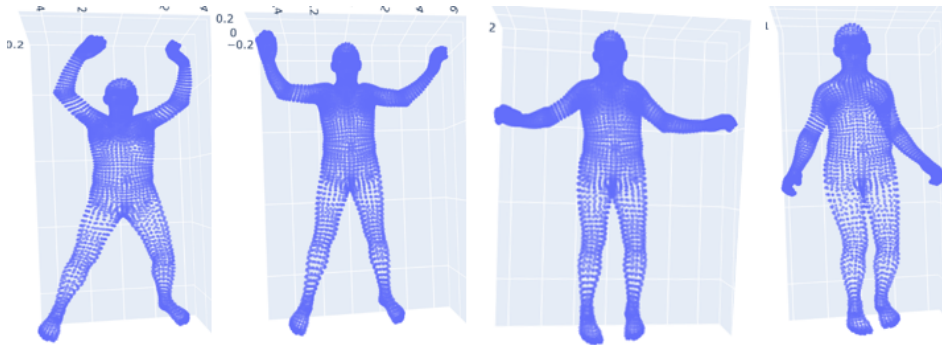


Figure 2.5: 3D keypoint sequence represented using the SMPL format.

A pretrained Human Mesh Recovery Model like Video Inference for Body Estimation (VIBE) model [23] or SMPLify [24] can be used to extract the SMPL parameters β and θ directly from RGB video data. These estimated parameters are then provided as input

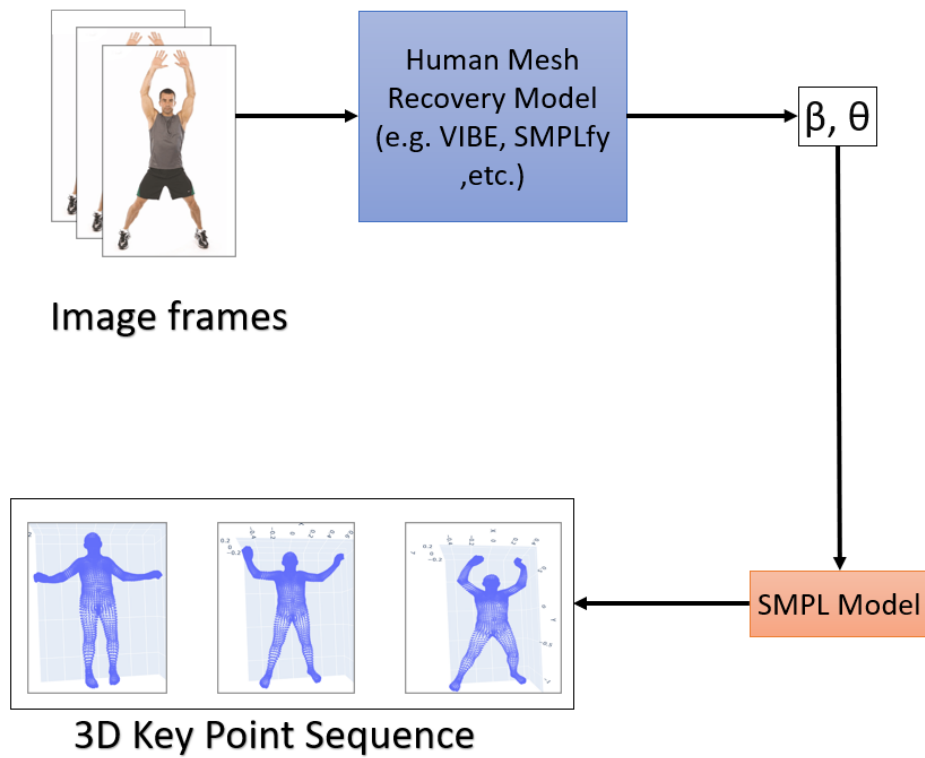


Figure 2.6: Pipeline illustrating the use of the VIBE model to estimate SMPL parameters from RGB video, followed by SMPL-based 3D keypoint sequence generation.

to the SMPL model to generate the corresponding 3D keypoint sequence. Figure 2.6 illustrates the combined operation of the Human Mesh Recovery model and SMPL model [22] for generating a 3D keypoint sequence in the SMPL representation.

2.2.2 Electromyographic (EMG) Signals

Electromyographic (EMG) [25] signals can be used to quantify the activation level of specific muscle groups. Using non-invasive surface EMG (sEMG) sensors comprising electrodes, the bioelectric activity produced during muscle contraction can be measured, typically in microvolt (μV) units. Figure 2.7 shows an example EMG plot recorded from eight muscle groups.

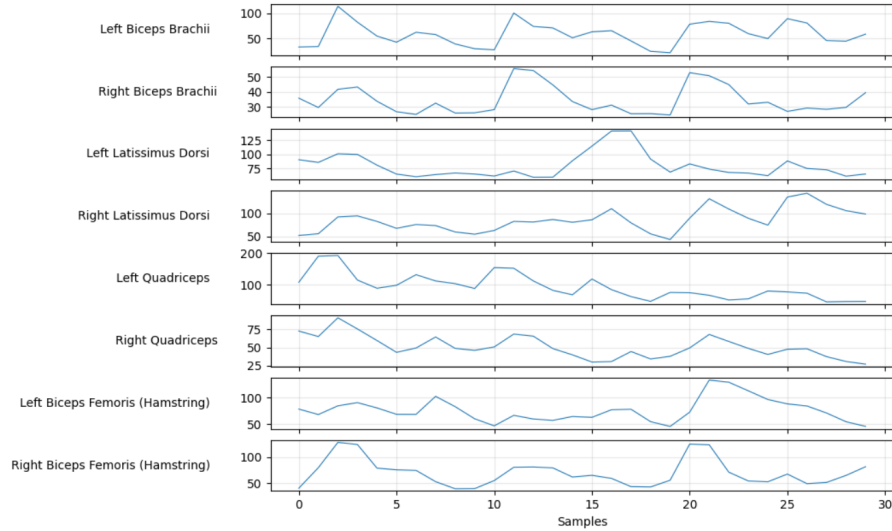


Figure 2.7: Electromyographic signals recorded from eight different muscle groups.

In the EMG-based dataset used in this research, signals were collected from the following eight muscle groups: left biceps brachii, right biceps brachii, left latissimus dorsi, right latissimus dorsi, left quadriceps, right quadriceps, left biceps femoris, and right biceps femoris.

EMG sensors can support exercise monitoring in fitness and rehabilitation settings by providing information about whether the target muscles are being activated as intended. By comparing the measured activation patterns with expected activation profiles, deviations in exercise execution can be identified. Figure 2.8 illustrates the use of sEMG sensors for measuring muscle activity during exercise.

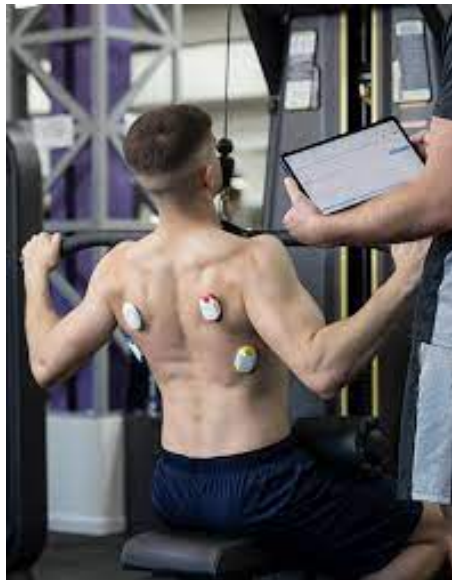


Figure 2.8: Example setup illustrating sEMG sensor placement and EMG signal acquisition for analyzing exercise form [1].

2.2.3 Inertial Measurement Unit (IMU) Signals

Inertial Measurement Unit (IMU) data are commonly used to capture the orientation and motion of body segments during different movements and exercises [26]. An IMU typically integrates a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer. These sensors measure linear acceleration, angular velocity, and orientation relative to the Earth’s magnetic field, respectively.

The accelerometer records acceleration along three orthogonal axes, producing three measurements at each time step. Likewise, the gyroscope provides three measurements corresponding to angular velocity about each axis. Orientation information from the magnetometer is represented as a 3×3 rotation matrix, resulting in nine values. As a result, each IMU sensor generates a total of $3 + 3 + 9 = 12$ features per time step.

Figure 2.9 shows an example of IMU sensor placement and data usage for gait (walking) analysis.



Figure 2.9: Use of IMU sensors to analyze gait patterns [2].

In the dataset used in this research, a total of 17 IMU sensors are deployed across the body. As a result, each time step in the dataset contains $17 \times 12 = 204$ IMU-derived features. A visualization of the collected IMU data using a heatmap representation is shown in Figure 2.10.

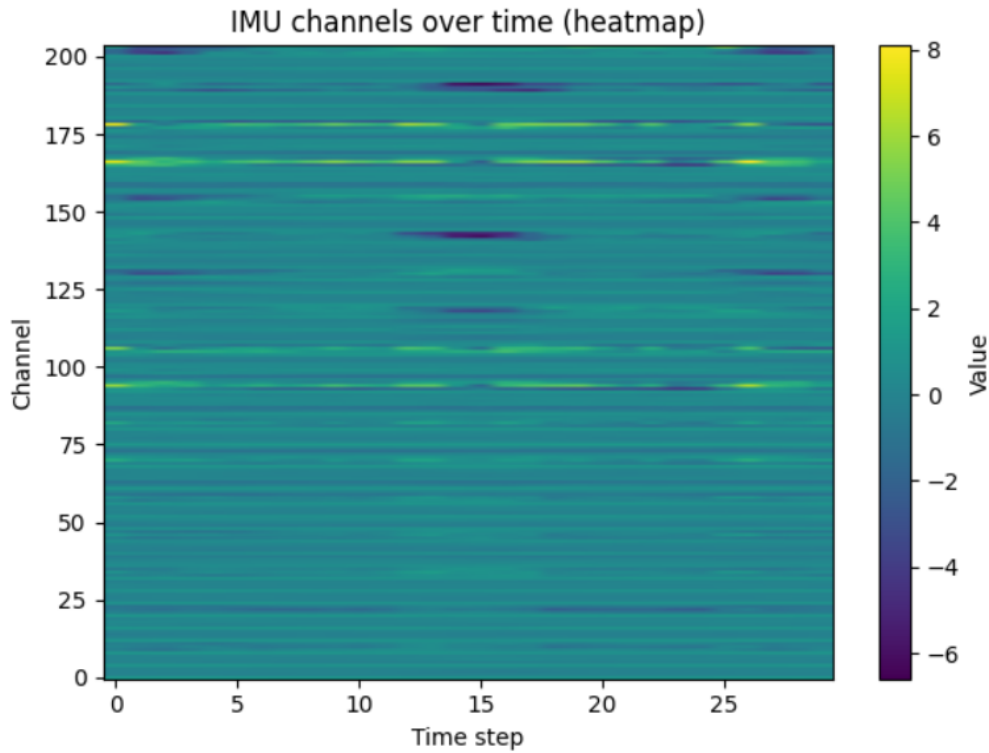


Figure 2.10: Heatmap visualization of IMU data collected from 17 sensors.

The 17 IMU sensors are positioned on key body segments, specifically the head, upper spine (chest), lower spine (pelvis), left and right shoulders, left and right upper arms, left and right forearms, left and right thighs, left and right shanks, left and right feet, and left and right toes, enabling comprehensive capture of full-body motion dynamics.

2.2.4 Text Describing Motion

Natural language has long been the primary medium for describing and analyzing human motion, particularly in fitness and rehabilitation contexts. Fitness professionals routinely use verbal or written descriptions to instruct, observe, and assess the movements performed by clients or patients. As a result, textual descriptions provide a high-level semantic representation of motion and serve as an important complementary modality in this research.

In this work, text is used to describe the intent and sequence of movements performed during an exercise. Figure 2.11 illustrates an example of a motion corresponding to the textual description: *“A person raises their hands above their head and then lowers them.”*

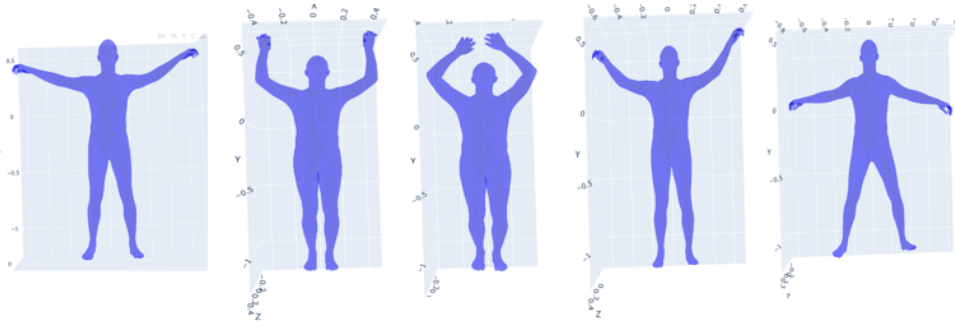


Figure 2.11: Example of a motion corresponding to the textual description: “A person raises their hands above their head and then lowers them.”

2.3 Literature Review

In recent years, several research efforts have focused on the problem of multimodal alignment, where information from different data modalities is projected into a shared representation space. One of the earliest and most influential works in this area is ImageBind by Girdhar et al. [3]. In this framework, six different modalities—audio, image frames, depth images, thermal images, text, and Inertial Measurement Unit (IMU) signals—are mapped into a common embedding space. An important observation from their work is that full multi-way paired data is not required during training. Instead, images are paired individually with each of the other modalities, and after training, all modalities can be successfully aligned within the same embedding space. The authors demonstrate that this shared space can be effectively used for several tasks, including cross-modal retrieval, embedding arithmetic, and even audio-to-image generation.

Following a similar line of work, Moon et al. proposed IMU2CLIP [4], where IMU signals, text, and video are aligned into a shared embedding space. Their experiments show that this alignment enables tasks such as retrieving IMU recordings using text queries and finding relevant video clips based on IMU data. In addition, they report that encoders trained through multimodal alignment can serve as strong pretrained models, leading to improved performance on downstream classification tasks.

Liu et al. introduced AlignRec [5], a multimodal recommendation framework designed to improve recommendation quality through alignment. Their system performs alignment at three different levels: between visual and textual content (for example, product images and descriptions), between content and categorical identifiers such as product IDs or category labels, and finally between users and items. By embedding these heterogeneous sources into a shared space, AlignRec is able to capture richer relationships between users and items.

In a related effort, Wang et al. [6] aligned images, text, audio, and 3D point clouds within a unified embedding space. A notable aspect of their approach is that it does not rely on fully paired multimodal data. Instead, alignment across modalities is achieved using overlapping modalities—such as text—that act as bridges between different modality

pairs. This allows the encoders to be trained even when direct pairing is unavailable.

Moving beyond global alignment, Yin et al. [7] proposed a fine-grained alignment strategy. Rather than aligning entire images and full text descriptions, their method decomposes text into individual tokens and segments images into smaller regions that act as visual tokens. Alignment is then performed at the token level, resulting in more detailed and precise associations between visual and textual elements.

Zhang et al. presented UniMTS (Unified Pre-training for Motion Time Series) [8], which focuses on aligning motion time-series data collected from smartphones and wearable devices with textual descriptions of human activities. Using a contrastive learning objective, motion signals and text are mapped into a shared embedding space. The resulting representations improve activity recognition performance and also support zero-shot classification, where activities unseen during training can still be recognized using text descriptions.

More recent work has begun to explore alignment together with generation capabilities. Yang et al. [9] proposed MAGE, which aims to bridge visual and semantic spaces in order to jointly improve multimodal alignment and generation. Their approach combines a text generation loss to maintain linguistic quality with an image-text distance loss in the embedding space, allowing for effective bidirectional alignment between image and language modalities.

Sudarsanam et al. [10] focus primarily on semantic alignment across language, audio, and visual modalities. Using the paired AVCaps dataset, which consists of synchronized audio, video frames, and captions, their work emphasizes learning robust aligned representations rather than supporting generation. Their results highlight the importance of well-curated paired datasets for cross-modal representation learning.

Liu et al. [27] introduced OLA, an omni-modal language model capable of handling text, video, and audio inputs. Their method relies on an anchor-based alignment strategy, where video serves as the central modality for grounding. While generation is mainly limited to text outputs, the work demonstrates how anchor-based alignment can scale to more general omni-modal settings.

Cicchetti et al. [28] took a more theoretical approach with Gramian multimodal representation learning. They proposed a new alignment metric that can be applied to more than two modalities at the same time, including text, video, audio, and depth. Rather than focusing on generation, their work contributes a general framework for measuring and evaluating multimodal alignment.

Yue et al. [29] explored instruction-augmented multimodal alignment for image-text matching. Their model is trained to predict alignment scores for image-text pairs and is optimized against human-annotated ground-truth scores. This approach improves fine-grained alignment performance, although it does not address multimodal generation.

Wang et al. [30] propose Ex-MCR to integrate separate multimodal embedding spaces into a unified representation space. CLAP provides an audio-text embedding space, ULIP provides a 3D-image embedding space, and CLIP provides an image-text embedding space.

Ex-MCR uses CLIP as the base space and learns lightweight projectors that map CLAP and ULIP features into the CLIP space, using their overlapping modalities: text for CLAP–CLIP alignment and image for ULIP–CLIP alignment. This approach preserves the original models while enabling audio, image, text, and 3D representations to be compared within a shared embedding space, even without direct paired data across all modalities.

Liu et al. [31] propose M3Bind, a method that brings different medical imaging models into one shared space by using text as the connector. Each medical model first has its own image-text space: one for X-rays and reports, one for CT scans and reports, one for retina images and text, one for ECG signals and text, and one for pathology images and text. Instead of needing direct pairs between these medical images, such as X-ray with CT or ECG with pathology, M3Bind aligns the text parts of these models. Because each image type is already connected to its own text, aligning the text spaces also helps connect the image modalities. The result is a shared medical embedding space where different medical data types can be compared through text.

Finally, MotionBind [14] extends multimodal alignment to the domain of human motion by incorporating text, image, video, audio, and 3D pose data. Using contrastive learning, motion is aligned with other modalities in a shared embedding space that is anchored to language representations. In addition, a motion decoder is introduced to enable 3D motion sequence generation, making MotionBind one of the few approaches that combine multimodal alignment and generation for embodied data.

2.4 Gap Analysis

The fields of sports and rehabilitation are increasingly benefiting from advances in artificial intelligence, supported by data collected from a variety of sensing modalities. Wearable and body-mounted sensors are commonly used to capture detailed information about human movement and physiological activity in both athletic training and rehabilitation settings. Inertial Measurement Unit (IMU) sensors are widely used for studying human motion, particularly in applications related to performance monitoring and rehabilitation. A substantial body of prior work has explored IMU signals in sports and rehabilitation settings [11, 15, 16]. These studies address a range of tasks, including classification problems such as activity recognition and terrain classification, as well as regression tasks, for example the generation of Functional Electrical Stimulation (FES) signals.

Alongside IMU data, Electromyography (EMG) signals have also been extensively investigated in these domains. EMG captures muscle activation patterns and has been used in several studies [12, 13] for regression tasks such as estimating muscle activation levels, as well as for classification tasks including exercise and motion recognition.

Another important representation for motion analysis is the use of 3D human body keypoint or pose sequences, which provide structured information about body posture and movement over time. Compared to raw visual data, 3D keypoints focus directly on motion while filtering out background and appearance-related factors. This makes them well

suited for applications such as motion tracking, activity classification, and rehabilitation assessment. Despite their relevance, relatively few works [6, 28] have explored 3D body representations in a multimodal learning setting, particularly in combination with other sensor-based modalities.

Text-based data has also played an important role in sports and rehabilitation applications [17, 18]. Advances in natural language processing (NLP) and conversational AI have enabled systems that support personalized rehabilitation plans, provide exercise instructions, and assist with patient monitoring. For instance, text-based interfaces can deliver guidance, respond to patient queries, and analyze written feedback to track progress over time.

Despite the extensive use of individual modalities such as IMU signals, EMG signals, 3D pose sequences, and text, to the best of our knowledge, no prior work has aligned all four modalities within a single shared embedding space. Such a unified alignment could offer significant advantages. In practical scenarios, data from all modalities may not always be available [19, 20, 21]; a shared embedding space would allow information from one modality (e.g., IMU signals) to be used to infer or support tasks typically reliant on other modalities (e.g., text descriptions or motion representations).

Furthermore, while many AI-based studies in sports and rehabilitation emphasize performance improvements, relatively little attention has been given to explainable artificial intelligence (XAI) within multimodal alignment frameworks. None of the aforementioned alignment-based approaches incorporate XAI techniques, despite their importance in addressing the black-box nature of deep learning models. Integrating XAI methods into multimodal alignment models could improve interpretability, increase user trust, and enhance reliability in real-world sports and rehabilitation applications.

The existing research gaps identified from the above discussion are summarized in Table 2.1, which provides a comparative overview of prior works with respect to the modalities considered, alignment strategies, generation capabilities, and the use of explainable AI. This comparison highlights the lack of unified alignment across IMU signals, EMG signals, 3D body keypoint sequences, and text, thereby motivating the need for a comprehensive multimodal framework.

Table 2.1: Comparison of multimodal capabilities across related works

Ref	Image / Video	3D Key Point Sequence	Text	IMU	Depth Image	Thermal Image	Audio	EMG	XAI	Multimodal Alignment	Multimodal Generation
[3]	✓	×	✓	✓	✓	✓	✓	×	×	✓	✓ (audio, image)
[4]	✓	×	✓	✓	×	×	×	×	×	✓	×
[5]	✓	×	✓	×	×	×	×	×	×	✓	×
[6]	✓	✓	✓	×	×	×	✓	×	×	✓	×
[7]	✓	×	✓	×	×	×	×	×	×	✓	✓ (text)
[8]	×	×	✓	✓	×	×	×	×	×	✓	×
[9]	✓	×	✓	×	×	×	×	×	×	✓	✓
[10]	✓	×	✓	×	×	×	✓	×	×	✓	×
[27]	✓	×	✓	×	×	×	✓	×	×	✓	✓ (text)
[28]	✓	×	✓	×	×	×	✓	×	×	✓	×
[29]	✓	×	✓	×	×	×	×	×	×	✓	✓ (3D motion)
[30]	✓	✓	✓	×	×	×	✓	×	×	✓	×
[31]	✓	×	×	×	×	×	×	×	×	✓	✓ (text)
[14]	✓	✓	✓	×	×	×	✓	×	×	✓	✓
Proposed Work	×	✓	✓	✓	×	×	×	✓	✓	✓	✓

2.5 Summary

This chapter introduced the core ideas needed to understand the thesis and placed the work within the context of existing research. It began by explaining how data from different modalities can be represented in a shared embedding space, enabling alignment and cross-modal generation. Key concepts such as encoders and decoders, multimodal alignment, unpaired learning, and the role of explainable AI were outlined to provide a clear conceptual foundation.

The chapter then described the four modalities used in this work—3D keypoints sequence or pose sequence, EMG signals, IMU signals, and text—and explained how each captures a different aspect of human movement. Together, these modalities provide complementary information that is particularly relevant for sports and rehabilitation applications.

A review of related work showed that prior studies have successfully aligned multiple modalities, but typically focus on a limited set of data types and rarely include explainability. The gap analysis highlighted that no existing work jointly aligns IMU, EMG, 3D keypoints, and text within a single framework, and that explainable AI is largely absent from multimodal alignment research. These gaps motivate the contributions of this thesis, which aims to build a unified and interpretable multimodal alignment framework.

Chapter 3

Methodology

This section presents the datasets used in this work along with the procedures adopted for data collection and preprocessing. It then describes the model architectures employed for learning, followed by details of the training process and the explainability algorithms used to interpret the model outputs.

3.1 Dataset

This section describes the datasets used in this study. Three publicly available datasets are employed, each providing human motion data in a different modality. HumanML3D provides paired 3D human pose sequences and natural language descriptions [32]. DIP-IMU contains synchronized 3D human poses and inertial measurement unit (IMU) signals [33]. The Muscles in Action dataset provides paired 3D human poses and surface electromyography (sEMG) signals recorded during physical activities [34].

3.1.1 Dataset Collection and Description

HumanML3D Dataset

HumanML3D contains around 14,000 3D human motion sequences collected from a variety of motion capture datasets and later unified under the AMASS framework [35]. The motion data are represented as 3D keypoint sequences in SMPL format and are sampled at a fixed rate of 20 Hz. The recordings originate from a large pool of 300+ subjects although individual subject identities are not explicitly retained in the final dataset. Each 3D keypoint sequence is paired with multiple natural language descriptions, which describe the same motion from different perspectives. These descriptions vary in terms of action type, motion style, and overall movement characteristics. By combining each motion sequence with all of its associated text descriptions, the dataset is organized into 39,714 pose-text pairs. Each pair consists of one motion sequence and one corresponding textual description.

DIP-IMU Dataset

The DIP-IMU dataset contains human motion data recorded using wearable inertial sensors and corresponding ground-truth 3D poses. The motion data are sampled at a rate of 60 Hz and were collected from 10 subjects performing a wide range of everyday movements and activities. The ground-truth human poses are represented as 3D keypoint sequences in SMPL format, using the standard SMPL joint structure. Each subject is equipped with 17 IMU sensors placed on different body parts, including the head, upper spine (chest), lower spine (pelvis), left and right shoulders, left and right upper arms, left and right forearms, left and right thighs, left and right shanks, left and right feet, and left and right toes. Each IMU sensor provides 12 values per time step, consisting of 3-axis linear acceleration, 3-axis angular velocity, and a 3×3 rotation matrix representing orientation. This results in a total of 204 IMU features per frame (17×12). After preprocessing and alignment, the dataset is organized into 3,208 pose-IMU pairs, where each pair consists of one SMPL-based 3D keypoint sequence and its corresponding IMU signal sequence.

Muscles in Action Dataset

The Muscles in Action dataset provides synchronized recordings of human motion and muscle activity collected during the execution of various physical actions. The dataset includes data from 10 human subjects, with video recordings temporally aligned with electromyography (EMG) signals and resampled to a common frequency of 10 Hz to ensure consistent processing. Human motion is represented using 3D keypoint sequences in SMPL format, obtained by applying SMPL-based pose estimation to the recorded videos. These pose sequences offer a structured representation of full-body movement over time. Muscle activity is captured using non-invasive surface electromyography (sEMG) sensors, which measure bioelectric signals generated during muscle contraction and are typically expressed in microvolt (μV) units. In the EMG data used in this work, signals are recorded from eight muscle groups: left and right biceps brachii, left and right latissimus dorsi, left and right quadriceps, and left and right biceps femoris. These measurements provide insight into muscle activation patterns during movement and are particularly useful for analyzing muscle engagement during physical actions. The complete dataset occupies approximately 52 GB of storage. Due to computational and storage constraints, a subset of 29.41 GB was used in this study, covering data from six subjects. Following preprocessing and temporal alignment, the data were organized into 6,000 pose-EMG pairs, with each pair consisting of an SMPL-based 3D pose sequence and its corresponding EMG signal sequence.

Dataset Summary

The summary of the datasets used in this study is presented in table 3.1. The datasets used in this study differ in their temporal sampling rates, which introduces challenges when aligning multimodal data streams. These differences require careful preprocessing to ensure temporal consistency across motion, sensor, and textual modalities.

Table 3.1: Summary of datasets used in this study.

Dataset	Rate	Participants	Used	Pairs	Data Type
HumanML3D	20 Hz	~300	~300	39,714	3D SMPL pose + text
DIP-IMU	60 Hz	10	10	3,208	3D SMPL pose + IMU
Muscles in Action	10 Hz	10	6	6,000	3D SMPL pose + sEMG

3.1.2 Data Preprocessing

Text Preprocessing

In order to process text, a numerical mapping was created for each word in the dataset, referred to as the vocabulary. The vocabulary also included numerical mappings for three special tokens: `<SOS>`, `<EOS>`, and `<PAD>`. The `<SOS>` and `<EOS>` tokens were used to indicate the beginning and end of a sentence, respectively, while the `<PAD>` token was used to standardize sequence lengths. Using this approach, a vocabulary of 9,568 tokens was constructed. Example mappings from the vocabulary include `<PAD>`: 0, `<SOS>`: 1, `<EOS>`: 2, `a`: 3, `man`: 4, and `kicks`: 5.

Each sentence in the dataset was tokenized by converting words into their corresponding numerical indices. Every tokenized sentence was prefixed with the `<SOS>` token and suffixed with the `<EOS>` token. A fixed sequence length of 137 tokens was selected, as this corresponds to the maximum sentence length observed in the dataset. Sentences shorter than this length were padded using the `<PAD>` token to ensure consistent input dimensions.

For example, the sentence *“a man kicks something or someone with his left leg.”* was converted into a sequence of numerical values beginning with the `<SOS>` token, followed by the indices corresponding to each word in the sentence, and ending with the `<EOS>` token. The remaining positions in the sequence were filled using padding tokens to reach the fixed length of 137. This representation preserves the original word order while enabling efficient batch processing of textual inputs.

Adjusting Different Sampling Frequencies

Since the 3D keypoint sequences, IMU signals, and EMG signals were sampled at different frequencies across datasets, a unified strategy was required to ensure that downstream models could learn robustly from inputs with varying temporal resolutions. To address this, a temporal resampling procedure was applied to the 3D keypoint sequences, EMG signals, and IMU values, as described in Algorithm ???. The approach randomly stretches or compresses a sequence along the time dimension by linearly interpolating it to a new length.

To preserve motion integrity and avoid excessive temporal distortion, the scaling factor was constrained within a fixed range, with a minimum scale of 0.5 and a maximum scale of 1.5. In addition, to prevent the resampling process from reducing a sequence to an excessively short duration, a minimum sequence length of 8 time steps was enforced. This strategy reduces sensitivity to the original sampling frequency while also acting as a form

of temporal data augmentation by introducing mild timing variations during training.

Algorithm 3.1: Temporal Scaling and Noise Augmentation

```

1: Input:  $data, min\_scale, max\_scale, min\_len, \sigma$ 
2: Output:  $augmented\_data$ 
3:  $augmented\_data \leftarrow data$ 
4: if random value  $\geq 0.5$  then
5:   Sample  $scale$  from  $[min\_scale, max\_scale]$ 
6:    $T \leftarrow$  number of time steps
7:    $T' \leftarrow \max(min\_len, \lfloor T \cdot scale \rfloor)$ 
8:   Resize  $augmented\_data$  to length  $T'$ 
9: end if
10: if random value  $< 0.5$  then
11:   Add Gaussian noise with standard deviation  $\sigma$ 
12: end if
13: return  $augmented\_data$ 

```

3D Keypoint Sequence Preprocessing

The 3D keypoint sequences were represented in SMPL format, where each motion sequence includes pose parameters (θ) and shape parameters (β). Since the objective of this work focuses on motion and muscle activation patterns rather than subject-specific body shape, the SMPL shape parameters (β) were removed considering the fact that muscle activation does not depend much on its respective size [36]. After removing β , the pose representation consists of the 72-dimensional pose vector θ , corresponding to 24×3 axis-angle rotations.

The axis-angle rotations are defined relative to the kinematic root, and the pelvis joint acts as the global reference. To further normalize the data and reduce unnecessary variability, the pelvis rotation was set to (0,0,0) for all sequences. Although axis-angle is compact, it has practical limitations for learning-based methods, such as discontinuities and ambiguity near rotations of π , which can lead to unstable gradients and poorer convergence when used directly as a regression target or input feature [37, 38].

To improve learning stability, each axis-angle rotation was converted into a 6D rotation representation [37, 38]. The 6D representation is commonly preferred in neural models because it provides a continuous parameterization and avoids discontinuities present in axis-angle and Euler angle forms, leading to smoother optimization and more reliable pose prediction. Thus the pose representation was converted into a vector of $24 \times 6 = 144$ representations.

Moreover, because the datasets are sampled at different temporal rates (Hz), the temporal resampling method described in Algorithm ?? was applied to SMPL pose sequences to randomly stretch or compress them in time. This step helps the model become less sensitive to dataset-specific sampling frequencies. It also functions as temporal data augmentation by introducing controlled timing noise during training, which encourages the model to learn more robust motion features.

IMU signal preprocessing

IMU preprocessing is performed to ensure that inertial measurements are numerically stable and suitable for learning. Each IMU sequence consists of a time series of 204-dimensional feature vectors, corresponding to the concatenated measurements from all IMU sensors at each time step. Any missing values in the sequence are handled by propagating the value from the previous time step to maintain temporal continuity.

For normalization, the mean and standard deviation are computed independently for each IMU channel across the dataset. These per-channel statistics are used to apply z-score normalization, where each IMU value is centered by subtracting its corresponding channel mean and scaled by its channel standard deviation. To avoid numerical instability, channels with zero variance are handled safely, and a small constant is added during normalization.

In addition to feature normalization, temporal robustness is introduced using the frame-rate adjustment procedure described in Algorithm ???. This algorithm randomly stretches or compresses IMU sequences along the time dimension using linear interpolation within predefined bounds. This step reduces sensitivity to variations in sampling frequency and acts as a form of temporal data augmentation by introducing mild timing variations during training.

EMG signal preprocessing

The EMG signals are processed to ensure that muscle activation measurements are stable and suitable for learning-based models. Each EMG recording is represented as a sequence of 8-dimensional vectors, where each dimension corresponds to the activation level of a specific muscle group at a given time step. Any invalid or missing values in the EMG sequences are replaced with valid numerical entries from the previous time step to preserve temporal continuity.

Normalization is performed on a per-channel basis by computing the mean and standard deviation independently for each EMG channel. This step helps reduce differences in signal scale across muscle groups and improves numerical stability. Channels with negligible variation are handled safely, and a small constant is included during scaling to prevent numerical issues.

In addition to normalization, the temporal resampling procedure described in Algorithm ??? is applied to the EMG sequences. By randomly stretching or compressing sequences through linear interpolation, this procedure introduces controlled temporal variations that act as a form of noise during training. This helps the model become less sensitive to small timing fluctuations and sensor noise, while also improving robustness to differences in sampling frequency across datasets.

3.2 Model Architectures

In this section, we describe the architectures of the four encoders and four decoders designed for each modality—3D keypoint sequences (pose), IMU signals, EMG signals, and text.

3.2.1 Pose Encoder

The pose encoder maps a single 3D body pose, represented by one time step of keypoints, into a compact embedding, as shown in Figure 3.1. As discussed in Section 3.1, for one pose instance, 72 pose parameters θ are extracted. These parameters are converted into a 6D rotation representation, resulting in a feature vector of size 114. The 6D representation is used to provide a continuous and stable encoding of joint rotations, avoiding discontinuities associated with angle-based pose representations.

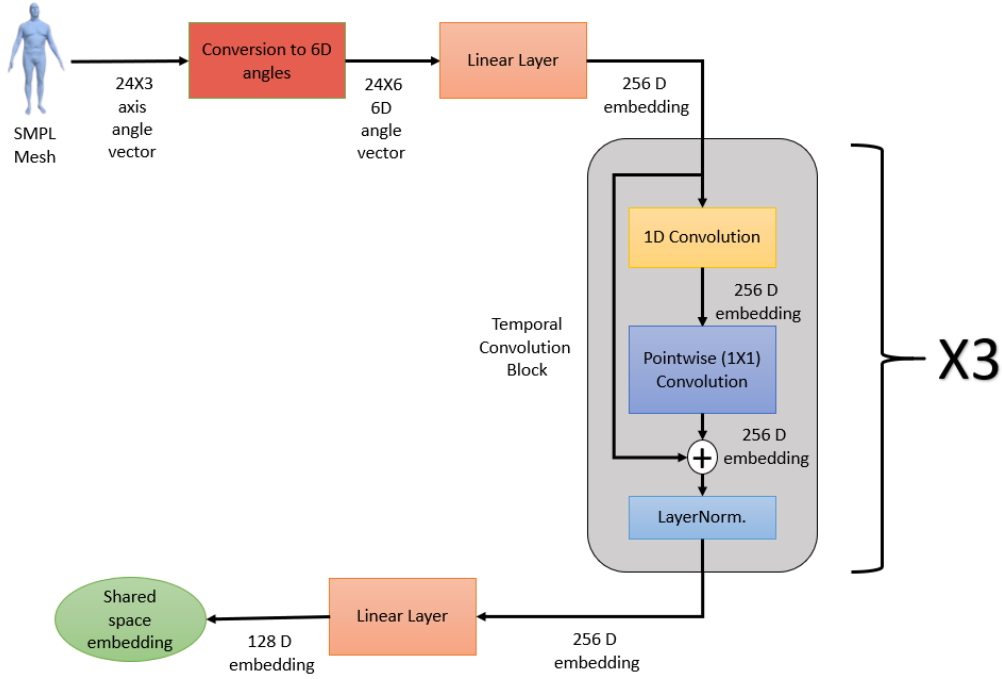


Figure 3.1: Architecture of the Pose Encoder used to map 3D keypoint sequences into a shared embedding space.

The 114-dimensional input vector is first passed through a linear projection layer that maps it to a 256-dimensional feature vector in the shared embedding space. This projection increases the representational capacity of the network.

The resulting feature vector is then processed by three stacked temporal convolutional containers. Each container consists of a one-dimensional convolution layer, followed by a pointwise (1×1) convolution and a layer normalization layer, all operating on the 256-dimensional feature space. The convolution layers act as learnable feature extractors that

refine and reweight pose features, while the pointwise convolution enables interactions between feature channels. A residual connection adds the input of each container to the output of the pointwise convolution, allowing the network to preserve original pose information and improve training stability. Layer normalization is applied to ensure stable feature distributions across samples.

Finally, the output of the third convolutional container is passed through a linear layer that reduces the feature dimensionality from 256 to 128. This produces a compact pose embedding, which serves as the final output of the pose encoder and is used as the shared latent representation. Since, the 3D Keypoint sequences do not have any time based dependencies, hence no sequential models or layers have been used in this architecture.

3.2.2 Pose Decoder

The pose decoder architecture is largely symmetric to the pose encoder and is responsible for reconstructing pose information from the shared embedding space, as illustrated in Figure 3.2. The decoder takes a 128-dimensional embedding from the shared latent space as input.

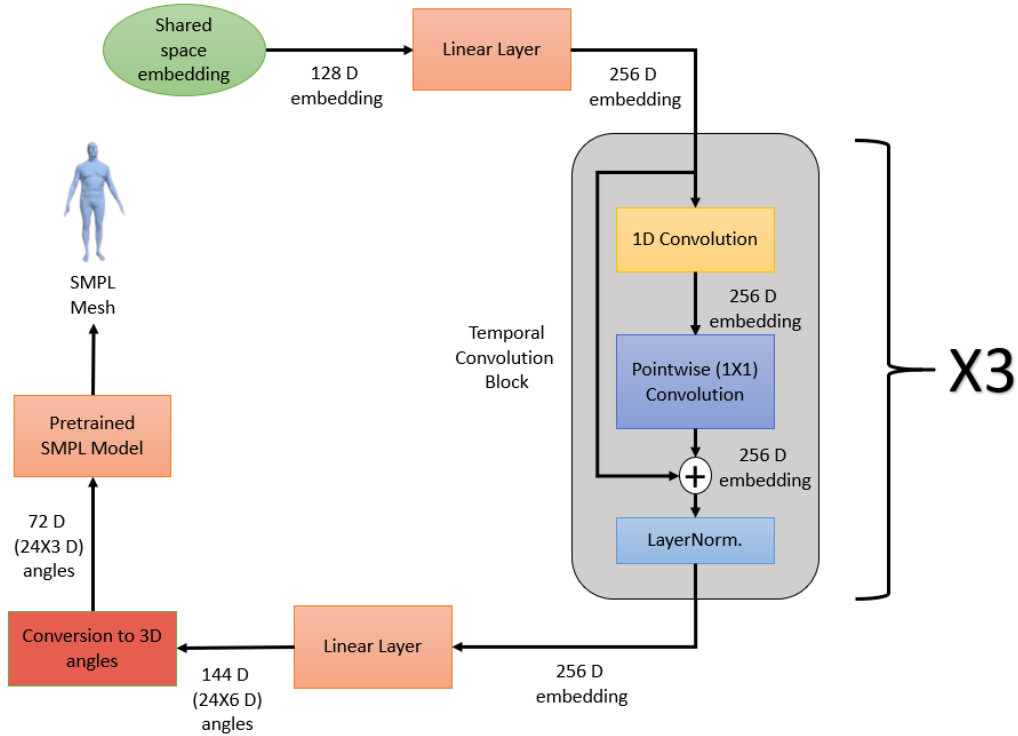


Figure 3.2: Architecture of the Pose Decoder, which reconstructs pose parameters from the shared embedding space.

The input embedding is first passed through a linear layer that expands the feature dimensionality to 256. This expansion restores the representational capacity required for pose reconstruction and aligns the feature size with the internal decoder blocks.

The resulting 256-dimensional feature vector is then processed by three consecutive temporal convolutional containers. Each container consists of a one-dimensional convolution layer, followed by a pointwise (1×1) convolution and a layer normalization layer, all operating on the 256-dimensional feature space. Similar to the pose encoder, these containers are used as structured feature transformation blocks rather than temporal sequence models, since no explicit temporal dependencies exist between pose instances. The convolution layers refine pose-related features, while the pointwise convolutions enable channel-wise interactions, and layer normalization ensures stable feature distributions during decoding.

After the three temporal convolutional containers, a 256-dimensional feature representation is obtained. This representation is then passed through a final linear layer that maps it to a 144-dimensional output. These 144 values correspond to the predicted 6D rotation representations of the pose parameters. The 6D rotations can be further converted to axis-angle representations, which are subsequently used by a pretrained SMPL model to generate the corresponding 3D human mesh.

3.2.3 IMU Encoder

The IMU encoder maps inertial measurement unit (IMU) signals into the shared embedding space, as shown in Figure 3.3. As discussed in Section 3.1, the IMU signal is represented as a sequence of size $(T, 204)$, where T denotes the number of time steps. Since IMU measurements exhibit strong temporal dependencies across time, a transformer-based architecture is employed.

The input IMU sequence is first passed through a linear projection layer that maps the feature dimensionality from 204 to 256, resulting in a $(T, 256)$ embedding. To preserve temporal ordering information, sinusoidal positional encodings of size $(T, 256)$ are added to the projected embeddings.

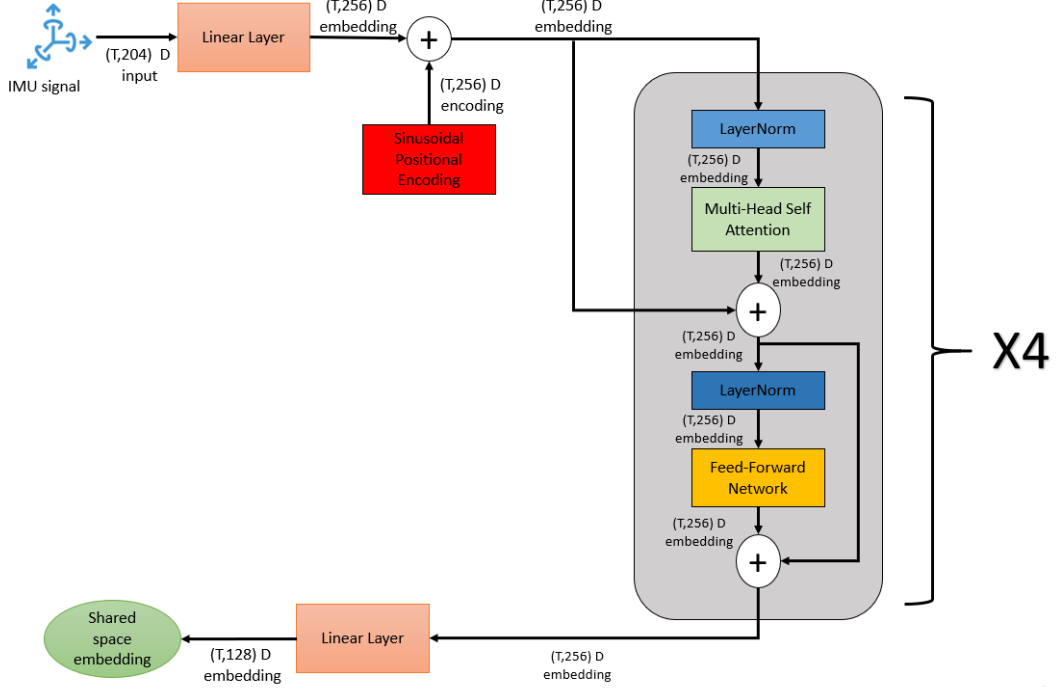


Figure 3.3: Transformer-based IMU Encoder architecture used to encode temporally dependent IMU signals into the shared embedding space.

The position-enhanced embeddings are then processed by four consecutive transformer encoder layers. Each encoder layer begins with layer normalization applied to the $(T, 256)$ input, followed by a multi-head self-attention (MHSA) module with 8 attention heads. For the i -th attention head, the self-attention operation is defined as

$$\text{Attention}_i(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d_k}}\right) V_i, \quad (3.1)$$

where d_k denotes the dimensionality of the key vectors.

The query, key, and value matrices for each attention head are obtained via learned linear projections of the layer input. Given the normalized input feature matrix $X \in \mathbb{R}^{T \times 256}$, these projections are computed as

$$Q_i = XW_i^Q, \quad K_i = XW_i^K, \quad V_i = XW_i^V, \quad (3.2)$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{256 \times d_k}$ are learnable weight matrices, and $d_k = 256/h$ with $h = 8$ attention heads. These projections allow each attention head to focus on different temporal relationships within the IMU sequence.

The outputs from all attention heads are concatenated and linearly projected to produce a $(T, 256)$ representation. This output is added to the input of the encoder layer through a residual connection. Layer normalization is then applied, followed by a position-

wise feed-forward network (FFN), defined as

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2, \quad (3.3)$$

which preserves the $(T, 256)$ dimensionality. A second residual connection adds the FFN output to its input, yielding the final output of the encoder layer.

This sequence of operations is repeated across all four transformer encoder layers, enabling the model to capture both short-term and long-range temporal dependencies in the IMU signals. After the final encoder layer, a linear projection maps the $(T, 256)$ embedding to a $(T, 128)$ representation. This output constitutes the IMU embedding in the shared embedding space and is used for downstream decoding and cross-modal alignment.

3.2.4 IMU Decoder

The IMU decoder reconstructs inertial sensor signals from the shared embedding space, as illustrated in Figure 3.4. The decoder takes a sequence of embeddings of size $(T, 128)$ as input.

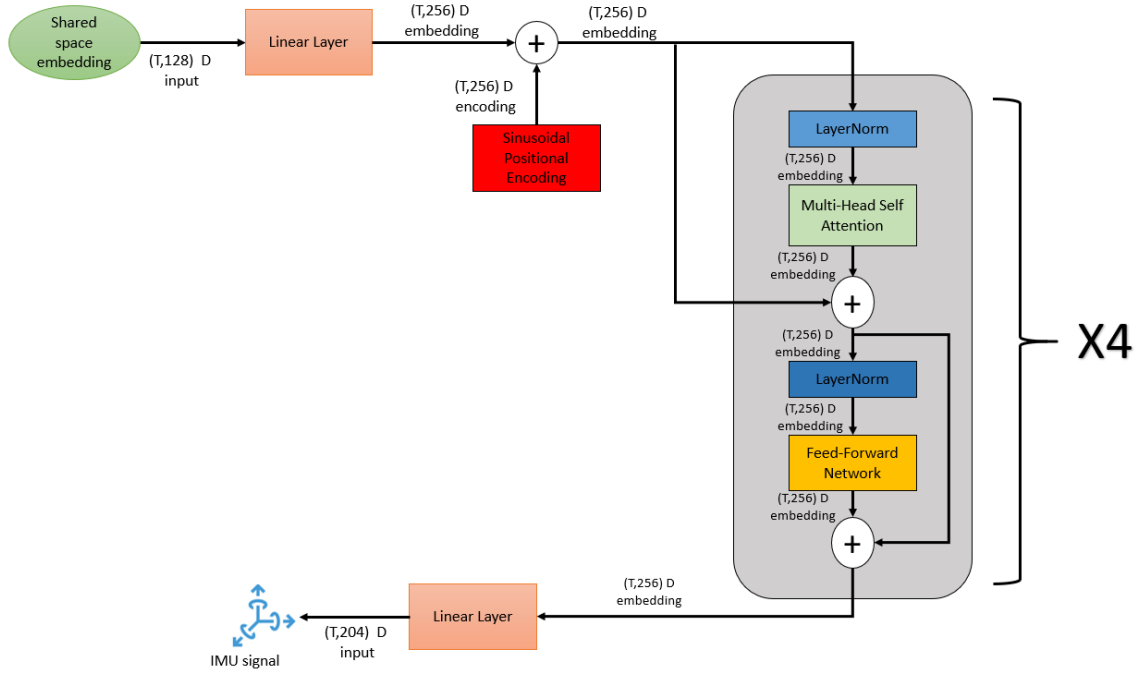


Figure 3.4: Architecture of the IMU Decoder used to reconstruct IMU signals from the shared embedding space.

The input embeddings are first passed through a linear layer that expands the feature dimensionality from 128 to 256, producing a $(T, 256)$ representation. To retain temporal ordering information, sinusoidal positional encodings are added to the expanded embeddings before further processing.

The position-enhanced features are then passed through four consecutive transformer

decoder blocks. Each decoder block is architecturally identical to the transformer encoder blocks used in the IMU encoder. Specifically, each block consists of an initial layer normalization, followed by a multi-head self-attention layer, a residual connection, a second layer normalization, and a position-wise feed-forward network, all operating in the 256-dimensional feature space. These blocks enable the decoder to model temporal dependencies and reconstruct temporally consistent IMU signals.

After the four decoder blocks, a $(T, 256)$ feature representation is obtained. This representation is finally passed through a linear layer that maps the features back to the original IMU signal space, producing a reconstructed IMU sequence of size $(T, 204)$.

3.2.5 EMG Encoder

The architecture of the EMG encoder is shown in Figure 3.5. The EMG encoder follows the same architectural design as the IMU encoder, differing only in the input dimensionality. For an EMG signal sequence of T time steps, the input is represented as a $(T, 8)$ feature sequence corresponding to the EMG channels.

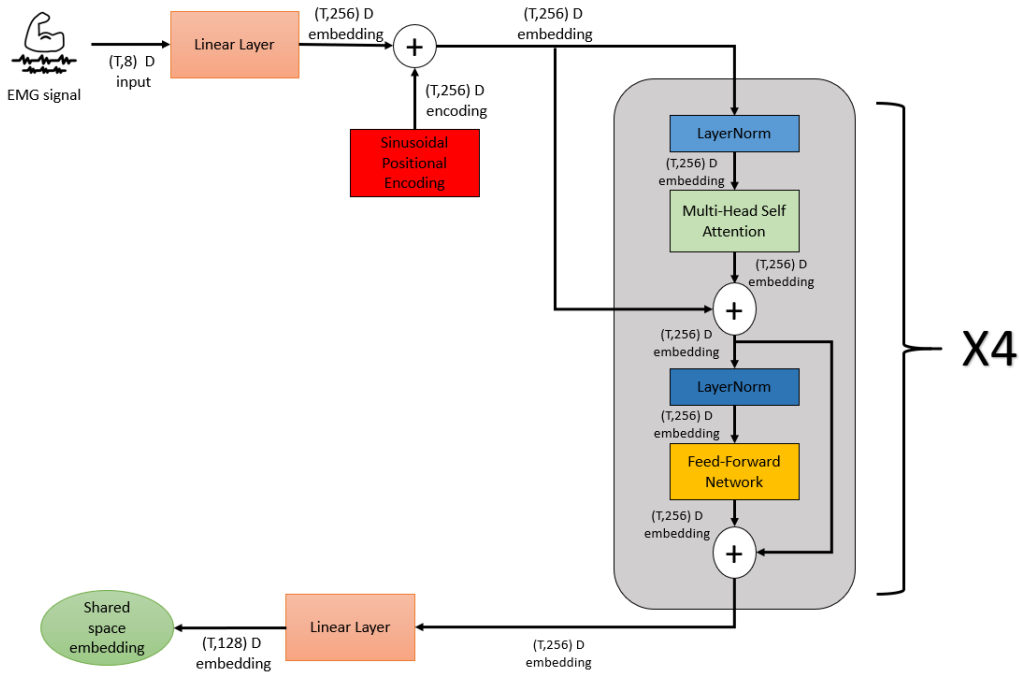


Figure 3.5: Architecture of the EMG Encoder. The network maps EMG signal sequences into the shared embedding space.

The input EMG sequence is first passed through a linear layer that projects the features into a higher-dimensional space, producing a $(T, 256)$ embedding. This projection allows the model to learn expressive representations of muscle activation patterns and aligns the EMG modality with the shared embedding space.

The projected features are then processed by a stack of temporal encoder containers

identical to those used in the IMU encoder. Each container consists of a normalization layer, followed by a Multi Head Self Attention Layer, residual connection, another normalization Layer, a Feed Forward Network and finally another Residual Connection.

The output of the temporal encoder stack is passed through a linear layer that reduces the dimensionality from 256 to 128, resulting in a $(T, 128)$ embedding. This embedding forms the final EMG representation in the shared embedding space and is used by downstream modules for decoding and cross-modal alignment.

3.2.6 EMG Decoder

The architecture of the EMG decoder is shown in Figure 3.6. The EMG decoder is responsible for reconstructing EMG signal sequences from the shared embedding representation. It follows an architecture symmetric to the EMG encoder, with the direction of information flow reversed.

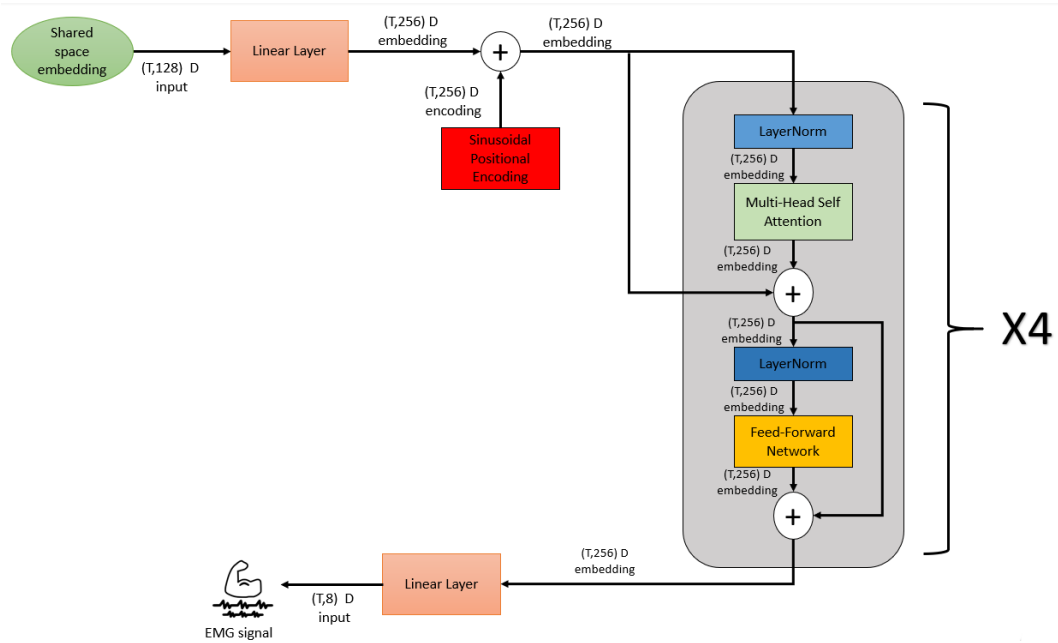


Figure 3.6: Architecture of the EMG Decoder. The network reconstructs EMG signal sequences from the shared embedding space.

The input to the decoder is a $(T, 128)$ embedding obtained from the shared latent space. This embedding is first passed through a linear layer that projects it into a higher-dimensional representation of size $(T, 256)$. This expansion provides sufficient capacity for reconstructing detailed temporal EMG patterns.

The expanded features are then processed by a stack of temporal decoder containers identical in structure to those used in the encoder. Each container consists of a normalization layer, followed by a Multi Head Self Attention Layer, residual connection, another normalization Layer, a Feed Forward Network and finally another residual connection. The shape of the feature map remains $(T, 256)$ throughout the decoder block.

Finally, the output of the temporal decoder stack is passed through a linear layer that maps the $(T, 256)$ representation to the EMG signal space, producing a reconstructed sequence of size $(T, 8)$. This output corresponds to the reconstructed EMG channels and forms the final output of the EMG decoder.

3.2.7 Text Decoder

The architecture of the text decoder is shown in Figure 3.7. The decoder follows a transformer-based design and generates text in an auto-regressive manner, where each next word is predicted based on previously generated tokens and a conditioning memory derived from the shared embedding space.

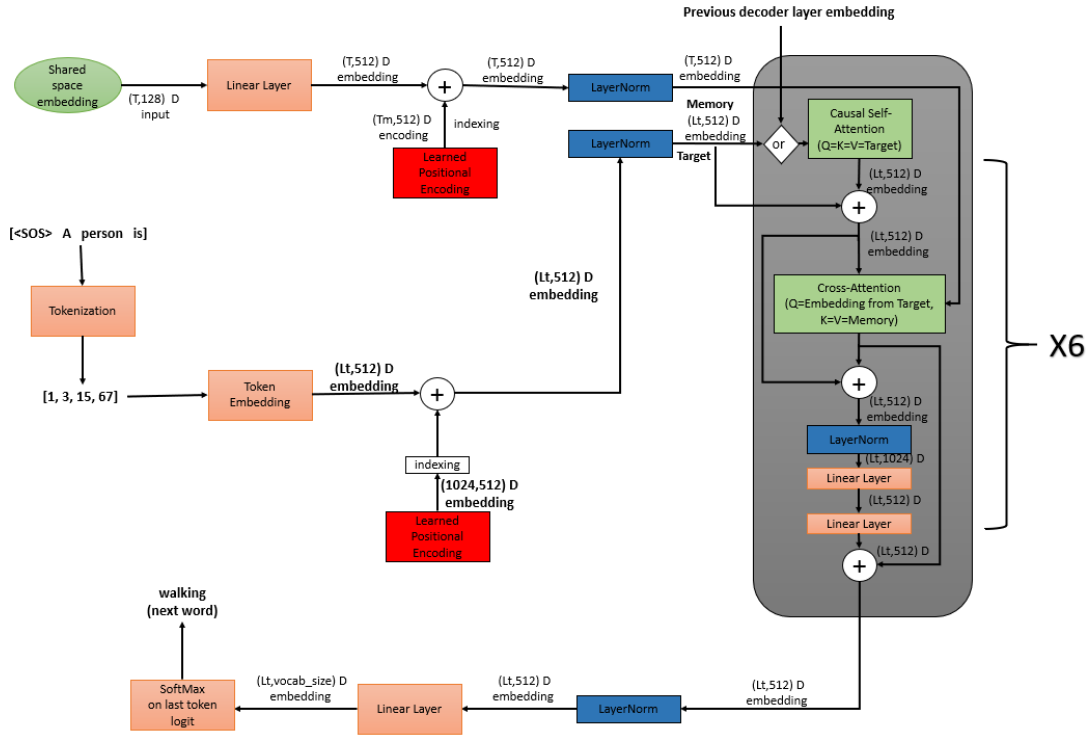


Figure 3.7: Transformer-based text decoder architecture. The decoder predicts the next token conditioned on previously generated text (target) and a memory sequence obtained from the shared embedding space.

Assume that the decoder has already generated the partial sentence “< sos> A person is” and is required to predict the next word. This partial sentence is tokenized into indices, for example $[1, 3, 15, 67]$, resulting in a target length $L_t = 4$. These token indices are mapped through a token embedding layer to obtain a sequence of embeddings of size $(L_t, 512)$. Token embeddings are used to convert discrete word indices into continuous representations suitable for neural processing.

To incorporate word-order information, a learned positional encoding table of maximum length $T_m = 1024$ and dimension 512 is used. The first L_t positional encodings are selected and added to the token embeddings. This addition allows the decoder to

distinguish between tokens appearing at different positions in the sentence. The resulting representation is passed through a layer normalization layer, producing the target embedding

$$\text{target} \in \mathbb{R}^{L_t \times 512}.$$

Layer normalization is applied here to stabilize training and ensure consistent feature scaling before attention operations.

In parallel, a memory sequence is constructed from the shared embedding space. The shared embedding has size $(T, 128)$, where T denotes the temporal length of the input representation. Since the decoder operates in a 512-dimensional space, the shared embedding is first projected using a linear layer to obtain a $(T, 512)$ representation. This projection aligns the modality-agnostic shared embedding with the text decoder feature dimension. Positional encodings from the same learned table are then added using the first T positions to preserve the temporal ordering of the memory sequence. The resulting features are passed through a layer normalization layer, yielding the memory embedding

$$\text{memory} \in \mathbb{R}^{T \times 512}.$$

The target and memory embeddings are then processed by a stack of six identical decoder blocks. Each decoder block consists of a causal self-attention layer, a cross-attention layer, and a position-wise feed-forward network, with residual additions and layer normalization applied between sub-layers.

Within each decoder block, the first operation is causal self-attention applied to the target embedding. Causal masking is used to ensure that each token attends only to itself and to previously generated tokens, enforcing the left-to-right generation constraint. Given the block input $H \in \mathbb{R}^{L_t \times 512}$, the query, key, and value matrices are computed as

$$Q = HW^Q, \quad K = HW^K, \quad V = HW^V, \quad (3.4)$$

and causal self-attention is defined as

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)V, \quad (3.5)$$

where M is a causal mask that blocks access to future positions. The output of this attention layer is added to the original target representation using a residual connection, which helps preserve linguistic context and improves gradient flow. Layer normalization is then applied to stabilize the resulting features.

Next, a cross-attention layer integrates information from the memory sequence into the target stream. In this layer, queries are obtained from the target representation, while keys and values are derived from the memory:

$$Q_c = H_t W^{Q_c}, \quad K_c = \text{memory} W^{K_c}, \quad V_c = \text{memory} W^{V_c}. \quad (3.6)$$

The cross-attention output allows each target token to attend to relevant parts of the shared embedding sequence, effectively grounding text generation in the input representation. This output is added to the target stream via a residual connection and followed by layer normalization to maintain stable feature distributions.

The final sub-layer in each decoder block is a position-wise feed-forward network (FFN), which applies two linear transformations with a non-linearity in between:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2, \quad (3.7)$$

where the intermediate dimensionality is expanded from 512 to 1024 and then reduced back to 512. This expansion–compression structure increases the representational capacity of the model and enables non-linear feature refinement beyond attention mechanisms. The FFN output is added residually to its input and normalized, completing one decoder block.

After passing through all six decoder blocks, the resulting representation has size $(L_t, 512)$. A final linear layer maps this representation to vocabulary logits of size $(L_t, |\mathcal{V}|)$. A softmax function converts these logits into token probabilities. During generation, the token with the highest probability at the last position is selected as the next word. In the given example, this process results in the prediction of the word “**walking**” as the next generated token.

3.2.8 Text Encoder

The text encoder pipeline is built largely from pretrained components, since training a complete text-to-motion system from scratch typically requires large paired datasets and significant compute, particularly to learn stable discrete motion representations and variable-length sequence generation. Figure 3.8 illustrates the overall architecture.

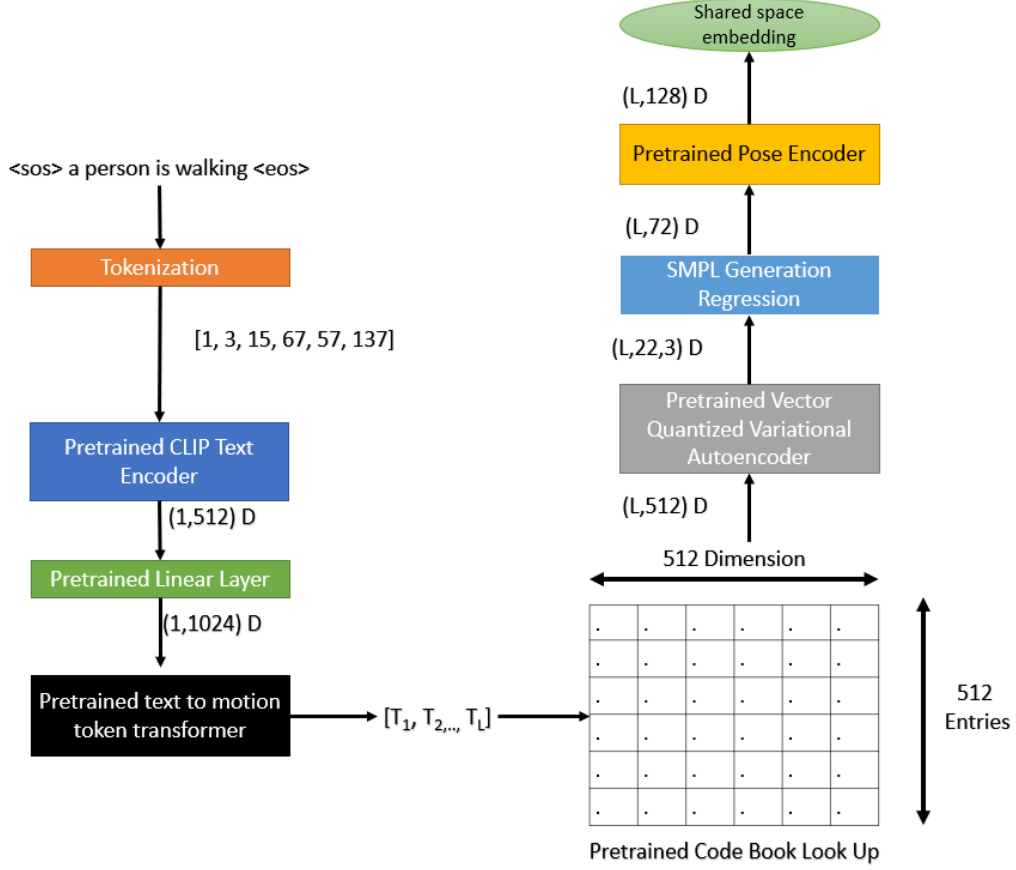


Figure 3.8: Text encoder pipeline composed of pretrained components. The input text is mapped to a fixed-dimensional semantic embedding using a CLIP text encoder, followed by a text-to-motion transformer that generates discrete motion tokens. These tokens are decoded into 3D joint trajectories using a VQ-VAE and subsequently converted to SMPL pose parameters via regression before being projected into the shared embedding space.

Given an input sentence, the text is first converted into a sequence of tokens using the tokenizer associated with the pretrained CLIP model. These tokens may have variable length depending on the input. The pretrained CLIP text encoder then maps the token sequence into a single global sentence embedding of size $(1, 512)$. This fixed-length embedding compresses the semantic content of the entire sentence into one vector, which makes it convenient to use as a conditioning signal for downstream modules.

Next, a pretrained linear projection maps the CLIP embedding from $(1, 512)$ to $(1, 1024)$. This projection is used to match the internal feature dimension expected by the pretrained text-to-motion transformer and to provide higher representational capacity for conditioning the token generator.

The projected embedding is then passed to a pretrained text-to-motion token transformer, which is trained to generate a variable-length sequence of discrete motion tokens of length L .

$$[T_1, T_2, \dots, T_L],$$

where each T_t is an integer index. These indices do not directly represent joint coordinates; instead, they refer to entries in a learned motion codebook.

A pretrained codebook lookup is applied next. The codebook contains $K = 512$ learned motion embeddings, each of dimension 512, i.e., a table of size $(512, 512)$. Each discrete token $T_t \in \{1, \dots, 512\}$ selects one corresponding codebook vector, producing a sequence of embeddings of size $(L, 512)$. This step converts the discrete token sequence into a continuous latent motion representation suitable for decoding.

The resulting latent sequence is decoded using a pretrained Vector Quantized Variational Autoencoder (VQ-VAE), producing a motion sequence in 3D joint space. Specifically, the decoder outputs 3D joint trajectories of shape $(L, 22, 3)$. Although this representation contains explicit 3D joints, it is not in SMPL parameter format. Therefore, the predicted joints are converted to SMPL pose parameters using the regression procedure described in Algorithm 3.2, resulting in an SMPL pose sequence of shape $(L, 72)$.

Algorithm 3.2: 3D Keypoint to SMPL Pose Regression

- 1: **Input:** 3D joint sequence $J \in \mathbb{R}^{T \times 22 \times 3}$
- 2: **Output:** SMPL pose parameters $\Theta \in \mathbb{R}^{T \times 72}$
- 3: Initialize SMPL shape parameters $\beta \in \mathbb{R}^{10}$ to zeros
- 4: Fix global orientation $R \in \mathbb{R}^{T \times 3}$ to zeros
- 5: Initialize body pose parameters $P \in \mathbb{R}^{T \times 69}$ to zeros
- 6: Load SMPL body model with fixed shape parameters
- 7: **for** each optimization iteration **do**
- 8: Generate SMPL joints $\hat{J}$ using fixed R and current P
- 9: Select the subset of SMPL joints corresponding to the 22 input joints
- 10: Compute joint reconstruction loss:

$$\mathcal{L} = \frac{1}{T} \sum_{i=1}^T \left\| \hat{J}_i - J_i \right\|_2^2$$

- 11: Update body pose parameters P using gradient-based optimization to minimize $\mathcal{L}$
- 12: **end for**
- 13: Construct SMPL pose vector by concatenating fixed zeros for global orientation with optimized body pose:

$$\Theta = [\mathbf{0}_{T \times 3} \mid P]$$

- 14: **return** Θ
-

Finally, the generated SMPL pose sequence is passed through the already trained pose encoder to obtain embeddings in the shared embedding space. This ensures that text-generated motion can be represented in the same latent space used by the other modalities and downstream modules.

3.3 Training Strategy

This section describes the training procedures for the pose encoder and decoder, the EMG encoder and decoder, the IMU encoder and decoder, and the text decoder.

Pose Encoder–Decoder Training Procedure

The pose encoder and pose decoder are trained jointly using a reconstruction-based objective, as summarized in Algorithm 3.3. The training process operates on pose sequences and their corresponding temporal masks, where the mask indicates valid frames within each sequence.

Algorithm 3.3: Training Procedure for Pose Encoder–Decoder

- 1: **Input:** Training set $\mathcal{D}_{train}$, validation set $\mathcal{D}_{val}$
 - 2: **Output:** Trained pose encoder E and pose decoder D
 - 3: Initialize encoder E and decoder D
 - 4: Initialize optimizer (AdamW) over parameters of E and D
 - 5: Initialize learning rate scheduler with warmup followed by cosine decay
 - 6: **for** each epoch = 1 to N **do**
 - 7: Set E, D to training mode
 - 8: **for** each batch (X, M) in $\mathcal{D}_{train}$ **do**
 - 9: Convert axis-angle poses X to 6D rotation representation
 - 10: Add noise to valid frames using mask M
 - 11: Encode poses: $Z = E(X, M)$
 - 12: Reconstruct poses: $\hat{X} = D(Z, M)$
 - 13: Compute reconstruction loss (L1) on valid frames
 - 14: Backpropagate loss and update E, D
 - 15: **end for**
 - 16: Set models to evaluation mode
 - 17: **for** each batch (X, M) in $\mathcal{D}_{val}$ **do**
 - 18: Convert poses to 6D representation
 - 19: Encode and decode to obtain reconstruction
 - 20: Compute validation reconstruction loss
 - 21: **end for**
 - 22: Update learning rate scheduler
 - 23: Save model checkpoint if validation loss improves
 - 24: **end for**
 - 25: **return** trained encoder E and decoder D
-

Both networks are optimized using the AdamW optimizer, together with a learning rate schedule that consists of an initial warmup phase followed by cosine decay. For each training batch, the input pose sequences are originally represented in axis-angle format and are first converted into a 6D rotation representation, which provides a continuous and numerically stable encoding of joint rotations.

To improve robustness and generalization, noise is added to the valid pose frames according to the temporal mask. The noisy pose sequence is then passed through the pose encoder to obtain a latent representation, which is subsequently reconstructed by the pose decoder. The training objective is defined as an L1 reconstruction loss computed in the 6D rotation space and evaluated only over valid frames. This loss, referred to as the *pose reconstruction loss*, is defined as

$$\mathcal{L}_{\text{pose}} = \frac{1}{\sum M} \sum_{t=1}^T M_t \left\| \hat{X}_t - X_t \right\|_1, \quad (3.8)$$

where X_t and $\hat{X}_t$ denote the ground-truth and reconstructed 6D pose representations at time step t , respectively, M_t is the binary validity mask, and $\sum M$ denotes the total number of valid frames. This formulation ensures that padded or invalid frames do not contribute to the loss.

After each epoch, the model is evaluated on the validation set using the same L1 pose reconstruction loss. The learning rate scheduler is updated at the end of each epoch, and model checkpoints are saved whenever an improvement in the validation loss is observed. This training procedure enables the pose encoder and decoder to learn a compact and consistent latent representation that supports accurate pose reconstruction and downstream cross-modal alignment.

IMU and EMG Encoder–Decoder Training Procedure

The training procedure for the IMU and EMG encoders is summarized in Algorithm 3.4. A teacher–student learning strategy is adopted, where the pose encoder and pose decoder serve as a fixed supervisory backbone. Since the pose encoder–decoder pair is pretrained to produce meaningful latent representations and accurate pose reconstructions, their parameters are kept frozen throughout this stage of training.

Algorithm 3.4: Training Procedure for IMU / EMG Encoder with Frozen Pose Networks

- 1: **Input:** Training set $\mathcal{D}_{train}$, validation set $\mathcal{D}_{val}$
 - 2: **Output:** Trained sensor encoder E_s (IMU or EMG)
 - 3: Load pretrained pose encoder E_p and pose decoder D_p
 - 4: Freeze parameters of E_p and D_p
 - 5: Initialize sensor encoder E_s
 - 6: Initialize optimizer (AdamW) over parameters of E_s
 - 7: Initialize learning rate scheduler (cosine annealing)
 - 8: **for** each epoch = 1 to N **do**
 - 9: Set E_s to training mode
 - 10: **for** each batch (X_p, X_s, M) in $\mathcal{D}_{train}$ **do**
 - 11: Convert pose inputs X_p to 6D rotation representation
 - 12: Compute pose embeddings $Z_p = E_p(X_p, M)$
 - 13: Compute sensor embeddings $Z_s = E_s(X_s, M)$
 - 14: Compute embedding alignment loss $\mathcal{L}_{emb} = \|Z_s - Z_p\|_2^2$ (masked)
 - 15: Reconstruct poses $\hat{X}_p = D_p(Z_s, M)$
 - 16: Compute pose reconstruction loss $\mathcal{L}_{pose} = \|\hat{X}_p - X_p\|_2^2$ (masked)
 - 17: Compute total loss $\mathcal{L} = \mathcal{L}_{emb} + \lambda\mathcal{L}_{pose}$
 - 18: Backpropagate loss and update E_s
 - 19: **end for**
 - 20: Set E_s to evaluation mode
 - 21: **for** each batch (X_p, X_s, M) in $\mathcal{D}_{val}$ **do**
 - 22: Compute validation losses $\mathcal{L}_{emb}$ and $\mathcal{L}_{pose}$
 - 23: **end for**
 - 24: Update learning rate scheduler
 - 25: Save model checkpoint if validation loss improves
 - 26: **end for**
 - 27: **return** trained sensor encoder E_s
-

For each training batch, paired pose and sensor data are provided along with a temporal validity mask. The pose sequences are first converted to a 6D rotation representation and passed through the frozen pose encoder to obtain pose embeddings. In parallel, the corresponding IMU or EMG signals are processed by the trainable sensor encoder to produce sensor embeddings in the same latent space. An embedding alignment loss is computed between the sensor embeddings and the pose embeddings, encouraging the sensor encoder to match the latent structure learned from pose data.

To further constrain the learned representations, the sensor embeddings are passed through the frozen pose decoder to reconstruct pose sequences. A pose reconstruction loss is computed by comparing the reconstructed poses with the ground-truth poses in the 6D rotation space, evaluated only over valid frames. The total training objective is defined as the sum of the embedding alignment loss and the pose reconstruction loss, ensuring that

the sensor encoder learns embeddings that are both latent-aligned and pose-consistent.

During validation, the same losses are computed without parameter updates to monitor generalization performance. A cosine annealing learning rate schedule is applied across epochs, and model checkpoints are saved based on validation performance. This training strategy allows the IMU and EMG encoders to learn modality-specific mappings into a shared pose-centric embedding space while leveraging the strong supervision provided by the frozen pose encoder-decoder.

IMU and EMG Decoder Training Procedure

The training procedure for the IMU and EMG decoders is summarized in Algorithm 3.5. In this stage, the goal is to learn a sensor-specific decoder that maps pose-centric latent representations back to the corresponding sensor signal space, while preserving consistency with the original pose information. To achieve this, the pose encoder, pose decoder, and the corresponding sensor encoder are kept frozen, and only the parameters of the sensor decoder are updated.

For each training batch, pose sequences are first converted to a 6D rotation representation and passed through the frozen pose encoder to obtain pose embeddings. These embeddings are then used as input to the trainable sensor decoder to reconstruct the corresponding IMU or EMG signals. A sensor reconstruction loss is computed between the reconstructed sensor signals and the ground-truth sensor measurements, evaluated only over valid time steps as indicated by the temporal mask.

To ensure that the reconstructed sensor signals remain informative with respect to human motion, the predicted sensor signals are passed through the frozen sensor encoder and subsequently decoded back into pose space using the frozen pose decoder. A pose consistency loss is computed by comparing the reconstructed poses with the original poses in the 6D rotation space. The total training objective is defined as a combination of the sensor reconstruction loss and the pose consistency loss, enforcing both signal-level accuracy and pose-level consistency.

Algorithm 3.5: Training Procedure for IMU / EMG Decoder with Frozen Encoders

- 1: **Input:** Training set $\mathcal{D}_{train}$, validation set $\mathcal{D}_{val}$
 - 2: **Output:** Trained sensor decoder D_s (IMU or EMG)
 - 3: Load pretrained pose encoder E_p , pose decoder D_p , and sensor encoder E_s
 - 4: Freeze parameters of E_p , D_p , and E_s
 - 5: Initialize sensor decoder D_s
 - 6: Initialize optimizer (AdamW) over parameters of D_s
 - 7: Initialize learning rate scheduler (cosine annealing)
 - 8: **for** each epoch = 1 to N **do**
 - 9: Set D_s to training mode
 - 10: **for** each batch (X_p, X_s, M) in $\mathcal{D}_{train}$ **do**
 - 11: Convert pose inputs X_p to 6D rotation representation
 - 12: Compute pose embeddings $Z_p = E_p(X_p, M)$
 - 13: Reconstruct sensor signals $\hat{X}_s = D_s(Z_p, M)$
 - 14: Compute sensor reconstruction loss $\mathcal{L}_{sens} = \|\hat{X}_s - X_s\|_2^2$ (masked)
 - 15: Compute sensor embeddings $Z_s = E_s(\hat{X}_s, M)$
 - 16: Reconstruct poses $\hat{X}_p = D_p(Z_s, M)$
 - 17: Compute pose consistency loss $\mathcal{L}_{pose} = \|\hat{X}_p - X_p\|_2^2$ (masked)
 - 18: Compute total loss $\mathcal{L} = \lambda_{sens}\mathcal{L}_{sens} + \lambda_{pose}\mathcal{L}_{pose}$
 - 19: Backpropagate loss and update D_s
 - 20: **end for**
 - 21: Set D_s to evaluation mode
 - 22: **for** each batch (X_p, X_s, M) in $\mathcal{D}_{val}$ **do**
 - 23: Compute validation losses $\mathcal{L}_{sens}$ and $\mathcal{L}_{pose}$
 - 24: **end for**
 - 25: Update learning rate scheduler
 - 26: Save model checkpoint if validation loss improves
 - 27: **end for**
 - 28: **return** trained sensor decoder D_s
-

During validation, the same loss terms are computed without updating model parameters to monitor generalization performance. A cosine annealing learning rate schedule is applied across training epochs, and model checkpoints are saved whenever an improvement in the validation loss is observed. This training strategy enables the IMU and EMG decoders to learn accurate inverse mappings from the shared latent space to sensor measurements while remaining consistent with the underlying pose representation.

Text Decoder Training Procedure

The training procedure for the pose-to-text model is summarized in Algorithm 3.6. In this stage, the objective is to train the text decoder to generate natural language descriptions

conditioned on motion information encoded by the pose encoder. The pose encoder is kept frozen throughout training and serves as a fixed feature extractor that maps pose sequences into a sequence of latent embeddings.

Algorithm 3.6: Training Procedure for Text Decoder

- 1: **Input:** Training set $\mathcal{D}_{train}$, validation set $\mathcal{D}_{val}$
 - 2: **Output:** Trained pose encoder E_p and text decoder D_t
 - 3: Initialize pose encoder E_p and text decoder D_t
 - 4: Freeze parameters of E_p
 - 5: Initialize optimizer (AdamW) over trainable parameters
 - 6: Initialize learning rate scheduler (cosine annealing)
 - 7: Define cross-entropy loss with padding ignored
 - 8: **for** each epoch = 1 to N **do**
 - 9: Set models to training mode
 - 10: **for** each batch (X_p, Y, M_t) in $\mathcal{D}_{train}$ **do**
 - 11: Build frame-valid mask M_p from pose sequence X_p
 - 12: Convert poses to 6D rotation representation if required
 - 13: Encode poses to obtain memory embeddings $Z = E_p(X_p, M_p)$
 - 14: Decode text logits conditioned on Z :
 - 15: $\hat{Y} = D_t(Z, Y, M_p, M_t)$
 - 16: Compute text generation loss $\mathcal{L}_{text}$ (cross-entropy)
 - 17: Backpropagate loss and update trainable parameters
 - 18: **end for**
 - 19: Set models to evaluation mode
 - 20: **for** each batch (X_p, Y, M_t) in $\mathcal{D}_{val}$ **do**
 - 21: Encode poses and decode text as above
 - 22: Compute validation text loss
 - 23: **end for**
 - 24: Update learning rate scheduler
 - 25: Save model checkpoint if validation loss improves
 - 26: Stop early if no improvement is observed for a fixed patience
 - 27: **end for**
 - 28: **return** trained models E_p and D_t
-

For each training batch, pose sequences are first processed to construct a frame-valid mask that identifies valid temporal steps. When required, the pose representations are converted to a 6D rotation format and passed through the frozen pose encoder to obtain pose embeddings. These embeddings act as the memory input to the text decoder and provide the motion context required for text generation.

The text decoder is trained to generate the corresponding textual description conditioned on the pose embeddings. Given the target text sequence, the decoder predicts token probabilities at each time step using cross-attention over the pose embeddings. The training objective is defined using a cross-entropy loss computed between the predicted

token distributions and the ground-truth tokens, with padding tokens excluded from the loss computation. Gradients are backpropagated only through the text decoder, while the pose encoder remains fixed.

After each epoch, the model is evaluated on a held-out validation set using the same loss function to monitor generalization performance. A cosine annealing learning rate schedule is applied during training, and model checkpoints are saved based on validation loss. This training strategy enables the text decoder to learn a mapping from pose-centric latent representations to natural language descriptions while leveraging the stable and informative embeddings produced by the pretrained pose encoder.

3.4 Explainability Analysis of Learned Embeddings

To analyze the contribution of individual joints and muscle activations to the learned embedding space, an occlusion-based explainability method is employed, as summarized in Algorithm 3.7. The objective of this analysis is to quantify how sensitive the produced embeddings are to perturbations in specific pose joints or EMG channels.

Algorithm 3.7: Occlusion-Based XAI for Pose and EMG Embeddings

- 1: **Input:** Embedding sequence $E \in \mathbb{R}^{T \times 128}$
 - 2: **Output:** Importance scores for EMG channels and pose joints
 - 3: Decode embedding E into EMG signals $\hat{X}_{emg} \in \mathbb{R}^{T \times 8}$
 - 4: Decode embedding E into pose parameters $\hat{X}_{pose} \in \mathbb{R}^{T \times 72}$
 - 5: Compute baseline EMG embedding $Z_{base}^{emg} = E_{emg}(\hat{X}_{emg})$
 - 6: Compute baseline pose embedding $Z_{base}^{pose} = E_{pose}(\hat{X}_{pose})$
 - 7: **for** each EMG channel $c = 1$ to 8 **do**
 - 8: Occlude channel c in $\hat{X}_{emg}$ by setting it to zero
 - 9: Encode occluded signal to obtain Z_c^{emg}
 - 10: Compute importance score $\Delta_c = \|Z_{base}^{emg} - Z_c^{emg}\|_2$
 - 11: **end for**
 - 12: **for** each pose joint $j = 1$ to 23 **do**
 - 13: Occlude joint j in $\hat{X}_{pose}$ by zeroing its rotation parameters
 - 14: Encode occluded pose to obtain Z_j^{pose}
 - 15: Compute importance score $\Delta_j = \|Z_{base}^{pose} - Z_j^{pose}\|_2$
 - 16: **end for**
 - 17: Normalize all importance scores to $[0, 1]$
 - 18: **return** normalized EMG and pose importance scores
-

Given a motion embedding sequence, the embedding is first decoded into both pose parameters and EMG signals using the corresponding pretrained decoders. These decoded signals serve as the baseline representations for the explainability analysis. The baseline embeddings are then recomputed by passing the decoded pose and EMG signals through their respective encoders.

For EMG-based analysis, each muscle channel is occluded independently by setting its

activation values to zero across all time steps. The occluded EMG signal is re-encoded, and the change in the resulting embedding is measured using the Euclidean distance from the baseline embedding. A larger change indicates a higher contribution of that muscle channel to the embedding.

Similarly, for pose-based analysis, individual joints are occluded by zeroing their rotation parameters while keeping the rest of the pose unchanged. The perturbed pose sequence is re-encoded, and the deviation from the baseline embedding is computed. This deviation serves as a measure of the joint’s influence on the learned representation. The root joint is excluded from this analysis to focus on relative joint contributions.

Finally, all importance scores are normalized to enable comparison across joints and muscle channels. This occlusion-based approach provides an intuitive and model-agnostic means of interpreting how different components of human motion influence the learned embedding space.

To provide a qualitative illustration of the proposed explainability method, a representative example is shown for the *jumping jack* motion. Figure 3.9 visualizes the normalized importance scores for both EMG channels and pose joints. The EMG importance highlights muscle groups that are strongly engaged during the exercise, and the joint importance emphasizes the contribution of the different joints whilst carrying out the motion.

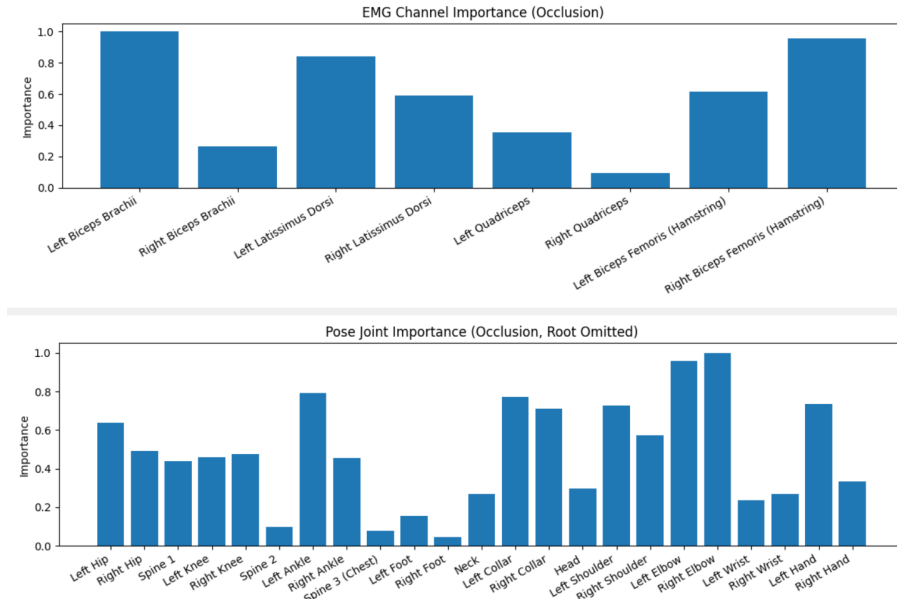


Figure 3.9: Occlusion-based explainability results for a jumping jack motion. Normalized importance scores indicate the relative contribution of individual EMG channels (top) and pose joints (bottom) to the learned embedding. Higher values correspond to greater sensitivity of the embedding to occlusion, highlighting key muscles and joints involved in the movement.

3.5 Graphical User Interface Development

In order to facilitate practical use of the encoders and decoders, a graphical user interface (GUI) has been developed, as shown in Figure 3.10. The interface provides access to all four encoders and four decoders within a unified environment, allowing users to easily perform cross-modal translations and reconstructions.

In addition to model inference, the GUI supports explainable AI (XAI) analysis and enables arithmetic operations directly in the latent embedding space. This makes it possible to inspect, manipulate, and interpret embeddings interactively, supporting both qualitative evaluation and exploratory analysis.

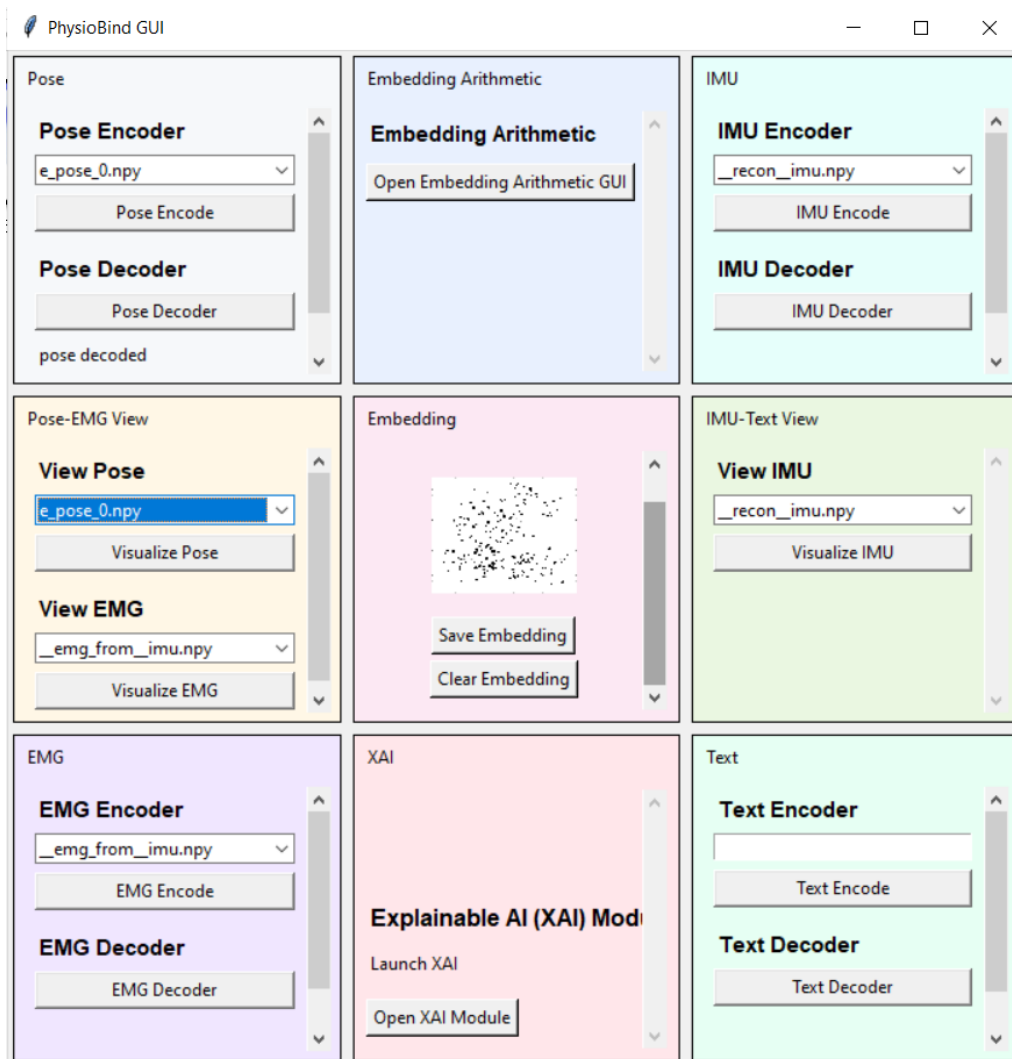


Figure 3.10: Graphical user interface developed for interacting with the encoders, decoders, and latent embedding space.

3.6 Embedding Arithmetic Module

The learned latent embeddings can be directly manipulated using arithmetic operations to generate meaningful cross-modal outputs. This capability enables the combination of motion characteristics originating from different sensing modalities within the shared embedding space. To support this functionality, an embedding arithmetic module was implemented as part of the developed graphical user interface.

An example of embedding arithmetic is shown in Figure 3.11. The first pose sequence is obtained from an IMU embedding and depicts a person bowing their head. The second pose sequence is derived from an EMG embedding and shows a person moving their feet up and down. By performing an element-wise addition of the two embeddings, a resultant embedding is produced and decoded into a pose sequence.

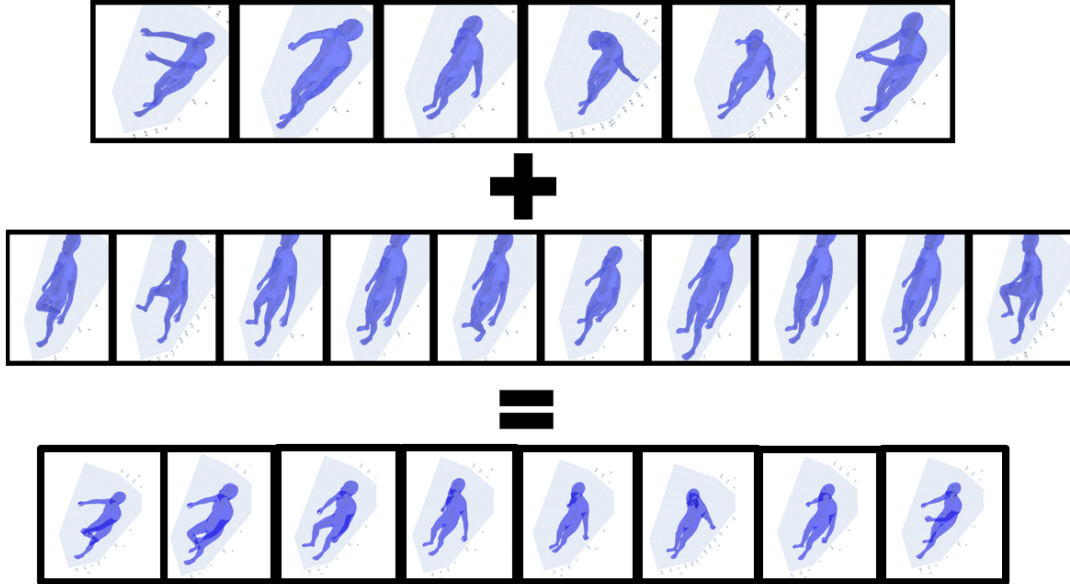


Figure 3.11: Example of embedding arithmetic. A pose generated from an IMU embedding (head bowing) and a pose generated from an EMG embedding (foot movement) are combined through embedding addition. The resulting pose exhibits both motions simultaneously.

An important practical property of the embedding arithmetic module is its robustness to differences in sequence length. When embeddings corresponding to motion sequences of unequal temporal duration are selected, the system automatically aligns them by restricting both embeddings to their shared temporal extent prior to performing arithmetic operations. This design choice ensures that addition and subtraction in the latent space remain well-defined and stable, without requiring manual temporal alignment or resampling. Consequently, embeddings derived from motion sequences of varying lengths can be combined seamlessly within the interface.

The reconstructed pose captures both motion components simultaneously, exhibiting coordinated foot movement along with head bowing. This demonstrates that the shared

embedding space supports meaningful compositional operations across modalities and preserves interpretable motion semantics when embeddings are combined.

Chapter 4

Results and Evaluation

This chapter presents the experimental results and is organized into three parts: (1) temporal multimodal representation and reconstruction of sequential sensor data, including 3D pose motion sequences, EMG, and IMU; (2) evaluation of the text encoder and decoder; and (3) comparison with state-of-the-art (SOTA) benchmarks.

4.1 Sequential Multimodal Sensor Representation and Reconstruction

This section evaluates the effectiveness of the proposed framework in encoding and decoding sequential data across multiple sensor modalities. The experiments are designed to assess how well each modality can be mapped into the shared latent embedding space and reconstructed back to its original form, thereby validating the quality of the learned representations. In addition to same-modality reconstruction, bidirectional cyclic cross-modal reconstruction is performed to examine the consistency and alignment of different modalities within the latent space.

Since the framework is trained using unpaired datasets, particular attention is given to cross-modal reconstruction between EMG and IMU signals, which are not temporally paired in the training data. These experiments serve to evaluate whether realistic and structurally consistent IMU signals can be generated from EMG representations, and vice versa, despite differences in sensor dimensionality and spatial coverage. Collectively, these evaluations provide insight into the robustness of the learned embeddings and the framework’s ability to support reliable multimodal translation and reconstruction.

4.1.1 Pose $\rightarrow$ Latent Embedding $\rightarrow$ Pose Reconstruction

To evaluate the pose autoencoder, we compare the original 3D pose motion sequences with their reconstructions produced by the pose decoder. Figure 4.1 shows a 2D UMAP projection of the latent embeddings for the ground-truth pose and reconstructed pose. The strong overlap between the two distributions indicates that the reconstructed samples

remain close to the originals in the learned representation space.

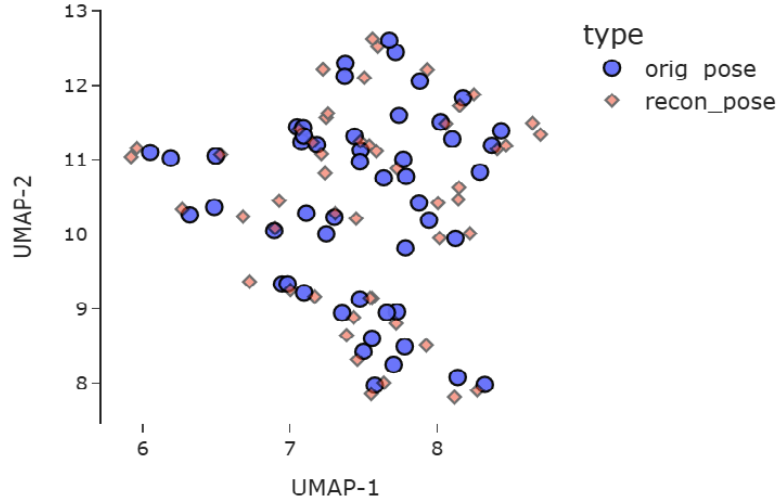


Figure 4.1: 2D UMAP visualization of pose and reconstructed pose

Quantitatively, the pose encoder-decoder achieves an average reconstruction MSE of **0.000636** and an average correlation of **0.996825**, indicating high-fidelity reconstruction.

To provide a qualitative view, Figure 4.2 shows the input 3D pose sequence (top row) before encoding, and the reconstructed sequence (bottom row) after passing through the pose encoder and pose decoder. Overall, the reconstructed motion closely matches the original, with only minor deviations. For instance, around the sixth frame, the lifted left hand shows a small shift in position compared to the ground truth.

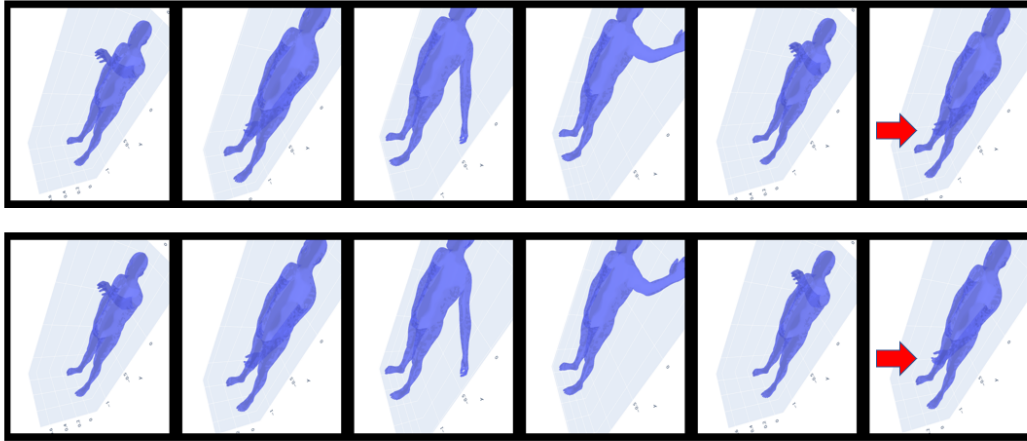


Figure 4.2: Qualitative comparison of the original (top) and reconstructed (bottom) 3D pose sequences across frames.

4.1.2 EMG $\rightarrow$ Latent Embedding $\rightarrow$ EMG Reconstruction

The EMG autoencoder is evaluated by comparing the original EMG sequences with their reconstructed outputs generated by the EMG decoder. Figure 4.3 shows a 2D UMAP projection of the latent embeddings for the ground-truth EMG data and the reconstructed EMG data. The substantial overlap between the two distributions indicates that the EMG encoder–decoder effectively preserves the temporal structure of muscle activation patterns in the latent space.

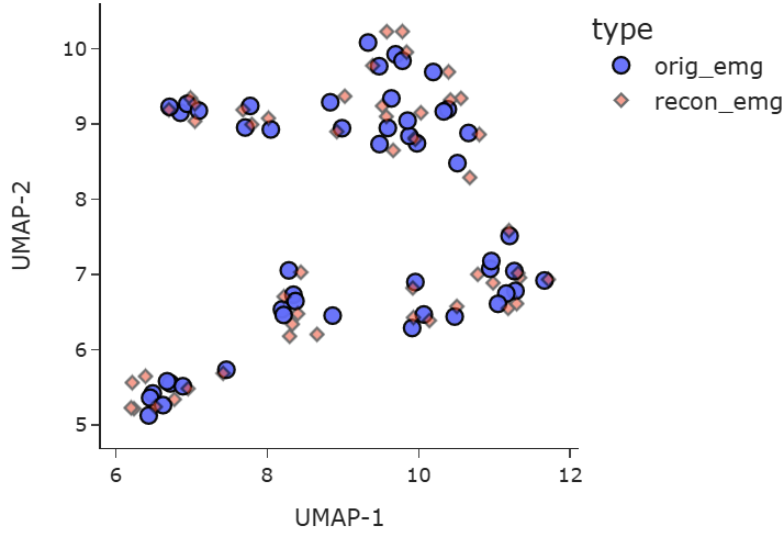


Figure 4.3: 2D UMAP visualization of EMG and reconstructed EMG embeddings.

Quantitatively, the EMG encoder–decoder achieves an average reconstruction MSE of **1.558622** and an average correlation of **0.999578**, indicating highly consistent reconstruction of EMG signals despite their lower channel count.

For qualitative inspection, Figure 4.4 presents the EMG signals from the eight recorded muscles before encoding (left) and after reconstruction by the EMG decoder (right). The reconstructed signals closely match the originals across all channels, confirming that the learned latent representation preserves the underlying EMG signal structure and amplitude information.

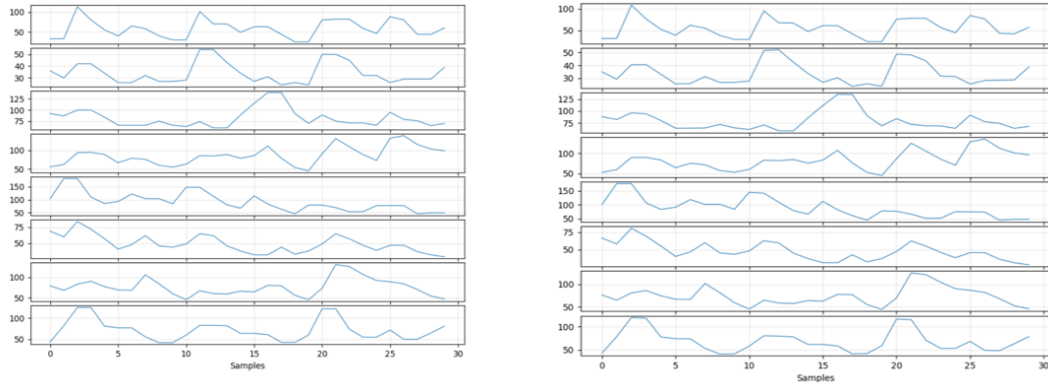


Figure 4.4: Qualitative comparison of original (left) and reconstructed (right) EMG signals across eight muscle channels.

4.1.3 IMU $\rightarrow$ Latent Embedding $\rightarrow$ IMU Reconstruction

The IMU autoencoder is evaluated by comparing the original IMU sequences with their reconstructions generated by the IMU decoder. Figure 4.5 presents a 2D UMAP projection of the latent embeddings for the ground-truth IMU data and the reconstructed IMU data. The close proximity of the two distributions indicates that the IMU encoder-decoder preserves the underlying temporal structure of the sensor signals in the latent space.

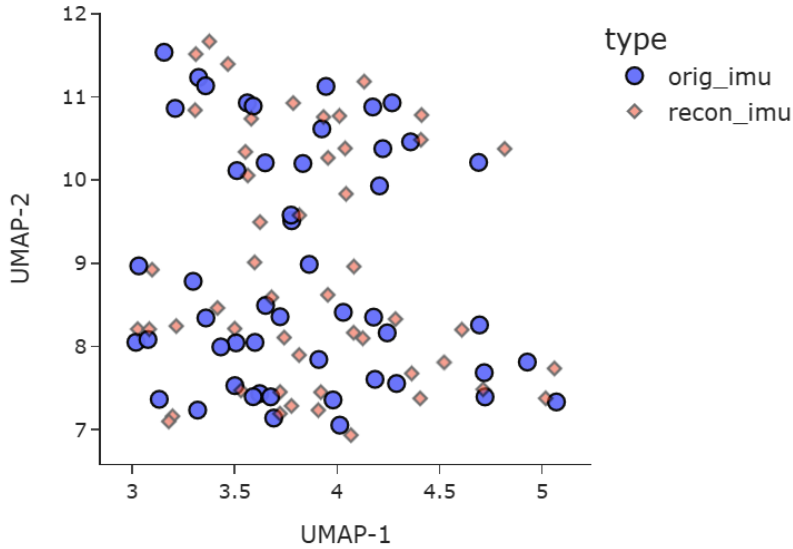


Figure 4.5: 2D UMAP visualization of IMU and reconstructed IMU embeddings.

Quantitatively, the IMU encoder-decoder achieves an average reconstruction MSE of **0.021327** and an average correlation of **0.991267**, demonstrating accurate reconstruction of IMU sensor sequences.

For qualitative evaluation, Figure 4.6 shows the IMU signals before encoding by the IMU encoder (left) and after reconstruction by the IMU decoder (right). The reconstructed signals closely align with the originals across all channels, confirming that the

latent embedding effectively preserves the IMU signal values.

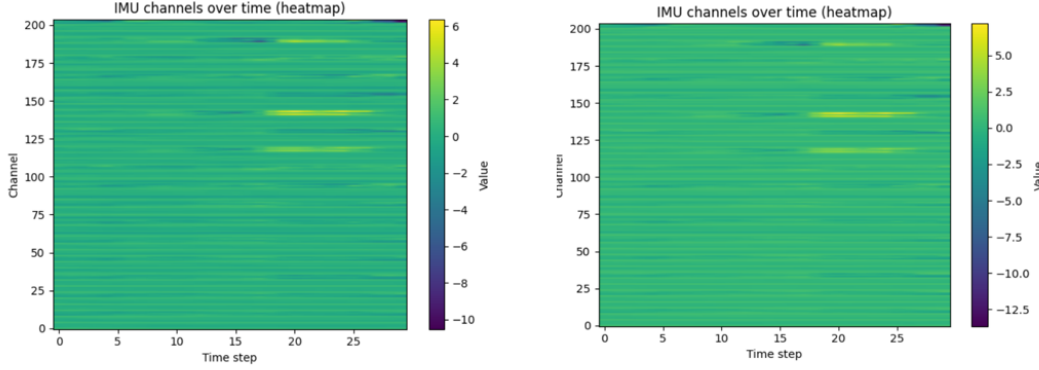


Figure 4.6: Qualitative comparison of original (left) and reconstructed (right) IMU signals.

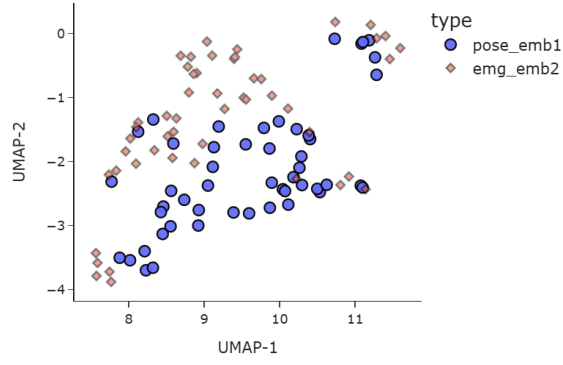
4.1.4 Bidirectional Cyclic Cross-Modal Reconstruction (Pose and EMG)

Bidirectional cyclic cross-modal reconstruction is performed between pose and EMG to ensure consistency and effective alignment between the two modalities within the shared embedding space. These experiments verify that pose and EMG representations can be reliably translated across modalities and reconstructed back to their original domains while preserving proximity in the latent space.

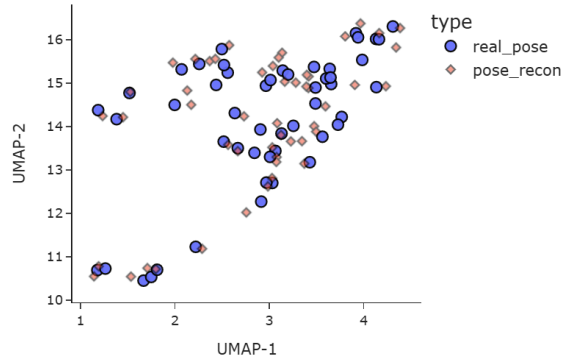
Pose $\rightarrow$ Embedding $\rightarrow$ EMG $\rightarrow$ Embedding $\rightarrow$ Pose In this experiment, pose sequences are encoded into the latent embedding space, translated into the EMG modality, re-embedded, and reconstructed back into the pose domain. This cyclic process enforces cross-modal consistency, ensuring that each pose sequence corresponds to a coherent EMG representation within the shared embedding space.

Quantitatively, the Pose $\rightarrow$ EMG $\rightarrow$ Pose cycle achieves an average embedding MSE of **0.053911**, an average pose reconstruction MSE of **0.003793**, and an average pose correlation of **0.9944**.

As shown in Figure 4.7a, the UMAP visualization demonstrates that pose embeddings and their translated EMG embeddings remain closely aligned in the latent space. Figure 4.7b further shows strong overlap between the original and reconstructed pose embeddings, indicating accurate reconstruction after translation through the EMG modality.



(a) 2D UMAP visualization showing proximity between pose embeddings and translated EMG embeddings.



(b) 2D UMAP visualization of original and reconstructed pose embeddings.

Figure 4.7: Embedding-space analysis for the Pose $\rightarrow$ EMG $\rightarrow$ Pose cycle.

For qualitative evaluation, Figure 4.8 illustrates the full cyclic reconstruction process. The original pose sequence is shown before conversion to the latent embedding and translation into EMG (top). The EMG signal is then re-encoded into the embedding space and decoded back into the pose domain. The reconstructed pose closely matches the original, with only minor deviations, which are highlighted in the figure.

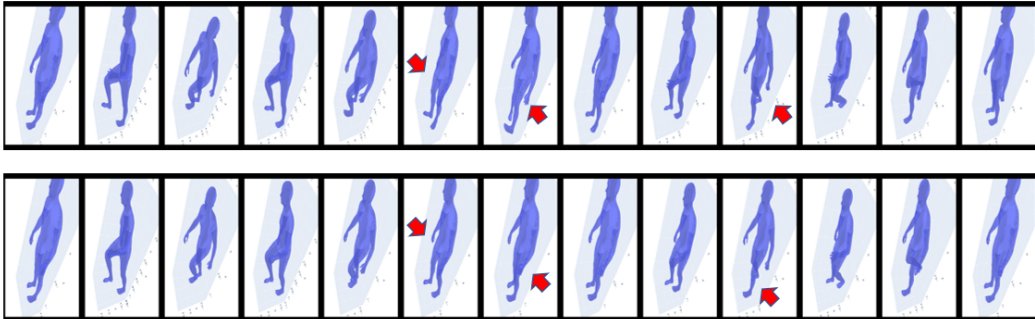
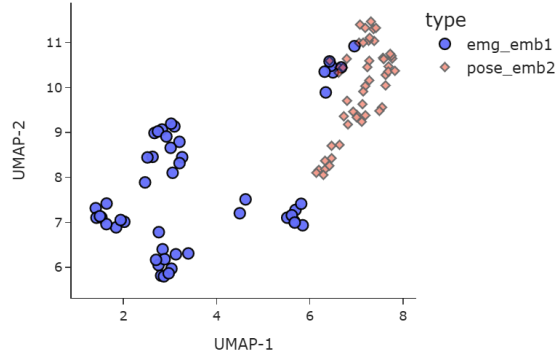


Figure 4.8: Qualitative visualization of the Pose $\rightarrow$ EMG $\rightarrow$ Pose cycle. The reconstructed pose closely matches the original, with minor deviations highlighted.

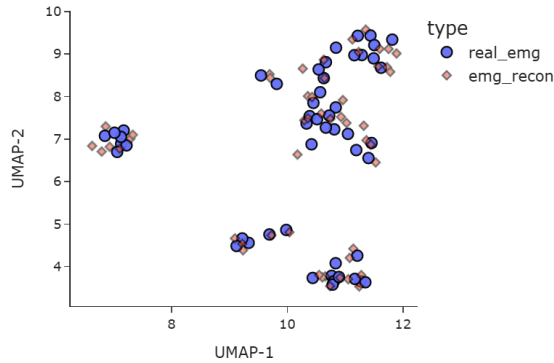
EMG $\rightarrow$ Embedding $\rightarrow$ Pose $\rightarrow$ Embedding $\rightarrow$ EMG In the reverse cycle, EMG sequences are encoded into the latent embedding space, translated into pose representations, re-embedded, and reconstructed back into the EMG domain. This evaluation examines whether EMG representations remain consistent after translation through the pose modality.

This cycle achieves an average embedding MSE of **0.072604**, an average EMG reconstruction MSE of **23.272789**, and an average EMG correlation of **0.977041**.

As shown in Figure 4.9a, the EMG embeddings and pose embeddings are relatively distant in the UMAP space. This separation arises because the EMG signals encode information from fewer sensing locations, whereas the pose embeddings capture motion data from a larger number of body joints. Nevertheless, the underlying motion from which the EMG signals were generated is largely preserved. This is evidenced in Figure 4.9b, where the UMAP projections of the original and reconstructed EMG embeddings exhibit close alignment.



(a) 2D UMAP visualization showing proximity between EMG embeddings and translated pose embeddings.



(b) 2D UMAP visualization of original and reconstructed EMG embeddings.

Figure 4.9: Embedding-space analysis for the EMG $\rightarrow$ Pose $\rightarrow$ EMG cycle.

For qualitative inspection, Figure 4.10 visualizes the full EMG $\rightarrow$ Pose $\rightarrow$ EMG cycle. Starting from the initial EMG signal, it is embedded and translated into a pose sequence (top). The pose is then re-embedded and converted back into EMG. To assess motion

integrity, the resulting representation is embedded once more and decoded using the pose decoder (bottom). The overall motion structure is largely preserved, with only a few deviations, which are highlighted in the figure.

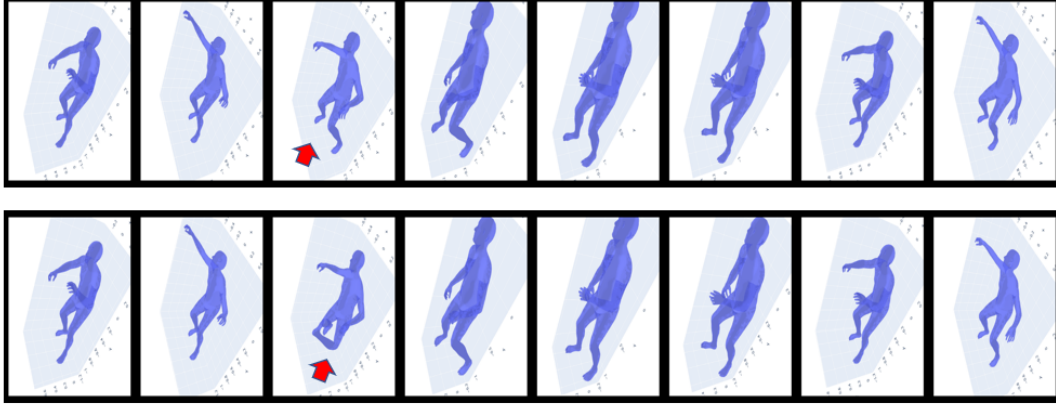


Figure 4.10: Qualitative visualization of the EMG $\rightarrow$ Pose $\rightarrow$ EMG cycle. Motion structure is largely preserved, with minor deviations highlighted.

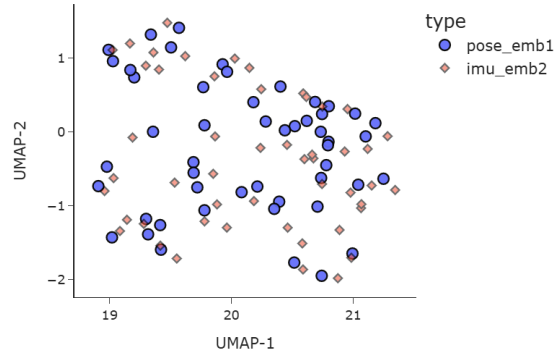
4.1.5 Bidirectional Cyclic Cross-Modal Reconstruction (Pose and IMU)

Bidirectional cyclic cross-modal reconstruction is performed between pose and IMU to evaluate consistency and alignment between kinematic pose representations and multi-sensor IMU signals within the shared embedding space. These experiments assess whether pose and IMU modalities support reliable bidirectional translation and reconstruction when passed through the latent embedding space.

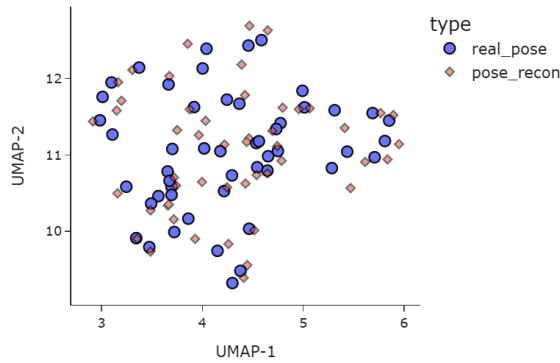
Pose $\rightarrow$ Embedding $\rightarrow$ IMU $\rightarrow$ Embedding $\rightarrow$ Pose In this experiment, pose sequences are encoded into the latent embedding space, translated into IMU representations, re-embedded, and reconstructed back into the pose domain. This cycle evaluates whether IMU signals can effectively capture and preserve pose variations through cross-modal translation.

Quantitatively, the Pose $\rightarrow$ IMU $\rightarrow$ Pose cycle achieves an average embedding MSE of **0.018454**, an average pose reconstruction MSE of **0.000767**, and an average pose correlation of **0.996383**, indicating highly accurate pose reconstruction through the IMU modality.

As shown in Figure 4.11a, the UMAP visualization demonstrates close alignment between pose embeddings and translated IMU embeddings. In addition, Figure 4.11b shows strong overlap between the original and reconstructed pose embeddings, confirming latent space consistency and reconstruction quality.



(a) 2D UMAP visualization showing proximity between pose embeddings and translated IMU embeddings.



(b) 2D UMAP visualization of original and reconstructed pose embeddings.

Figure 4.11: Embedding-space analysis for the Pose $\rightarrow$ IMU $\rightarrow$ Pose cycle.

For qualitative evaluation, Figure 4.12 illustrates the full Pose $\rightarrow$ IMU $\rightarrow$ Pose cycle. The original pose sequence is shown at the top. It is first encoded into the latent embedding, translated into IMU signals using the IMU decoder, and then re-encoded back into the embedding space. Finally, the pose decoder reconstructs the pose sequence. The reconstructed pose closely matches the original and preserves motion integrity, with only minor deviations highlighted in the figure.

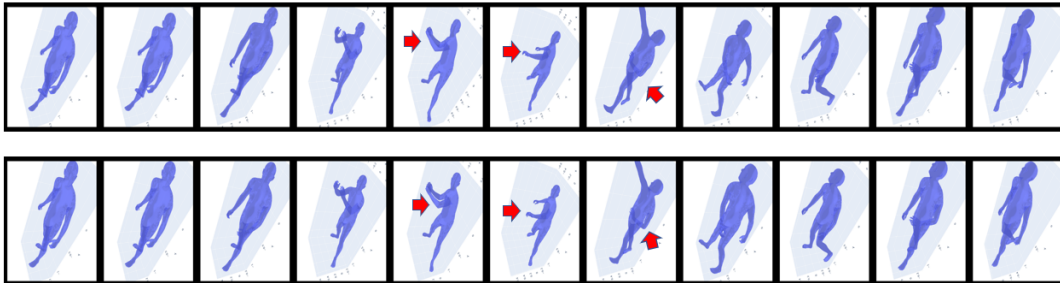
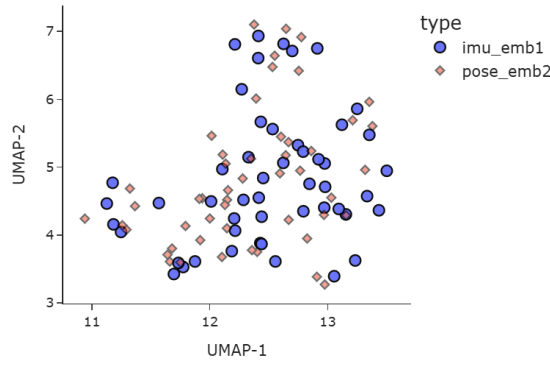


Figure 4.12: Qualitative visualization of the Pose $\rightarrow$ IMU $\rightarrow$ Pose cycle. The reconstructed pose preserves the original motion pattern, with minor deviations highlighted.

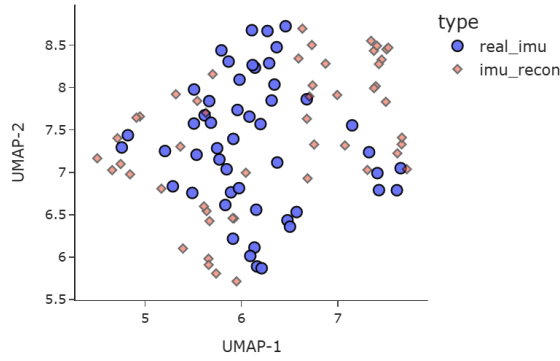
IMU $\rightarrow$ Embedding $\rightarrow$ Pose $\rightarrow$ Embedding $\rightarrow$ IMU In the reverse cycle, IMU sequences are encoded into the latent embedding space, translated into pose representations, re-embedded, and reconstructed back into the IMU domain. This evaluation examines the consistency of IMU reconstruction after translation through the pose modality.

This cycle achieves an average embedding MSE of **0.346965**, an average IMU reconstruction MSE of **1.456685**, and an average IMU correlation of **0.656515**.

As illustrated in Figure 4.13a, the UMAP projection shows that IMU embeddings translated through the pose modality remain structured but exhibit increased dispersion compared to the pose-to-IMU direction. Figure 4.13b shows that the reconstructed IMU embeddings follow the overall distribution of the original IMU embeddings.



(a) 2D UMAP visualization showing proximity between IMU embeddings and translated pose embeddings.



(b) 2D UMAP visualization of original and reconstructed IMU embeddings.

Figure 4.13: Embedding-space analysis for the IMU $\rightarrow$ Pose $\rightarrow$ IMU cycle.

The lower correlation observed in the IMU $\rightarrow$ Pose $\rightarrow$ IMU cycle can be attributed to fundamental differences between the two modalities. IMU measurements encode acceleration, angular velocity, and magnetic orientation information, which capture dynamic and sensor-specific motion characteristics. In contrast, pose representations primarily describe kinematic structure through joint axis-angle parameters and do not explicitly preserve these inertial quantities. As a result, although the reconstructed IMU signals retain the overall motion trends, discrepancies in acceleration- and orientation-related components

lead to a reduced correlation.

For qualitative evaluation, Figure 4.14 illustrates the full $\text{IMU} \rightarrow \text{Pose} \rightarrow \text{IMU}$ cycle. Starting from the embedded IMU signal, a pose sequence is generated (top). The pose is then encoded back into the embedding space and decoded into IMU signals. To assess motion-based consistency, the reconstructed IMU is embedded once more and decoded into a pose sequence. The resulting motion pattern is largely preserved, with noticeable errors and deviations highlighted in the figure.

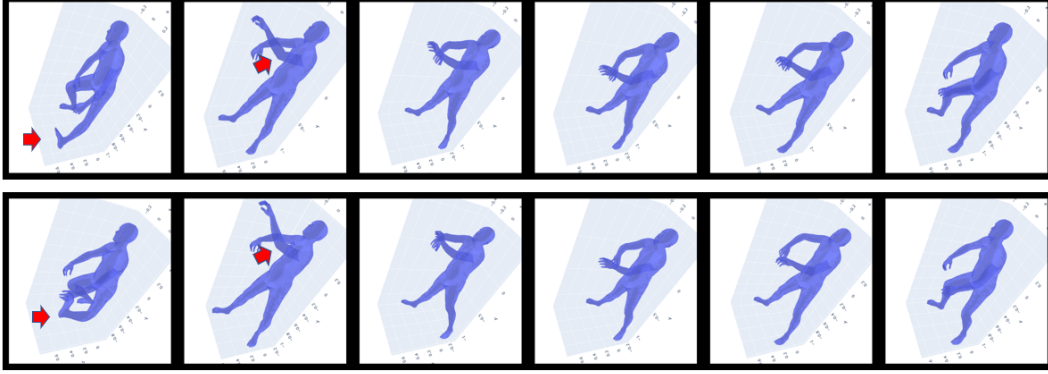


Figure 4.14: Qualitative visualization of the $\text{IMU} \rightarrow \text{Pose} \rightarrow \text{IMU}$ cycle. Motion is largely preserved, with visible deviations highlighted.

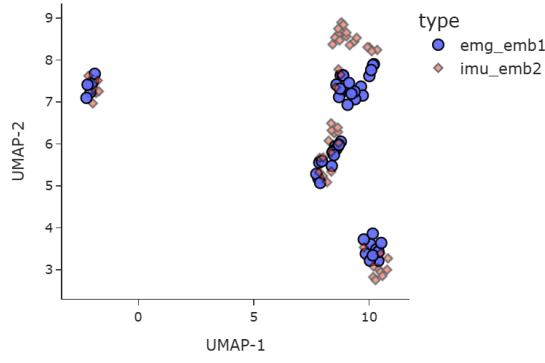
4.1.6 Bidirectional Cyclic Cross-Modal Reconstruction (EMG and IMU)

Bidirectional cyclic cross-modal reconstruction is performed between EMG and IMU to evaluate alignment and consistency between muscle activation signals and multi-sensor inertial measurements within the shared embedding space. These experiments assess whether the latent space supports reliable translation and reconstruction between modalities with substantially different sensing characteristics.

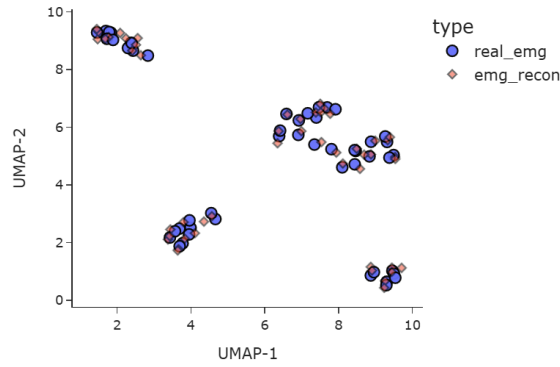
EMG $\rightarrow$ Embedding $\rightarrow$ IMU $\rightarrow$ Embedding $\rightarrow$ EMG In this experiment, EMG sequences are encoded into the latent embedding space, translated into IMU representations, re-embedded, and reconstructed back into the EMG domain. This cycle evaluates whether IMU representations derived from EMG preserve sufficient information to enable consistent EMG reconstruction.

Quantitatively, this cycle achieves an average embedding MSE of **0.038686**, an average EMG reconstruction MSE of **12.439726**, and an average EMG correlation of **0.986233**, indicating that EMG signals can be reconstructed with high correlation after translation through the IMU modality.

As shown in Figure 4.15a, the UMAP visualization illustrates that EMG embeddings and their translated IMU embeddings remain structured and closely aligned in the latent space. Furthermore, Figure 4.15b shows substantial overlap between original and reconstructed EMG embeddings, confirming effective cyclic reconstruction.



(a) 2D UMAP visualization showing proximity between EMG embeddings and translated IMU embeddings.



(b) 2D UMAP visualization of original and reconstructed EMG embeddings.

Figure 4.15: Embedding-space analysis for the EMG $\rightarrow$ IMU $\rightarrow$ EMG cycle.

For qualitative evaluation, Figure 4.16 illustrates the full EMG $\rightarrow$ IMU $\rightarrow$ EMG cycle. Starting from the embedded EMG signal, a pose sequence is generated to assess motion consistency (top). The embedding is then translated into IMU signals, re-encoded into the latent space, and decoded back into EMG. The motion integrity of the reconstructed EMG is largely preserved across the cycle, with a few errors and deviations highlighted in the figure.

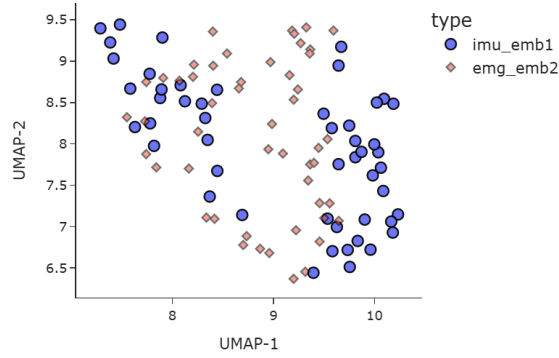


Figure 4.16: Qualitative visualization of the EMG $\rightarrow$ IMU $\rightarrow$ EMG cycle. Motion integrity is largely preserved, with minor deviations highlighted.

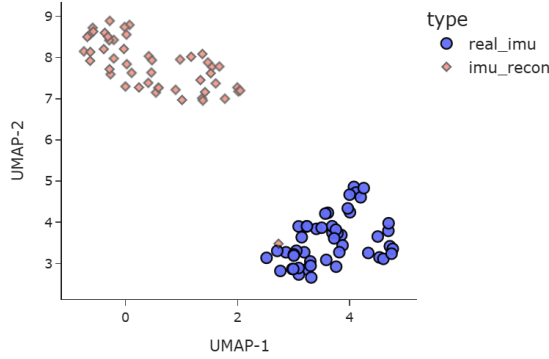
IMU $\rightarrow$ Embedding $\rightarrow$ EMG $\rightarrow$ Embedding $\rightarrow$ IMU In the reverse cycle, IMU sequences are encoded into the latent embedding space, translated into EMG representations, re-embedded, and reconstructed back into the IMU domain. This evaluation examines the consistency of IMU reconstruction after translation through the EMG modality.

This cycle achieves an average embedding MSE of **0.519857**, an average IMU reconstruction MSE of **1.870389**, and an average IMU correlation of **0.447831**.

As shown in Figure 4.17a, the IMU embeddings and their translated EMG embeddings exhibit noticeable misalignment in the UMAP space, indicating a weaker correspondence between the two modalities in this direction. Furthermore, Figure 4.17b shows that the UMAP projections of the original and reconstructed IMU embeddings are more widely separated, suggesting that although the reconstruction preserves the overall distribution, fine-grained IMU-specific details are not fully recovered.



(a) 2D UMAP visualization showing proximity between IMU embeddings and translated EMG embeddings.



(b) 2D UMAP visualization of original and reconstructed IMU embeddings.

Figure 4.17: Embedding-space analysis for the IMU $\rightarrow$ EMG $\rightarrow$ IMU cycle.

The lower correlation and UMAP misalignment observed in the IMU $\rightarrow$ EMG $\rightarrow$ IMU cycle is primarily attributable to differences in sensor dimensionality and spatial coverage between the two modalities. The IMU setup consists of 17 sensors distributed across the body, capturing global orientation and motion cues, including head and shoulder movements. In contrast, the EMG modality comprises 8 channels and does not capture

activity from the head or shoulders. As a result, information related to these regions is lost during translation into EMG. However, the underlying motion is preserved considerably as shown in the qualitative evaluation.

For qualitative evaluation, Figure 4.18 illustrates the full IMU $\rightarrow$ EMG $\rightarrow$ IMU cycle. The IMU signal is first embedded and converted into a pose sequence to assess motion consistency (top). The embedding is then translated into EMG, re-encoded into the latent space, and reconstructed back into IMU signals. To evaluate motion integrity, the reconstructed IMU is again decoded into a pose sequence. The resulting pose shows preserved arm motion, while head and shoulder movements are diminished, consistent with the modality limitations.

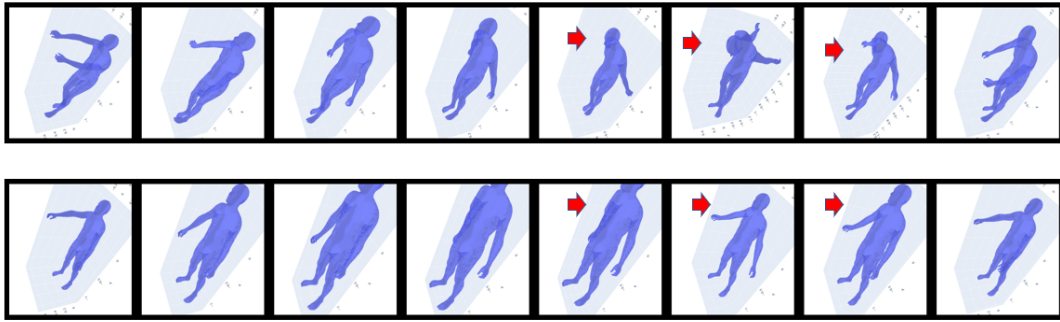


Figure 4.18: Qualitative visualization of the IMU $\rightarrow$ EMG $\rightarrow$ IMU cycle. Arm motion is preserved, while head and shoulder motion is reduced due to modality limitations.

4.2 Text Encoder and Decoder Evaluation

In this section the evaluation process and results of the text decoder and text encoder is described.

4.2.1 Text Decoder Evaluation

The text decoder is trained to generate textual descriptions from the learned latent representations. Quantitative evaluation is performed using the negative log-likelihood loss on the training and validation sets. The text decoder achieves a training loss of **2.2657** and a validation loss of **2.8488**, indicating stable convergence and effective generalization.

The vocabulary size used for training is **9,568** unique tokens. When interpreted relative to the logarithm of the vocabulary size, as defined in [39], the observed loss values indicate that the decoder effectively models the underlying language distribution and captures meaningful semantic structure. As summarized in Table 4.1, both training and validation losses fall within the very low range, reflecting strong language modeling performance. The moderate gap between training and validation loss further suggests consistent performance without overfitting.

Table 4.1: Interpretation of negative log-likelihood (NLL) loss relative to vocabulary size ($|\mathcal{V}| = 9,568$, $\log(|\mathcal{V}|) \approx 9.17$). Lower values indicate better language modeling performance.

NLL Loss Range	Relative to $\log(\mathcal{V})$	Approx. Numeric Range	Interpretation
$< 0.3 \cdot \log(\mathcal{V})$	Very low	< 2.75	Strong language modeling, high semantic coherence
$0.3 - 0.5 \cdot \log(\mathcal{V})$	Low	$2.75 - 4.59$	Good generalization and stable decoding
$0.5 - 0.7 \cdot \log(\mathcal{V})$	Moderate	$4.59 - 6.42$	Acceptable performance, limited uncertainty
$0.7 - \log(\mathcal{V})$	High	$6.42 - 9.17$	Weak modeling, limited semantic structure
$> \log(\mathcal{V})$	Very high	> 9.17	Poor convergence or undertrained decoder

For qualitative evaluation, Figure 4.19 shows an example of text generation from a pose input. A pose depicting a person raising and waving a hand is first encoded using the pose encoder and then decoded into text using the text decoder. The generated description is: *“a person waves with their right hand”*. Notably, this pose was not part of the HumanML3D pose-text training dataset, indicating that the text decoder generalizes beyond training samples.

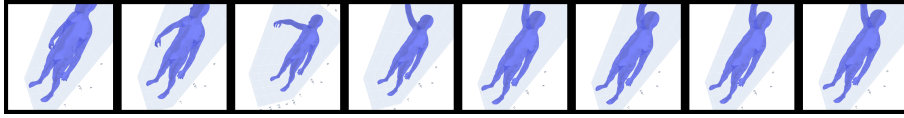


Figure 4.19: Qualitative example of text generation from a pose input. The pose is encoded into the latent space and decoded into a natural language description.

4.2.2 Text Encoder Evaluation

The text encoder is evaluated by reconstructing motion sequences from textual inputs. Given a textual description describing a specific action (e.g., a person running), the text encoder maps the input sentence into the shared latent embedding space, which is subsequently decoded into a motion sequence.

As shown in Figure 4.20, the reconstructed motions capture the overall semantic intent of the input text, indicating that the text encoder successfully embeds high-level semantic information relevant to the described action. The generated motion sequences are temporally coherent and clearly reflect the intended activities.

However, observable distortions are present in joint-level configurations. While the primary motion is carried out, several irregular limb movements are evident, which considerably distort the naturalness of the motion. This suggests that although the text encoder captures global motion semantics, detailed kinematic constraints are not strongly enforced when motion is generated solely from textual input.

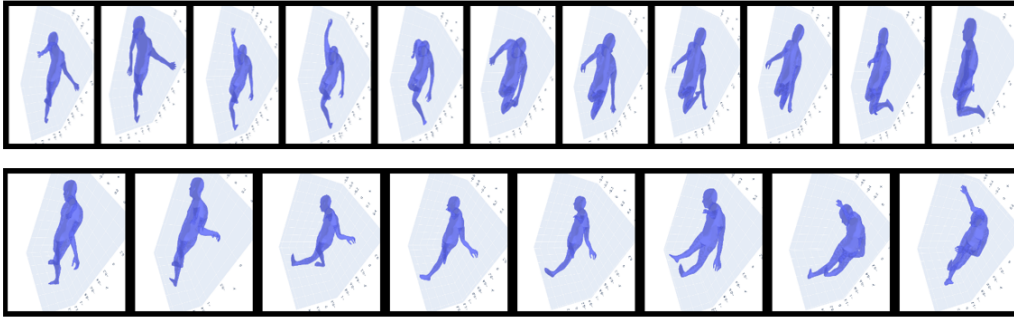


Figure 4.20: Qualitative evaluation of the text encoder. Two pose sequences generated from embeddings produced by the text encoder using the phrases “a person is kneeling down” and “a person is kicking a ball”. Although the intended actions are recognizable, the motions exhibit irregular limb movements that significantly distort the overall motion.

4.3 Benchmark Comparison with State-of-the-Art Methods

In this section, the performance of the developed models are compared against the performance of the state of the art models on the datasets used for this research.

4.3.1 Comparative Performance on the Pose–Text HumanML3D Dataset

This subsection presents a benchmark comparison on the Pose–Text *HumanML3D* dataset using standard captioning metrics. We report BLEU-1, BLEU-4, ROUGE-L, and BERTScore, where higher values indicate better agreement between predicted and reference captions.

Table 4.2: Comparative performance on the Pose–Text HumanML3D dataset. Higher is better for all metrics.

Model	BLEU1	BLEU4	ROUGE-L	BERT Score
Our model	27.18	7.95	35.69	40.57
Guided Attention for Interpretable Motion Captioning [40]	69.9	25.0	55.3	40.3
TM2T [41]	61.7	22.3	49.2	37.8

As shown in Table 4.2, our model surpasses prior state-of-the-art methods in terms of BERTScore, achieving a score of **40.57** on the HumanML3D dataset. This indicates that the pose encoder and text decoder effectively preserve the underlying semantics of the textual descriptions. Although the BLEU-1, BLEU-4, and ROUGE-L scores are comparatively lower, this can be attributed to increased lexical diversity in the generated captions. The model often produces varied wording relative to the ground-truth annotations while still maintaining strong semantic consistency, which is more accurately reflected by the BERTScore metric.

4.3.2 Comparative Performance on the Muscle in Action Dataset

This subsection reports a benchmark comparison on the *Muscle in Action* dataset using the 3D keypoint position RMSE metric. Lower RMSE indicates more accurate 3D pose

reconstruction.

Table 4.3: Comparative performance on the Muscle in Action dataset using 3D keypoint position RMSE (lower is better).

Model	3D keypoint position RMSE
Our model	0.0873
Original authors and dataset collectors [34]	0.0518

As shown in Table 4.3, our model achieves a 3D keypoint position RMSE of **0.0873** on the Muscle in Action dataset. This result provides a direct quantitative comparison with the original baseline reported for the dataset. While the current state-of-the-art achieves a lower RMSE of 0.0518, our performance remains competitive and relatively close, indicating near state-of-the-art accuracy and confirming the overall quality and reliability of the proposed approach.

4.3.3 Comparative Performance on the DIP-IMU Dataset

This subsection presents a benchmark comparison on the Pose-IMU *DIP-IMU* dataset using standard rotation error metrics. We report the average rotation error and the corresponding standard deviation, both measured in degrees. Lower values indicate better pose estimation accuracy and stability.

Table 4.4: Comparative performance on the DIP-IMU dataset using average rotation error (lower is better).

Model	Average rotation error ($^{\circ}$)	Average rotation std ($^{\circ}$)
Our model	17.6589	5.1887
Original authors and dataset collectors [33]	17.54	11.54

As shown in Table 4.4, our model achieves an average rotation error of 17.6589° on the DIP-IMU dataset, which is closely comparable to the state-of-the-art result of 17.54° reported by the original authors. Despite the similar mean error, our method produces more consistent rotation estimates, as evidenced by a substantially lower standard deviation of 5.1887° compared to 11.54° for the state-of-the-art approach. This indicates improved stability and reliability in pose estimation from IMU data.

Chapter 5

Conclusion

This chapter concludes the thesis by summarizing the main contributions of the work, reflecting on its limitations, and outlining directions for future research.

5.1 Summary

In this thesis, we have introduced a multimodal framework for aligning and generating data across 4 modalities including 3D keypoint sequence, EMG, IMU and text. To the best of our knowledge, this is the first research work where all of these four modalities have been included together. Moreover, we have used XAI techniques which was not used in any of the previous works reviewed during the survey.

A key challenge in this work was the use of unpaired datasets but despite this constraint, the combination of reconstruction experiments, cyclic cross-modal consistency evaluations, and latent-space analyses demonstrated that meaningful and stable alignment and generation was done. Moreover, this framework managed to outperform some of the previous state of the art research works and in other cases closely matched their performance in terms of several metrics.

5.2 Limitations

While the proposed framework shows encouraging performance, several limitations remain. Firstly, although the text encoder captures the overall semantic nature of motion descriptions, it does not always sufficiently model joint-level motion. As a result, deviations in joint configurations are occasionally observed, particularly in limb movements.

Moreover, although four modalities are considered in this study, other potentially useful sources of information—such as vision-based features, force or pressure sensing, or additional physiological signals—were not included. Incorporating these modalities could further enrich the shared latent representation and improve robustness.

Furthermore, the text encoder and the text decoder could not be utilized for cyclic cross-modal consistency evaluations along with the 3D pose sequence, EMG and IMU

signal due to the limitations of the text encoder.

5.3 Future Work

Future research can extend this framework in several directions. One natural extension is to incorporate additional modalities into the shared embedding space, thereby increasing the scope of multimodal generation and improving generalization. In addition, in the future works we will focus on improving text encoding strategies to better motion generation and reduce pose angle deviations. This improvement will also help in conducting cross-modal consistency evaluations for the text encoder and text decoder. Together, these directions have the potential to further strengthen the robustness and applicability of the proposed multimodal framework.

References

- [1] Callibri. Callibri muscle tracker: Fitness, rehabilitation, and research. <https://muscle-tracker.com/sectors>, 2026. Accessed 5 January 2026.
- [2] STT Systems. Sensor placement guide. https://docs.stt-systems.com/isen/2025/es/5_Sensor_placement.html, 2026. Accessed 5 January 2026.
- [3] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15180–15190, 2023.
- [4] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Aparajita Saraf, Amy Bearman, and Babak Damavandi. Imu2clip: language-grounded motion sensor translation with multimodal contrastive learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13246–13253, 2023.
- [5] Yifan Liu, Kangning Zhang, Xiangyuan Ren, Yanhua Huang, Jiarui Jin, Yingjie Qin, Ruilong Su, Ruiwen Xu, Yong Yu, and Weinan Zhang. Alignrec: Aligning and training in multimodal recommendations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1503–1512, 2024.
- [6] Zehan Wang, Yang Zhao, Haifeng Huang, Jiageng Liu, Aoxiong Yin, Li Tang, Linjun Li, Yongqi Wang, Ziang Zhang, and Zhou Zhao. Connecting multi-modal contrastive representations. *Advances in Neural Information Processing Systems*, 36:22099–22114, 2023.
- [7] Yuanyang Yin, Yaqi Zhao, Yajie Zhang, Yuanxing Zhang, Ke Lin, Jiahao Wang, Xin Tao, Pengfei Wan, Wentao Zhang, and Feng Zhao. Sea: Supervised embedding alignment for token-level visual-textual integration in mllms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1058–1070, 2025.
- [8] Xiyuan Zhang, Diyan Teng, Ranak Roy Chowdhury, Shuheng Li, Dezhi Hong, Rakesh Gupta, and Jingbo Shang. Unimts: Unified pre-training for motion time series. *Advances in Neural Information Processing Systems*, 37:107469–107493, 2024.

- [9] Yuchen Yang, Jiaheng Wu, Yan Zhang, Tiejun Zhao, Ziyang Chen, et al. Mage: Multimodal alignment and generation enhancement via bridging visual and semantic spaces. *arXiv preprint arXiv:2507.21741*, 2025.
- [10] Parthasaarathy Sudarsanam, Irene Martín-Morató, and Tuomas Virtanen. Representation learning for semantic alignment of language, audio, and visual modalities. *arXiv preprint arXiv:2505.14562*, 2025.
- [11] Christina Kranzinger, Severin Bernhart, Wolfgang Kremser, Verena Venek, Harald Rieser, Sebastian Mayr, and Stefan Kranzinger. Classification of human motion data based on inertial measurement units in sports: a scoping review. *Applied Sciences*, 13(15):8684, 2023.
- [12] Mia Chiquier and Carl Vondrick. Muscles in action. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22091–22101, 2023.
- [13] Shehla Inam, Sana Al Harmain, Shehzaib Shafique, Mafia Afzal, Arfa Rabail, Faisal Amin, and Muhammad Waqar. A brief review of strategies used for emg signal classification. In *2021 international conference on artificial intelligence (ICAI)*, pages 140–145. IEEE, 2021.
- [14] Kaleab A Kinfu and Rene Vidal. Motionbind: Multi-modal human motion alignment for retrieval, recognition, and generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- [15] Clyde Matthew CONDOR and Luis Gerardo CANETE. Development of an imu sensor-based finger flexion approximation system for fes-based finger assistive rehabilitation systems—basic kinematic analysis and sensor fusion testing—. In *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec) 2022*, pages 2P2–I03. The Japan Society of Mechanical Engineers, 2022.
- [16] Taylor Knuth and Paul Groves. Imu based context detection of changes in the terrain topography. In *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 680–690. IEEE, 2023.
- [17] Amanda L Zaleski, Rachel Berkowsky, Kelly Jean Thomas Craig, and Linda S Pescatello. Comprehensiveness, accuracy, and readability of exercise recommendations provided by an ai-based chatbot: mixed methods study. *JMIR medical education*, 10(1):e51308, 2024.
- [18] José Blasco, Sergio Roig-Casasús, Celedonia Igual-Camacho, Beatriz Díaz-Díaz, and José Pérez-Maletski. Conversational chatbot to promote adherence to rehabilitation after total knee replacement: implementation and feasibility. *Archives of Physical Medicine and Rehabilitation*, 103(12):e125, 2022.

- [19] Francesco Felici and Alessandro Del Vecchio. Surface electromyography: what limits its use in exercise and sport physiology? *Frontiers in neurology*, 11:578504, 2020.
- [20] Oliwia Ptaszyk, Tarek Boutefnouchet, Gerard Cummins, Jin Min Kim, and Ziyun Ding. Wearable devices for the quantitative assessment of knee joint function after anterior cruciate ligament injury or reconstruction: A scoping review. *Sensors*, 25(18):5837, 2025.
- [21] Logan Wade, Laurie Needham, Polly McGuigan, and James Bilzon. Applications and limitations of current markerless motion capture methods for clinical gait biomechanics. *PeerJ*, 10:e12995, 2022.
- [22] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.
- [23] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020.
- [24] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European conference on computer vision*, pages 561–578. Springer, 2016.
- [25] Carlo J De Luca. The use of surface electromyography in biomechanics. *Journal of applied biomechanics*, 13(2):135–163, 1997.
- [26] Oliver J Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, Computer Laboratory, 2007.
- [27] Zuyan Liu, Yuhao Dong, Jiahui Wang, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Ola: Pushing the frontiers of omni-modal language model. *arXiv preprint arXiv:2502.04328*, 2025.
- [28] Giordano Cicchetti, Eleonora Grassucci, Luigi Sigillo, and Danilo Comminiello. Gramian multimodal representation learning and alignment. *arXiv preprint arXiv:2412.11959*, 2024.
- [29] Xinli Yue, JianHui Sun, Junda Lu, Liangchao Yao, Fan Xia, Tianyi Wang, Fengyun Rao, Jing Lyu, and Yuetang Deng. Instruction-augmented multimodal alignment for image-text and element matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1379–1388, 2025.
- [30] Ziang Zhang, Zehan Wang, Luping Liu, Rongjie Huang, Xize Cheng, Zhenhui Ye, Wang Lin, Huadai Liu, Haifeng Huang, Yang Zhao, et al. Extending multi-modal

- contrastive representations. *Advances in Neural Information Processing Systems*, 37:91880–91903, 2024.
- [31] Yunhao Liu, Suyang Xi, Shiqi Liu, Hong Ding, Chicheng Jin, Chong Zhong, Junjun He, Catherine C Liu, and Yiqing Shen. Multimodal medical image binding via shared text embeddings. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1610–1620, 2026.
- [32] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, June 2022.
- [33] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser learning to reconstruct human pose from sparseinertial measurements in real time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 37(6):185:1–185:15, November 2018.
- [34] Mia Chiquier and Carl Vondrick. Muscles in action. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22091–22101, 2023.
- [35] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.
- [36] Dario Farina, Roberto Merletti, and Roger M Enoka. The extraction of neural strategies from the surface emg. *Journal of applied physiology*, 96(4):1486–1495, 2004.
- [37] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.
- [38] Sitao Xiang and Hao Li. Revisiting the continuity of rotation representations in neural networks. *arXiv preprint arXiv:2006.06234*, 2020.
- [39] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.
- [40] Karim Radouane, Julien Lagarde, Sylvie Ranwez, and Andon Tchechmedjiev. Guided attention for interpretable motion captioning. *arXiv preprint arXiv:2310.07324*, 2023.
- [41] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pages 580–597. Springer, 2022.