

# THREE-DIMENSIONAL VIEW INVARIANT GAIT RECOGNITION

by

Dağlar Berk Erdem

B.S., Chemical Engineering, Boğaziçi University, 2021

Submitted to the Institute for Graduate Studies in  
Science and Engineering in partial fulfillment of  
the requirements for the degree of  
Master of Science

Graduate Program in Computer Engineering  
Boğaziçi University

2025

## ACKNOWLEDGEMENTS

I would like to thank my supervisor, Berk Gokberk, for his consistent support and guidance during my master's education. I would like to express my gratitude to my family, who encouraged and supported me in every way throughout my life.



## ABSTRACT

# THREE-DIMENSIONAL VIEW INVARIANT GAIT RECOGNITION

This thesis investigates the robustness of gait recognition systems to viewpoint changes. The performance of gait recognition systems is often affected by the camera’s angle relative to the subject. To assess the severity of this issue, we propose a novel three-fold protocol to determine the effect of the camera’s angle on viewpoint robustness. First, we divided gait recognition datasets into  $10^\circ$  bins and assigned each gait sequence to a bin. Second, we restructured the datasets such that each gait sequence appears only in side-profile views in the training set and front or back views in the test set. Finally, we rotated the test set to the training viewpoints and evaluated the gait recognition frameworks. Specifically, we applied our protocol to top-performing gait recognition frameworks in the Gait3D and GREW datasets. We also reproduced and validated the results of these gait recognition methods. Additionally, we have provided angular labels of each gait sequence in the Gait3D and GREW datasets. Also proposed two novel gait representations, 1-channel Depth Image and RGB Depth Image. Our results indicate that all models achieved higher accuracy when evaluated on angles closer to those used in their training, and performance is decreased as test angles diverged further. Furthermore, using 3D gait representations combined with 2D representations did not yield improved robustness compared to their 2D counterparts when evaluated with our protocol. Notably, rotating test sets to align with training viewpoints improved performance. Most importantly, we show that using RGB Depth Images in SkeletonGait++ outperformed its Silhouette-based and 1-channel Depth Image-based counterparts. Lastly, we showed that using silhouettes or 2D poses generated from SMPLs or 3D poses leads to a decline in performance.

## ÖZET

# ÜÇ BOYUTLU BAKIŞ AÇISINDAN BAĞIMSIZ YÜRÜYÜŞ TANIMA

Bu tez, yürüyüş tanıma sistemlerinin bakış açısı değişikliklerine karşı bağlılığını araştırmaktadır. Yürüyüş tanıma sistemlerinin performansı, genellikle kameranın bakış açısından etkilenmektedir. Bu sorunun ciddiyetini değerlendirmek amacıyla, kamera açısının bakış açısı dayanıklılığına olan etkisini saptamak için özgün, üç aşamalı bir protokol önerilmektedir. İlk olarak, yürüyüş tanıma veri setlerini 10 derecelik bölmelere ayırdık ve her bir yürüyüş sekansını bir bölmeye atadık. İkinci olarak, veri setlerini, her yürüyüş sekansının eğitim setinde yalnızca yan profil görünümünde, test setinde ise ön veya arka görünümde yer alacak şekilde yeniden yapılandırdık. Son olarak, test setini eğitimdeki bakış açılarına göre döndürerek yürüyüş tanıma modellerini değerlendirdik. Protokolümüzü, özellikle Gait3D ve GREW veri setlerindeki en yüksek performanslı yürüyüş tanıma modellerine uyguladık. Ayrıca, bu yürüyüş tanıma yöntemlerinin sonuçlarını yeniden üreterek doğruladık. Bununla birlikte, Gait3D ve GREW veri setlerindeki her bir yürüyüş sekansı için açısız etiketler de sağladık. Ayrıca, Tek Kanallı Derinlik Görüntüsü ve RGB Derinlik Görüntüsü adında iki özgün yürüyüş temsili önerdik. Sonuçlarımız, tüm modellerin, eğitimde kullanılan açılara daha yakın açılarda değerlendirildiğinde daha yüksek doğruluk elde ettiğini ve test açısı farklılaştıkça performansın düştüğünü göstermektedir. Ayrıca, 3B yürüyüş temsillerinin 2B temsillerle birlikte kullanılmasının, protokolümüzle değerlendirildiğinde 2B muadillerine kıyasla dayanıklılıkta bir iyileşme sağlamadığı görülmüştür. En önemlisi, SkeletonGait++ modelinde RGB Derinlik Görüntüleri kullanmanın, Silüet tabanlı ve Tek Kanallı Derinlik Görüntüsü tabanlı muadillerine göre daha yüksek performans sağladığını gösterdik. Son olarak, SMPL'lerden veya 3B pozlardan oluşturulan silüetlerin ya da 2B pozların kullanılmasının performansta düşüşe yol açtığını gösterdik.

## TABLE OF CONTENTS

|                                                                |      |
|----------------------------------------------------------------|------|
| ACKNOWLEDGEMENTS . . . . .                                     | iii  |
| ABSTRACT . . . . .                                             | iv   |
| ÖZET . . . . .                                                 | v    |
| LIST OF FIGURES . . . . .                                      | viii |
| LIST OF TABLES . . . . .                                       | xii  |
| LIST OF SYMBOLS . . . . .                                      | xvi  |
| LIST OF ACRONYMS/ABBREVIATIONS . . . . .                       | xvii |
| 1. INTRODUCTION . . . . .                                      | 1    |
| 1.1. Contributions . . . . .                                   | 8    |
| 1.2. Organization of the Thesis . . . . .                      | 9    |
| 2. RELATED WORK . . . . .                                      | 10   |
| 2.1. 2D Methods . . . . .                                      | 10   |
| 2.2. 2D + 3D Methods . . . . .                                 | 23   |
| 3. METHODOLOGY . . . . .                                       | 28   |
| 3.1. SMPLGait . . . . .                                        | 28   |
| 3.2. GaitBase . . . . .                                        | 33   |
| 3.3. SwinGait . . . . .                                        | 36   |
| 3.4. SkeletonGait . . . . .                                    | 41   |
| 3.5. Proposed Protocol . . . . .                               | 45   |
| 3.5.1. Angular Separation and Restructuring Datasets . . . . . | 45   |
| 3.5.2. Assessment of Viewpoint Invariance . . . . .            | 48   |
| 3.5.3. Rotating Gait Representations . . . . .                 | 53   |
| 3.5.4. Generating Gait Representations . . . . .               | 54   |
| 4. EXPERIMENTS AND RESULTS . . . . .                           | 58   |
| 4.1. Datasets . . . . .                                        | 58   |
| 4.1.1. Gait3D . . . . .                                        | 58   |
| 4.1.2. GREW . . . . .                                          | 59   |
| 4.2. Quantitative Results . . . . .                            | 60   |

|                                                   |    |
|---------------------------------------------------|----|
| 4.2.1. Gait3D Experiments . . . . .               | 61 |
| 4.2.2. Interpretation of Gait3D Results . . . . . | 68 |
| 4.2.3. GREW Experiments . . . . .                 | 76 |
| 4.2.4. Interpretation of GREW Results . . . . .   | 84 |
| 5. CONCLUSION . . . . .                           | 86 |
| 6. FUTURE WORK . . . . .                          | 89 |
| REFERENCES . . . . .                              | 90 |



## LIST OF FIGURES

|             |                                                                                                                                                                                                       |    |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1.1. | Biometric Modalities. . . . .                                                                                                                                                                         | 2  |
| Figure 1.2. | A Typical Gait Recognition Pipeline. . . . .                                                                                                                                                          | 5  |
| Figure 1.3. | Examples of 2D and 3D gait representations. . . . .                                                                                                                                                   | 6  |
| Figure 3.1. | SMPLGait Framework. . . . .                                                                                                                                                                           | 32 |
| Figure 3.2. | GaitBase Framework. . . . .                                                                                                                                                                           | 36 |
| Figure 3.3. | SwinGait Framework. . . . .                                                                                                                                                                           | 38 |
| Figure 3.4. | 2D/3D residual units. . . . .                                                                                                                                                                         | 39 |
| Figure 3.5. | Two successive 2D/3D Swin Transformer blocks. . . . .                                                                                                                                                 | 40 |
| Figure 3.6. | SkeletonGait Framework. . . . .                                                                                                                                                                       | 44 |
| Figure 3.7. | Gait3D dataset divided into different angles using the angular separation protocol. Silhouettes and corresponding SMPL representations from different individuals are shown for each angle range. . . | 48 |
| Figure 3.8. | GREW dataset divided into different angles using the angular separation protocol. Silhouettes and 2D Pose representations from different individuals are shown for each angle range. . . . .          | 49 |

|              |                                                                                                                                                                                                                                                                                                                                                                                                                          |    |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 3.9.  | Distribution of Gait Sequences of restructured Gait3D dataset by our protocol and categorized into 10-degree angular bins. Data is taken from Table 3.5. . . . .                                                                                                                                                                                                                                                         | 50 |
| Figure 3.10. | Distribution of Gait Sequences of restructured GREW dataset by our protocol and categorized into 10-degree angular bins. Data is taken from Table 3.6. . . . .                                                                                                                                                                                                                                                           | 50 |
| Figure 3.11. | Demonstration of the rotation protocol on gait representations. The original representations are shown on the left, and the rotated images are on the right of each image. (a) contains the original silhouette on the left and a rotated version on the right. (b) and (c) contains the same configuration for 2D Skeletons and SMPLs, respectively. Similarly, (d) and (e) contain Depth and RGB Depth images. . . . . | 55 |
| Figure 4.1.  | Sample gait representations from the Gait3D dataset. . . . .                                                                                                                                                                                                                                                                                                                                                             | 59 |
| Figure 4.2.  | Sample gait representations from the GREW dataset. . . . .                                                                                                                                                                                                                                                                                                                                                               | 60 |
| Figure 4.3.  | The rank-1 accuracies of 2D and 3D Gait Recognition methods in Gait3D dataset obtained from Experiment 1. The data is taken from Table 4.3. . . . .                                                                                                                                                                                                                                                                      | 62 |
| Figure 4.4.  | The Rank-1 accuracies of Gait Recognition methods in rotated and restructured Gait3D dataset. The data is taken from Table 4.3 and Table 4.5. . . . .                                                                                                                                                                                                                                                                    | 66 |

|              |                                                                                                                                                                                                                                                                                                               |    |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 4.5.  | Different modalities of Gait3D dataset. Each row contains original Silhouettes, original SMPLs, silhouettes generated from SMPLs, depth images, RGB depth images, rotated SMPLs, silhouettes generated from rotated SMPLs, rotated depth images, and finally, rotated RGB depth images, respectively. . . . . | 66 |
| Figure 4.6.  | The Rank-1 accuracies of pre-trained Gait Recognition methods in the mapped and restructured Gait3D dataset. Models are trained in Experiment 1. The data is taken from Table 4.3 and Table 4.7.                                                                                                              | 68 |
| Figure 4.7.  | The Rank-1 accuracies of top-performing gait recognition methods in the generated and RGB Depth Gait3D dataset. The data is taken from Table 4.7 and Table 4.9. . . . .                                                                                                                                       | 72 |
| Figure 4.8.  | The Rank-1 accuracies of top-performing gait recognition methods in the original and Depth Gait3D dataset. The data is taken from Table 4.3 and Table 4.11. . . . .                                                                                                                                           | 73 |
| Figure 4.9.  | The Rank-1 accuracies of top-performing gait recognition methods in the Depth Gait3D dataset. The data is taken from Table 4.12. .                                                                                                                                                                            | 74 |
| Figure 4.10. | The rank-1 accuracies of Gait Recognition methods in the GREW dataset based on angles. The data is taken from Table 4.15. . . . .                                                                                                                                                                             | 78 |
| Figure 4.11. | Sample Poses from the GREW dataset. The first row contains 2D Poses generated from 3D poses, and the second row contains rotated 2D Poses generated from rotated 3D poses. . . . .                                                                                                                            | 80 |
| Figure 4.12. | The Rank-1 accuracies of Gait Recognition methods in rotated and restructured GREW dataset. The data is taken from Table 4.15 and Table 4.17. . . . .                                                                                                                                                         | 81 |

Figure 4.13. The R-1 accuracies of pre-trained Gait Recognition methods in the mapped and restructured GREW dataset. The data is taken from Table 4.19 and Table 4.15. . . . . 81



## LIST OF TABLES

|            |                                                                                                                                                                                                   |    |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Table 2.1. | Summary of Gait Recognition frameworks. . . . .                                                                                                                                                   | 27 |
| Table 3.1. | Details of the SLN Network. . . . .                                                                                                                                                               | 29 |
| Table 3.2. | Details of the 3D-STN Network (Input Size: (85,)). . . . .                                                                                                                                        | 30 |
| Table 3.3. | Architecture of ResNet-like backbone. k, c, b denote kernel size,<br>channels, and number of blocks. . . . .                                                                                      | 34 |
| Table 3.4. | Architecture of 10-layer DeepGaitV2-3D. . . . .                                                                                                                                                   | 37 |
| Table 3.5. | Distribution of Gait Sequences of restructured Gait3D dataset by<br>our protocol and categorized into 10-degree angular bins. . . . .                                                             | 51 |
| Table 3.6. | Distribution of Gait Sequences of restructured GREW dataset by<br>our protocol and categorized into 10-degree angular bins. . . . .                                                               | 52 |
| Table 4.1. | Statistics of Gait Recognition Benchmarks. . . . .                                                                                                                                                | 58 |
| Table 4.2. | Experiment-1 results. . . . .                                                                                                                                                                     | 61 |
| Table 4.3. | Evaluation of gait recognition frameworks on the Gait3D dataset.<br>Angular bins are constructed according to the angular separation<br>protocol. Values correspond to rank-1 accuracies. . . . . | 63 |
| Table 4.4. | Statistical interpretation of the Experiment 2 results. Data is taken<br>from Table 4.3 . . . . .                                                                                                 | 64 |

|             |                                                                                                                                                                                                                                                                                                         |    |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Table 4.5.  | Evaluation of top-performing gait recognition frameworks on the rotated Gait3D test set. Gait recognition frameworks are sourced from Experiment 2. Angular bins are constructed according to the angular separation protocol. Values correspond to rank-1 accuracies.                                  | 67 |
| Table 4.6.  | Comparison of Experiment 3 results against Experiment 2. Models are trained in Experiment 2 with the reconstructed Gait3D dataset. Data in LHS is taken from Table 4.4 and RHS is taken from Table 4.5                                                                                                  | 67 |
| Table 4.7.  | Evaluation of gait recognition frameworks on the mapped Gait3D test set. Models are sourced from Experiment 1. Values correspond to rank-1 accuracies. . . . .                                                                                                                                          | 69 |
| Table 4.8.  | Comparison of Experiment 4 results against Experiment 2. Data in LHS is taken from Table 4.4 and RHS is taken from Table 4.7. . .                                                                                                                                                                       | 69 |
| Table 4.9.  | Evaluation of top-performing gait recognition frameworks on the RGB Depth Gait3D test set. Values correspond to rank-1 accuracies.                                                                                                                                                                      | 70 |
| Table 4.10. | Comparison of Experiment 5 results against Experiment 4. Data in LHS is taken from Table 4.7 and RHS is taken from Table 4.9 . . .                                                                                                                                                                      | 70 |
| Table 4.11. | Comparison of Experiment 5 results against Experiment 1. Models on the LHS are sourced from Experiment 1, which are trained with Silhouettes, whereas models on the RHS are trained in Experiment 6 with RGB Depth images. Data in LHS is taken from Table 4.4 and RHS is taken from Table 4.9. . . . . | 71 |

|             |                                                                                                                                                                                                                                                                                             |    |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Table 4.12. | Evaluation results of SkeletonGait++ framework using 1-Channel Depth and 3-Channel (RGB) Depth images of Gait3D dataset. Angular bins are constructed, and models are trained using our protocol with 1-Channel and 3-Channel Depth Images. Values correspond to rank-1 accuracies. . . . . | 71 |
| Table 4.13. | Comparison of SkeletonGait++ rank-1 accuracy performances. The data in LHS is taken from Table 4.6 and RHS is taken from Table 4.12.                                                                                                                                                        | 73 |
| Table 4.14. | Reproducing gait recognition frameworks on the GREW dataset. .                                                                                                                                                                                                                              | 76 |
| Table 4.15. | R-1 accuracies of gait recognition frameworks on the restructured GREW dataset. Angular bins are defined according to the angular separation protocol. . . . .                                                                                                                              | 79 |
| Table 4.16. | Statistical interpretation of the Experiment 8 results. Models on the left-hand side (LHS) are sourced from Experiment 7 and evaluated using our protocol. Data is taken from Table 4.15. Values correspond to rank-1 accuracies. . . . .                                                   | 80 |
| Table 4.17. | Evaluation of SkeletonGait framework on the rotated GREW test set. Models are trained in Experiment 8 with the restructured GREW dataset. Values correspond to rank-1 accuracies. . . . .                                                                                                   | 82 |
| Table 4.18. | Comparison of SkeletonGait performance in Experiment 9 against Experiment 8. Models are trained in Experiment 8 with the restructured GREW dataset. Data in LHS is taken from Table 4.16 and RHS is taken from Table 4.17. . . . .                                                          | 82 |

|             |                                                                                                                                                                                                                                                                                                                                                                     |    |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Table 4.19. | The R-1 accuracies of SkeletonGait in the mapped GREW dataset. Models are trained in Experiment 8 with the restructured GREW dataset. . . . .                                                                                                                                                                                                                       | 83 |
| Table 4.20. | Comparison of SkeletonGait performance in Experiment 10 against Experiment 8. Models are trained in Experiment 8 with the restructured GREW dataset. Data in LHS is taken from Table 4.16 and RHS is taken from Table 4.19. The Skeletons on the RHS correspond to 2D skeletons and are generated from 3D Skeletons in the GREW dataset using our protocol. . . . . | 83 |

## LIST OF SYMBOLS

|                |                                     |
|----------------|-------------------------------------|
| $J$            | Joint Map                           |
| $L$            | Limb Map or Loss                    |
| $L_{ID}$       | Identity Loss                       |
| $L_{Triplet}$  | Triplet Loss                        |
| $P_g$          | 3D Pose information of Gait Frame g |
| $R$            | Resolution of the Skeleton Map      |
| $R_y$          | Rotation Matrix around y-axis       |
| $\theta$       | Angle in radians                    |
| $\hat{\theta}$ | Angle in degrees                    |
| $\Theta$       | Pose Parameters of the SMPL         |
| $\hat{\Theta}$ | Angular bin boundaries in degrees   |
| $\phi_{train}$ | Training angles of our protocol     |

## LIST OF ACRONYMS/ABBREVIATIONS

|         |                                                   |
|---------|---------------------------------------------------|
| 2D      | Two Dimensional                                   |
| 3D      | Three Dimensional                                 |
| 3D-STN  | 3D Spatial Transformation Network                 |
| ARME    | Adaptive Region-based Motion Extractor            |
| BN      | Batch Normalization                               |
| ASTP    | Adaptive Spatio-Temporal Pooling                  |
| CA-STT  | Canonical Alignment Spatial-Temporal Transformer  |
| CCA     | Canonical Correlation Analysis                    |
| CGI     | Chrono-Gait Image                                 |
| CIL     | Counterfactual Intervention Learning              |
| CSTL    | Context-Sensitive Temporal Learning network       |
| CNN     | Convolutional Neural Network                      |
| DCDC    | Diversity-Constrained Dynamic Convolution         |
| DI      | Depth Image                                       |
| FC      | Fully Connected layer                             |
| FConv   | Focal Convolution                                 |
| FPFE    | Frame-level Part Feature Extractor                |
| FTA     | Frame-level Temporal Aggregation                  |
| GCM     | Global Cross-granularity Module                   |
| GCN     | Graph Convolutional Network                       |
| GEI     | Gait Energy Image                                 |
| GP-Gait | Generalized Pose-based Gait Recognition framework |
| GPS     | Gait Parsing Sequence                             |
| GeM     | Generalized Mean                                  |
| GHI     | Gait History Image                                |
| GLFE    | Global and Local Feature Extractor                |
| GMPA    | Global Motion Pattern Aggregator                  |
| HOD     | Human-Oriented Descriptors                        |

|           |                                                                |
|-----------|----------------------------------------------------------------|
| HOT       | Human-Oriented Transformation                                  |
| HPM       | Horizontal Pyramid Mapping                                     |
| HP        | Horizontal Pooling                                             |
| HPP       | Horizontal Pyramid Pooling                                     |
| HSTL      | Hierarchical Spatio-Temporal Representation Learning framework |
| LCMB      | Local Conv-Mixing Block                                        |
| LDA       | Linear Discriminant Analysis                                   |
| LHS       | Left Hand Side                                                 |
| LN        | Layer Normalization                                            |
| LTA       | Local Temporal Aggregation                                     |
| MAE       | Mixture of Action Experts                                      |
| MLP       | Multi-Layer Perceptron                                         |
| MSA       | Multi-head Self-Attention                                      |
| MTE       | Mixture of Temporal Experts                                    |
| MTSGait   | Multi-hop Temporal Switch network                              |
| PAGCN     | Part-Aware Graph Convolutional Network                         |
| PCM       | Part Cross-granularity Module                                  |
| PTM       | Phase-aware Temporal Modelling                                 |
| QACE      | Quality Adaptive Cross Entropy                                 |
| QATriplet | Quality Adaptive Margin Triplet                                |
| RHS       | Right Hand Side                                                |
| RGa       | Relation-aware Global Attention                                |
| SP        | Set Pooling                                                    |
| SM        | Skeleton Map                                                   |
| SLN       | Silhouette Learning Network                                    |
| SiLD      | Silhouette-guided Deformation                                  |
| SSFL      | Salient Spatial Feature Learning                               |
| SMPL      | Skinned Multi-Person Linear model                              |
| SW-MSA    | Shifted Window Multi-head Self Attention                       |
| TP        | Temporal Pooling                                               |

|       |                                     |
|-------|-------------------------------------|
| TSL   | Temporal Similarity Learning module |
| VPNet | Visual Prompt Network               |
| ViT   | Vision Transformer                  |



# 1. INTRODUCTION

The escalating need for robust and reliable person identification systems in modern society has become more apparent. As the demand for such systems grows, so too does the need for strong and dependable authentication methods that can function effectively in diverse and often challenging environments [1, 2]. This is particularly evident in critical areas such as national security, law enforcement, surveillance, and access control systems [1–5]. Traditional identification methods often rely on tokens, such as ID cards, or knowledge, like passwords, which can be lost, stolen, or forgotten. This underscores the need for more inherent forms of identity recognition, leading to the widespread adoption and development of biometric technologies [2], [5–7].

In response to this demand, biometric authentication has emerged as a powerful paradigm. Biometrics, at its core, is the science of recognizing individuals based on their unique physiological or behavioral characteristics [2], [5–7]. These characteristics serve as distinctive traits or identifiers for authentication or identification purposes. They can be reliably captured and processed for decision-making in identity management systems [8]. Biometric modalities are typically categorized into two main types: physiological and behavioral. Physiological biometrics are related to the physical structure of the body, such as fingerprints, face, iris, retinal patterns, and hand geometry [6–8]. On the other hand, behavioral biometrics are based on patterns of behavior, including signature, voice-print, keystroke dynamics, and gait [6–8]. While many physiological biometrics are already commercially available and widely deployed, behavioral modalities like keystroke dynamics, gait, odor, ear shape, and DNA are still areas of active research for online identification systems [8]. The aforementioned biometric taxonomy and modalities are illustrated in Figure 1.1.

Among the behavioral biometrics, gait recognition has garnered significant attention recently, which aims to identify individuals based on their unique walking patterns [8,9]. Similarly, gait refers to the unique manner in which a person walks [10], [6].

It is a distinctive personal trait that involves both the surface geometry of the body and the dynamic motions of the joints [11]. A key advantage of gait recognition compared to other biometric modalities is its ability to identify individuals from a distance, often using standard video equipment, without requiring explicit cooperation from the subject [3, 12–15]. This contrasts with many other biometric methods, such as fingerprint or iris scanning, which require close contact or specific positioning [1, 6, 8]. From a surveillance perspective, gait can be captured stealthily [7] and is less likely to be obscured compared to biometrics like face or palm [3, 8, 12]. Furthermore, recognizing individuals by their gait can be particularly useful in scenarios where other identifying features are masked, such as when people wear face coverings [2]. The system can also be mounted out of reach, making it more resistant to tampering [1]. While the uniqueness of gait compared to biometrics like fingerprint or iris is a subject of ongoing study, the inherent characteristics of an individual’s walking style make it a valuable and often irreplaceable biometric modality in visual surveillance and other applications [3, 16].

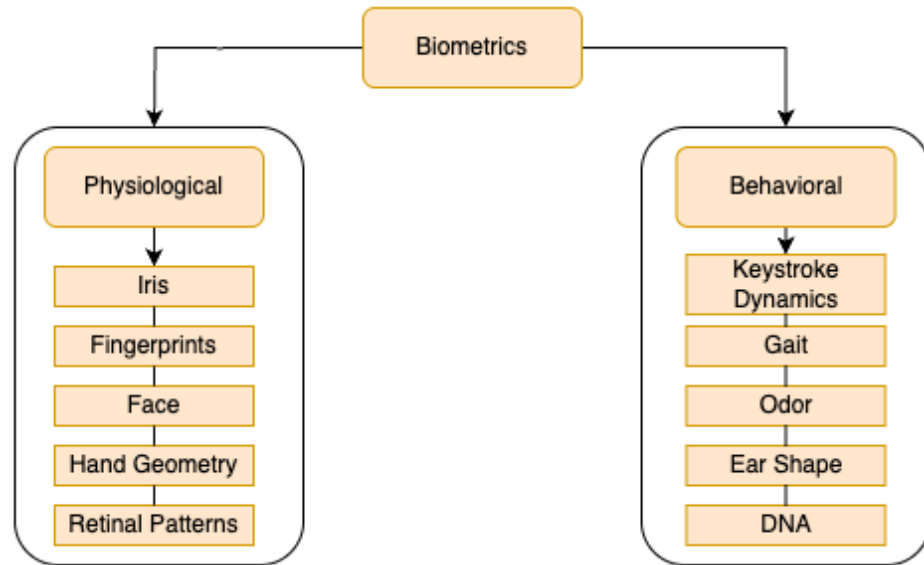


Figure 1.1. Biometric Modalities.

The study of human gait has a long history, extending back decades within fields such as biomechanics. Early research, dating at least to the 1960s, focused primarily on clinical gait analysis to identify abnormalities in lower extremity joints [8], [1],

[6]. Pioneers utilized techniques involving marker strips and strobe units to record limb trails and derive basic biomechanical models [1, 6]. Psychological studies also demonstrated that humans possess an impressive ability to perceive biological motion and even recognize individuals based on gait, sometimes without relying on visual cues like shape [17].

However, the application of gait specifically for human identification is a relatively recent attempt [6, 8]. The journey towards automated gait recognition began in the 1990s, with the first systems appearing around 1994 [6]. The field gained significant momentum with initiatives like the HumanID Gait Challenge Problem, which provided standardized datasets and benchmarks to evaluate algorithm performance and understand factors affecting gait recognition [17]. This historical progression highlights the transition from gait analysis as a medical or psychological phenomenon to its development as a viable biometric modality for security and identification purposes [1, 6, 8].

A traditional gait recognition system typically follows a structured pipeline comprising several phases, which are data acquisition, feature representation, dimension reduction, and classification [6, 18]. This modular pipeline enables researchers to focus on improving specific components, although modern deep learning approaches often integrate several of these steps into an end-to-end system [2]. The gait recognition pipeline is illustrated in Figure 1.2.

The initial step involves collecting human gait data, which is inherently spatio-temporal, capturing the dynamic motion of a person over time [6]. Traditionally, this has relied heavily on video data captured by standard cameras [8], [5], [17]. However, gait data can be acquired using various sensors, including accelerometers, floor sensors, and continuous wave radar [6]. Recent advancements have introduced new mediums, such as depth cameras, which provide 3D structural information, and LiDAR, capable of capturing precise 3D point clouds [2], [14]. While RGB images from cameras contain rich information, their direct use can introduce superficial noises like clothing texture, which are irrelevant to the underlying gait pattern [2]. This often requires subsequent

processing steps to extract relevant gait features.

Once the raw data is acquired, it needs to be converted into a suitable format, or representation, where distinctive gait features can be extracted [6]. This is a critical step, and methods are broadly categorized into appearance-based, model-based, and those utilizing other modalities. Appearance-based methods primarily rely on processing 2D representations like silhouette images, which are binary masks of the human body obtained by removing the background through techniques like background subtraction or advanced segmentation [11], [19]. Examples of such silhouettes are shown in Figure 1.3. Silhouettes simplify the data by discarding color and texture, but can lose some internal body structure and are vulnerable to variations caused by clothing or camera viewpoint [2]. Within this category, some approaches compress the sequence of silhouettes into a single image template, such as the Gait Energy Image (GEI), which averages the silhouettes over a walking cycle to reduce dimensionality and noise [10]. Other templates include Motion Energy Image (MEI), Gait History Image (GHI), and Chrono-Gait Image (CGI) [2, 3].

On the other hand, Model-based approaches attempt to model the human body structure and capture the motion patterns of body parts [18–20]. This often involves using articulated body models or deriving features from body joints and skeletons [21], [15]. Examples of such skeletons and human mesh representations are shown in Figure 1.3. The aim here is to extract features that are more robust to changes in viewpoint or clothing compared to silhouette-based methods [10], [2]. More recently, 3D representations such as human meshes are being explored to capture more comprehensive structural details. These representations contain dense 3D information about body shape, viewpoint, and dynamics. The use of model-based methods, particularly those based on human skeletons or 3D meshes, is also seen as a promising direction for achieving viewpoint robustness [19], [21]. Beyond silhouettes and skeletons, other representations derived from different sensors or processing techniques are used, including optical flow images to capture pixel-level motion dynamics, depth images, and point clouds from LiDAR [14]. Human parsing, which identifies different body parts, offers

another granular representation [22, 23].

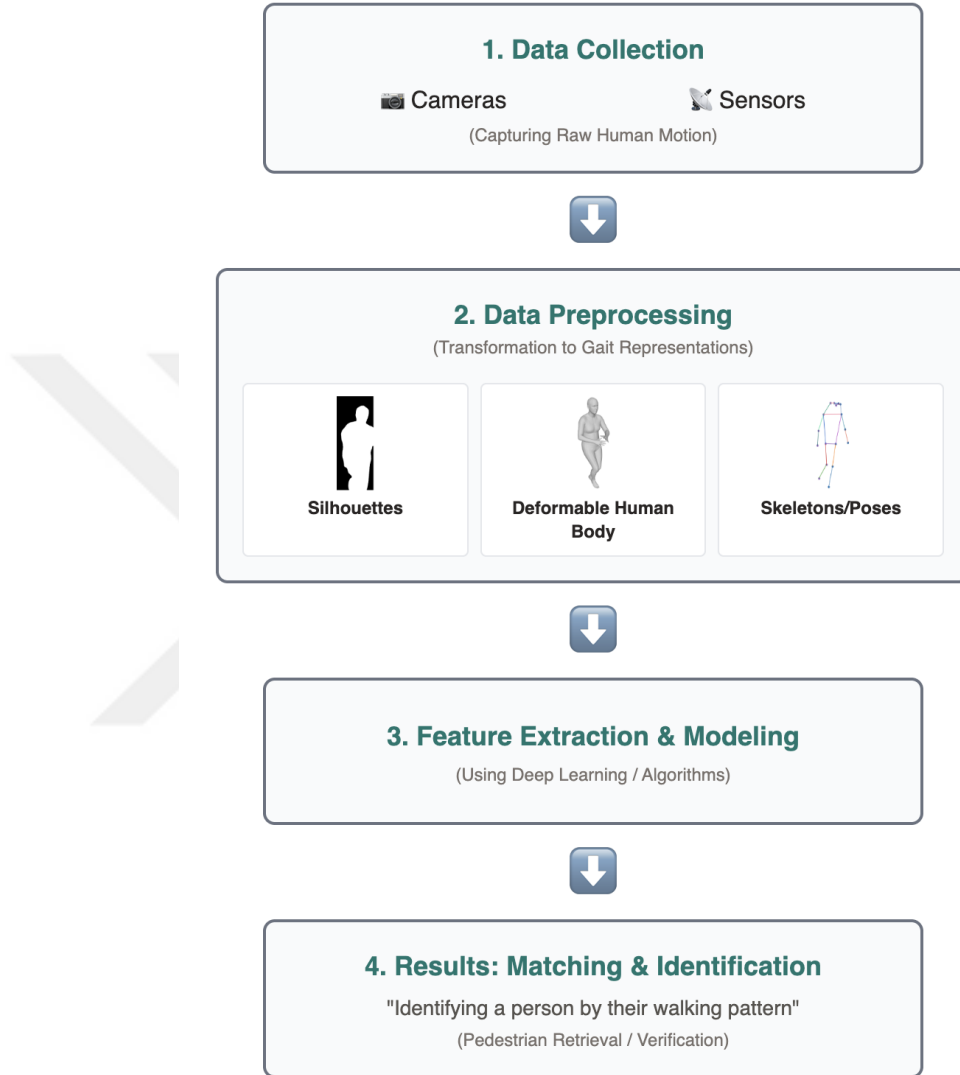


Figure 1.2. A Typical Gait Recognition Pipeline.

Following the representation step, relevant gait features are extracted from the prepared data. This involves identifying characteristic patterns of movement or shape that are unique to an individual. Traditional methods might analyze spatio-temporal patterns or parameters derived from fitted body models [5, 8]. Dimension reduction techniques are often applied to obtain a more compact and manageable feature space while preserving discriminative information [6]. Techniques such as Canonical Correlation Analysis (CCA) or Linear Discriminant Analysis (LDA) have been used, though

deep learning models now commonly handle this implicitly by learning compressed feature representations [6]. The common goal is to derive features that are distinctive for each individual while being invariant to external factors. On the other hand, deep learning-based gait recognition methods inherently learn to extract features from input representations [2].

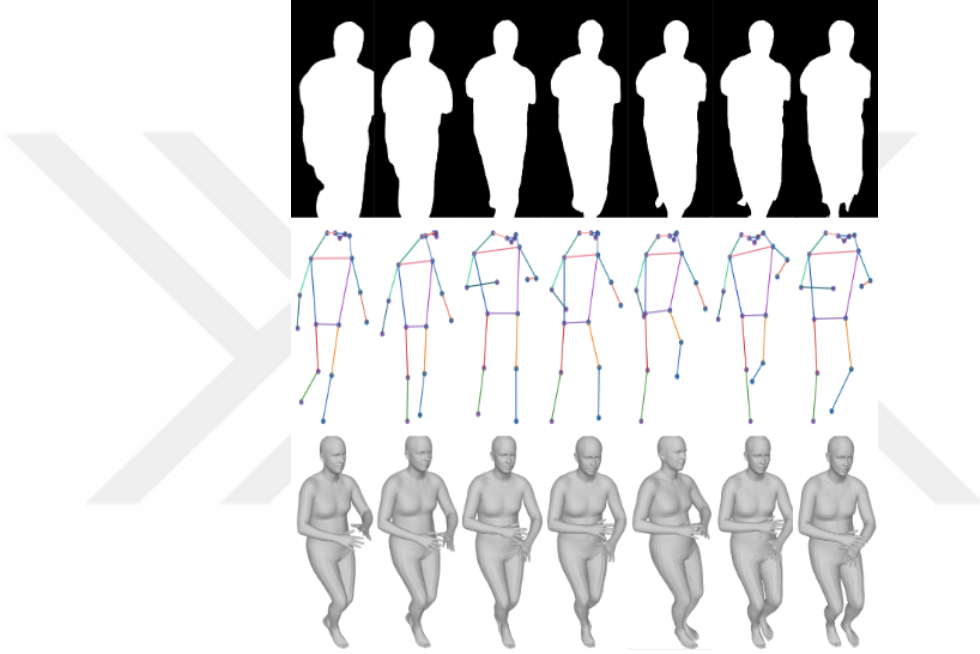


Figure 1.3. Examples of 2D and 3D gait representations.

The final phase involves using the extracted features to identify the individual. This typically involves comparing the features from an unknown gait sequence (the probe) against a database of known individuals (the gallery) [18]. This comparison can be a one-to-one verification process or a one-to-many identification task, where the system retrieves the most similar identity from the gallery [2]. Similarity is often measured using metrics like Euclidean distance [13]. With the advancements in deep learning, sophisticated models are trained to learn highly discriminative features and perform direct similarity matching or classification within the learned feature space [2].

The unique advantages of gait recognition make it highly significant for numerous real-world applications. Its ability to identify individuals at a distance and without

requiring cooperation is invaluable for video surveillance, allowing monitoring and identification in public spaces [24]. It plays a vital role in law enforcement and forensic identification, assisting in suspect retrieval and criminal investigations where other biometric evidence may be limited or unavailable [24]. Access control in semi-constrained environments, such as walkways in airports where paths can be somewhat controlled, is another potential application. Beyond security, gait analysis has long been and continues to be crucial in the medical field for diagnosing mobility disorders and guiding patient rehabilitation [2]. The growing importance of gait recognition is further highlighted in scenarios where traditional biometrics might be compromised, such as when individuals wear masks or protective gear [2].

Despite its advantages, gait recognition faces several challenges that limit its performance, particularly in real-world scenarios [18]. A significant challenge is the deviation of gait patterns, which is affected by numerous external factors. Changes in clothing, footwear, or carrying objects like backpacks or bags can change a person's appearance significantly. Environmental conditions such as varying lighting, cluttered backgrounds, or occlusions (when parts of the body are hidden) also pose considerable problems for data acquisition and processing. Walking speed is another factor that can affect gait dynamics and recognition accuracy [2]. Furthermore, temporary physical conditions, such as injuries or conscious attempts to disguise one's walk, can alter the gait pattern. Another important challenge is that variations in the camera viewpoint from which the gait is captured. While many gait recognition datasets provide well-structured data and annotated views, there are notable limitations when it comes to real-world datasets like Gait3D [21] and GREW [24]. For instance, in-the-lab datasets such as CASIA-B [25] and OUMVLP [26] include angular labels, which are essential for analyzing viewpoint dependency. However, Gait3D and GREW lack such labels, making it challenging to evaluate the robustness of gait recognition frameworks to varying viewpoints in real-life conditions. These challenges highlight the complexity of gait recognition and motivate ongoing research efforts to develop more resilient systems.

### 1.1. Contributions

In this thesis, we propose an evaluation protocol to evaluate the viewpoint robustness of top-performing gait recognition frameworks. Additionally, we evaluate pre-trained models available in the OpenGait model zoo [27] to assess the advantages of our protocol. The Gait3D [21] and GREW [24] datasets were restructured into training and testing subsets following our protocol, facilitating a more systematic and reliable evaluation. Subsequently, we conducted training and evaluation experiments using state-of-the-art gait recognition models on the restructured datasets. Using these restructured datasets, we conducted experiments with state-of-the-art gait recognition models, including SMPLGait [21], GaitBase [27], SwinGait-3D [28], and SkeletonGait [15], to assess performance improvements and viewpoint adaptability under the proposed protocol. Then, we rotated the test set to the training angles and evaluated the models using the rotated test set. Most importantly, we utilized 1-channel and 3-channel Depth Images as novel gait representation and re-evaluated those frameworks. Finally, we analyzed the change in recognition accuracy when using silhouettes or 2D poses derived from SMPLs or 3D poses.

The main contributions can be summarized as follows:

- We improved the performance of the best-performing gait recognition method, SkeletonGait++, by utilizing RGB Depth Images.
- We have demonstrated that rotating the test set to align with the training viewpoints improves the accuracy.
- We have reproduced and validated the results of state-of-the-art gait recognition methods. These methods include SMPLGait, GaitBase, SwinGait, and SkeletonGait.
- We proposed a three-fold protocol to assess the robustness to viewpoint changes of existing gait recognition methods through experiments conducted on the Gait3D and GREW datasets.
- We have demonstrated that state-of-the-art gait recognition frameworks are not

robust to viewpoint changes, contrary to their claim.

- New viewpoint annotations for gait sequences of Gait3D and GREW datasets are introduced, enabling more detailed and systematic evaluations of these datasets.
- We have demonstrated that the usage of silhouettes or 2D poses that are generated from SMPLs or 3D poses leads to a decay in performance.
- We proposed two novel gait representations, 1-channel Depth Image and RGB Depth Image, and demonstrated their superior performance in top-performing gait recognition methods.

## 1.2. Organization of the Thesis

The rest of the thesis is organized as follows. In Section 2, we review the previous research in gait recognition. We categorize and summarize existing methods based on whether they use solely 2D or combine 2D information with 3D. In Section 3, we describe the architectures of state-of-the-art gait recognition methods that are utilized in this thesis and the details of the proposed three-fold protocol that assesses viewpoint invariance. In Section 4, we present the details of the gait recognition datasets used in this thesis. Moreover, we report the quantitative results and their interpretations from training and evaluating gait recognition frameworks using our proposed protocol. Finally, in Section 5, we summarize the findings, highlight the contributions regarding viewpoint dependency in gait recognition, and discuss the effects of the experimental outcomes.

## 2. RELATED WORK

Gait recognition is a biometric authentication technique that identifies individuals based on their unique gait patterns. According to traditional taxonomy, gait recognition methods are commonly categorized into two types: appearance-based methods and model-based methods. The appearance-based methods are centered around body shape and take silhouettes, GEIs, or gait parsing sequences as input. They are considered more prominent methods due to their stronger performance compared to model-based methods. Model-based methods, on the other hand, emphasize a deformable human body. They have adopted skeletons over time and progressed to incorporating representations of meshes as inputs. The methods are designed to be more resistant to changes in clothing or texture, as well as noise from the environment. However, these methods heavily rely upon how and how effectively human pose estimation is employed, which extracts features from the raw gait data. Moreover, some gait recognition frameworks utilize a hybrid approach by combining appearance and model-based modalities. In this thesis, we will examine them based on their input modality type. We will categorize gait recognition frameworks into two groups: those that utilize 2D input information (e.g., silhouettes, human parsing, or skeletons) and those that combine 2D and 3D inputs (e.g., SMPLs or Point Clouds). Methods for each group in our taxonomy are clearly explained in Section 2.1 and Section 2.2.

### 2.1. 2D Methods

Chao et al. (2019) proposed GaitSet, a gait recognition framework that treats gait itself as an unordered collection of silhouette frames, rather than standard sequential information [13]. GaitSet is inspired by the drawbacks of prior gait recognition methods, which are strict temporal constraints and the loss of fine-grained spatial information. The methodology centers around a convolutional neural network (CNN) that extracts frame-wise features independently from individual silhouette frames and then applies a Set Pooling (SP) process. Then, it aggregates frame-wise features into

a single representation of a set and feeds it to a Horizontal Pyramid Mapping (HPM) module, which adds discriminative ability through spatial hierarchies extracted from gait data. Adopting such a set-based approach substantially increased gait recognition robustness against changes in angles, clothing, and sparse frame availability. Nevertheless, the need for high-quality silhouettes might reduce GaitSet’s applicability under severe occlusions or variable background settings.

Fan et al. (2020) proposed GaitPart, which is a temporal part-based model [29]. GaitPart is designed to model the entire human body as a single entity for gait recognition. The authors recognized that individual body parts have different shapes and movement patterns. Consequently, GaitPart contains two novel modules: the Frame-level Part Feature Extractor (FPFE) and Micro-motion Capture Module (MCM). The FPFE employs Focal Convolution (FConv) layers to promote fine-grained spatial feature learning by focusing on local part information within silhouette sequences. The MCM captures short-range spatio-temporal features and local micro-motion behaviors. Therefore, MCM differentiates among different gait patterns. The proposed boosts recognition accuracy across benchmark datasets, outperforming state-of-the-art approaches.

Line et al. (2021) proposed GaitGL, which aims to enhance discriminative feature representation by using both global visual information and local region details [30]. The authors are motivated by the fact that global features may overlook fine-grained gait nuances, and local features may ignore the contextual relations among regions. To address this issue, the authors utilize a dual-branch architecture centered on the Global and Local Feature Extractor (GLFE) and the Local Temporal Aggregation (LTA) module. The GLFE module utilizes GLConv layers that ensemble both global and local gait features in a principled fashion. The latter replaces traditional spatial pooling by aggregating temporal information in localized clips, hence preserving higher spatial resolution. Moreover, GaitGL further integrates Generalized Mean (GeM) pooling for adaptive spatial feature mapping and utilizes a combination of triplet and cross-entropy loss functions to optimize the learning process. Nevertheless, the model’s

effectiveness heavily depends on high-quality silhouette inputs. Moreover, its 3D CNN-based structure is computationally resource-intensive, which may limit its scalability in resource-constrained environments.

Huang et al. (2021) proposed CSTL, a Context-Sensitive Temporal Learning network [31]. CSTL is developed to improve gait recognition by effectively aggregating multi-scale temporal features in a context-aware manner. The motivation behind CSTL stems from the need to derive significant motion information from silhouettes of walking individuals, primarily when temporal patterns differ between individuals. A three-branch temporal architecture of CSTL extracts frame-level, short-term, and long-term features, each representing different motion granularities. The chief innovation lies in its relation modeling process, which dynamically assesses the relevance of these scales and adaptively aggregates them to produce temporally attentive representations. To complement it, CSTL incorporates a Salient Spatial Feature Learning (SSFL) module to address spatial misalignment caused by temporal convolution. SSFL chooses the most informative regions over frames to reconstruct good-quality recombinant frames, which improves spatial consistency. The relation between adaptive temporal aggregation and spatial saliency exploitation allows CSTL to represent fine-grained variations in walking patterns across conditions effectively. As an appearance-based approach based on silhouette sequences, CSTL is endowed with rich spatio-temporal learning abilities and provides enhanced adaptability to intricate motion flows. Although authors are keen to stress the complementary strengths of CSTL’s temporal and spatial modules, dependency upon precise part-based silhouette division could hinder performance under poor segmentation conditions.

Zheng et al. (2022) introduced the Multi-hop Temporal Switch Network (MTSGait) [32]. MTSGait aims to improve gait recognition performance in complex, real-world environments, which is motivated by the limitations of traditional 3D convolutional techniques such as feature misalignment, high computational complexity, and optimization difficulties. MTSGait employs a novel multi-hop temporal switch technique, which efficiently captures temporal dynamics at multiple scales by utilizing 2D

convolutional layers in conjunction with channel-switching techniques across frames. MTSGait also presents a non-cyclic, continuous sampling technique designed to enhance the resilience of temporal feature extraction in situations involving diverse walking patterns, irregular viewpoints, and occlusions. The method highlights areas for future improvement, even if it is quite effective and computationally efficient, since it is sensitive to severe occlusions and abrupt temporal discontinuities.

Teepe et al. (2022) introduced GaitGraph2, which is a skeleton-based gait recognition framework [33]. GaitGraph2 is designed to mitigate the shortcomings of the silhouette-based methods, particularly susceptibility to visual artifacts and sensitivity to appearance variations. This method aims to capture the pure gait dynamics and emphasize temporal and structural gait features over visual details. Therefore, the authors utilized Graph Convolutional Networks (GCNs) with higher-order input representations, which are derived from human body keypoints. To model detailed spatial-temporal gait dynamics, GaitGraph2 integrates a multi-branch graph convolutional architecture that incorporates joint positions, bone lengths, angles, and motion velocities. The main contribution of this framework is that it demonstrates considerable performance improvements over existing skeleton-based methods.

Zheng et al. (2023) introduced ParsingGait, which is a novel framework in terms of input modality. The authors recognized that traditional silhouette- and skeleton-based gait recognition methods often miss out on essential details and came up with a new gait representation called Gait Parsing Sequence (GPS). This representation consists of fine-grained human segmentation sequences, capturing rich semantic details and dynamics of individual body parts. The ParsingGait framework, on the other hand, is built on a powerful CNN-based framework and followed by two modules. The first one captures the general structure of the body, while the other one analyzes the relationships and movements between parts in detail via graph convolutional networks (GCN). The proposed framework shows significant performance improvements over traditional methods on benchmark datasets. However, it can be stated that the accuracy of the system depends on the precision of the initial segmentation. Moreover,

its efficiency may decrease in cases with severe occlusions or incorrect segmentations. These situations also highlight the difficulties that ParsingGait may face in real-world applications.

Fu et al. (2023) proposed the Generalized Pose-based Gait Recognition (GP-Gait) framework to enhance the generalization ability of pose-based approaches [34]. The authors recognized that pose-based methods for gait recognition experience significant performance losses when tested across different setups due to variations such as camera angles and distances. Therefore, they proposed a Human-Oriented Transformation (HOT) to obtain a unified and stable pose representation. The transformation addresses issues caused by environmental variations via affine transformation, body rescaling, and alignment. The framework further presents Human-Oriented Descriptors (HOD) that capture discriminative and robust features such as joint, bone, and angle information. At the core of GPGait is a Part-Aware Graph Convolutional Network (PAGCN), which effectively extracts local and global spatial features by dividing the human body graph into separate segments. The most significant contribution of GPGait is enhanced cross-domain performance achieved over benchmarks. Although GPGait provides significant improvement in generalization and resilience, it is highly dependent upon pose estimation accuracy and could be ineffective under extreme occlusions or heavily noisy conditions.

Ma et al. (2023) presented DANet, a Dynamic Aggregated Network. DANet enhances gait recognition effectiveness by dynamically aggregating local and global movement patterns over silhouette sequences [35]. The authors are motivated by the fact that traditional models are based on fixed-weight convolutional layers that fail to capture spatial and temporal relationships effectively. To address this problem, DANet proposes two modules: the Local Conv-Mixing Block (LCMB) and the Global Motion Pattern Aggregator (GMPA). LCMB captures fine-grained local movement patterns through complex-valued vectorial representations and dynamic neighboring pixel-to-pixel attention. GMPA exploits a self-attention mechanism to select and pool representative local features into a strong global descriptor. The method efficiently

boosts discrimination by conforming to movement cues and filtering out noise. The model excels at richly extracting gait dynamics and condensing them into a global feature space. However, the proposed method is computationally intensive due to complex-valued operations and multi-staged processing. Consequently, it can be an obstacle for real-time deployment scenarios.

Dou et al. (2023) proposed GaitGCI, which is a generative counterfactual intervention framework that is capable of generating counterfactuals [36]. The goal of GaitGCI is to handle confounders in gait recognition. The existing silhouette-based approaches are prone to attention misalignment and focus on irrelevant regions in the input data under variations like viewpoint or walking conditions. To address this issue, GaitGCI has combined two modules, which are Counterfactual Intervention Learning (CIL) and Diversity-Constrained Dynamic Convolution (DCDC). The former is based on causal inference theory and seeks to maximize the differences between counterfactual and factual likelihoods. This helps the model prioritize genuinely discriminating features over confounder-tainted features. The latter generates sample-wise attention through a combination of low-rank matrix decomposition and a rank-based diversity constraint, and introduces adaptive convolution with minimal computational load. The advantages of GaitGCI include improved robust interpretability, flexibility, and efficiency. Additionally, GCI is model-agnostic and can be easily plugged into existing gait recognition pipelines to enhance performance. Nevertheless, its reliance upon singular value decomposition and dataset-specific hyperparameters like the number of channels may hinder efficiency and generalization ability when scaling to heterogeneous or high-dimensional data sources.

The Hierarchical Spatio-Temporal Representation Learning (HSTL) framework was proposed by Wang et al. (2023) [37]. The motivation behind HSTL was to solve the limitations of traditional gait recognition techniques, which often neglect the hierarchical dependencies between local motion patterns. Consequently, the authors introduced a multi-level hierarchical approach for gait representations, recognizing that body parts exhibit different yet related motion qualities. The core method comprises an adaptive

region-based motion extractor (ARME) to capture independent, region-specific motion features, an adaptive spatio-temporal pooling (ASTP) module for hierarchical feature mapping, and a frame-level temporal aggregation (FTA) module to reduce redundancy through multi-scale temporal downsampling. Their primary contributions are the explicit modeling of hierarchical relationships within gait dynamics and the improvement of accuracy in benchmark datasets. HSTL exhibits remarkable resilience and efficiency. However, it may encounter difficulties when relying on accurate region segmentation and consistent gait cycles, particularly under highly occluded or irregular gait patterns.

Fan et al. (2023) proposed OpenGait to fill an important gap in the field of gait recognition [27]. It has been observed that most of the conventional gait recognition methods fail to perform sufficiently under real-world conditions due to the difficulties encountered while gathering real-world data, such as noise, occlusion, and variable illumination. In this direction, OpenGait has been developed as a flexible and comprehensive codebase that supports various data types, including silhouettes, RGB images, human body models such as SMPL [38], and skeletal data. This codebase supports many learning paradigms including end-to-end, multi-modality, and contrastive learning frameworks as well as many input modalities including silhouettes, RGB images, and human body models such as SMPL [38] and skeleton data. This work mostly contributes in two ways: first in offering a consistent platform enabling thorough assessments over several indoor and outdoor datasets; second in suggesting GaitBase framework as a straightforward but strong baseline model. GaitBase shows amazing flexibility, surpassing current techniques greatly in outdoor environments, so laying a strong basis for next investigations. Notwithstanding these advantages, OpenGait admits shortcomings including its reliance on silhouettes, which would suffer with complicated occlusions and the low-quality segmentation inherent in real-world data.

Zhu et al. (2023) proposed GaitRef, which is a multi-modal gait recognition framework [39]. GaitRef is designed to address the limitations of using a single modality as input, such as skeletons or silhouettes alone. This method aims to address the

inherent noise in frame-wise joint predictions and the variability of silhouettes. Therefore, the authors introduced a dual-modality framework that utilizes both skeleton- and silhouette-based gait representations. The frameworks have a baseline method called Gaitmix. GaitMix encodes silhouettes and skeletons through separate encoders and fuses their embeddings. Using the GaitMix architecture as a baseline, GaitRef further enhances performance by refining joint positions in skeleton sequences using silhouette features. Specifically, it contains a correction network that predicts frame-wise adjustments to skeleton joints by combining silhouette features, raw joint positions, and skeleton embeddings. The refinement ensures temporal coherence and retains identity-specific cues throughout the sequence. Unlike previous methods that employ temporal smoothing or rely on RGB images, GaitRef refines skeletons based solely on silhouettes, thereby improving both durability and privacy. The approach prevents inconsistent poses from being inserted and provides structural integrity improvement without further annotation. A strength of GaitRef lies in its end-to-end approach, which harmoniously combines silhouette-informed refinement and skeleton-based representations. However, performance is expected to be highest when it relies on good silhouettes and initial pose estimates, potentially degrading under highly occluded or very low-resolution conditions.

Fan et al. (2024) developed a comprehensive set of deep learning models for practical gait recognition applications, including CNN-based DeepGaitV2 and Transformer-based SwinGait series [28]. Considering the inadequacy of traditional shallow models, especially in real-world conditions such as different viewing angles, clothing changes, and occlusion, researchers conducted a systematic study on deeper and more robust network structures. DeepGaitV2 series achieves significant improvements in gait feature extraction by explicitly modeling temporal dependencies using multi-layer 2D and 3D CNNs and pseudo-3D convolutions. In parallel, SwinGait series presents a hybrid CNN-Transformer architecture, where convolution layers are first applied to reduce the influence of image regions without information, and then 2D and 3D Swin Transformers are used for detailed spatiotemporal analysis. These methods have demonstrated significant performance gains on in-the-wild datasets, proving their validity and broad

applicability under realistic conditions. However, the high-quality silhouette preprocessing required by these approaches means that they may be vulnerable to severe occlusions or corrupted input data.

Ma et al. (2024) presented the Visual Prompt Network (VPNet) to address fundamental challenges in gait recognition, such as the complexities caused by real-world changes such as viewpoint, clothing, occlusions, and pedestrian movement patterns. The authors were driven by the need for robust techniques that can generalize to real-world conditions after realizing that conventional convolutional methods are sometimes insufficient due to their rigid parameterization and low adaptability. The VPNet approach is based on a structure developed using a different method. In this structure, a trainable, part-based prompt pool is used together with a dynamically operating transformer architecture. One of the striking aspects of the method is that it can dynamically select task-relevant prompt templates according to the conditions. In this way, the model can more effectively extract important gait features from silhouette data. When the application results are examined, it is seen that VPNet performs quite robustly, especially in outdoor conditions, and provides significant improvements over current methods on some benchmark datasets. However, despite all these positive aspects, the success of the system largely depends on the quality of the selected requests. Therefore, if the training data is scarce or the conditions are extremely variable, the method may experience a decrease in performance.

Zhu et al. (2024) proposed DyGait, which is a dynamic gait recognition framework to explicitly model the temporal evolution of gait patterns across different phases of a gait cycle [40]. Traditional approaches often emphasize spatial cues while underrepresenting the gait dynamics. Unlike traditional approaches, DyGait aims to enhance the modeling of gait motion over time. DyGait combines two functionalities, which are the Phase-aware Temporal Modeling (PTM) mechanism and the Temporal Similarity Learning (TSL) module. The PTM decomposes the gait cycle into multiple temporal phases, which helps the model to learn phase-specific motion representations that more accurately reflect temporal semantics. TSL is responsible for comparing gait dynamics

between subjects across corresponding phases, which enhances discriminative capacity in temporal domains. Notably, another contribution of this framework is its Phase-Aware Gait Set, which is a flexible representation to adapt phase-aligned comparisons. Regarding comprehensive experiments, DyGait outperforms existing state-of-the-art methods in both lab and wild conditions. Moreover, it achieves better phase-consistent matching across viewpoints. Although DyGait has strengths in temporal modeling and cross-view generalization, its dependence on exact phase estimation creates a possible restriction, especially in unconstrained or real-world data, where gait cycles may be irregular or incomplete.

Huang et al. (2024) introduced GaitMoe [41]. GaitMoe is designed to handle gait sequences under occluded scenarios. The authors approached gait recognition from the perspective of action detection. They recognized that occlusions in real-world environments often lead to missing information, noisy data, and misalignment issues. GaitMoE utilizes a Mixture of Temporal Experts (MTE) to construct temporal action anchors from silhouette sequences. It effectively handles temporal occlusions using dilated convolutions with varying dilation ratios. Moreover, the Mixture of Action Experts (MAE) adaptively integrates similar actions across gait cycles and refines these action anchors into action proposals. The primary contributions of GaitMoE are that it pioneers an action detection approach for gait sequences and introduces the comprehensive OccGait dataset, which provides extensive occlusion scenarios and explicit annotations. Although GaitMoe’s superior performance in occluded scenarios, its effectiveness is dependent on accurate initial silhouette extraction and is limited in environments with severe irregular occlusions.

Wang et al. (2024) proposed QAGait which is a novel quality-oriented gait recognition framework. The authors realized that traditional silhouette-based methods face severe performance losses in challenging scenarios like abnormal segmentation, severe occlusions, and non-standard postures. Hence, they introduced a comprehensive quality assessment and improvement framework. In particular, QAGait uses an alignment technique designed to correct non-standard postures and Maximal Connect Area and

Template Match techniques to remove background noise and unidentifiable silhouettes, respectively. Moreover, the framework combines two quality-aware loss functions dynamically adjusting optimization depending on silhouette quality: Quality Adaptive Margin Cross Entropy (QACE) and Quality Adaptive Margin Triplet (QATriplet). These methods exhibit higher accuracy and robustness, which helps to improve performance generally over several gait datasets. Despite these advantages, the quality of the first silhouette segmentation determines the effectiveness of QAGait, thus it could have some limitations in cases with highly ambiguous or severely compromised inputs.

Fan et al. (2024) introduced the SkeletonGait framework, which aims to address the challenges related to effectively combining silhouettes and skeleton data [15]. The authors highlight the limited use of structural cues from skeleton data and the sensitivity of silhouettes to noise, and they propose a new representation called the skeleton map. Using Gaussian approximations, this approach converts human joint coordinates into silhouette-like images, preserving structural information while minimizing the noise related to silhouettes. SkeletonGait++ is a multi-branch architecture combining both skeleton maps and silhouettes, whereas SkeletonGait is a baseline architecture modified from image-based gait recognition methods. The main contributions are proving the better ability of structural elements for robust gait recognition, particularly in situations including low illumination and occlusion. SkeletonGait still has difficulties with viewpoint diversity and cross-view recognition, especially in challenging real-world scenarios even if it greatly improves state-of-the-art performance over many datasets.

Zheng et al. (2024) proposed the XGait framework to overcome the main constraints of current gait recognition methods [42], such as the difficulties with the integration of silhouette and human parsing data under real-world conditions. The authors were driven by the trade-offs between parsing sequences that are rich in detail but prone to segmentation errors, and those known for their robustness but low information entropy. Consequently, the authors developed a new cross-granularity alignment approach. Dual CNN-based encoders, separately handling parsing and silhouette sequences, are used in this framework. Core components are the Part Cross-granularity

Module (PCM), which aligns dynamics of body parts using parsing’s higher information entropy, and the Global Cross-granularity Module (GCM), which improves parsing quality by using silhouette-derived global features. XGait shows notable improvements on large-scale datasets such as Gait3D and CCPG, attesting to its exceptional accuracy and robustness especially in situations involving occlusions and clothing changes. Notwithstanding these benefits, the method’s reliance on accurate segmentation of parsing data can still restrict its efficacy in very noisy or highly occluded conditions.

Zheng et al. (2024) developed the GaitSTR model, presenting a Sequential Two-Stream Refinement strategy to address limitations of traditional methods that rely on silhouette or skeleton inputs [43]. The authors identified the inherent challenges posed by silhouette and skeleton modalities. They realized that models that use silhouettes lack detailed part information. Similarly, models that use skeletons suffer from frame-wise inconsistencies due to jittery detections. Therefore, the authors decided to combine these modalities. GaitSTR employs silhouette sequences to bridge temporal consistency and skeleton accuracy through internal skeleton refinement and cross-modal refinement through silhouette-skeleton fusion. Key contributions are the introduction of a skeleton correction network and a cross-modal adapter module that effectively refines joints and bones via sequential self-correction and silhouette-informed fine-tuning. The power of GaitSTR resides not just in its demonstrated competitive or even higher performance over difficult datasets. The requirement of highly accurate initial pose estimation for skeleton refinement is, however, a potential weakness, especially under scenarios of heavy occlusions or poor-quality inputs.

Wu et al. (2025) proposed DAGait, a skeleton-led gait recognition framework. DAGait aims to resolve spatio-temporal distribution inconsistencies present in in-the-wild datasets [44]. The authors are motivated by the limitations of traditional silhouette-based methods in handling different viewpoints. Hence, they proposed an in-place innovative preprocessing and alignment process to their framework. A key feature of DAGait is a skeleton-led silhouette alignment approach, which applies affine transformations to align posture corrections based on keypoint positions in the torso.

Alignment provides vertical alignment and invariant scaling over frames, significantly enhancing spatial-temporal coherence. The preprocessing-aligned silhouettes are then passed through standard backbone networks for feature learning and recognition. Notably, DAGait outperformed alternative alignment methods such as bounding-box-based or random rotation approaches. Even though DAGait improves generalization performance, it relies on trustworthy skeleton extraction, which can be a barrier when images have poor resolution or significant occlusions.

Liu et al. (2025) presented the GaitRGA framework, especially meant to improve gait recognition accuracy under real-world, complex, and dynamic scenarios [45]. The authors proposed a new Relation-Aware Global Attention (RGA) mechanism after realizing the major performance degradation of traditional approaches. Namely, they suffer from problems including viewpoint changes, occlusions, and variations in clothing. GaitRGA leverages global structural relationships within gait silhouettes by combining two specialized attention modules—Spatial Attention (RGA-S) and Channel Attention (RGA-C). The RGA-C module models inter-channel relationships to improve feature representation across multiple channels. Whereas, the RGA-S module uses an edge enhancement mechanism to capture and emphasize spatial relationships within silhouette frames. GaitRGA makes contributions mostly in terms of its efficient integration of global context, which improves gait recognition performance on in-the-wild scenarios. Nevertheless, the method’s reliance on accurate silhouette extraction could limit its robustness in highly occluded or noisy environments.

Xu et al. (2025) introduced PSGait, which is a novel framework in terms of input modality. The authors recognized that traditional silhouette- and skeleton-based gait recognition approaches often lack detailed structural information and have low information entropy, and came up with a new gait representation called Parsing Skeleton. This representation captures detailed semantic part-level dynamics and structural connectivity, thereby enhancing the informational content of the representation. The PSGait framework, on the other hand, combines Parsing Skeleton and silhouette data either by overlay or multi-channel concatenation, and then feeds this combined data

into CNNs for feature extraction. The biggest strengths of PSGait are its lightweight design, strong performance, and efficiency, thanks to its use of different types of input data. Still, the method can struggle in situations where people are heavily covered or wearing very different clothes, or when the pose estimates aren't accurate, which makes it less reliable in unpredictable, real-world conditions.

Jin et al. (2025) developed the MultiGait framework, which aims to break new ground in gait recognition systems [23]. Conventional gait recognition systems are primarily based on silhouettes, which can only reflect an individual's movement characteristics to a limited extent. Building on this shortcoming, researchers sought to create a more comprehensive gait representation by combining various types of data. MultiGait combines silhouette data that presents the general body lines of the individual with human parcellation outputs that carry detailed anatomical region information and optical flow images that capture the subtleties of movement, providing a multi-dimensional representation. MultiGait++, the advanced version of the system, uses a unique fusion strategy called C2Fusion that integrates these three different data sources more effectively. Thanks to this strategy, both common structures and modality-specific differences are highlighted, thus significantly increasing the system's ability to distinguish individuals. The obtained results show that MultiGait++ outperforms existing methods by a wide margin and exhibits a much more robust performance, especially in real-world conditions. However, the success of the system is directly related to the quality of the data used. Errors that may occur during the optical flow and parcellation stages can negatively affect recognition accuracy. This reveals that the system should be tested in scenarios with dense crowds, excessive overlaps, or low image quality.

## 2.2. 2D + 3D Methods

Shen et al. (2023) introduced LidarGait, an innovative framework specifically developed to leverage LiDAR data for gait recognition in real-world environments [14]. Recognizing the limitations of conventional RGB-based methods, particularly in challenging scenarios involving poor lighting conditions, occlusions, and viewpoint vari-

ations, the authors proposed utilizing LiDAR point clouds as a robust modality for capturing detailed spatial information. The core methodology of LidarGait includes a novel point cloud processing pipeline featuring voxelization and spatial-temporal feature extraction via CNNs. Additionally, the authors developed a unique point-cloud-based gait dataset to facilitate research in this relatively unexplored modality. The key contribution of LidarGait lies in demonstrating the feasibility and effectiveness of LiDAR-based gait recognition, significantly enhancing accuracy under complex real-world conditions. Nonetheless, the framework’s dependence on high-quality point cloud data and computational resources poses potential constraints, particularly in dynamic or resource-constrained environments.

Shen et al. (2023) introduced LidarGait, which is an innovative framework specifically developed to utilize LiDAR data [14]. The authors highlighted the limitations of conventional RGB-based techniques under challenging scenarios such as poor lighting conditions, occlusions, and viewpoint changes. Hence, they proposed using point clouds as a potential modality for capturing spatial information. LidarGait framework contains a novel point cloud processing pipeline to extract spatial-temporal features using CNNs and voxelization. To encourage research in this modality rather than conventional modalities, the authors also created a special point-cloud-based gait dataset named SUSTech1K. LidarGait primarily contributes by demonstrating the feasibility and efficiency of LiDAR-based gait recognition. Nevertheless, the dependency on high-quality point cloud data and computational resources poses potential constraints.

Zheng et al. (2022) proposed SMPLGait, a new 3D gait recognition framework [21]. SMPLGait is built on the Skinned Multi-Person Linear (SMPL) [38] model to solve the constraints of 2D silhouette- or skeleton-based approaches in unconstrained environments. The authors noted that 2D projections sometimes overlook critical 3D information, including viewpoint, shape, and motion dynamics, especially in in-the-wild scenarios. To overcome these limitations, SMPLGait combines two branches: a 3D Spatial Transformation Network (3D-STN) using dense 3D SMPL meshes to capture and normalize viewpoint and shape variations, and a silhouette-based network

for learning appearance features. The 3D-STN improves cross-view consistency by producing transformation matrices that align the appearance features across views in the latent space. The authors also introduced Gait3D, the first large-scale 3D gait dataset with over 25,000 sequences from 4,000 subjects featuring high-quality 3D meshes, 2D/3D keypoints, and silhouettes recorded across 39 cameras in a supermarket environment. This combination of data and methodology enabled the evaluation of SMPLGait against existing 2D and 3D methods, revealing the potential of dense 3D representations in complex real-world scenes. SMPLGait depends mostly on accurate 3D mesh recovery, even though it presents a convincing direction for gait recognition in the wild. It could suffer when low resolution or occlusion causes pose estimation to fail.

Dong et al. (2024) proposed HybridGait, which is designed to address the challenges associated with cloth-changing gait recognition under realistic, in-the-wild scenarios [46]. The authors proposed a hybrid approach that combines temporal dynamics with 2D projections of 3D human meshes, recognizing that existing gait recognition systems often fail to handle clothing variations over time and across different viewpoints. Three complementary branches comprise the framework: an appearance branch using silhouette data, a temporal branch with the Canonical Alignment Spatial-Temporal Transformer (CA-STT), and a projection branch using a Silhouette-guided Deformation (SilD) technique. Especially, CA-STT records intrinsic temporal dynamics independent of clothes, while SilD improves projected 3D human meshes to match original silhouettes. HybridGait’s main contribution is its improved recognition performance and durability shown on demanding datasets including CCGait and Gait3D. But its performance mostly relies on accurate initial mesh projections and silhouette quality, implying possible sensitivity to highly occluded or greatly distorted data.

Wang et al. (2025) proposed VM-Gait, a comprehensive multi-modal framework for gait recognition. The authors designed VM-Gait to address limitations inherent in both 2D and 3D representation-based methods [47]. Moreover, their study is motivated by problems including viewpoint variation, partial occlusion, and computational com-

plexity that often limit the efficacy of current approaches. VM-Gait introduces a novel 3D representation called Virtual Marker, which greatly simplifies the process of extracting dynamic topological features from 3D mesh sequences. The framework combines a CNN-based encoder to process silhouette sequences with a point-based transformer for 3D mesh data, effectively integrating these complementary modalities to improve recognition accuracy. VM-Gait shows great resilience in many scenarios, including challenging real-world environments, and achieves state-of-the-art performance across several benchmark datasets. Nevertheless, the framework may still suffer in cases of strong occlusions or noisy inputs since it mostly relies on exact silhouette and mesh data, so stressing the ongoing difficulties in multi-modal fusion for gait recognition.

Considering the provided sources, the literature on gait recognition has been dominated mainly by silhouette-based methods, primarily due to their effectiveness, efficiency, and the fact that silhouettes have been the most popular gait data. These methods, often categorized under appearance-based approaches, directly extract features from visual input without constructing human models. However, as gait recognition research has advanced, frameworks have evolved to increasingly incorporate multi-modal inputs, such as combining silhouettes with skeletons, mesh, or flow data. The complementary strengths of different modalities drive this shift; for example, while silhouettes capture body shape, they are sensitive to variations like clothing and camera view, whereas skeletons emphasize intrinsic structure and are more robust to clothing changes, though they may lack shape details. Hybrid methods leveraging these combined modalities aim to achieve advanced and more robust performance by compensating for individual disadvantages. The explained methods are summarized in Table 2.1.

Table 2.1. Summary of Gait Recognition frameworks.

| Framework              | Year | Input Type | Input Modality          | Backbone        | Datasets                                  |
|------------------------|------|------------|-------------------------|-----------------|-------------------------------------------|
| GaitSet [13]           | 2019 | 2D         | Silhouette              | CNN             | CASIA-B,OUMVLP                            |
| GaitPart [29]          | 2020 | 2D         | Silhouette              | FConv           | CASIA-B,OUMVLP                            |
| GaitGL [30]            | 2021 | 2D         | Silhouette              | CNN             | CASIA-B,OU-MVLP                           |
| CSTL [31]              | 2021 | 2D         | Silhouette              | CNN             | CASIA-B,OU-MVLP                           |
| SMPLGait-2D [21]       | 2022 | 2D         | Silhouette              | CNN             | Gait3D                                    |
| MTSGait [32]           | 2022 | 2D         | Silhouette              | CNN             | Gait3D,GREW                               |
| GaitGraph2 [33]        | 2022 | 2D         | Skeleton                | ResGCN          | CASIA-B,OUMVLP-Pose                       |
| ParsingGait [48]       | 2023 | 2D         | Parsing                 | ResNet          | Gait3D-Parsing                            |
| GPGait [34]            | 2023 | 2D         | Skeleton                | ResGCN          | CASIA-B,OUMVLP-Pose,GREW,Gait3D           |
| DANet [35]             | 2023 | 2D         | Silhouette              | CNN             | CASIA-B,OUMVLP,Gait3D                     |
| GaitGCI [36]           | 2023 | 2D         | Silhouette              | CNN             | OUMVLP,CASIA-B,GREW,Gait3D                |
| HSTL [37]              | 2023 | 2D         | Silhouette              | ARME            | CASIA-B,OUMVLP,GREW,Gait3D                |
| GaitBase [27]          | 2023 | 2D         | Silhouette              | ResNet          | CASIA-B,OUMVLP,GREW,Gait3D                |
| GaitMix [39]           | 2023 | 2D+2D      | Silhouette+Skeleton     | CNN             | CASIA-B,OUMVLP,GREW,Gait3D                |
| GaitRef [39]           | 2023 | 2D+2D      | Silhouette+Skeleton     | CNN             | CASIA-B,OUMVLP,GREW,Gait3D                |
| SwinGait-Series [28]   | 2024 | 2D         | Silhouette              | Transformer     | CASIA-B,OUMVLP,GREW,Gait3D,CCPG,SUSTech1K |
| DeepGaitV2-Series [28] | 2024 | 2D         | Silhouette              | CNN             | CASIA-B,OUMVLP,GREW,Gait3D,CCPG,SUSTech1K |
| VPNet [49]             | 2024 | 2D         | Silhouette              | ResNet          | CASIA-B,OUMVLP,GREW,Gait3D                |
| DyGait [40]            | 2024 | 2D         | Silhouette              | Custom          | CASIA-B,OUMVLP,GREW,Gait3D                |
| GaitMoE [41]           | 2024 | 2D         | Silhouette              | MoE             | OccGait,OccCASIA-B,Gait3D,GREW            |
| QAGait [50]            | 2024 | 2D         | Silhouette              | ResNet          | CASIA-B,GREW,Gait3D                       |
| SkeletonGait [15]      | 2024 | 2D         | Skeleton Map            | CNN             | OUMVLP,GREW,Gait3D,CCPG,SUSTech1K         |
| SkeletonGait++ [15]    | 2024 | 2D+2D      | Silhouette+Skeleton Map | CNN             | OUMVLP,GREW,Gait3D,CCPG,SUSTech1K         |
| XGait [42]             | 2024 | 2D+2D      | Silhouette+Parsing      | ResNet          | Gait3D,CCPG                               |
| GaitSTR [43]           | 2024 | 2D+2D      | Silhouette+Skeleton     | CNN             | CASIA-B,OUMVLP,GREW,Gait3D                |
| DAGait [44]            | 2025 | 2D+2D      | Silhouette+Skeleton     | -               | Gait3D,CCPG,SUSTech1K                     |
| GaitRGA [45]           | 2025 | 2D         | Silhouette              | ResNet          | GREW,Gait3D,SUSTech1K                     |
| PSGait [51]            | 2025 | 2D+2D      | Silhouette+Skeleton     | ResNet          | Gait3D,CCPG,SUSTech1K                     |
| MultiGait [23]         | 2025 | 2D+2D+2D   | Silhouette+Parsing+Flow | CNN             | GREW,Gait3D,CCPG,SUSTech1K                |
| SMPLGait-3D [21]       | 2022 | 2D+3D      | Silhouette+SMPL         | CNN             | Gait3D                                    |
| LidarGait [14]         | 2023 | 3D         | Point Cloud             | ResNet          | SUSTech1K                                 |
| HybridGait [46]        | 2024 | 2D+3D      | Silhouette+SMPL         | CNN+Transformer | Gait3D,CCGait                             |
| VM-Gait [47]           | 2025 | 2D+3D      | Silhouette+SMPL         | CNN+Transformer | Gait3D,OUMVLP-Mesh                        |

### 3. METHODOLOGY

In this section, we provide a comprehensive overview of the gait recognition frameworks utilized in this study. We introduce the core concepts and principles behind these methods, followed by detailed discussions of their architectures and operational mechanisms. Then, we explain how these frameworks process input data and extract discriminative gait features to achieve robust recognition performance. Finally, we describe our proposed protocols.

#### 3.1. SMPLGait

Zheng et al. (2022) proposed SMPLGait, a new 3D gait recognition framework [21]. SMPLGait is built on the Skinned Multi-Person Linear (SMPL) [38] model to solve the constraints of 2D silhouette- or skeleton-based approaches in unconstrained environments. The authors noted that 2D projections sometimes overlook critical 3D information, including viewpoint, shape, and motion dynamics, especially in in-the-wild scenarios. To overcome these limitations, SMPLGait combines two branches: a 3D Spatial Transformation Network (3D-STN) using SMPLs to capture and normalize viewpoint and shape variations, and a Silhouette Learning Network (SLN), a silhouette-based network for learning appearance-based features.

The SLN processes 2D silhouettes extracted from video sequences, which provide critical appearance-based information such as global motion patterns and general body contours. This branch extracts features using a CNN that progressively reduces the dimensionality of the input silhouettes while retaining key features. The extracted features capture high-level attributes like the subject’s clothing and motion trajectory. The input to the SLN branch is a sequence of silhouettes, denoted as  $X_{sil} = \{x_i\}_{i=1}^L$ . Each frame’s silhouette  $x_i$  has a dimension of  $H \times W$ , which corresponds to height and width. A CNN-based backbone, referred to as SLN, processes each silhouette  $x_i$  to extract a frame-level feature map  $F_i$ . The shape of this feature map is  $F_i \in \mathbb{R}^{h \times w}$ ,

where  $h$  and  $w$  are the height and width of the feature map, respectively.

Table 3.1. Details of the SLN Network.

| Layers           | Kernel # | Kernel Size  | Stride | Padding |
|------------------|----------|--------------|--------|---------|
| Conv1            | 64       | $5 \times 5$ | 1      | 2       |
| LeakyReLU (0.01) | -        | -            | -      | -       |
| Conv2            | 64       | $3 \times 3$ | 1      | 1       |
| LeakyReLU (0.01) | -        | -            | -      | -       |
| Max Pooling      | -        | $2 \times 2$ | 2      | 0       |
| Conv3            | 128      | $3 \times 3$ | 1      | 1       |
| LeakyReLU (0.01) | -        | -            | -      | -       |
| Conv4            | 128      | $3 \times 3$ | 1      | 1       |
| LeakyReLU (0.01) | -        | -            | -      | -       |
| Max Pooling      | -        | $2 \times 2$ | 2      | 0       |
| Conv5            | 256      | $3 \times 3$ | 1      | 1       |
| LeakyReLU (0.01) | -        | -            | -      | -       |
| Conv6            | 256      | $3 \times 3$ | 1      | 1       |
| LeakyReLU (0.01) | -        | -            | -      | -       |

In parallel, the 3D-STN utilizes the SMPL model to process dense 3D data, including shape and pose parameters. Shape parameters ( $\beta$ ) encode the static attributes of the human body, such as height and proportions, while pose parameters ( $\theta$ ) represent the dynamic aspects of walking, including joint rotations and body movements. These parameters encode the fine-grained details of body structure and joint dynamics. The input to the 3D-STN branch is a sequence of SMPL parameters, denoted as  $Y_{sp} = \{y_i\}_{i=1}^L$ . Each  $y_i$  is a vector representing the SMPL parameters for the  $i$ -th frame. The dimension of the SMPL vector, denoted by  $D$ , includes parameters for 3D body pose ( $24 \times 3$  dimensions), 3D body shape (10 dimensions), and camera scale/translation (3 dimensions). These vectors are concatenated and fed into the 3D-STN, which comprises three fully connected layers followed by batch normalization and

ReLU activation functions. To prevent overfitting, dropout layers are added to the last two layers of the 3D-STN. Finally, the 3D-STN outputs a frame-level transformation vector  $g_i$  for each input SMPL vector  $y_i$ . The vector  $g_i$  has a dimension of  $h \times w$ .

Table 3.2. Details of the 3D-STN Network (Input Size: (85,)).

| Layers | Neuron #     | Dropout Rate |
|--------|--------------|--------------|
| FC1    | 128          | 0.0          |
| BN1    | -            | -            |
| ReLU   | -            | -            |
| FC2    | 256          | 0.2          |
| BN1    | -            | -            |
| ReLU   | -            | -            |
| FC3    | $h \times w$ | 0.2          |
| BN1    | -            | -            |
| ReLU   | -            | -            |

The core of the framework lies in integrating these two branches. The 3D Spatial Transformation Module applies the learned transformation vector  $g_i$  to the 2D feature map  $F_i$ . First,  $g_i$  is reshaped into a matrix  $G_i \in \mathbb{R}^{w \times h}$ . Then, the transformation is performed using matrix multiplication as

$$\hat{F}_i = F_i \cdot (I + G_i), \quad (3.1)$$

where  $I$  is an identity matrix, and  $\cdot$  denotes matrix multiplication. Notably,  $F_i$  and  $G_i$  are expanded to square matrices by zero padding and  $h = w = \max(h, w)$ .  $\hat{F}_i \in \mathbb{R}^{h \times w}$  is the resulting transformed feature map.

After feature extraction and alignment, the outputs of the 3D Spatial Transformation Module are fused through a pooling mechanism that aggregates spatial and temporal features into a comprehensive representation. The Feature Aggregation module uses set pooling and horizontal pyramid pooling (HPP), which were first introduced

in GaitSet [13].

The Set Pooling process begins with a sequence of frame-level features, say  $\hat{F}_i \in \mathbb{R}^{h \times w}$  for each frame  $i = 1, \dots, T$ , extracted by the 3D Spatial Transformation module. Set Pooling then aggregates these  $T$  feature maps into a single feature map  $\hat{F}_{set} \in \mathbb{R}^{h \times w}$ . For each specific element  $(h, w)$  in the feature map, the value in  $F_{set}$  is the maximum value found across all frames  $t$  at that exact location

$$F_{set}(h, w) = \max_{t=1}^T F_t(h, w). \quad (3.2)$$

This max-pooling operation selects the most prominent feature response across the entire set for each spatial location.

Following the temporal aggregation via the Set Pooling operation, the resulting set-level feature is processed by the HPP. HPP is a spatial pooling strategy that aims to capture features from different horizontal parts or segments of the human body, recognizing that different body parts contribute distinctly to the overall gait pattern. HPP operates on the aggregated spatial feature map  $F_{set} \in \mathbb{R}^{h \times w}$ . It divides the feature map height  $h$  into  $P$  horizontal segments. For each stripe  $p = 1, \dots, P$ , a max and average pooling operation is applied to the features within that segment to produce a feature vector  $v_p$ . Let  $F_{set}^{(p)} \in \mathbb{R}^{(h/P) \times w}$  denotes the feature map corresponding to the  $p$ -th horizontal stripe. The pooled vector  $v_p$  is derived from  $F_{set}^{(p)}$ :

$$v_p = \text{MaxPool}(F_{set}^{(p)}) + \text{AveragePool}(F_{set}^{(p)}), \quad (3.3)$$

where MaxPool represents the maximum pooling function and AveragePool represents average pooling function. These pooling methods aggregate features across the spatial dimensions of the stripe ( $h/P$  and  $w$ ). The final sequence-level gait feature for the pyramid scale  $s$ ,  $F^s$ , is obtained by concatenating the pooled vectors from all  $P$  horizontal stripes:

$$F^s = [v_1, v_2, \dots, v_P] \in \mathbb{R}^P. \quad (3.4)$$

This operation is performed for each pyramid scale  $s$ , and the resulting feature vector is obtained by

$$F = [F^1, F^2, \dots, F^S], \quad (3.5)$$

where  $S$  is the number of pyramid scales. This structure allows the network to learn and utilize spatial features from different body parts, contributing to a more discriminative gait representation.

The final stage of the framework is the classification module, where the aggregated feature representation is processed through fully connected layers to reduce dimensionality and map it to the output space. The framework is trained end-to-end using a loss function

$$L = \alpha L_{tri} + \beta L_{ce}, \quad (3.6)$$

where  $L_{tri}$  is the triplet loss and  $L_{ce}$  is the cross entropy loss.  $\alpha$  and  $\beta$  are weighting parameters. During inference, the similarity between query and gallery sequences is measured using cosine similarity based on the aggregated sequence-level features.

While the novel framework is the two-branch SMPLGait, the authors also introduce a variation used in ablation studies called SMPLGait w/o 3D. This variant effectively represents the SLN operating in isolation, without the spatial normalization provided by the 3D-STN branch. It serves as a baseline to demonstrate the contribution of integrating the 3D SMPL information. The overall diagram of the SMPLGait framework is shown in Figure 3.1

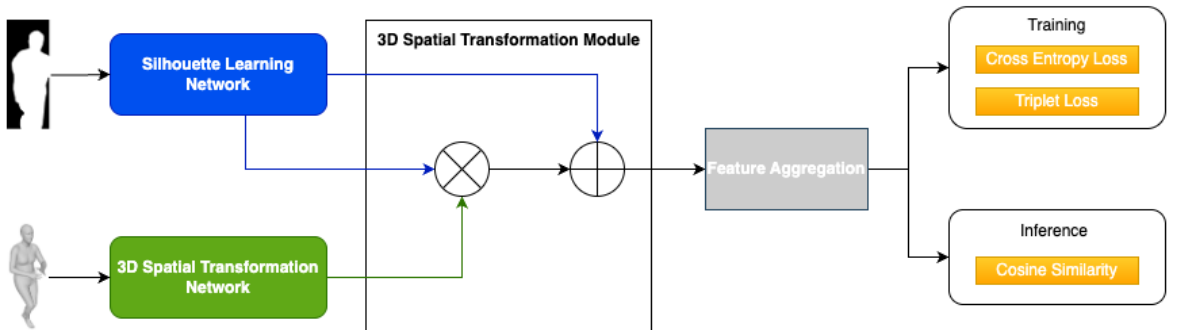


Figure 3.1. SMPLGait Framework.

### 3.2. GaitBase

The design of GaitBase employs several easy and widely accepted modules. This framework consists of a backbone network responsible for extracting features, followed by components for temporal pooling (TP), horizontal pooling (HP), a head, and loss functions. The backbone of GaitBase is built around a ResNet-like [52] network that processes input silhouette sequences.

The input to the GaitBase framework is a sequence of human silhouettes. Each input silhouette frame  $x_j$  is fed into the ResNet-like backbone. The purpose of this backbone is to extract initial frame-level features. A standard ResNet block learns a residual mapping  $F(x)$  such that the output is  $y = F(x) + x$ , where  $x$  is the input to the block [52]. This residual connection helps in training deep networks. The ResNet-like backbone in GaitBase consists of an initial convolution layer followed by four basic residual blocks with increasing channel counts:  $64 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512$ . The layers use  $3 \times 3$  kernels followed by Batch Normalization and ReLU activations. Downsampling is handled by convolutions with a stride in the last three layers. If an input silhouette  $x_j$  has dimensions  $C_{in} \times H_{in} \times W_{in}$  (where  $C_{in} = 1$  for silhouettes), the backbone transforms it into a 3D feature map  $f_j$  with dimensions  $C_{out} \times H_{out} \times W_{out}$ . Let's denote the backbone output for frame  $j$  as

$$f_j \in \mathbb{R}^{C \times H \times W}. \quad (3.7)$$

After extracting feature maps  $\{f_j\}_{j=1}^n$  for all  $n$  silhouettes from the input set, a Temporal Pooling module is used to aggregate these frame-level features into a single set-level 3D feature map, denoted as  $F \in \mathbb{R}^{C \times H \times W}$ . Following the approach in GaitSet [13], this aggregation is performed by taking the maximum along the temporal dimension. If we consider the gait sequence with  $n$  frames as a 4D tensor  $F_{frames} \in \mathbb{R}^{n \times C \times H \times W}$ , the temporal pooling operation can be expressed as:

$$F_{c,h,w} = \max_{j=1}^n (f_j)_{c,h,w}, \quad (3.8)$$

where  $(f_j)_{c,h,w}$  is the value at channel  $c$ , height  $h$ , and width  $w$  of the feature map for frame  $j$ , and  $F_{c,h,w}$  is the corresponding value in the aggregated set-level feature map  $F$ . This operation is permutation invariant, meaning the order of frames in the

input set does not affect the resulting set-level feature. Notably, silhouettes are binary images and inherently contain a single channel. Therefore, the  $C$  dimension will be ignored.

Table 3.3. Architecture of ResNet-like backbone.  $k$ ,  $c$ ,  $b$  denote kernel size, channels, and number of blocks.

| Layer  | Structure $[k \times k, c] \times b$   |                        |
|--------|----------------------------------------|------------------------|
| Conv0  | $3 \times 3, 64$                       | $\times 1, stride = 1$ |
| Block1 | $3 \times 3, 64$<br>$3 \times 3, 64$   | $\times 1, stride = 1$ |
| Block2 | $3 \times 3, 64$<br>$3 \times 3, 128$  | $\times 1, stride = 2$ |
| Block3 | $3 \times 3, 128$<br>$3 \times 3, 256$ | $\times 1, stride = 2$ |
| Block1 | $3 \times 3, 256$<br>$3 \times 3, 512$ | $\times 1, stride = 1$ |

The aggregated set-level feature map  $F \in \mathbb{R}^{h \times w}$  is then fed to the Horizontal Pooling layer. This technique is inspired by the part-based paradigm, and it aims to capture discriminative features from horizontal body parts. The feature map  $F$  is divided into  $P$  horizontal parts. For each part, pooling is applied spatially to reduce the dimensions and yield a part-specific feature vector. Unlike some more complex part-based methods that use pyramid structures involving multiple scales of horizontal divisions (like the HPM in GaitSet [13] and SMPLGait [21]), GaitBase explicitly uses horizontal pooling without a pyramid structure [27, 53]. This simplifies the architecture significantly.

For horizontal pooling without a pyramid, the feature map  $F \in \mathbb{R}^{H \times W}$  is divided into  $P$  fixed horizontal parts. For each part  $p \in \{1, \dots, P\}$ , the max pooling and average pooling operations are applied across the spatial dimensions  $(H/P) \times W$ . The

resulting feature vector for part  $p$  would be  $v_p$ , which can be denoted as

$$v_p = \text{MaxPool}(F_{set}^{(p)}) + \text{AveragePool}(F_{set}^{(p)}), \quad (3.9)$$

where MaxPool represents the maximum pooling function and AveragePool represents average pooling function. This process results in a sequence of  $P$  part-level feature vectors:  $v_1, v_2, \dots, v_P$ . The  $P$  part-level feature vectors  $v_1, \dots, v_P$  are then transformed into a discriminative feature space suitable for identification. Each part feature vector  $v_p$  is passed through its own separate fully connected (FC) layer. These FC layers project the part features into a new dimension  $d$ . Following these FC layers, a BNNeck module is applied.

The BNNeck structure, introduced in [54], is a crucial component for learning discriminative embeddings. BNNeck is particularly useful in identification and metric learning losses. It typically comprises FC layers without bias terms, followed by Batch Normalization (BN) layers. In the context of training with identity loss (cross-entropy) and triplet loss, features before the BN layer ( $f_t$ ) might be used for triplet loss calculation, while features after the BN layer ( $f_i$ ) are used for identity loss and for inference (retrieval).

For each part  $p$ , the part-level feature vector  $v_p$  is first projected by an FC layer into  $\mathbb{R}^d$ . Let this projected feature before BN be  $f_t^p \in \mathbb{R}^d$ . This is followed by a BN layer. The output of the BN layer for part  $p$  is  $f_i^p \in \mathbb{R}^d$ . The BN operation can be defined as

$$\hat{f}_t^p = \frac{f_t^p - \mathbb{E}[f_t^p]}{\sqrt{\text{Var}[f_t^p] + \epsilon}}, \quad (3.10)$$

$$f_i^p = \gamma \hat{f}_t^p + \beta, \quad (3.11)$$

where  $\mathbb{E}[f_t^p]$  and  $\text{Var}[f_t^p]$  are the mean and variance of the mini-batch features (calculated per dimension), and  $\gamma$  and  $\beta$  are learned scale and shift parameters. In the inference stage, the features  $f_i^p$  are used for distance calculation to perform identity retrieval. The final gait representation can be seen as the collection or concatenation of the BN-normalized part features  $f_i^1, \dots, f_i^P$ , resulting in a feature vector of dimension

$P \times d$ .

GaitBase is trained using a combined loss function that contains an identity (cross-entropy) loss and a triplet loss function. This is a common strategy in person re-identification and gait recognition to encourage both classification accuracy and metric space separation [2, 6]. The total loss  $L$  is typically the sum of these components:

$$L = L_{ID} + L_{Triplet}. \quad (3.12)$$

The identity loss  $L_{ID}$  is a standard cross-entropy loss where the model is trained to classify each input sample to its correct identity. This loss is typically computed using the features after the BN layer in the BNNeck. The triplet loss  $L_{Triplet}$  is a metric learning loss designed to minimize the distance between embeddings of positive pairs (samples from the same person) and maximize the distance between embeddings of negative pairs (samples from different people). This combined loss function helps the model to learn features that are not only discriminative for classification but also well-structured in the embedding space for effective retrieval. The overall diagram of the GaitBase framework is shown in Figure 3.2

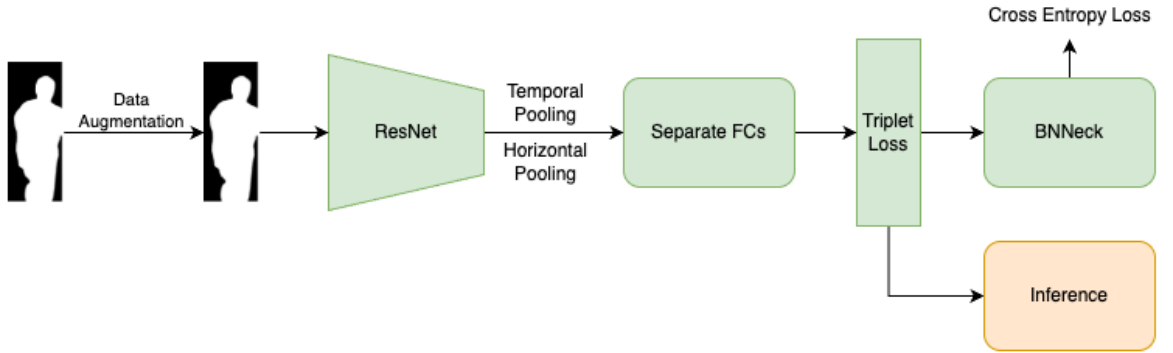


Figure 3.2. GaitBase Framework.

### 3.3. SwinGait

Vision Transformer (ViT) [55] typically divides an input image into non overlapping patches and treats each as a token. For gait silhouettes, a significant proportion

of these patches can be entirely black or entirely white, which lacks any distinguishing edge or shape information. These are called 'dumb' patches. The self-attention mechanism, which is a core component of Transformers, performs computations involving pairs of these tokens. When a 'dumb' patch is involved in a query-key pair, it can lead to useless gradients, particularly in the lower layers of the network. SwinGait's architecture is based on the Swin Transformer [56], which utilizes a shifted window attention mechanism. This design enables the model to process hierarchically non-overlapping patches of silhouettes. SwinGait addresses the 'dumb' patch issue by integrating convolution blocks into the initial stages of the network [28]. CNNs inherently handle dense feature extraction and expand their receptive fields through sliding windows and hierarchical structures. By employing convolution blocks for the initial layers, SwinGait transforms the sparse binary input into a dense, float-encoded feature map. Consequently, it mitigates the 'dumb patch' issue before introducing the Transformer layers.

Table 3.4. Architecture of 10-layer DeepGaitV2-3D.

| Layers | Output feature map size | DeepGaitV2-3D                                                                    |
|--------|-------------------------|----------------------------------------------------------------------------------|
| Conv0  | $(T, C, 64, 44)$        | $3 \times 3, stride = 1$                                                         |
| Stage1 | $(T, C, 64, 44)$        | $\begin{bmatrix} 3 \times 3, C \\ 3 \times 3, C \end{bmatrix}$                   |
| Stage2 | $(T, 2C, 32, 22)$       | $\begin{bmatrix} 3 \times 3 \times, 2C \\ 3 \times 3 \times 3, 2C \end{bmatrix}$ |
| Stage3 | $(T, 4C, 16, 11)$       | $\begin{bmatrix} 3 \times 3 \times, 4C \\ 3 \times 3 \times 3, 4C \end{bmatrix}$ |
| Stage4 | $(T, 8C, 16, 11)$       | $\begin{bmatrix} 3 \times 3 \times, 8C \\ 3 \times 3 \times 3, 8C \end{bmatrix}$ |
| TP     | $(1, 8C, 16, 11)$       | Temporal Pooling                                                                 |
| HP     | $(1, 8C, 16, 1)$        | Horizontal Pooling                                                               |
| Head   | $(1, 8C, 16, 1)$        | Flatten, 16 separate FCs and BNNecks                                             |

The SwinGait framework is presented in two variants: SwinGait-2D and its 3D counterpart, SwinGait-3D. These variants are built upon a general gait framework structure [27], which typically comprises a Backbone network, followed by TP, HP, and a Head module [2, 6, 18]. The overall architecture is illustrated in Figure 3.3. It shows the complete information flow starting with an input that represents a sequence of silhouettes. The initial input can be thought of as having the shape  $T \times C \times H \times W$ , where  $T$  is the number of frames (temporal dimension),  $C$  is the channel dimension (typically 1 for binary silhouettes),  $H$  is the height, and  $W$  is the width. The first two stages of the SwinGait backbone are instantiated as convolution blocks. These layers process the silhouette data, extract low-level structural features, and transform the input representation into a feature map. Following these initial convolution stages, the architecture employs Swin Transformer blocks for deeper feature learning.

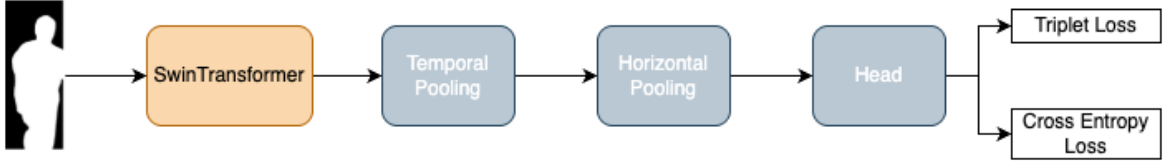


Figure 3.3. SwinGait Framework.

The key difference between SwinGait-2D and SwinGait-3D is the dimensionality of the residual connections in the Swin Transformer blocks, which is shown in Figure 3.4

- SwinGait-2D utilizes 2D Swin Transformer blocks, primarily operating on the spatial dimensions of the feature maps at each time step. This variant processes the gait sequence in a manner similar to set-based methods, where explicit temporal dependencies between successive frames are not necessarily modeled within the core feature extraction blocks.
- SwinGait-3D incorporates 3D Swin Transformer blocks, which operate across both spatial and temporal dimensions. This design is specifically aimed at capturing explicit temporal changes and dependencies between successive gait frames, a crucial aspect for understanding walking patterns.

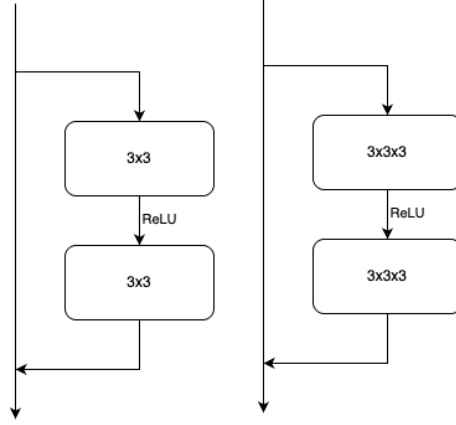


Figure 3.4. 2D/3D residual units.

The core of the SwinGait backbone (after the initial convolutional stages) lies in the Swin Transformer blocks. These blocks are designed to build hierarchical feature maps. A typical Swin Transformer block contains the following operations:

- Layer Normalization (LN): Applied before the multi-head self-attention (MSA) module and the Multi-Layer Perceptron (MLP). This helps stabilize training.
- Multi-head Self-Attention (MSA): Unlike standard self-attention computed globally across all tokens, the Swin Transformer computes self-attention within non-overlapping local windows that partition the feature map. This window-based approach significantly reduces the computational complexity from quadratic to linear with respect to the image size.
- Shifted Window (SW-MSA): To allow for connections between different windows across layers, the Swin Transformer introduces a shifted window partitioning approach in successive blocks. One block uses standard window partitioning, while the next uses a configuration where the windows are shifted, enabling cross-window connections. This alternating scheme is crucial for the model's representational power. The blocks are referred to as W-MSA (Window Multi-head Self-Attention) and SW-MSA.
- Residual Connections: Standard residual connections are employed around the MSA and MLP modules, similar to ResNet architectures, helping with training

deeper models.

- Multi-Layer Perceptron (MLP): A feed-forward network is applied after the self-attention module, typically consisting of two linear layers with a non-linear activation in between.

Figure 3.5 shows two successive 2D/3D Swin Transformer blocks, the structure with LN, MSA/SW-MSA, residual connections, and MLP. The 3D variant extends these operations to include the temporal dimension, performing self-attention within 3D windows that span spatial and temporal axes, which helps capture dynamic gait features.

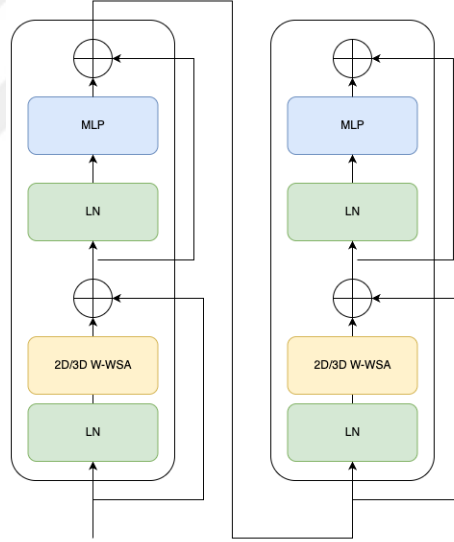


Figure 3.5. Two successive 2D/3D Swin Transformer blocks.

Following the backbone network (initial convolutions + Swin Transformer blocks), the general framework includes Temporal Pooling to aggregate features along the temporal axis, followed by Horizontal Pooling to divide and extract spatial features, and a BNNeck as Head to map the features into a discriminative space for identification. The training process typically utilizes loss functions such as triplet loss and cross-entropy loss. These parts are identical to the proposed general gait framework architecture in Section 3.2.

### 3.4. SkeletonGait

SkeletonGait is a gait recognition framework that uses skeletal representations to extract robust and discriminative features for person identification. The framework processes raw joint coordinate data and transforms it into structured feature representations through a pipeline consisting of preprocessing, skeleton map generation, feature extraction, temporal modeling, and classification.

The generation of the skeleton map starts with the raw coordinates of human joints  $(x_k, y_k, c_k)$ , where  $(x_k, y_k)$  represents the 2D location of the  $k$ -th joint and  $c_k$  its confidence score. In order to transform these raw coordinates into a meaningful gait representation, the following processing steps are employed. Firstly, to mitigate the influence of identity-unrelated factors such as walking trajectory and camera distance, the raw coordinates undergo center-normalization and scale-normalization steps. The center of the human body is approximated using the center point of the two hips, denoted as  $(x_{core}, y_{core}) = (\frac{x_{11}+x_{12}}{2}, \frac{y_{11}+y_{12}}{2})$ . The raw coordinates  $(x_k, y_k)$  are then normalized to  $(\hat{x}_k, \hat{y}_k)$  using the following transformations:

$$\hat{x}_k = x_k - x_{core} + \frac{R}{2} \quad \text{and} \quad \hat{y}_k = y_k - y_{core} + \frac{R}{2}, \quad (3.13)$$

where  $R$  is the resolution of the skeleton map. Subsequently, the body height is normalized to a predefined value  $H$  based on the maximum and minimum heights of the human joints, denoted as  $y_{max} = \max_k y_k$  and  $y_{min} = \min_k y_k$ . The coordinates are further scaled using:

$$x'_k = \frac{\hat{x}_k - y_{min}}{y_{max} - y_{min}} \times H \quad \text{and} \quad y'_k = \frac{\hat{y}_k - y_{min}}{y_{max} - y_{min}} \times H. \quad (3.14)$$

These normalized coordinates  $(x'_k, y'_k)$  effectively position the body's center of mass to  $(R/2, R/2)$  and standardize the body height.

After normalization, the skeleton map is rendered by representing the joints and limbs as heatmaps with a Gaussian approximation. Initially, a blank image of size  $R \times R$  is created. A joint map ( $J$ ) is generated by summing Gaussian maps centered

at each normalized joint position  $(x'_k, y'_k)$ , weighted by their confidence scores  $c_k$ :

$$J(i, j) = \sum_{k=1}^K e^{-\frac{(i-x'_k)^2 + (j-y'_k)^2}{2\sigma^2}} \times c_k, \quad (3.15)$$

where  $(i, j)$  represents a pixel coordinate on the  $R \times R$  map, and  $\sigma$  is a hyperparameter controlling the variance of the Gaussian distribution.

Similarly, a limb map ( $L$ ) is created by aggregating Gaussian approximations along each limb segment. For the  $n$ -th limb defined by connecting joints  $n-$  and  $n+$ , the contribution to a pixel  $(i, j)$  is based on the Euclidean distance  $D((i, j), S[n-, n+])$  from the pixel to the limb segment  $S[n-, n+]$ :

$$L(i, j) = \sum_{n=1}^N e^{-\frac{D((i, j), S[n-, n+])^2}{2\sigma^2}} \times \min(c_n, c_{n+}), \quad (3.16)$$

where  $N$  is the total number of limbs, and  $\min(c_n, c_{n+})$  uses the minimum confidence score of the connected joints as a weight. The final skeleton map is obtained by stacking the joint map and the limb map. This results in a gait representation with a size of  $2 \times R \times R$  (two channels). Lastly, to obtain a concrete representation and remove unnecessary blank regions, a subject-centered cropping operation is applied. The vertical range is determined by the non-zero pixel values, while the horizontal range is set from  $\frac{R-H}{2}$  to  $\frac{R+H}{2}$ . The cropped skeleton maps are then resized to a standard input size of  $2 \times 64 \times 64$ . As a result, the generated skeleton map has become a grid-based representation that explicitly distributes structural characteristics like limb length, ratio, and movement across spatial and temporal dimensions, inherently matching the locality requirements of spatiotemporal gait description.

SkeletonGait is the baseline model, which is constructed by building upon the successful DeepGaitV2 architecture [28]. SkeletonGait replaces DeepGaitV2's standard silhouette input with the newly developed skeleton map by changing the input channel of the initial convolutional layer to accommodate the two channels of the skeleton map. The reason behind this design choice is that it provides a direct means to compare the representational capabilities of structural features and structural features (silhouette). Moreover, DeepGaitV2 serves as a strong, state-of-the-art baseline

suitable for benchmarking. The data flow within the SkeletonGait framework largely mirrors that of DeepGaitV2, processing the sequence of skeleton maps through a series of stages to extract discriminative gait features. Starting with the input skeleton map sequence, which has a shape of  $T \times 2 \times 64 \times 44$ , where  $T$  is the sequence length, 2 is the channel count, and  $64 \times 44$  are the spatial dimensions. This input is processed by the initial convolutional layer, Conv0, which uses a  $3 \times 3$  kernel with a stride of 1 and outputs a feature map of size  $T \times C \times 64 \times 44$ , where  $C$  is the channel dimension for subsequent stages. This is followed by Stage1, composed of residual units, which maintains the spatial dimensions and channel count, resulting in a feature map of size  $T \times C \times 64 \times 44$ . Down-sampling occurs in Stage2 and Stage3, utilizing 3D residual units (or pseudo 3D units by default for computational efficiency), which involve 1D temporal and 2D spatial convolutions. Stage2 employs a stride of 2 for down-sampling, producing a feature map of size  $T \times 2C \times 32 \times 22$ . Stage3 further down-samples with a stride of 2, yielding a feature map of size  $T \times 4C \times 16 \times 11$ . Stage4, also using 3D residual units, maintains the spatial size with a stride of 1, increasing channels to  $8C$ , resulting in a feature map of size  $T \times 8C \times 16 \times 11$ . Following these convolutional stages, a TP module aggregates the feature map sequence along the temporal dimension, producing a set-level representation of shape  $1 \times 8C \times 16 \times 11$ . The principles and mathematics behind TP are detailed in Section 3.2. Subsequently, HP divides this pooled map horizontally into parts and extracts feature vectors from each part. The specifics of HP are also provided in Section 3.2. Finally, these feature vectors are passed through a Head module, consisting of separate fully connected layers and Batch Normalization (BN) Necks, which map the features into a discriminative metric space for recognition. The mathematical foundation for BNNecks can be found in Section 3.2. Building on the observation that body shape and structural features offer complementary information [21], the SkeletonGait++ framework is introduced as a multi-branch architecture designed to integrate both silhouette and skeleton map representations. This framework contains two branches. First branch is silhouette-based, which processes standard silhouette inputs similar to DeepGaitV2, and a skeleton branch, which processes skeleton maps similar to SkeletonGait. Both branches share similar network architectures, particularly in their early stages, like Conv0 and Stage1. A crucial

component of SkeletonGait++ is the fusion module, responsible for aggregating the features learned by the two branches. This fusion occurs frame-by-frame, although the diagram shown in Figure 3.6 might depict a single frame for simplicity. Various fusion mechanisms were considered, including element-wise Add, channel-wise Concatenate, and Attention-based fusion, which learns how to optimally combine the features from both branches. Notably, in this study, we used Attention-based fusion. Following the fusion point, the subsequent stages (Stage 2, 3, 4, TP, HP, and Head) process the combined feature representation using architectures similar to those in SkeletonGait. The initial silhouette input to the silhouette branch is  $T \times 1 \times 64 \times 44$ , and the skeleton map input to the skeleton branch is  $T \times 2 \times 64 \times 44$ . After the initial stages (Conv0, Stage1), both branches produce feature maps, which are then combined by the chosen fusion method before being passed to the later stages for further processing and final feature extraction via TP, HP, and the Head module, identical to the procedure in SkeletonGait.

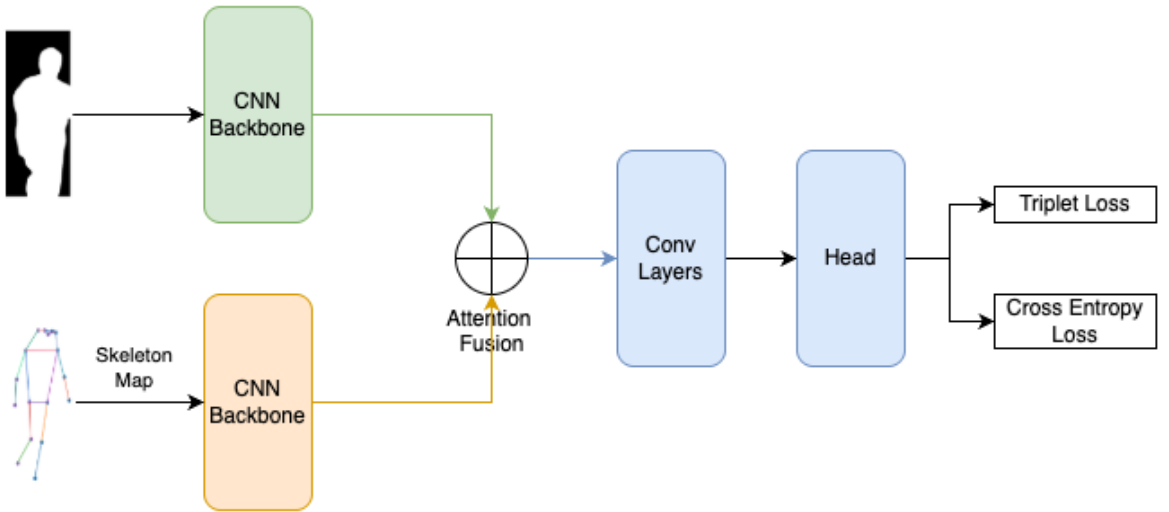


Figure 3.6. SkeletonGait Framework.

In essence, SkeletonGait and SkeletonGait++ explore the utility of structural gait features captured by the novel skeleton map representation, either in isolation or combined with traditional silhouette features, offering a systematic approach to leverage different modalities for improved gait recognition performance in varied environments.

The underlying network components like residual units, TP, HP, and BNNeck are foundational building blocks whose operational details are elaborated upon in Section 3.2.

### 3.5. Proposed Protocol

To assess viewpoint dependency regarding varying viewpoints in real-world scenarios, angular annotations of each gait sequence must be provided. However, existing benchmarks either lack such information or are constructed in controlled environments. For example, the CASIA-B dataset contains angular annotations for each gait sequence [25], and cameras are separated by  $18^\circ$ . Similarly, the OUMVLP dataset contains angular annotations for each gait sequence [26], and cameras are separated by  $15^\circ$ . However, these datasets are collected under a controlled environment. Therefore, to assess the viewpoint dependency of gait recognition methods under real-world scenarios, we selected the widely accepted in-the-wild GREW [24] and Gait3D [21] datasets. But the angular annotations for these datasets are not provided. Therefore, we propose a three-fold protocol to assess the effect of changing the camera viewpoint on gait recognition methods. First, we divided the aforementioned gait recognition datasets into  $10^\circ$  bins and assigned each gait sequence to a bin based on the most frequently occurring bin of its frames. Second, we restructured the datasets such that each gait sequence appears only in side-profile views in the training set and front or back views in the test set. The side-profile view mainly corresponds to  $180 \pm 30^\circ, 0 \pm 30^\circ$ . Lastly, we mapped the test set into training angles by rotating with the help of the 3D information. The implementation details of each protocol step are explained below.

#### 3.5.1. Angular Separation and Restructuring Datasets

Gait sequences from the Gait3D and GREW datasets are labeled with viewpoint angles and categorized into 10-degree angular bins using the pose information provided by the datasets. As a baseline assumption, we defined the viewpoint angle such that a pedestrian’s right-side profile corresponds to  $0^\circ$ , and the angles increase in the counterclockwise direction around the subject. For each pose corresponding to a

frame in the dataset, three points from the torso were selected. Then we calculated the corresponding angle for the gait frame using these points.

For the Gait3D dataset, OpenPose25 [57] format 3D keypoints are provided, and we utilized the central hip, right shoulder, and left shoulder keypoints for the angular annotation protocol. We call these points  $p_1 = (x_{p1}, y_{p1}, z_{p1})$ ,  $p_2 = (x_{p2}, y_{p2}, z_{p2})$ , and  $p_3 = (x_{p3}, y_{p3}, z_{p3})$ , respectively. We draw two vectors from the shoulder to hip keypoints for the keypoints of the frame, and the resulting vectors are called  $p_1\vec{p}_2$  and  $p_1\vec{p}_3$ . Specifically, using the coordinates of the selected keypoints, two vectors were constructed for the frame using

$$p_i\vec{p}_j = \begin{bmatrix} x_{pj} - x_{pi} \\ y_{pj} - y_{pi} \\ z_{pj} - z_{pi} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}. \quad (3.17)$$

The cross product of these vectors was then computed to obtain a normal vector  $\vec{n}_{Gait3D}$  perpendicular to the plane formed by them using

$$\vec{n}_{Gait3D} = p_1\vec{p}_2 \times p_1\vec{p}_3 = \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix}, \quad (3.18)$$

where  $p_1\vec{p}_2 = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$  and  $p_1\vec{p}_3 = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$  and  $\vec{n}_{Gait3D}$  is the normal vector of the frame from the Gait3D dataset. Since the gait is primarily along the x-axis, the angle  $\theta_{Gait3D}$  between the normal vector  $\vec{n}_{Gait3D}$  and the unit vector along the x-axis is calculated using

$$\theta_{Gait3D} = \arccos \left( \frac{\vec{n}_{Gait3D} \cdot \vec{\hat{x}}}{\|\vec{n}_{Gait3D}\| \|\vec{\hat{x}}\|} \right), \quad (3.19)$$

where  $\vec{\hat{x}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$  and  $\theta_{Gait3D}$  is in radians. Since  $\theta_{Gait3D} \in [0, \pi]$ , we needed to check the sign of the  $z$  dimension and map the output value such that  $\theta_{Gait3D} \in [0, 2\pi]$ . This

procedure can be denoted as

$$\text{If } n_{Gait3D,z} > 0, \text{ then } \theta_{Gait3D} = 2\pi - \theta_{Gait3D}, \quad (3.20)$$

where a positive  $z$  value corresponds to an orientation in the negative direction since the angles are assumed to be increasing in the counterclockwise direction. Finally, we convert  $\theta_{Gait3D}$  to degrees as

$$\hat{\theta}_{Gait3D} = \theta_{Gait3D} \times \frac{180}{\pi}, \quad (3.21)$$

where  $\hat{\theta}_{Gait3D}$  is the angular annotation in degrees of the frame from the Gait3D dataset. We also express this angle as  $\hat{\Theta}_{Gait3D} = [l_{lower}, l_{upper}]$  where  $l_{lower}$  and  $l_{upper}$  represent the lower and upper boundary of the  $10^\circ$ -wide angular bin. We repeated these steps for each frame of the gait sequence in the Gait3D dataset. Repetition of these steps for every gait frame allowed each frame to be labeled with its corresponding angle and boundaries. Then, we apply a majority voting technique to assign each gait sequence to an angular bin. Let a gait sequence be defined as a set of gait frames  $G = \{g_1, g_2, \dots, g_n\}$  where each  $g_i$  corresponds to a gait frame characterized by some angular bin. To determine the most recurring angular bin over the entire gait sequence, we apply a majority voting operation based on the median over the set  $G$ , denoted as

$$\hat{\Theta}_G = \text{median}(\{\hat{\Theta}_{g_1}, \hat{\Theta}_{g_2}, \dots, \hat{\Theta}_{g_n}\}), \quad (3.22)$$

where  $\hat{\Theta}_{g_i}$  denotes the angular bin of the gait frame  $g_i$ . We then label the entire gait sequence  $G$  with this most recurring angular bin  $\hat{\Theta}_G$ . This procedure is applied to all gait sequences in the Gait3D dataset, and Figure 3.7 illustrates examples of the Gait3D dataset divided into  $10^\circ$  angular segments following our separation protocol.

COCO-17 [57] 3D skeleton keypoints are provided in the GREW dataset, and we utilized the chin, left hip, and right hip keypoints for the angular annotation protocol. We call these points  $k_1 = (x_{k1}, y_{k1}, z_{k1})$ ,  $k_2 = (x_{k2}, y_{k2}, z_{k2})$ , and  $k_3 = (x_{k3}, y_{k3}, z_{k3})$ , respectively. Using the coordinates of the selected points, we draw two vectors from the hip to the chin for the frame from the GREW dataset, and the resulting vectors are called  $k_1\vec{k}_2$  and  $k_1\vec{k}_3$ . The cross product of these vectors was then computed to obtain a normal vector  $\vec{n}_{GREW}$  perpendicular to the plane formed by them using  $\vec{n}_{GREW} = k_1\vec{k}_2 \times k_1\vec{k}_3$ . Since the gait is primarily along the x-axis, the angle  $\theta$  between

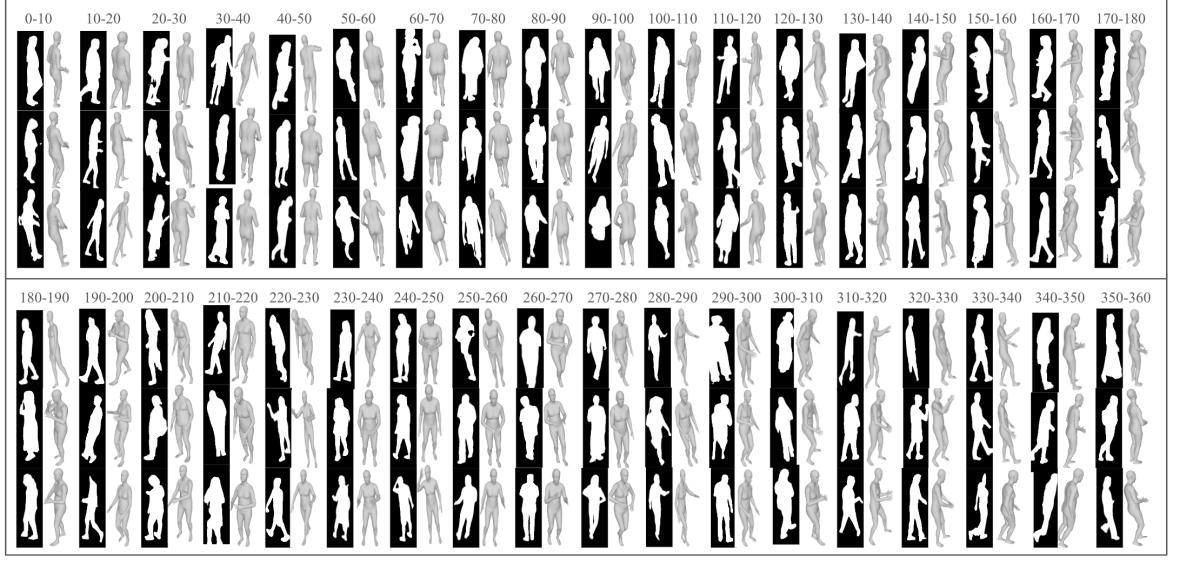


Figure 3.7. Gait3D dataset divided into different angles using the angular separation protocol. Silhouettes and corresponding SMPL representations from different individuals are shown for each angle range.

the normal vector  $\vec{n}_{GREW}$  and the unit vector along the x-axis  $\hat{x}$  is calculated and denoted as  $\theta_{GREW}$ . Then, we arranged the correct sign regarding the  $z$  dimension and mapped the output value such that  $\theta_{GREW} \in [0, 2\pi]$ . Finally, we converted the value from radians to degrees and denoted it as  $\hat{\theta}_{GREW}$ . As a result, the frame is labeled with the corresponding angle and boundaries. Similarly, this procedure is repeated for each frame of the gait sequence in the GREW dataset, and  $\hat{\Theta}_G$  is calculated for all gait sequences in the GREW dataset. Figure 3.8 illustrates examples from the GREW dataset divided into  $10^\circ$  angular segments following our separation protocol.

### 3.5.2. Assessment of Viewpoint Invariance

In the second phase, we split the Gait3D and GREW datasets into training and testing subsets using our assessment protocol. We aim to evaluate viewpoint invariance of gait recognition methods by using restructured datasets. This structuring procedure involves individuals' gait sequences, which appear in side profiles for the training set and in front or back profiles for the test set. To achieve this, angular

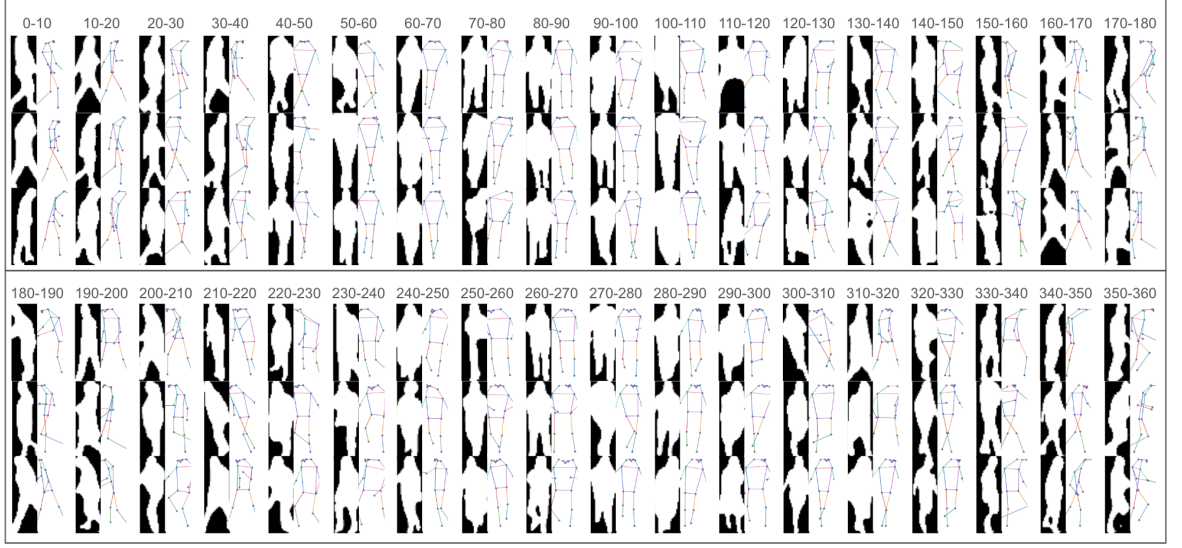


Figure 3.8. GREW dataset divided into different angles using the angular separation protocol. Silhouettes and 2D Pose representations from different individuals are shown for each angle range.

ranges of  $10^\circ$  increments were defined and gait sequences are fitted into corresponding ranges in Section 3.5.1. Regarding our protocol, gait sequences within the  $0 \pm 30^\circ$  and  $180 \pm 30^\circ$  angular ranges were considered as the training set, while sequences outside these ranges were marked as the test set. A detailed distribution of the restructured Gait3D and GREW datasets and the train-test split are provided in Table 3.5 and Table 3.6, respectively. Moreover, visual interpretation of the distributions are shown in Figure-3.9 and Figure-3.10, respectively. Then, we train the best-performing gait recognition methods on our restructured Gait3D and GREW datasets (Gait3D, Gait-Base, SkeletonGait++, SwinGait) and evaluate them on the restructured test set that contains only gait sequences in unseen viewpoints. This procedure reveals the view-point invariance of gait recognition methods by highlighting each model’s ability to generalize to unseen angles.

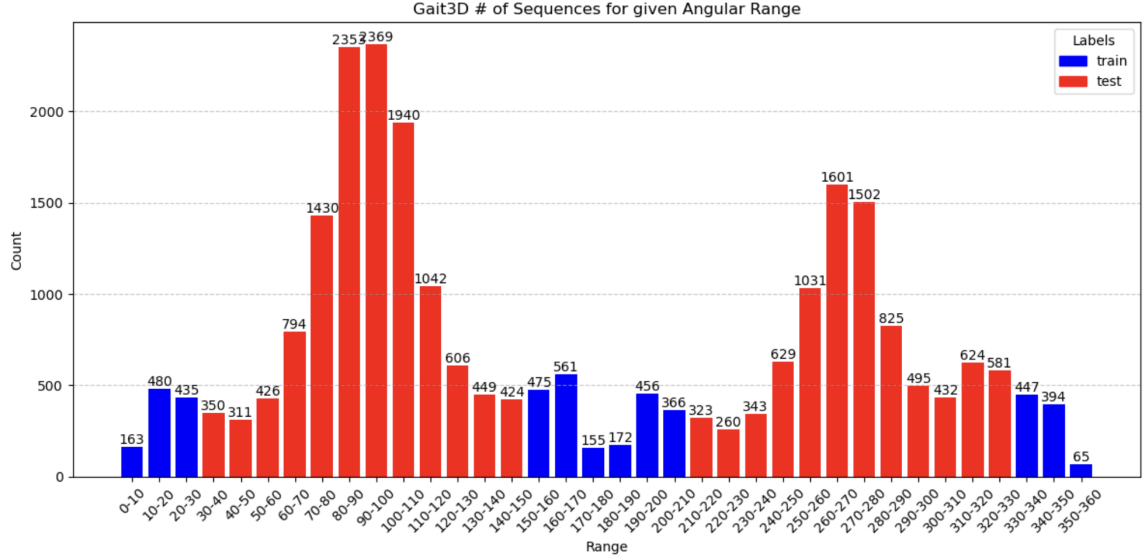


Figure 3.9. Distribution of Gait Sequences of restructured Gait3D dataset by our protocol and categorized into 10-degree angular bins. Data is taken from Table 3.5.

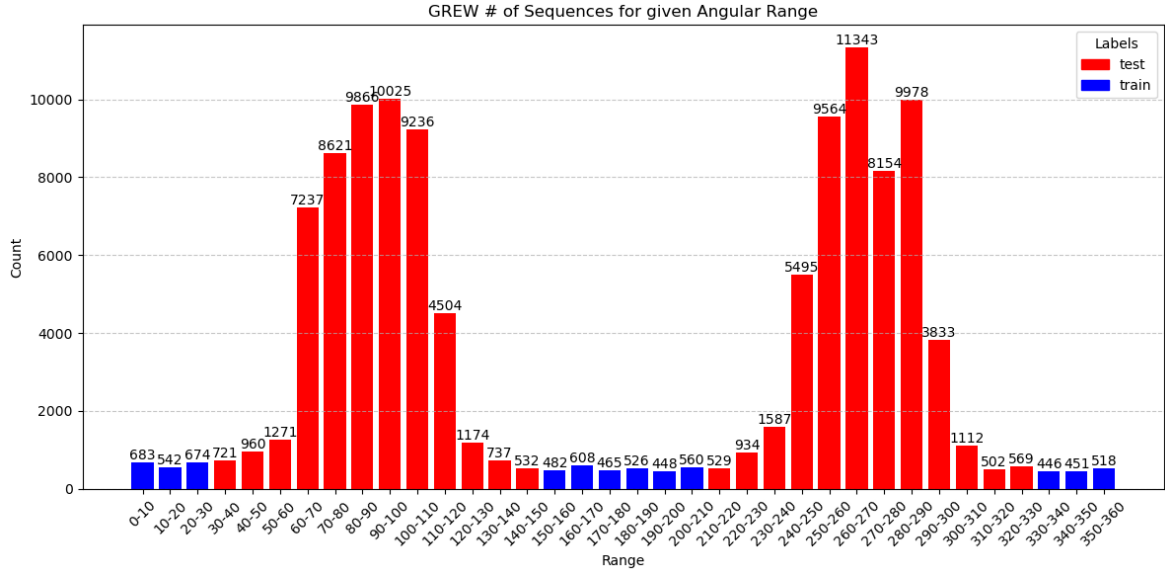


Figure 3.10. Distribution of Gait Sequences of restructured GREW dataset by our protocol and categorized into 10-degree angular bins. Data is taken from Table 3.6.

Table 3.5. Distribution of Gait Sequences of restructured Gait3D dataset by our protocol and categorized into 10-degree angular bins.

| Angle Range | # of Gait Sequences | # of Gait Frames |
|-------------|---------------------|------------------|
| 0-10        | 163                 | 22809            |
| 10-20       | 480                 | 67168            |
| 20-30       | 435                 | 59665            |
| 30-40       | 370                 | 51776            |
| 40-50       | 341                 | 55809            |
| 50-60       | 516                 | 70776            |
| 60-70       | 837                 | 105834           |
| 70-80       | 1495                | 169645           |
| 80-90       | 2318                | 255005           |
| 90-100      | 2338                | 267028           |
| 100-110     | 1845                | 200116           |
| 110-120     | 1167                | 127529           |
| 120-130     | 670                 | 87511            |
| 130-140     | 511                 | 68027            |
| 140-150     | 451                 | 61916            |
| 150-160     | 475                 | 65152            |
| 160-170     | 561                 | 79948            |
| 170-180     | 155                 | 21260            |
| 180-190     | 172                 | 23592            |
| 190-200     | 456                 | 62546            |
| 200-210     | 366                 | 50201            |
| 210-220     | 426                 | 53454            |
| 220-230     | 411                 | 54168            |
| 230-240     | 486                 | 65784            |
| 240-250     | 739                 | 90583            |
| 250-260     | 1097                | 141414           |
| 260-270     | 1470                | 196526           |
| 270-280     | 1348                | 180662           |
| 280-290     | 865                 | 120497           |
| 290-300     | 589                 | 83078            |
| 300-310     | 573                 | 73226            |
| 310-320     | 576                 | 78231            |
| 320-330     | 423                 | 75300            |
| 330-340     | 447                 | 61311            |
| 340-350     | 394                 | 54042            |
| 350-360     | 65                  | 8778             |

Table 3.6. Distribution of Gait Sequences of restructured GREW dataset by our protocol and categorized into 10-degree angular bins.

| Angle Range | # of Gait Sequences | # of Gait Frames |
|-------------|---------------------|------------------|
| 0-10        | 683                 | 50069            |
| 10-20       | 542                 | 52580            |
| 20-30       | 674                 | 64334            |
| 30-40       | 721                 | 78392            |
| 40-50       | 960                 | 123855           |
| 50-60       | 1271                | 221212           |
| 60-70       | 3198                | 703123           |
| 70-80       | 8963                | 948209           |
| 80-90       | 10099               | 1054216          |
| 90-100      | 13967               | 1032210          |
| 100-110     | 8782                | 843861           |
| 110-120     | 4439                | 507308           |
| 120-130     | 1174                | 171636           |
| 130-140     | 737                 | 95084            |
| 140-150     | 532                 | 66709            |
| 150-160     | 482                 | 54597            |
| 160-170     | 508                 | 53638            |
| 170-180     | 465                 | 47256            |
| 180-190     | 526                 | 46209            |
| 190-200     | 448                 | 47857            |
| 200-210     | 560                 | 56316            |
| 210-220     | 529                 | 67344            |
| 220-230     | 934                 | 113877           |
| 230-240     | 1587                | 224660           |
| 240-250     | 4240                | 569709           |
| 250-260     | 9124                | 929675           |
| 260-270     | 10551               | 1079011          |
| 270-280     | 13280               | 935900           |
| 280-290     | 6381                | 933799           |
| 290-300     | 6381                | 468465           |
| 300-310     | 1112                | 180098           |
| 310-320     | 502                 | 86150            |
| 320-330     | 569                 | 66181            |
| 330-340     | 446                 | 51114            |
| 340-350     | 451                 | 45211            |
| 350-360     | 518                 | 48944            |

### 3.5.3. Rotating Gait Representations

To further investigate the viewpoint invariance in gait recognition methods, we rotated the gait representations in the Gait3D and GREW datasets. Specifically, we rotated each gait representation to the closest training angle and then evaluated using our protocol described in Section 3.5.2. We selected the rotation axis as the  $y$ -axis to align gait representations with the predefined training angles. The 3D gait representations provided in the datasets were then rotated accordingly.

For the Gait3D dataset, pose results and human mesh coordinates were provided as 3D gait representations. We define our rotation matrix  $R_y(\theta) \in \mathbb{R}^{3 \times 3}$  around the  $y$ -axis by  $\theta$  degrees as

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}. \quad (3.23)$$

To align a test sample with the closest training viewpoint, we determine the required rotation angle  $\hat{\theta}_{g_i, rot}$  of the gait frame  $g_i \in G$  by computing the minimum distance to the nearest training angle as

$$\phi_{train} = \arg \min_{\phi \in \{0^\circ, 180^\circ, 360^\circ\}} |\phi - \hat{\theta}_{G, mean}|, \quad (3.24)$$

$$\hat{\theta}_{g_i, rot} = \phi_{train} - \hat{\theta}_{g_i}, \quad (3.25)$$

where  $\hat{\theta}_{G, mean} = \frac{l_{upper} + l_{lower}}{2}$  is the average value of the angular bin of the gait sequence  $G$  such that  $\hat{\theta}_G = [l_{lower}, l_{upper}]$  as described in Section 3.5.1,  $\phi \in \{0^\circ, 180^\circ, 360^\circ\}$  are the training angles as described in Section 3.5.2 and  $\phi_{train}$  is the closest angle to the gait sequence  $G$ . SMPL representation is created by concatenating three matrices provided by the Gait3D dataset, which are pose (24, 3), shape (10, ), and cam (3, ) parameters. The resulting array is (85, ) shaped. We rotate SMPL information by simply rotating the shape coordinates, as the cam parameters and betas are irrelevant to the viewpoint. For the given gait frame  $g_i$ , the rotated SMPL representation can be obtained as

$$SMPL_{g_i, rot} = [SMPL_{g_i} \times R_y(\hat{\theta}_{g_i, rot}), cam_G, betas_G], \quad (3.26)$$

where  $SMPL_{g_i,rot} \in \mathbb{R}^{85 \times 1}$  is the rotated SMPL representation for the gait frame  $g_i$ . Similarly, we rotate 3D pose information  $P$  for the gait frame  $g_i$  as

$$P_{g_i,rot} = P \times R_y(\hat{\theta}_{g_i,rot}), \quad (3.27)$$

where  $P_{g_i,rot} \in \mathbb{R}^{25 \times 3}$  is the rotated pose representation for the gait frame  $g_i$ . Then, we ignored the  $z$  dimension of the  $P_{g_i,rot}$  to obtain the corresponding rotated 2D pose representations  $P_{g_i,rot}^{2D} \in \mathbb{R}^{25 \times 2}$ . Finally, we utilized SMPL representations to obtain both silhouettes created from original SMPLs and those created from rotated SMPLs as described in Section 3.5.4, which will be used in further experiments. We used the provided 3D coordinates of the human mesh  $M_{g_i} \in \mathbb{R}^{6890 \times 3}$  from the Gait3D dataset and created a binary image (silhouette) using a thresholding technique as follows

$$B(x, y) = \begin{cases} 1 & \text{if } \alpha(x, y) > T \\ 0 & \text{otherwise} \end{cases}, \quad (3.28)$$

where  $\alpha(x, y)$  is the alpha channel value of the rasterized mesh at pixel  $(x, y)$ , and  $T$  is the threshold. For the sake of simplicity, we set  $T$  as 0.5 in our study. Similarly, we rotated the human mesh  $M_g$  using  $M_{g_i,rot} = M_{g_i} \times R_y(\hat{\theta}_{g_i,rot})$  and obtained rotated silhouette by substituting rasterized mesh  $M_{g_i,rot}$  into equation 3.28. Figure 3.11 illustrates examples from the gait representations following our rotation protocol.

For the GREW dataset, only the pose information was provided as a 3D gait representation. For a given gait frame  $g_i$  of a gait sequence  $G$  from the GREW dataset, we rotate 3D pose information  $P_{g_i} \in \mathbb{R}^{17 \times 3}$  by using equation 3.27. Then, we simply ignore the  $z$  dimension to obtain the rotated 2D pose  $P_{g_i,rot}^{2D} \in \mathbb{R}^{17 \times 2}$ .

### 3.5.4. Generating Gait Representations

The generation of 1-channel and RGB Depth Images from 3D meshes involves rendering a mesh from a specific camera viewpoint and capturing depth information at each pixel. This process leverages principles of computer graphics rendering. Let a 3D mesh be defined by its vertices  $V = \{v_j \in \mathbb{R}^3\}_{j=1}^N$  and faces  $F = \{f_k \subset V\}_{k=1}^M$ . A camera's position and orientation are defined by a rotation matrix  $R \in SO(3)$  and

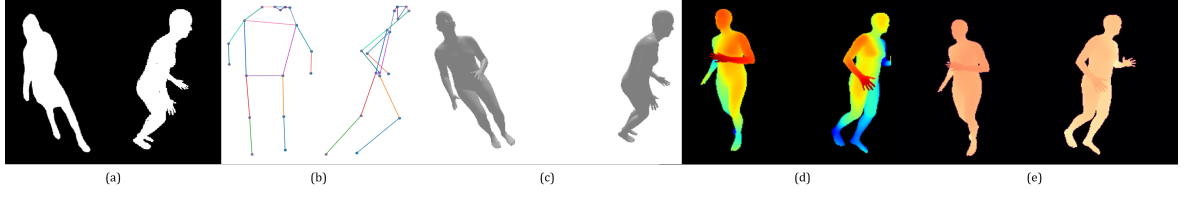


Figure 3.11. Demonstration of the rotation protocol on gait representations. The original representations are shown on the left, and the rotated images are on the right of each image. (a) contains the original silhouette on the left and a rotated version on the right. (b) and (c) contains the same configuration for 2D Skeletons and SMPLs, respectively. Similarly, (d) and (e) contain Depth and RGB Depth images.

a translation vector  $T \in \mathbb{R}^3$ . These parameters determine the transformation from world coordinates to camera coordinates. A 3D point  $P_{world} \in \mathbb{R}^3$  is transformed to  $P_{cam} \in \mathbb{R}^3$  as:

$$P_{cam} = RP_{world} + T. \quad (3.29)$$

The depth of the point from the camera is typically represented by the third component of  $P_{cam}$ , denoted as  $Z_{cam}$ . The rasterization process projects the 3D mesh onto a 2D image plane of dimensions  $W \times H$ . For each pixel  $(u, v)$  on this plane, a depth buffer  $D_{raw}(u, v)$  stores the minimum  $Z_{cam}$  value among all mesh points that project onto that pixel:

$$D_{raw}(u, v) = \min\{Z_{cam} \mid P_{world} \text{ on mesh, projects to } (u, v)\}. \quad (3.30)$$

If no mesh fragment projects to a pixel,  $D_{raw}(u, v)$  is assigned a designated background value (e.g., a very large number or zero).

To generate a 1-channel (grayscale) depth image, the raw depth values  $D(u, v)$  from  $D_{raw}$  are processed as follows:

- (i) Identify Valid Depth Values: Filter out background values. Let  $D_{valid}$  be the set of valid depth values, i.e., those representing actual mesh points.
- (ii) Determine Min and Max Valid Depths: Calculate the minimum ( $D_{min}$ ) and maximum ( $D_{max}$ ) values within  $D_{valid}$ :

$$D_{min} = \min_{(u,v) \in D_{valid}} D(u, v). \quad (3.31)$$

$$D_{max} = \max_{(u,v) \in D_{valid}} D(u, v). \quad (3.32)$$

(iii) Normalize Valid Depths: Scale the valid depth values to the range  $[0, 1]$ :

$$D_{norm}(u, v) = \frac{D(u, v) - D_{min}}{D_{max} - D_{min} + \epsilon}, \quad (3.33)$$

where  $\epsilon$  is a small positive constant (e.g.,  $10^{-8}$ ) to prevent division by zero.

(iv) Set Background: Pixels corresponding to the background are typically assigned a specific normalized value (e.g., 1.0 for the farthest depth):

$$D_{norm}(u, v) = 1.0 \quad \text{if } (u, v) \text{ is a background pixel.} \quad (3.34)$$

(v) Scale to Grayscale Range: Convert the normalized values to an 8-bit grayscale intensity range  $[0, 255]$ :

$$D_{1C}(u, v) = \lfloor D_{norm}(u, v) \times 255 \rfloor. \quad (3.35)$$

The output  $D_{1C}(u, v)$  is a single-channel image where pixel intensity directly represents the normalized depth.

For an RGB Depth Image, the normalized depth values are mapped to a color space.

(i) Extract and Normalize Depth Values: Similar to the 1-channel process, raw depth values  $D(u, v)$  are extracted from  $D_{raw}$ , and valid depths are normalized to  $D_{norm}(u, v) \in [0, 1]$ .

(ii) Optional Inversion: For visual clarity (e.g., making closer objects brighter with certain colormaps), the normalized depth can be inverted:

$$D_{norm\_inv}(u, v) = 1.0 - D_{norm}(u, v). \quad (3.36)$$

(iii) Color Mapping: A colormap function  $\text{Colormap} : [0, 1] \rightarrow \mathbb{R}^3$  (e.g., a "jet" colormap) is applied to transform the scalar normalized depth values into RGB color triplets:

$$D_{RGB}(u, v) = \text{Colormap}(D_{norm\_inv}(u, v)). \quad (3.37)$$

(iv) Set Background Color: Background pixels are typically rendered as black  $(0, 0, 0)$ .

The resulting  $D_{RGB}(u, v)$  is a 3-channel RGB image, providing a visually rich representation of depth.

Notably, 1-channel and 3-channel depth images are named as 2.5D. For the rest of this thesis, we will refer to silhouette-based representations as 2D, deformable human body-based (SMPL) representations as 3D, and Depth Image-based representations as 2.5D. The generation of silhouettes from 3D meshes involves rendering the mesh from a specific viewpoint, with a particular focus on extracting the object’s outline rather than its texture or lighting.

## 4. EXPERIMENTS AND RESULTS

### 4.1. Datasets

There are numerous datasets available in the field of gait recognition. In this study, we utilized the widely accepted in-the-wild datasets, GREW [24] and Gait3D [21]. Both of these datasets not only provide large-scale, unconstrained gait data but also include valuable 3D information. In this section, we provide a detailed description of these two datasets. Additionally, we summarize other notable gait recognition datasets that have been introduced in the literature in Table 4.1.

Table 4.1. Statistics of Gait Recognition Benchmarks.

| Dataset             | Year | Data Types                    | # of Subjects | # of Sequences | # of Views |
|---------------------|------|-------------------------------|---------------|----------------|------------|
| CASIA-B [25]        | 2006 | RGB, Sil.                     | 126           | 13,640         | 11         |
| CASIA-C [58]        | 2006 | Infrared, Sil.                | 153           | 1,530          | 1          |
| KY4D [59]           | 2010 | Sil., RGB, 3D Volumetrics     | 42            | 168            | 16         |
| SZTAKI-LGA [60]     | 2016 | 3D Point Cloud                | 28            | 11             | 1          |
| OU-MVLP [26]        | 2018 | Sil.                          | 10,307        | 288,596        | 14         |
| FVG [61]            | 2019 | RGB                           | 226           | 2,856          | 3          |
| PCG [62]            | 2020 | 3D Point Cloud                | 30            | 60             | 1          |
| GREW [24]           | 2021 | Sil., 2D/3D Skeleton, Flow    | 26,345        | 128,671        | 882        |
| Gait3D [21]         | 2022 | Sil., 2D/3D Skeleton, 3D Mesh | 4,000         | 25,309         | 39         |
| OUMVLP-Mesh [63]    | 2022 | 3D Mesh                       | 10,307        | 288,596        | 14         |
| Gait3D-Parsing [48] | 2023 | Parsing                       | 4,000         | 25,309         | 39         |
| SUSTech1K [14]      | 2023 | RGB, Sil., 3D Point Cloud     | 1,050         | 25,239         | 12         |
| OccCASIA-B [64]     | 2023 | RGB, Sil.                     | 124           | 13,640         | 11         |
| CCPG [65]           | 2024 | RGB, Sil.                     | 200           | 15,566         | 10         |
| CCGait [46]         | 2024 | Sil., 2D/3D Skeleton, 3D Mesh | 1,495         | 4,828          | 9          |
| OccGait [41]        | 2024 | Sil.                          | 101           | 80,000+        | 8          |

#### 4.1.1. Gait3D

The Gait3D dataset is a large-scale benchmark for gait recognition in real-world, unconstrained environments. It contains 25,309 gait sequences from 4,000 subjects, averaging 129 frames per sequence, captured using 39 cameras in various indoor settings.

Unlike many earlier datasets, Gait3D incorporates dense 3D representations, including silhouettes, SMPL models, and skeletons. The Gait3D dataset provides richer information about body shape, viewpoint, and motion dynamics, making it particularly suitable for studying the challenges of viewpoint invariance, occlusions, and diverse gait variations in real-world applications. Example gait representations from the Gait3D dataset are shown in Figure 4.1.

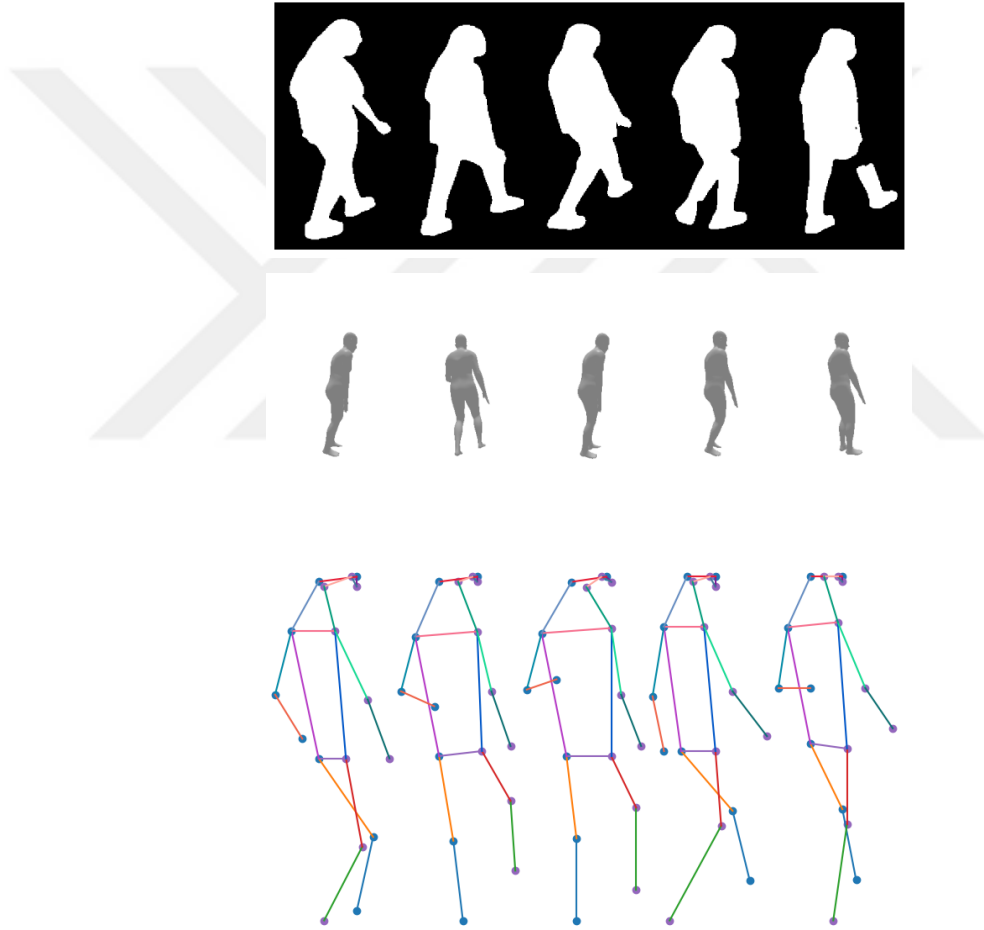


Figure 4.1. Sample gait representations from the Gait3D dataset.

#### 4.1.2. GREW

The GREW dataset is a large-scale benchmark for gait recognition in real-world, unconstrained settings. It comprises 128,671 gait sequences from 26,345 subjects, av-

eraging 98 frames per sequence, collected from outdoor surveillance footage. GREW contains varying data types such as silhouettes, 2D/3D poses, and GEIs. The dataset addresses challenges like varying viewpoints, clothing, and occlusions, making it valuable for testing gait recognition models in practical scenarios. Sample gait representations from the Gait3D dataset are shown in Figure 4.2.

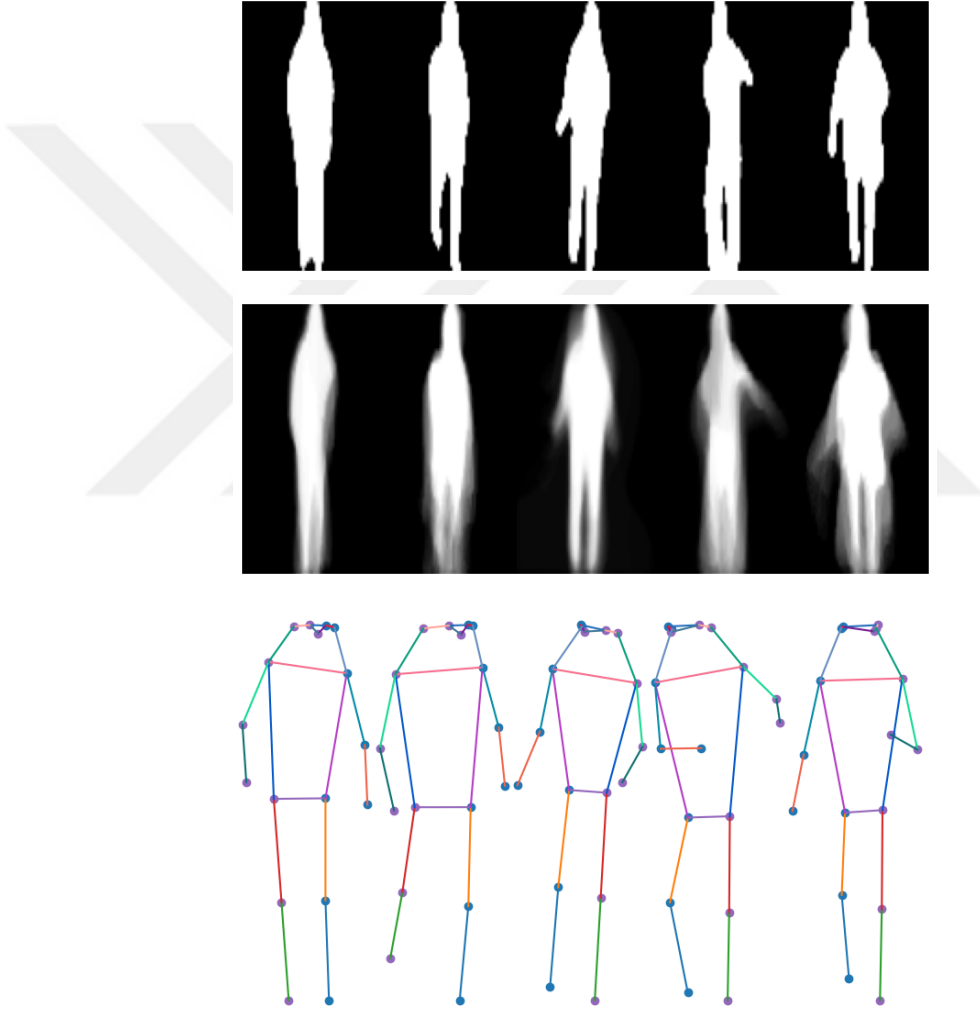


Figure 4.2. Sample gait representations from the GREW dataset.

## 4.2. Quantitative Results

The experimental setup was designed to assess the viewpoint invariance by training top-performing gait recognition frameworks using our protocol, which are SMPL-

Gait, GaitBase, SwinGait, and SkeletonGait. These models were selected due to their leading performance on the Gait3D and GREW datasets in prior studies. For these experiments, the Gait3D and GREW datasets were divided into angular bins using our protocol in Section 3.5, as shown in Figure 3.9 and Figure 3.10. We then rotated the test set to the closest training angle and evaluated these models on the rotated dataset. The default training and evaluation configurations were employed during training. In the following subsections, we present the experiments and interpret the results obtained using the Gait3D and GREW datasets, respectively.

#### 4.2.1. Gait3D Experiments

In Experiment 1, we aim to validate the reproducibility of the reported results of the top-performing gait recognition frameworks, which are SMPLGait, GaitBase, SwinGait, and SkeletonGait. We trained these gait recognition frameworks from scratch and reported the rank-1 accuracies in the Gait3D dataset, which is shown in Table 4.2. Notably, SMPLGait-2D represents the baseline version of SMPLGait-3D, which is trained only with 2D data. Moreover, SwinGait-3D does not utilize 3D information, it uses only 2D gait representation (Silhouettes). The 3D suffix is added to this method due to its usage of temporal information as a third dimension.

Table 4.2. Experiment-1 results.

| Framework    | Input Data              | Reported R-1 Accuracy | Our R-1 Accuracy |
|--------------|-------------------------|-----------------------|------------------|
| SMPLGait-2D  | Silhouette              | 42.9                  | 43.6             |
| SMPLGait     | Silhouette+SMPL         | 46.3                  | 46.3             |
| GaitBase     | Silhouette              | 64.6                  | 64.6             |
| SwinGait-3D  | Silhouette              | 75.0                  | 75.0             |
| SkeletonGait | Skeleton Map+Silhouette | 77.6                  | 77.4             |

In Experiment 2, we aim to assess the robustness to viewpoint changes of gait recognition frameworks. First, we restructured the Gait3D dataset into  $10^\circ$  angular bins. We separated it into a training and testing subset according to the protocol de-

scribed in Section 3.5 and named it the restructured Gait3D dataset. Then, we trained the same gait recognition frameworks of Experiment 1 from scratch with the restructured Gait3D dataset. We reported the evaluation results in the restructured Gait3D test set in Table 4.3 and statistical interpretation of the results in Table 4.4. Notably, we have calculated the weighted average using the corresponding # of gait sequences in each bin and denoted it as average, which is provided in Table 3.5. In Table 4.3, we present the input types of each method in the fourth row, including Silhouettes (Sil.), SMPL parameters (SMPL), and Skeleton Maps (SM). For example, SMPLGait-3D uses Silhouettes and SMPL parameters as input representation. Table 4.3 represents a comparative analysis of four gait recognition models retrained using the proposed training and evaluation protocol. Additionally, Table 4.3 highlights the performance of pre-trained models across different angular ranges, and it demonstrates the advantages of incorporating gait sequences from diverse viewpoints in models trained with the proposed protocol. A visual representation of Experiment 2 results can be found in Figure 4.3. Notably, dashed lines in Figure 4.3 correspond to the evaluation results of the models on the restructured Gait3D dataset that are trained for replication in Experiment 1.

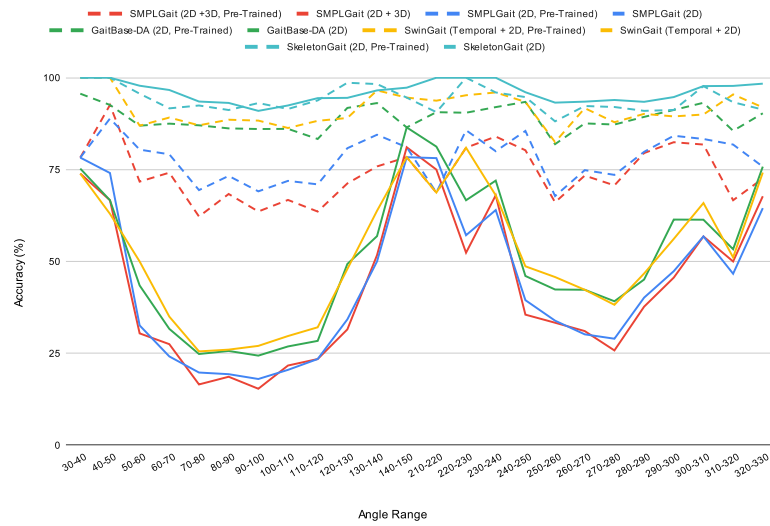


Figure 4.3. The rank-1 accuracies of 2D and 3D Gait Recognition methods in Gait3D dataset obtained from Experiment 1. The data is taken from Table 4.3.

Table 4.3. Evaluation of gait recognition frameworks on the Gait3D dataset. Angular bins are constructed according to the angular separation protocol. Values correspond to rank-1 accuracies.

| Variant        | Pre-Trained |      |      |          |       |       | Trained with Gait3D dataset using our protocol |         |         |          |      |         |
|----------------|-------------|------|------|----------|-------|-------|------------------------------------------------|---------|---------|----------|------|---------|
|                | SMPLGait    |      |      | SMPLGait |       |       | SMPLGait                                       |         |         | SMPLGait |      |         |
|                | Sil. + SMPL | 2D   | 3D   | Sil.     | 2D    | 3D    | Sil.                                           | 2D      | 3D      | Sil.     | 2D   | 3D      |
| Input Type     | Sil. + SMPL | Sil. | Sil. | Sil.     | Sil.  | Sil.  | Sil.                                           | Sil.    | Sil.    | Sil.     | Sil. | Sil.    |
| Input Modality | 2D + 3D     | 2D   | 2D   | 2D       | 2D    | 2D    | 2D + 2D                                        | 2D + 3D | 2D + 3D | 2D       | 2D   | 2D + 2D |
| 30-40          | 78.2        | 78.2 | 95.6 | 100.0    | 100.0 | 100.0 | 100.0                                          | 73.9    | 78.2    | 73.9     | 73.9 | 73.9    |
| 40-50          | 92.5        | 88.8 | 92.5 | 100.0    | 100.0 | 100.0 | 100.0                                          | 66.6    | 74.0    | 66.6     | 62.9 | 85.1    |
| 50-60          | 71.7        | 80.4 | 86.9 | 86.9     | 86.9  | 95.6  | 95.6                                           | 30.4    | 32.6    | 43.4     | 52.1 | 52.1    |
| 60-70          | 74.1        | 79.1 | 87.5 | 89.1     | 89.1  | 91.6  | 91.6                                           | 27.5    | 24.1    | 31.6     | 34.9 | 36.6    |
| 70-80          | 62.2        | 69.4 | 87.0 | 87.0     | 87.0  | 92.4  | 92.4                                           | 16.5    | 19.7    | 24.8     | 23.3 | 27.6    |
| 80-90          | 68.3        | 73.2 | 86.2 | 88.5     | 88.5  | 91.2  | 91.2                                           | 18.6    | 19.3    | 25.6     | 24.9 | 30.7    |
| 90-100         | 63.6        | 69.0 | 86.0 | 88.3     | 88.3  | 93.1  | 93.1                                           | 15.3    | 18.0    | 24.3     | 26.5 | 32.6    |
| 100-110        | 66.7        | 71.9 | 86.0 | 86.3     | 86.3  | 91.5  | 91.5                                           | 21.6    | 20.5    | 27.3     | 29.0 | 35.2    |
| 110-120        | 63.5        | 70.9 | 83.3 | 88.2     | 88.2  | 93.8  | 93.8                                           | 23.4    | 23.4    | 27.7     | 32.0 | 38.8    |
| 120-130        | 71.2        | 80.8 | 91.7 | 89.0     | 89.0  | 98.6  | 98.6                                           | 31.5    | 34.2    | 49.3     | 45.2 | 53.4    |
| 130-140        | 75.8        | 84.4 | 93.1 | 96.5     | 96.5  | 98.2  | 98.2                                           | 51.7    | 50.0    | 56.8     | 62.0 | 60.3    |
| 140-150        | 78.3        | 81.0 | 86.4 | 94.5     | 94.5  | 94.5  | 94.5                                           | 81.0    | 78.3    | 86.4     | 78.3 | 64.8    |
| 210-220        | 68.7        | 68.7 | 90.6 | 93.7     | 93.7  | 90.6  | 90.6                                           | 75.0    | 78.1    | 81.2     | 68.7 | 68.7    |
| 220-230        | 80.9        | 85.7 | 90.4 | 95.2     | 95.2  | 100.0 | 100.0                                          | 52.3    | 57.1    | 66.6     | 76.1 | 76.1    |
| 230-240        | 83.9        | 80.0 | 92.0 | 95.9     | 95.9  | 95.9  | 95.9                                           | 68.0    | 63.9    | 72.0     | 68.0 | 68.0    |
| 240-250        | 80.2        | 85.5 | 93.4 | 93.4     | 93.4  | 94.7  | 94.7                                           | 35.5    | 39.4    | 46.0     | 47.3 | 64.4    |
| 250-260        | 66.1        | 67.7 | 81.9 | 82.4     | 82.4  | 88.1  | 88.1                                           | 33.3    | 33.8    | 42.3     | 44.0 | 43.5    |
| 260-270        | 73.3        | 74.8 | 87.5 | 91.7     | 91.7  | 92.3  | 92.3                                           | 31.0    | 30.1    | 42.3     | 41.7 | 43.7    |
| 270-280        | 70.7        | 73.5 | 87.2 | 87.8     | 87.8  | 92.0  | 92.0                                           | 25.7    | 28.9    | 39.4     | 38.5 | 42.6    |
| 280-290        | 79.5        | 79.8 | 89.3 | 90.1     | 90.1  | 90.9  | 90.9                                           | 37.7    | 40.1    | 45.0     | 45.9 | 54.0    |
| 290-300        | 82.4        | 84.2 | 91.2 | 89.4     | 89.4  | 91.2  | 91.2                                           | 45.6    | 47.3    | 61.4     | 56.1 | 63.1    |
| 300-310        | 81.8        | 83.3 | 93.1 | 89.9     | 89.9  | 97.7  | 97.7                                           | 56.8    | 56.8    | 61.3     | 61.3 | 68.1    |
| 310-320        | 66.6        | 81.8 | 85.5 | 95.4     | 95.4  | 93.3  | 93.3                                           | 50.0    | 46.6    | 53.3     | 54.4 | 51.1    |
| 320-330        | 72.5        | 75.8 | 90.3 | 91.9     | 91.9  | 91.3  | 91.3                                           | 67.7    | 64.5    | 75.8     | 74.1 | 62.9    |

Table 4.4. Statistical interpretation of the Experiment 2 results. Data is taken from Table 4.3

| Input Type     | Pre-Trained |             |             |             |                | Trained with Gait3D dataset using our protocol |             |             |             |                |
|----------------|-------------|-------------|-------------|-------------|----------------|------------------------------------------------|-------------|-------------|-------------|----------------|
|                | SMPLGait-3D | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ | SMPLGait-3D                                    | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ |
|                | Sil. + SMPL | Sil.        | Sil.        | Sil.        | Sil. + SM      | Sil. + SMPL                                    | Sil.        | Sil.        | Sil.        | Sil. + SM      |
| Input Modality | 2D + 3D     | 2D          | 2D          | 2D          | 2D + 2D        | 2D + 3D                                        | 2D          | 2D          | 2D          | 2D + 2D        |
| Average        | 70.9        | 75.4        | 87.7        | 89.7        | 93.1           | 33.1                                           | 34.0        | 41.3        | 41.6        | 45.7           |

In Experiment 3, we aim to assess the effect of rotating test samples into training viewpoints. First, we rotated the test split of the restructured Gait3D dataset into training angles using our protocol, as described in Section 3.5, and named it the rotated test set. Then, we evaluated the gait recognition frameworks sourced from Experiment 2 using the rotated Gait3D dataset. Finally, we compared the results with the models trained using our protocol in Experiment 2. We have illustrated sample gait sequences from the rotated test set in Figure 4.5. Moreover, the results of Experiment 3 are shown in Table 4.5 and statistical interpretation of the results in Table 4.6. A visual comparison with Experiment 2 results is demonstrated in Figure 4.4. Notably, the straight lines in Figure 4.4 correspond to the models that are trained with the proposed protocol in Experiment 2.

In Experiment 4, we aim to assess the effect of generating silhouettes and 2D Skeletons from SMPLs and 3D Skeletons, respectively. First, we generated gait representations from the restructured Gait3D dataset from the SMPLs as described in Section 3.5. Then, we mapped 3D poses of the restructured Gait3D dataset to 2D poses and named it the mapped Gait3D dataset. We evaluated the gait recognition frameworks that were trained from scratch in Experiment 1 using the mapped Gait3D dataset. We reported the evaluation results of gait recognition frameworks sourced from Experiment 1 in the mapped Gait3D dataset in Table 4.7. Finally, we compare the results of Experiment 4 with those of Experiment 2 to demonstrate the effect of using silhouettes generated from SMPLs and 2D poses generated from 3D poses, and present the results in Table 4.8. Note that we evaluated pre-trained gait recognition methods sourced from Experiment 1 in Experiment 2 using our restructured Gait3D dataset. Moreover, a visual comparison of the experiments on the mapped Gait3D dataset and

restructured Gait3D dataset can be found in Figure 4.6. Sample silhouettes that are generated from SMPLs are demonstrated in Figure 4.5.

In Experiment 5, we propose a novel gait representation called RGB Depth Images (DI) as described in Section 3.5. First, we generated RGB Depth Images using SMPLs in the Gait3D dataset and named it the RGB Depth Gait3D dataset. Then, we train top-performing gait recognition models using the RGB Depth Gait3D dataset and present the evaluation results of gait recognition frameworks in Table 4.9. In Table 4.9, we present the input types of each method in the third row, including RGB Depth Images (RGB DI), SMPL parameters, and Skeleton Maps (SM). For example, SMPLGait-3D uses RGB Depth Images and SMPL parameters as input representation. Then, we compare the results of Experiment 5 with those of Experiment 4 to demonstrate the effect using RGB Depth Images over Silhouettes generated SMPLs Table 4.10 and also visualized in Figure 4.7. Additionally, we compare the results of Experiment 5 with those of Experiment 1 to investigate the benefits of using RGB Depth Images against original silhouettes in Table 4.11 and visualize them in Figure 4.8. Sample depth images that are generated from SMPLs are demonstrated in Figure 4.5.

In Experiment 6, we aim to assess the effect of rotation and generating Depth Images (DI) on gait recognition accuracy. First, we generate three-channel (RGB) and single-channel depth images of the Gait3D dataset and train the SkeletonGait++ models on both modalities using our protocol described in Section 3.5 and present the results in Table 4.12. Then, we rotate the test set to the training viewpoint and evaluate the newly trained SkeletonGait++ models on both datasets and present the results in Table 4.12. Finally, we present the results for each angular bin in Figure 4.9 and compare the performance of using RGB and single-channel Depth Images against original silhouettes in Table 4.13. The results presented in the second and third column of the Table 4.12, corresponds to 1-channel and RGB Depth Images, respectively. Similarly, in fourth and fifth columns we present results obtained using 3-channel (RGB) Depth images. Moreover, we illustrate sample single-channel and RGB depth images in Figure 4.5.

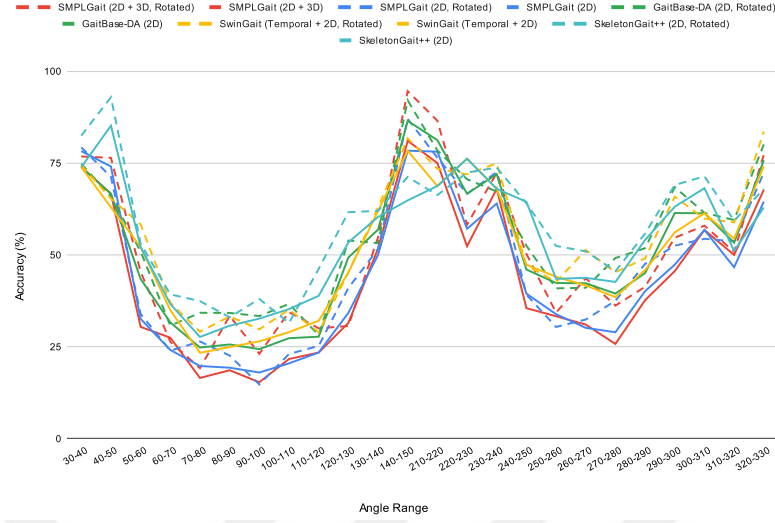


Figure 4.4. The Rank-1 accuracies of Gait Recognition methods in rotated and restructured Gait3D dataset. The data is taken from Table 4.3 and Table 4.5.

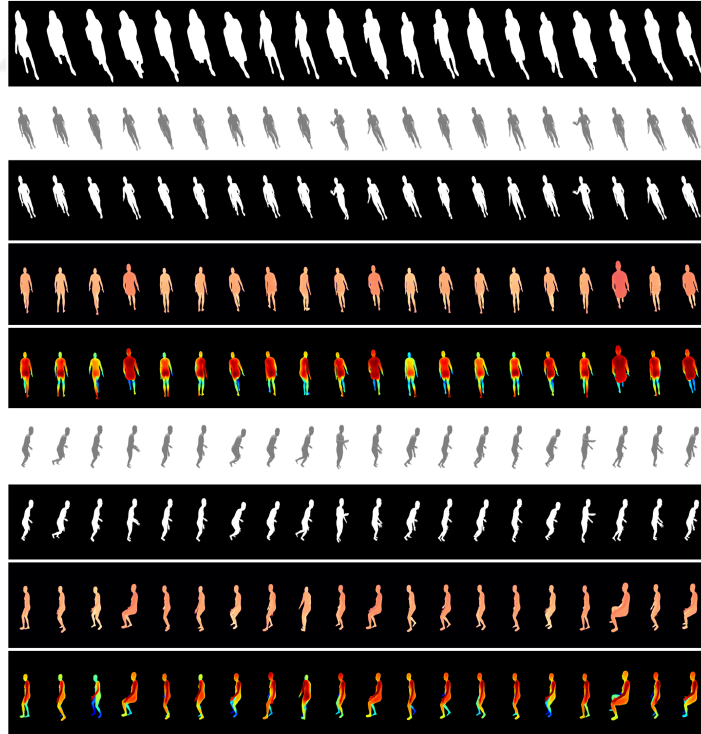


Figure 4.5. Different modalities of Gait3D dataset. Each row contains original Silhouettes, original SMPLs, silhouettes generated from SMPLs, depth images, RGB depth images, rotated SMPLs, silhouettes generated from rotated SMPLs, rotated depth images, and finally, rotated RGB depth images, respectively.

Table 4.5. Evaluation of top-performing gait recognition frameworks on the rotated Gait3D test set. Gait recognition frameworks are sourced from Experiment 2. Angular bins are constructed according to the angular separation protocol. Values correspond to rank-1 accuracies.

|                | SMPLGait-3D | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ |
|----------------|-------------|-------------|-------------|-------------|----------------|
| Input Type     | Sil. + SMPL | Sil.        | Sil.        | Sil.        | Sil + SM       |
| Input Modality | 2D + 3D     | 2D          | 2D          | 2D          | 2D + 2D        |
| 30-40          | 76.8        | 79.2        | 74.8        | 74.4        | 82.4           |
| 40-50          | 76.4        | 71.2        | 65.8        | 65.3        | 92.9           |
| 50-60          | 45.4        | 33.9        | 50.9        | 58.3        | 53.0           |
| 60-70          | 26.3        | 23.9        | 30.7        | 37.0        | 39.2           |
| 70-80          | 19.1        | 26.4        | 34.2        | 29.1        | 37.4           |
| 80-90          | 33.4        | 22.6        | 34.2        | 33.3        | 33.0           |
| 90-100         | 23.1        | 14.7        | 33.3        | 29.7        | 38.0           |
| 100-110        | 34.4        | 23.0        | 36.6        | 35.1        | 31.4           |
| 110-120        | 30.0        | 25.2        | 28.1        | 28.9        | 46.2           |
| 120-130        | 30.6        | 41.0        | 53.9        | 45.2        | 61.6           |
| 130-140        | 54.8        | 51.6        | 53.2        | 62.9        | 62.0           |
| 140-150        | 94.5        | 87.0        | 92.0        | 81.7        | 71.2           |
| 210-220        | 86.5        | 76.4        | 78.4        | 73.4        | 66.3           |
| 220-230        | 58.2        | 66.6        | 70.6        | 71.9        | 72.4           |
| 230-240        | 72.3        | 72.5        | 67.2        | 74.9        | 73.7           |
| 240-250        | 50.3        | 39.2        | 52.5        | 47.1        | 64.0           |
| 250-260        | 34.3        | 30.4        | 40.9        | 42.6        | 52.4           |
| 260-270        | 43.6        | 32.4        | 41.0        | 51.3        | 51.0           |
| 270-280        | 36.2        | 37.5        | 49.1        | 45.3        | 45.6           |
| 280-290        | 41.3        | 47.5        | 51.8        | 49.0        | 55.8           |
| 290-300        | 54.6        | 52.5        | 68.4        | 65.8        | 69.0           |
| 300-310        | 57.9        | 54.4        | 61.4        | 59.8        | 71.3           |
| 310-320        | 50.8        | 53.7        | 59.5        | 58.8        | 59.3           |
| 320-330        | 77.2        | 72.3        | 80.0        | 83.5        | 67.9           |
| Average        | 41.0        | 36.7        | 46.2        | 45.7        | 50.1           |

Table 4.6. Comparison of Experiment 3 results against Experiment 2. Models are trained in Experiment 2 with the reconstructed Gait3D dataset. Data in LHS is taken from Table 4.4 and RHS is taken from Table 4.5

|                | Evaluated with Gait3D dataset in Exp. 2 |             |             |             |                | Evaluated with rotated Gait3D dataset in Exp. 3 |             |             |             |                |
|----------------|-----------------------------------------|-------------|-------------|-------------|----------------|-------------------------------------------------|-------------|-------------|-------------|----------------|
|                | SMPLGait-3D                             | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ | SMPLGait-3D                                     | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ |
| Input Type     | Sil. + SMPL                             | Sil.        | Sil.        | Sil.        | Sil. + SM      | Sil. + SMPL                                     | Sil.        | Sil.        | Sil.        | Sil. + SM      |
| Input Modality | 2D + 3D                                 | 2D          | 2D          | 2D          | 2D             | 2D + 3D                                         | 2D          | 2D          | 2D          | 2D             |
| Average        | 33.1                                    | 34.0        | 41.3        | 41.6        | 45.7           | 41.0                                            | 36.7        | 46.2        | 45.7        | 50.1           |

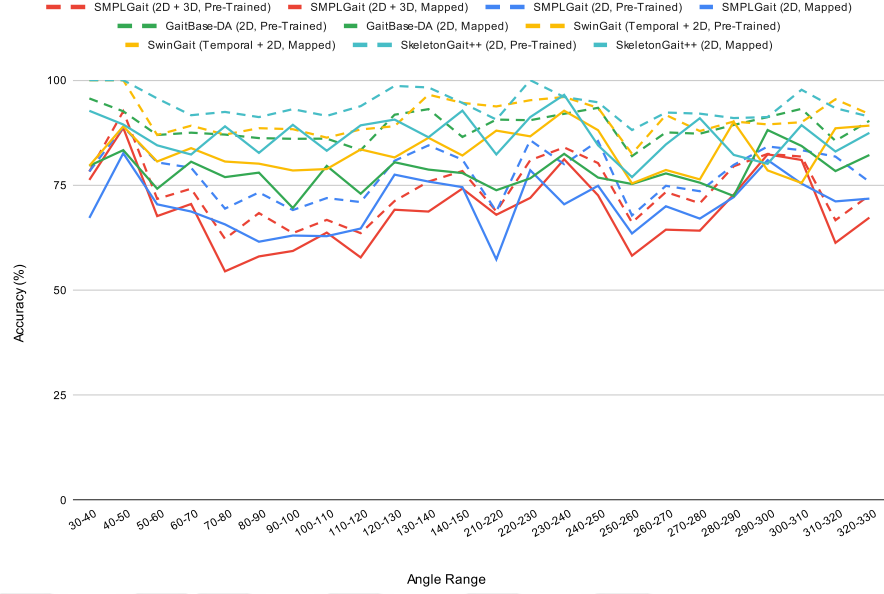


Figure 4.6. The Rank-1 accuracies of pre-trained Gait Recognition methods in the mapped and restructured Gait3D dataset. Models are trained in Experiment 1. The data is taken from Table 4.3 and Table 4.7.

#### 4.2.2. Interpretation of Gait3D Results

Upon examining the results shown in Figure 4.3, it was observed that the pre-trained models outperformed the retrained models. As shown in Table 4.3, pre-trained SMPLGait and SMPLGait-2D methods exhibited an average performance improvement of 30% and 33% over their retrained counterparts. This can be attributed to the fact that 75% of the test dataset overlaps with the training data used for pre-trained models.

Furthermore, the performance of pre-trained models was found to decline as the evaluation angles deviated from the training set’s angular range and to improve as the test angles approached the training viewpoints. For instance, the pre-trained SMPLGait and SMPLGait-2D methods achieved 78% accuracy within the 30-40 degree range, while their accuracy dropped to approximately 70% in the 80-90 and 270-280 degree ranges, which were the farthest from the training set angles.

Table 4.7. Evaluation of gait recognition frameworks on the mapped Gait3D test set. Models are sourced from Experiment 1. Values correspond to rank-1 accuracies.

|                | SMPLGait-3D | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ |
|----------------|-------------|-------------|-------------|-------------|----------------|
| Input Type     | Sil. + SMPL | Sil.        | Sil.        | Sil.        | Sil.           |
| Input Modality | 2D + 3D     | 2D          | 2D          | 2D          | 2D             |
| 30-40          | 76.2        | 67.2        | 79.6        | 79.6        | 92.7           |
| 40-50          | 88.6        | 82.6        | 83.3        | 89.0        | 89.4           |
| 50-60          | 67.6        | 70.4        | 74.1        | 80.6        | 84.4           |
| 60-70          | 70.5        | 68.7        | 80.5        | 83.8        | 82.3           |
| 70-80          | 54.4        | 65.6        | 76.9        | 80.6        | 89.0           |
| 80-90          | 58.0        | 61.5        | 77.9        | 80.1        | 82.6           |
| 90-100         | 59.3        | 63.0        | 69.5        | 78.5        | 89.3           |
| 100-110        | 63.7        | 62.8        | 79.5        | 78.8        | 83.2           |
| 110-120        | 57.8        | 64.6        | 72.9        | 83.5        | 89.2           |
| 120-130        | 69.1        | 77.5        | 80.4        | 81.6        | 90.6           |
| 130-140        | 68.7        | 75.8        | 78.7        | 86.2        | 86.4           |
| 140-150        | 74.1        | 74.5        | 77.8        | 82.0        | 92.7           |
| 210-220        | 67.9        | 57.3        | 73.8        | 87.9        | 82.3           |
| 220-230        | 71.9        | 78.5        | 76.6        | 86.6        | 91.2           |
| 230-240        | 81.1        | 70.4        | 82.4        | 92.7        | 96.5           |
| 240-250        | 72.5        | 74.8        | 76.8        | 88.0        | 84.4           |
| 250-260        | 58.2        | 63.5        | 75.2        | 75.4        | 76.9           |
| 260-270        | 64.4        | 69.9        | 77.8        | 78.6        | 84.7           |
| 270-280        | 64.1        | 67.0        | 75.6        | 76.3        | 90.9           |
| 280-290        | 72.7        | 72.1        | 72.4        | 90.0        | 82.2           |
| 290-300        | 82.3        | 80.9        | 88.1        | 78.5        | 80.0           |
| 300-310        | 81.0        | 75.3        | 84.3        | 75.5        | 89.2           |
| 310-320        | 61.2        | 71.1        | 78.3        | 88.5        | 83.0           |
| 320-330        | 67.2        | 71.8        | 82.1        | 89.1        | 87.4           |
| Average        | 65.2        | 67.9        | 77.1        | 81.5        | 86.1           |

Table 4.8. Comparison of Experiment 4 results against Experiment 2. Data in LHS is taken from Table 4.4 and RHS is taken from Table 4.7.

|                | Evaluated with Gait3D dataset |             |             |             |                | Evaluated with generated Gait3D dataset |             |             |             |                |
|----------------|-------------------------------|-------------|-------------|-------------|----------------|-----------------------------------------|-------------|-------------|-------------|----------------|
|                | SMPLGait-3D                   | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ | SMPLGait-3D                             | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ |
| Input Type     | Sil. + SMPL                   | Sil.        | Sil.        | Sil.        | Sil. + SM      | Sil. + SMPL                             | Sil.        | Sil.        | Sil.        | Sil. + SM      |
| Input Modality | 2D + 3D                       | 2D          | 2D          | 2D          | 2D + 2D        | 2D + 3D                                 | 2D          | 2D          | 2D          | 2D + 2D        |
| Average        | 70.9                          | 75.4        | 87.7        | 89.7        | 93.1           | 65.2                                    | 67.9        | 77.1        | 81.5        | 86.1           |

Table 4.9. Evaluation of top-performing gait recognition frameworks on the RGB Depth Gait3D test set. Values correspond to rank-1 accuracies.

|                | SMPLGait-3D   | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ |
|----------------|---------------|-------------|-------------|-------------|----------------|
| Input Type     | RGB DI + SMPL | RGB DI      | RGB DI      | RGB DI      | RGB DI + SM    |
| Input Modality | 2.5D + 3D     | 2.5D        | 2.5D        | 2.5D        | 2.5D + 2D      |
| 30-40          | 77.5          | 72.4        | 83.9        | 83.5        | 96.9           |
| 40-50          | 91.3          | 84.1        | 85.6        | 94.0        | 99.4           |
| 50-60          | 73.4          | 74.7        | 76.7        | 86.6        | 91.2           |
| 60-70          | 77.2          | 73.2        | 82.4        | 92.3        | 88.0           |
| 70-80          | 59.6          | 70.4        | 79.8        | 85.8        | 98.6           |
| 80-90          | 60.0          | 65.3        | 81.2        | 86.9        | 91.8           |
| 90-100         | 62.7          | 69.9        | 74.5        | 84.1        | 94.3           |
| 100-110        | 64.9          | 69.2        | 80.9        | 84.5        | 91.9           |
| 110-120        | 62.7          | 68.1        | 76.3        | 87.0        | 94.1           |
| 120-130        | 71.9          | 81.2        | 85.2        | 85.2        | 96.5           |
| 130-140        | 71.4          | 82.8        | 84.6        | 89.8        | 93.1           |
| 140-150        | 79.7          | 77.0        | 83.4        | 90.9        | 97.6           |
| 210-220        | 74.8          | 62.8        | 75.0        | 91.4        | 90.4           |
| 220-230        | 76.7          | 81.9        | 80.2        | 91.0        | 100.0          |
| 230-240        | 83.3          | 75.3        | 84.5        | 100.0       | 100.0          |
| 240-250        | 75.6          | 81.5        | 79.3        | 96.6        | 94.0           |
| 250-260        | 60.5          | 66.4        | 76.7        | 82.5        | 86.8           |
| 260-270        | 69.6          | 74.2        | 83.2        | 86.1        | 93.1           |
| 270-280        | 67.2          | 68.8        | 80.9        | 81.8        | 97.7           |
| 280-290        | 78.4          | 78.8        | 77.9        | 98.8        | 88.5           |
| 290-300        | 88.5          | 84.0        | 92.3        | 87.6        | 85.0           |
| 300-310        | 84.5          | 77.6        | 86.0        | 79.1        | 93.8           |
| 310-320        | 65.1          | 74.7        | 82.1        | 92.1        | 89.9           |
| 320-330        | 73.3          | 73.1        | 85.1        | 98.0        | 95.4           |
| Average        | 69.0          | 72.4        | 80.6        | 87.6        | 93.4           |

Table 4.10. Comparison of Experiment 5 results against Experiment 4. Data in LHS is taken from Table 4.7 and RHS is taken from Table 4.9

|                | Evaluated with generated Gait3D dataset in Exp. 4 |             |             |             |                | Evaluated with the RGB Depth Gait3D dataset in Exp.5 |             |             |             |                |
|----------------|---------------------------------------------------|-------------|-------------|-------------|----------------|------------------------------------------------------|-------------|-------------|-------------|----------------|
|                | SMPLGait-3D                                       | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ | SMPLGait-3D                                          | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ |
| Input Type     | Sil. + SMPL                                       | Sil.        | Sil.        | Sil.        | Sil. + SM      | RGB DI + SMPL                                        | RGB DI      | RGB DI      | RGB DI      | RGB DI + SM    |
| Input Modality | 2D + 3D                                           | 2D          | 2D          | 2D          | 2D + 2D        | 2.5D + 3D                                            | 2.5D        | 2.5D        | 2.5D        | 2.5D + 2D      |
| Average        | 65.2                                              | 67.9        | 77.1        | 81.5        | 86.1           | 69.0                                                 | 72.4        | 80.6        | 87.6        | 93.4           |

Table 4.11. Comparison of Experiment 5 results against Experiment 1. Models on the LHS are sourced from Experiment 1, which are trained with Silhouettes, whereas models on the RHS are trained in Experiment 6 with RGB Depth images. Data in LHS is taken from Table 4.4 and RHS is taken from Table 4.9.

| Input Type     | Evaluated with Gait3D dataset using our protocol |             |             |             |                | Evaluated with RGB Depth Images using our protocol |             |             |             |                |
|----------------|--------------------------------------------------|-------------|-------------|-------------|----------------|----------------------------------------------------|-------------|-------------|-------------|----------------|
|                | SMPLGait-3D                                      | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ | SMPLGait-3D                                        | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ |
|                | Sil. + SMPL                                      | Sil.        | Sil.        | Sil.        | Sil. + SM      | RGB DI + SMPL                                      | RGB DI      | RGB DI      | RGB DI      | RGB DI + SM    |
| Input Modality | 2D + 3D                                          | 2D          | 2D          | 2D          | 2D + 2D        | 2.5D + 3D                                          | 2.5D        | 2.5D        | 2.5D        | 2.5D + 2D      |
| Average        | 70.9                                             | 75.4        | 87.7        | 89.7        | 93.1           | 69.0                                               | 72.4        | 80.6        | 87.6        | 93.4           |

Table 4.12. Evaluation results of SkeletonGait++ framework using 1-Channel Depth and 3-Channel (RGB) Depth images of Gait3D dataset. Angular bins are constructed, and models are trained using our protocol with 1-Channel and 3-Channel Depth Images. Values correspond to rank-1 accuracies.

| Input Type | Trained with Depth Images |                 | Trained with RGB Depth Images |                     |
|------------|---------------------------|-----------------|-------------------------------|---------------------|
|            | DI + SM                   | Rotated DI + SM | RGB DI + SM                   | Rotated RGB DI + SM |
| 30-40      | 73.7                      | 77.1            | 79.6                          | 84.3                |
| 40-50      | 86.9                      | 91.9            | 85.4                          | 88.6                |
| 50-60      | 51.2                      | 51.5            | 52.6                          | 56.4                |
| 60-70      | 38.1                      | 41.2            | 42.0                          | 42.8                |
| 70-80      | 27.3                      | 29.4            | 33.0                          | 36.0                |
| 80-90      | 29.6                      | 32.0            | 32.8                          | 37.4                |
| 90-100     | 32.3                      | 34.7            | 33.7                          | 35.7                |
| 100-110    | 33.8                      | 36.5            | 41.0                          | 46.3                |
| 110-120    | 39.1                      | 40.3            | 42.7                          | 47.4                |
| 120-130    | 54.6                      | 54.8            | 58.3                          | 62.9                |
| 130-140    | 59.2                      | 62.6            | 60.7                          | 65.2                |
| 140-150    | 64.2                      | 66.7            | 67.4                          | 71.2                |
| 210-220    | 69.7                      | 73.1            | 68.7                          | 73.2                |
| 220-230    | 75.5                      | 78.7            | 78.5                          | 79.8                |
| 230-240    | 69.2                      | 73.4            | 71.6                          | 75.6                |
| 240-250    | 65.0                      | 66.8            | 66.6                          | 68.7                |
| 250-260    | 44.8                      | 45.0            | 47.2                          | 47.3                |
| 260-270    | 42.8                      | 45.6            | 45.3                          | 50.4                |
| 270-280    | 42.5                      | 45.6            | 47.8                          | 50.7                |
| 280-290    | 53.9                      | 57.7            | 58.2                          | 59.1                |
| 290-300    | 64.9                      | 68.5            | 67.0                          | 72.4                |
| 300-310    | 67.0                      | 67.2            | 72.6                          | 74.1                |
| 310-320    | 51.5                      | 52.1            | 52.5                          | 55.8                |
| 320-330    | 61.4                      | 65.6            | 68.5                          | 68.7                |
| Average    | 45.5                      | 47.9            | 49.0                          | 52.3                |

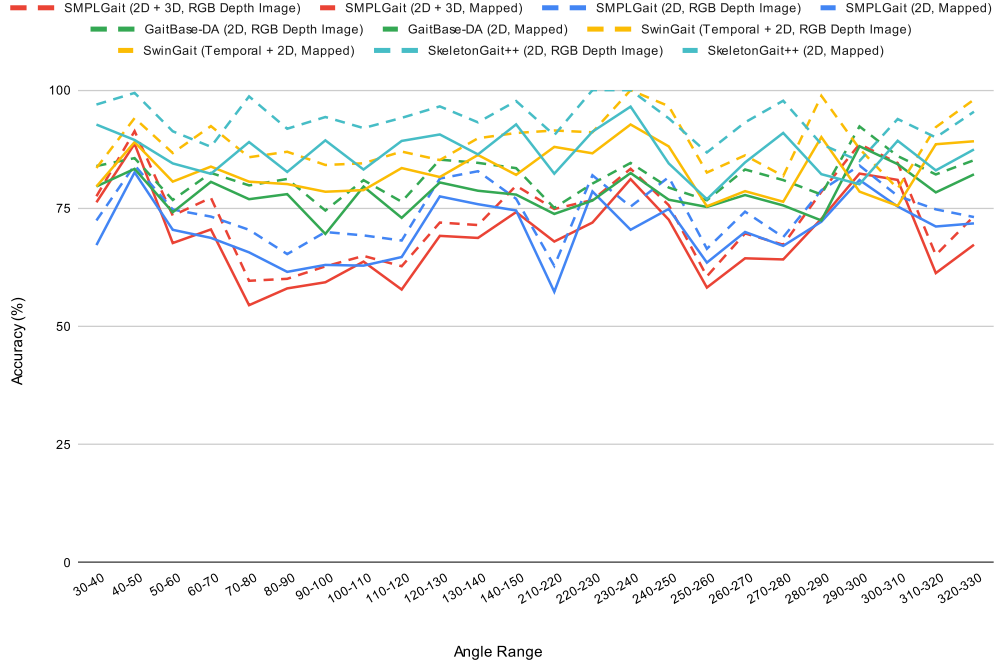


Figure 4.7. The Rank-1 accuracies of top-performing gait recognition methods in the generated and RGB Depth Gait3D dataset. The data is taken from Table 4.7 and Table 4.9.

Interestingly, among the pre-trained models, the one trained with only 2D information outperformed the model trained with 3D data. For example, comparing the test results of SMPLGait (3D) and SMPLGait-2D methods, the largest performance gap of 15% was observed in the 310-320 degree range. Across all test results, the SMPLGait-2D method demonstrated an average 4% higher accuracy than the SMPLGait method using 3D data. The SkeletonGait model stands out among all the evaluated methods, achieving the highest accuracy across the angular ranges. When trained using the proposed protocol, SkeletonGait demonstrates remarkable viewpoint invariance, with superior performance compared to models trained with our protocol.

Evaluating the performance of models trained using the proposed angular separation protocol revealed that all models achieved higher accuracy on test angles closer to the angular ranges used in training, with performance declining as test angles deviated further. For instance, retrained models achieved an average accuracy of 80% in the

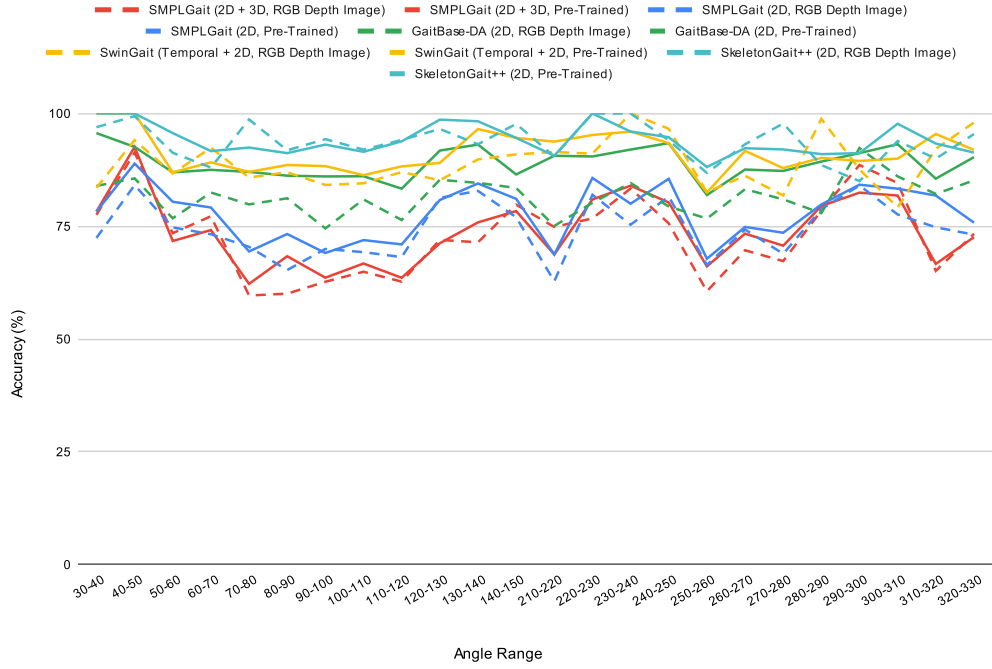


Figure 4.8. The Rank-1 accuracies of top-performing gait recognition methods in the original and Depth Gait3D dataset. The data is taken from Table 4.3 and Table 4.11.

Table 4.13. Comparison of SkeletonGait++ rank-1 accuracy performances. The data in LHS is taken from Table 4.6 and RHS is taken from Table 4.12.

| Input Type | Trained with Silhouettes in Exp. 2 |                   | Trained with Depth Images in Exp. 6 |                 | Trained with RGB Depth Images in Exp. 5 |                     |
|------------|------------------------------------|-------------------|-------------------------------------|-----------------|-----------------------------------------|---------------------|
|            | Sil. + SM                          | Rotated Sil. + SM | DI + SM                             | Rotated DI + SM | RGB DI + SM                             | Rotated RGB DI + SM |
| Average    | 45.7                               | 50.1              | 45.5                                | 47.9            | 49.0                                    | 52.3                |

30-40 degree range, while this dropped to 36% in the 80-90 degree range. Similarly, for test angles close to the training set, such as 140-150, 210-220, and 320-330 degrees, the average accuracy was 84%, 80%, and 76%, respectively, while for the farthest ranges, like 260-270 and 270-280 degrees, performance fell to 47% and 45%.

Based on the results presented in Figure 4.4, it can be stated that rotating the gait representations improved the performance of the models remarkably. Specifically, among all angular bins for all models, the maximum observed improvement is 21%, while the minimum is -4%. For example, the performance of GaitBase in the 70-80

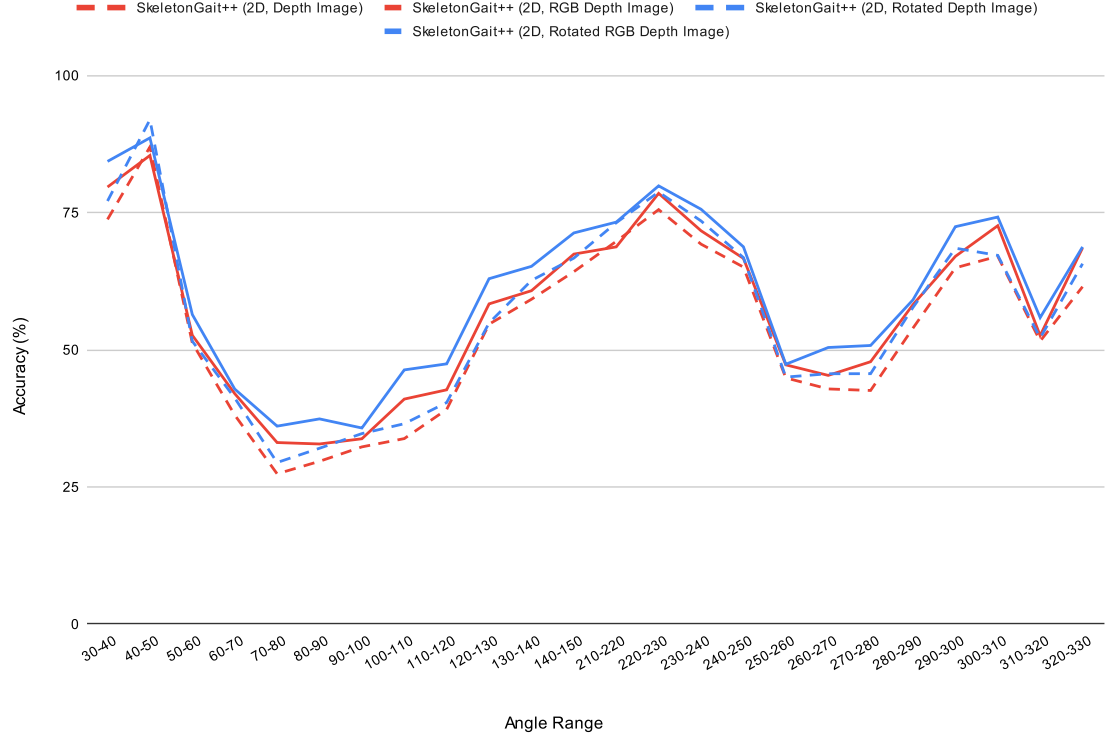


Figure 4.9. The Rank-1 accuracies of top-performing gait recognition methods in the Depth Gait3D dataset. The data is taken from Table 4.12.

degrees range improved by 10%. Similarly, the performance of SkeletonGait improved by 9% in the 250-260 degree range. Furthermore, the SMPLGait framework, which utilizes 3D information, achieved the highest improvement in weighted average R-1 accuracy at 7%, whereas SMPLGait-2D improved by only 3%. Additionally, the SkeletonGait framework, which utilizes both rotated silhouette and 2D pose, achieved a 5% increase in the weighted average R-1 accuracy.

Considering the results shown in Figure 4.6, it can be stated that using silhouettes that are mapped from SMPLs resulted in a decay in performance in general. For example, the performance of GaitBase and SwinGait in the 30-40 degrees range is decreased by 16% and 20%, respectively. Similarly, the weighted average R-1 accuracy of SMPLGait-2D dropped by 7%, whereas SMPLGait, which combines 3D with 2D information, experienced a relatively smaller reduction of only 5%. Among all

methods, GaitBase exhibited the largest decline in weighted average R-1 accuracy, which is around 11%. Additionally, mapping 3D poses into 2D resulted in about a 7% performance decay in terms of weighted average R-1 accuracy for the SkeletonGait framework.

Results in Table 4.9 indicate that SkeletonGait++ is the best performing model with a 93.4% accuracy while using RGB Depth Images with Skeleton Maps. Moreover, using RGB depth images outperforms the performance of the SkeletonGait++ method, which is trained with our protocol and evaluated using silhouettes generated from SMPLs. To be specific, SkeletonGait++ with RGB Depth Images outperforms its counterpart by 7.3% with a 93.4% average accuracy, as shown in Table 4.10. Similarly, SkeletonGait++ outperforms its pre-trained counterpart by 0.3% with a 93% average accuracy, as shown in Table 4.11. Additionally, the SMPLGait-3D method achieves 69% average accuracy, which is 3.8% over its counterpart that is trained with our protocol and evaluated with Silhouettes that are generated from SMPLs as shown in Table 4.10. Notably, the average accuracy of SwinGait-3D methods decreases by 7% when using RGB Depth Images compared to its pre-trained counterpart that is evaluated with our protocol. In general, the accuracy of gait recognition methods decreases by 2.8% on average when using RGB Depth Images compared to the original Gait3D dataset, as shown in Table 4.11. Similarly, the accuracy of gait recognition methods increases by 5% on average when using RGB Depth Images compared to Silhouettes that are generated from SMPLs.

Considering the results in Table 4.12, SkeletonGait++ that uses 1-channel Depth Images performed 45% average accuracy, whereas SkeletonGait++ that uses 3-channel Depth Images performed 49% accuracy. In other words, using RGB Depth Images increases the average accuracy by 3.5%. Similarly, rotating test sets of 1-channel Depth Images and RGB Depth Images yield 48% and 52.3% average accuracies, respectively. The second and third columns indicate that rotating the Depth Images in the test set increases the average accuracy by around 2.5%. The fourth and fifth columns indicate that rotating the RGB Depth Images in the test set increases the average accuracy

by around 3%. The third and fifth columns indicate that using rotated 3-channel Depth Images outperforms their 1-channel counterparts by 4% on average accuracy. The findings in Table 4.13 indicate that using Depth Images over Silhouettes does not yield an increase in average accuracy. Moreover, the third and fifth columns show that rotating Silhouettes outperforms rotating 1-channel Depth Images by 2%. Conversely, when RGB Depth Images are utilized over Silhouettes, the performance is observed to be increased by 3%. The second, fourth, and sixth columns indicate that using RGB Depth Images as input modality results in the best average accuracy with a 49%. Similarly, the third, fifth, and seventh columns indicate that using rotated RGB DI outperforms its counterparts with a 52.3% average accuracy.

The training for Gait3D was conducted on a computer equipped with an A100 GPU and 83 GB of DRAM. The training durations were 7 hours for the SMPLGait-2D model, 10 hours for SMPLGait, 9 hours for GaitBase-DA, 19 hours for SwinGait, and 28 hours for SkeletonGait.

#### 4.2.3. GREW Experiments

In Experiment 7, we aim to validate the reproducibility of the reported results of the top-performing gait recognition frameworks, which are SMPLGait, GaitBase, SwinGait, and SkeletonGait. We trained these gait recognition frameworks from scratch and reported the evaluation results in the GREW dataset in Table 4.14. Notably, the SMPLGait-2D represents the baseline model of SMPLGait, which is trained only with 2D data.

Table 4.14. Reproducing gait recognition frameworks on the GREW dataset.

| Framework    | Input Data               | Reported R-1 Accuracy | Our R-1 Accuracy |
|--------------|--------------------------|-----------------------|------------------|
| SMPLGait-2D  | Silhouette               | -                     | 52.5             |
| GaitBase     | Silhouette               | 60.1                  | 60.1             |
| SwinGait-3D  | Silhouette               | 79.3                  | 79.3             |
| SkeletonGait | Skeleton Map+ Silhouette | 85.8                  | 85.8             |

In Experiment 8, we aim to assess the robustness to viewpoint changes of the gait recognition frameworks reproduced in Experiment 5. First, we restructured the GREW dataset into  $10^\circ$  angular bins and separated it into training and testing subsets using the protocol described in Section 3.5, which we named the restructured GREW dataset. Then, we trained the reproduced gait recognition frameworks in Experiment 7 from scratch with the restructured GREW dataset. We reported the evaluation results in the restructured GREW dataset in Table 4.15 and statistical interpretation of the results in Table 4.16, which contains the evaluation of both models trained in Experiment 5 and Experiment 6 using the restructured GREW test set. Notably, we have calculated the weighted average using the corresponding # of gait sequences in each bin and denoted it as average, which is provided in Table 3.6. A visual interpretation of the results on the restructured GREW dataset can be found in Figure 4.10. Similarly, dashed lines in Figure 4.10 correspond to the evaluation results of the models that are reproduced in Experiment 5.

In Experiment 9, we aim to assess the effect of rotating test samples into training angles. Since the 3D pose information was the only 3D gait representation in the GREW dataset, we rotated the 3D poses of the restructured GREW dataset. Then, we downsampled the rotated 3D poses into 2D poses, as described in Section 3.5. We demonstrated sample 2D poses that are generated by rotating and downscaling 3D poses in Figure 4.11. Then, we evaluated the baseline variant of the SkeletonGait framework sourced from Experiment 6 using the rotated test set. Finally, we compared these results with those obtained from the model trained using our proposed protocol in Experiment 8, and present the results in Table 4.18. The results are shown in Table 4.17. The comparison of evaluation on rotated and non-rotated GREW dataset on each angular bin is illustrated in Figure 4.12. Notably, straight lines in Figure 4.12 correspond to the evaluation results on the restructured GREW dataset of the models that are trained with our protocol in Experiment 8.

In Experiment 10, we aim to assess the effect of generating 2D poses from the 3D poses. First, we generate the 2D poses of the restructured GREW dataset from the

3D Skeletons as described in Section 3.5, and name it the mapped GREW dataset. Then, we evaluated the baseline version of the SkeletonGait framework, which was trained with our protocol in Experiment 8 using the mapped GREW dataset. We reported the evaluation results in the mapped GREW dataset in Table 4.19. Lastly, we compared the results with those of Experiment 6 to understand the effect of using 2D skeletons generated from 3D skeletons, and presented the results in Table 4.20. A visual comparison of the results of experiments on the mapped GREW dataset and restructured GREW dataset can be found in Figure 4.13, where the data is taken from Table 4.19 and Table 4.15. Notably, straight lines in Figure 4.12 correspond to the results of the SkeletonGait trained on the restructured GREW dataset that was trained with the proposed protocol in Experiment 8. Sample 2D poses that are generated from 3D poses are demonstrated in Figure 4.11.

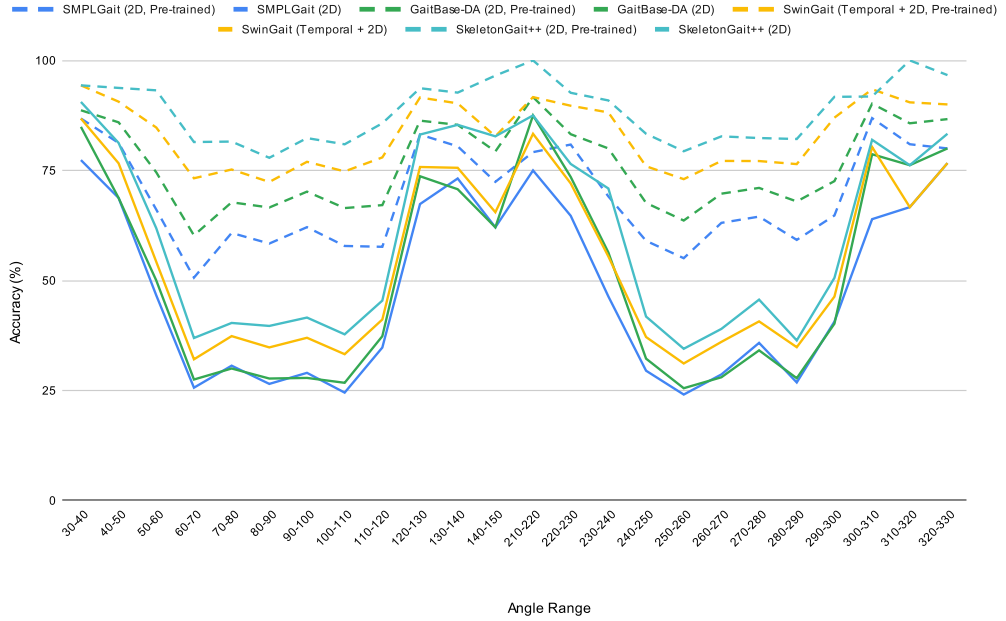


Figure 4.10. The rank-1 accuracies of Gait Recognition methods in the GREW dataset based on angles. The data is taken from Table 4.15.

Table 4.15. R-1 accuracies of gait recognition frameworks on the restructured GREW dataset. Angular bins are defined according to the angular separation protocol.

| Input Type     | Pre-Trained |       |             |       |             | Trained in GREW dataset using our protocol |             |             |             |              |                |         |
|----------------|-------------|-------|-------------|-------|-------------|--------------------------------------------|-------------|-------------|-------------|--------------|----------------|---------|
|                | SMPLGait-2D |       | GaitBase-DA |       | SwinGait-3D | SkeletonGait++                             | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait | SkeletonGait++ |         |
|                | Sil.        | 2D    | Sil.        | 2D    | Sil.        | Sil. + SM                                  | Sil.        | 2D          | Sil.        | 2D           | Sil.           | 2D + 2D |
| Input Modality |             |       |             |       |             |                                            |             |             |             |              |                |         |
| 30-40          | 86.7        | 88.6  | 94.3        | 94.3  | 77.3        | 94.3                                       | 84.9        | 86.7        | 73.5        | 73.5         | 90.5           | 90.5    |
| 40-50          | 81.2        | 85.9  | 90.6        | 90.6  | 68.7        | 93.7                                       | 68.7        | 76.6        | 70.3        | 70.3         | 81.2           | 81.2    |
| 50-60          | 66.1        | 74.5  | 84.7        | 84.7  | 46.6        | 93.2                                       | 50.0        | 54.2        | 50.0        | 50.0         | 61.8           | 61.8    |
| 60-70          | 50.6        | 60.2  | 73.2        | 73.2  | 25.6        | 81.4                                       | 27.5        | 32.0        | 20.8        | 20.8         | 36.9           | 36.9    |
| 70-80          | 60.         | 67.7  | 75.2        | 75.2  | 30.6        | 81.5                                       | 30.0        | 37.3        | 22.7        | 22.7         | 40.3           | 40.3    |
| 80-90          | 58.3        | 66.5  | 72.3        | 72.3  | 26.5        | 77.8                                       | 27.7        | 34.7        | 19.5        | 19.5         | 39.6           | 39.6    |
| 90-100         | 62.1        | 70.1  | 76.9        | 76.9  | 29.0        | 82.3                                       | 27.8        | 36.9        | 22.2        | 22.2         | 41.5           | 41.5    |
| 100-110        | 57.8        | 66.4  | 74.7        | 74.7  | 24.5        | 80.9                                       | 26.7        | 33.2        | 20.0        | 20.0         | 37.7           | 37.7    |
| 110-120        | 57.6        | 67.1  | 77.9        | 77.9  | 34.7        | 85.7                                       | 37.3        | 41.1        | 30.2        | 30.2         | 45.4           | 45.4    |
| 120-130        | 83.1        | 86.3  | 91.5        | 91.5  | 67.3        | 93.6                                       | 73.6        | 75.7        | 64.2        | 64.2         | 83.1           | 83.1    |
| 130-140        | 80.4        | 85.3  | 90.2        | 90.2  | 73.1        | 92.6                                       | 70.7        | 75.6        | 68.2        | 68.2         | 85.3           | 85.3    |
| 140-150        | 72.4        | 79.3  | 82.7        | 82.7  | 62.0        | 96.5                                       | 62.0        | 65.5        | 72.4        | 72.4         | 82.7           | 82.7    |
| 210-220        | 79.16       | 91.66 | 91.66       | 91.66 | 75.0        | 100.0                                      | 87.5        | 83.3        | 62.5        | 62.5         | 87.5           | 87.5    |
| 220-230        | 80.8        | 83.2  | 89.7        | 89.7  | 64.7        | 92.6                                       | 73.5        | 72.0        | 72.0        | 72.0         | 76.4           | 76.4    |
| 230-240        | 69.0        | 80.0  | 88.1        | 88.1  | 46.3        | 90.9                                       | 56.3        | 55.4        | 45.4        | 45.4         | 70.9           | 70.9    |
| 240-250        | 59.0        | 67.6  | 75.9        | 75.9  | 29.5        | 83.3                                       | 32.2        | 37.1        | 26.7        | 26.7         | 41.8           | 41.8    |
| 250-260        | 55.0        | 63.6  | 73.0        | 73.0  | 24.0        | 79.3                                       | 25.5        | 31.1        | 18.6        | 18.6         | 34.4           | 34.4    |
| 260-270        | 63.0        | 69.7  | 77.1        | 77.1  | 28.6        | 82.7                                       | 28.0        | 36.0        | 18.9        | 18.9         | 39.0           | 39.0    |
| 270-280        | 64.5        | 71.0  | 77.1        | 77.1  | 35.8        | 82.3                                       | 34.1        | 40.7        | 26.2        | 26.2         | 45.6           | 45.6    |
| 280-290        | 59.2        | 67.9  | 76.4        | 76.4  | 26.8        | 82.1                                       | 27.7        | 34.8        | 17.3        | 17.3         | 36.3           | 36.3    |
| 290-300        | 64.7        | 72.5  | 86.9        | 86.9  | 40.6        | 91.7                                       | 40.1        | 46.3        | 38.0        | 38.0         | 50.5           | 50.5    |
| 300-310        | 86.8        | 90.1  | 93.4        | 93.4  | 63.9        | 91.8                                       | 78.6        | 80.3        | 72.1        | 72.1         | 81.9           | 81.9    |
| 310-320        | 80.95       | 85.71 | 90.47       | 90.47 | 66.6        | 100.0                                      | 76.1        | 66.6        | 76.1        | 76.1         | 76.1           | 76.1    |
| 320-330        | 80.0        | 86.6  | 90.0        | 90.0  | 76.6        | 96.6                                       | 80.0        | 76.6        | 76.6        | 76.6         | 83.3           | 83.3    |

Table 4.16. Statistical interpretation of the Experiment 8 results. Models on the left-hand side (LHS) are sourced from Experiment 7 and evaluated using our protocol. Data is taken from Table 4.15. Values correspond to rank-1 accuracies.

|                | Pre-Trained |             |             |                | Trained in GREW dataset using our protocol |             |             |              |                |
|----------------|-------------|-------------|-------------|----------------|--------------------------------------------|-------------|-------------|--------------|----------------|
|                | SMPLGait-2D | GaitBase-DA | SwinGait-3D | SkeletonGait++ | SMPLGait-2D                                | GaitBase-DA | SwinGait-3D | SkeletonGait | SkeletonGait++ |
| Input Type     | Sil.        | Sil.        | Sil.        | Sil. + SM      | Sil.                                       | Sil.        | Sil.        | SM           | Sil. + SM      |
| Input Modality | 2D          | 2D          | 2D          | 2D + 2D        | 2D                                         | 2D          | 2D          | 2D           | 2D + 2D        |
| Average        | 62.02       | 69.71       | 77.61       | 83.34          | 32.91                                      | 33.78       | 26.9        | 40.09        | 44.46          |

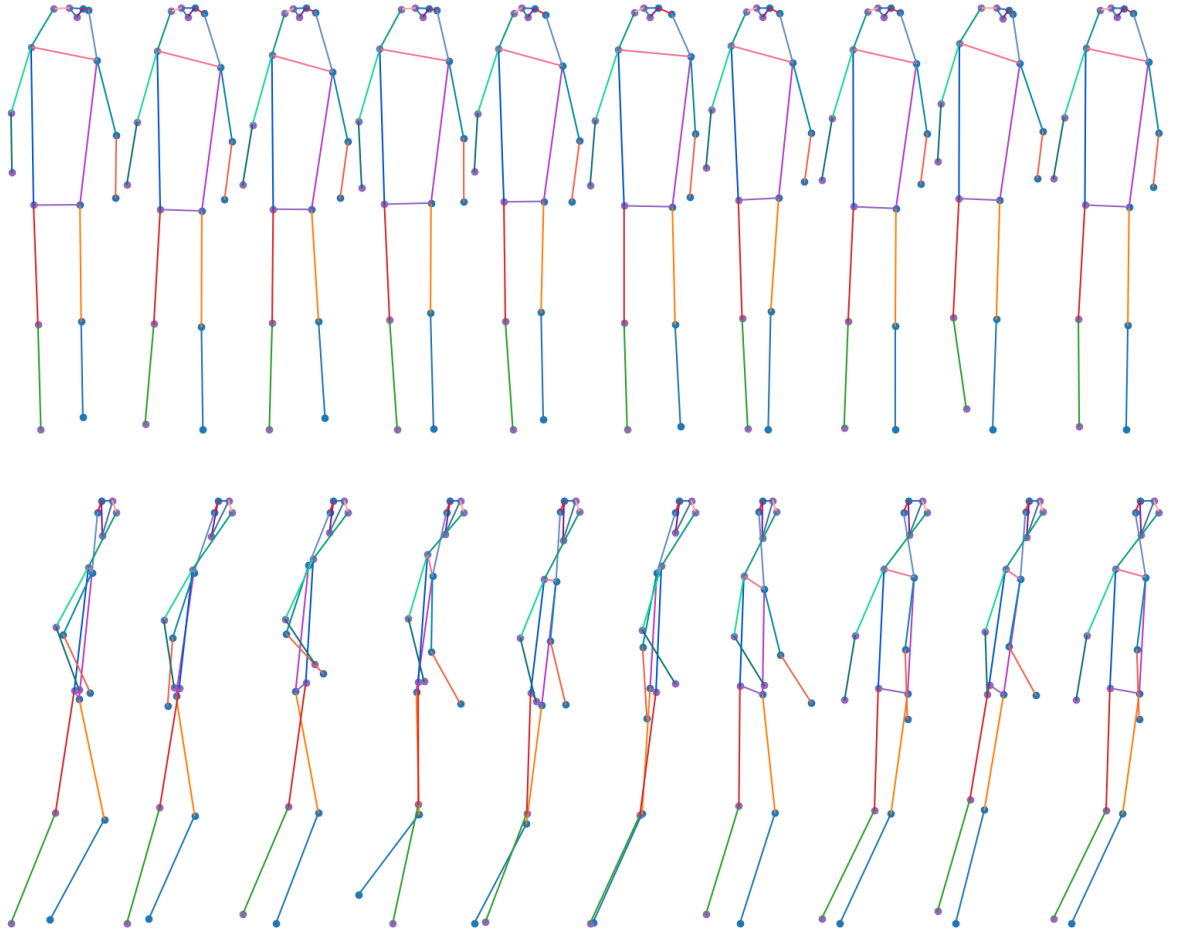


Figure 4.11. Sample Poses from the GREW dataset. The first row contains 2D Poses generated from 3D poses, and the second row contains rotated 2D Poses generated from rotated 3D poses.

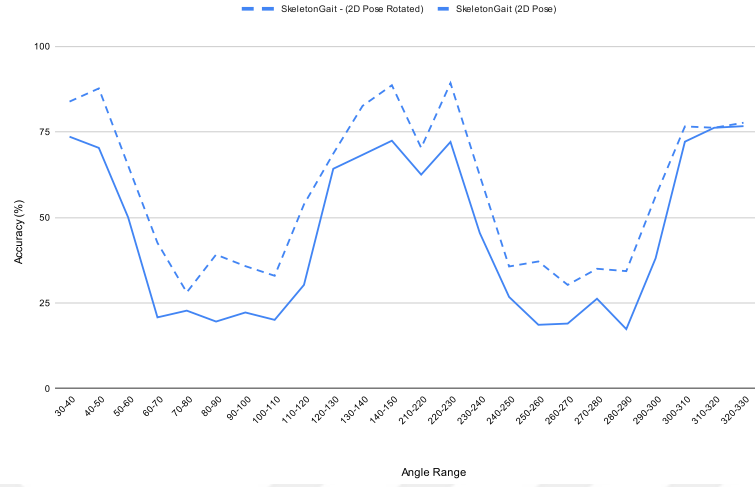


Figure 4.12. The Rank-1 accuracies of Gait Recognition methods in rotated and restructured GREW dataset. The data is taken from Table 4.15 and Table 4.17.

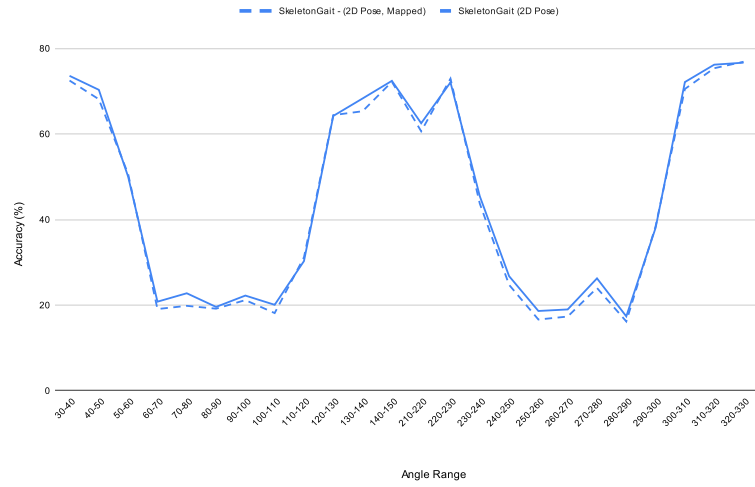


Figure 4.13. The R-1 accuracies of pre-trained Gait Recognition methods in the mapped and restructured GREW dataset. The data is taken from Table 4.19 and Table 4.15.

Table 4.17. Evaluation of SkeletonGait framework on the rotated GREW test set. Models are trained in Experiment 8 with the restructured GREW dataset. Values correspond to rank-1 accuracies.

| Angles  | Accuracy |
|---------|----------|
| 30-40   | 83.8     |
| 40-50   | 87.6     |
| 50-60   | 65.1     |
| 60-70   | 42.5     |
| 70-80   | 28.1     |
| 80-90   | 39.1     |
| 90-100  | 35.7     |
| 100-110 | 32.9     |
| 110-120 | 53.8     |
| 120-130 | 68.6     |
| 130-140 | 82.5     |
| 140-150 | 88.6     |
| 210-220 | 70.3     |
| 220-230 | 89.2     |
| 230-240 | 62.2     |
| 240-250 | 35.6     |
| 250-260 | 37.1     |
| 260-270 | 30.2     |
| 270-280 | 35.0     |
| 280-290 | 34.3     |
| 290-300 | 56.1     |
| 300-310 | 76.5     |
| 310-320 | 76.1     |
| 320-330 | 77.6     |
| Average | 40.6     |

Table 4.18. Comparison of SkeletonGait performance in Experiment 9 against Experiment 8. Models are trained in Experiment 8 with the restructured GREW dataset. Data in LHS is taken from Table 4.16 and RHS is taken from Table 4.17.

|         | Evaluated with original Skeletons | Evaluated with rotated Skeletons |
|---------|-----------------------------------|----------------------------------|
| Average | 26.9                              | 40.6                             |

Table 4.19. The R-1 accuracies of SkeletonGait in the mapped GREW dataset.

Models are trained in Experiment 8 with the restructured GREW dataset.

|         |      |
|---------|------|
| 30-40   | 72.4 |
| 40-50   | 68.1 |
| 50-60   | 50.6 |
| 60-70   | 19.0 |
| 70-80   | 19.8 |
| 80-90   | 19.1 |
| 90-100  | 21.1 |
| 100-110 | 18.1 |
| 110-120 | 31.1 |
| 120-130 | 64.4 |
| 130-140 | 65.3 |
| 140-150 | 72.1 |
| 210-220 | 60.6 |
| 220-230 | 72.9 |
| 230-240 | 43.7 |
| 240-250 | 24.7 |
| 250-260 | 16.6 |
| 260-270 | 17.3 |
| 270-280 | 24.0 |
| 280-290 | 16.1 |
| 290-300 | 38.4 |
| 300-310 | 70.5 |
| 310-320 | 75.3 |
| 320-330 | 76.8 |
| Average | 25.5 |

Table 4.20. Comparison of SkeletonGait performance in Experiment 10 against Experiment 8. Models are trained in Experiment 8 with the restructured GREW dataset. Data in LHS is taken from Table 4.16 and RHS is taken from Table 4.19. The Skeletons on the RHS correspond to 2D skeletons and are generated from 3D Skeletons in the GREW dataset using our protocol.

|         | Evaluated with original Skeletons | Evaluated with generated Skeletons |
|---------|-----------------------------------|------------------------------------|
| Average | 26.9                              | 25.5                               |

#### 4.2.4. Interpretation of GREW Results

Considering the results shown in Figure 4.10, it was observed that the pre-trained models outperformed the retrained models, similar to prior observations in Gait3D. As shown in Table 4.15, pre-trained SMPLGait-2D and SwinGait methods exhibited an average performance improvement of 34% and 45% over their retrained counterparts. This can be attributed to the fact that 92% of the test dataset overlaps with the training data used for pre-trained models.

Furthermore, the performance of pre-trained models improved as the gallery size increased. For instance, the pre-trained SMPLGait-2D and SwinGait methods achieved 50% and 66% accuracy within the 50-60 degree range, where the gallery contains 1,271 sequences. In contrast, their accuracy increased to approximately 60% and 75% in the 180-190 and 320-330 degree ranges, which are the bins with the smallest gallery size around 447.

Evaluating the performance of models trained using the proposed angular separation protocol revealed that all models achieved higher accuracy on test angles closer to the angular ranges used in training, with performance declining as test angles deviated further. For instance, retrained models achieved an average accuracy of 54,53% in the 50-60 degree range, while this dropped to 39% in the 0-10 degree range. Similarly, for test angles close to the training set, such as 120-130, 230-240, and 300-310 degrees, the average accuracy was 55%, 53%, and 54%, respectively, while for the farthest ranges, like 170-180 and 350-360 degrees, performance fell to 43% and 41%.

Based on the results presented in Figure 4.12, it can be stated that the rotated test set resulted in better performance compared to the non-rotated test set. Specifically, among all angular bins for all models, the maximum observed improvement is 24%, while the minimum is 0%. Rotating the poses improved the performance of the SkeletonGait model remarkably. For example, the performance of SkeletonGait in the 60-70 and 280-290 degrees range is improved by 22% and 17%, respectively. Further-

more, the weighted average R-1 accuracy of SkeletonGait framework is increased by 13%.

Considering the results shown in Figure 4.13, it can be stated that using 2D poses that are mapped from 3D poses resulted in a slight decay in performance in general. For example, the performance of SkeletonGait in the 30-40 and 130-140 degrees range is decreased by 1% and 3%, respectively. Additionally, mapping 3D poses into 2D resulted in about a 2% performance decay in terms of weighted average R-1 accuracy for the SkeletonGait.

The training for GREW was conducted on a computer with a 2xH100 GPU, 160 vCPU cores, and 640GB of RAM. The training durations were 3,6 days for the SMPLGait-2D model, 2,9 days for GaitBase-DA, 4 days for SwinGait and 6 days for SkeletonGait.

## 5. CONCLUSION

In this thesis, we assessed the viewpoint robustness of advanced gait recognition frameworks by introducing a comprehensive evaluation protocol applied to two large-scale, real-world datasets, Gait3D and GREW. We proposed a three-fold protocol to assess the effect of changing the camera viewpoint on gait recognition methods. First, we divided the aforementioned gait recognition datasets into  $10^\circ$  bins and assigned each gait sequence to a bin based on the most frequently occurring bin of its frames. Second, we restructured the datasets such that each gait sequence appears only in side-profile views in the training set and front or back views in the test set. The side-profile view mainly corresponds to  $180 \pm 30^\circ, 0 \pm 30^\circ$ . Lastly, we rotated the test set to align with the training viewpoint and evaluated the gait recognition frameworks with the rotated dataset. Specifically, we applied our protocol to top-performing gait recognition frameworks, including SMPLGait, GaitBase, SwinGait, and SkeletonGait, in the Gait3D and GREW datasets to analyze their robustness against viewpoint variations. We also reproduced and validated the results of these gait recognition methods. Additionally, we have provided angular labels for each gait sequence in the Gait3D and GREW datasets, which address a missing component and enable more precise and detailed analyses of viewpoint variations and their effects. We also proposed two novel gait representations, called 1-channel Depth Image and RGB Depth Image, which show superior performance in top-performing gait recognition methods. Most importantly, we have managed to improve the performance of the best-performing gait recognition method, SkeletonGait++, by using the RGB Depth Image.

Our analysis of models trained using the proposed angular separation protocol showed that all models achieved higher accuracy when evaluated on angles closer to those used in training, with performance decreasing as the test angles diverged further. This sensitivity to viewpoint variations was consistent across all methods. Interestingly, the use of 3D gait representations such as SMPL parameters combined with 2D representations did not improve robustness compared to their 2D-only counter-

parts. For instance, SMPLGait-2D outperformed SMPLGait-3D on the reconstructed Gait3D dataset with a 1% higher accuracy. The outperformance of pre-trained models over retrained models on both Gait3D and GREW was also a significant finding. For example, on the Gait3D dataset, the pre-trained SMPLGait-3D performed 71%, whereas our protocol-trained counterpart performed 43.2% accuracy. This behavior can be attributed to the overlap between the test dataset and the training data used for pre-trained models (75% for Gait3D and 92% for GREW).

Rotating test sets to align with training viewpoints significantly enhanced recognition accuracy. This approach demonstrated the positive effect of viewpoint alignment on performance, particularly for methods that combine both 3D and 2D gait representations. On the Gait3D dataset, SMPLGait-3D demonstrated an 8% improvement in average R-1 accuracy when the test set was rotated, while SkeletonGait, which uses both rotated silhouettes and 2D poses, achieved a 4% increase. This can be attributed to the superior information gain from 3D information when properly aligned. Similarly, on GREW, rotating poses improved SkeletonGait’s average R-1 accuracy by 13%.

Furthermore, our experiments demonstrated that recognition accuracy declines when using silhouettes or 2D poses derived from SMPLs or 3D poses, which indicated the importance of utilizing original or well-aligned data rather than processed 2D projections for robust gait recognition. On Gait3D, using SMPL-mapped silhouettes resulted in a general decline in performance. For example, GaitBase and SwinGait experienced 16% and 20% drops, respectively, in specific angular ranges. SkeletonGait also demonstrated a 7% performance decay when mapping 3D poses to 2D. Similarly, on GREW, mapping 3D poses to 2D resulted in about a 2% performance decay for SkeletonGait.

Regarding our proposed gait representations, SkeletonGait++ achieved the highest accuracy of 93.4% using RGB Depth Images with Skeleton Maps on Gait3D, outperforming its counterpart evaluated with SMPL-generated silhouettes by 7.3% and pre-trained version by 0.3%. Using 1-channel Depth Images over silhouettes did not

consistently increase average accuracy, however utilizing RGB Depth Images as input modality generally yielded the best average accuracy, both with and without rotation. Notably, rotated 3-channel Depth Images outperformed their 1-channel counterparts by 4% on average accuracy.

In conclusion, we have observed that SkeletonGait++ is the top-performing model among all experiments. Moreover, our findings contribute significantly to gait recognition by supporting the development of more accurate and viewpoint-invariant biometric systems. We have quantified the viewpoint sensitivity of state-of-the-art methods, highlighted the benefits of viewpoint alignment, and demonstrated the potential of novel gait representations, particularly RGB Depth images, for enhancing recognition performance. Future research should focus on developing methods that inherently address viewpoint variations without requiring explicit test-set rotation or relying solely on pre-trained models with extensive training data overlap.

## 6. FUTURE WORK

Current gait recognition frameworks mostly use Set Pooling (SP) [66] to aggregate frame-level features into a single set-level representation. Set Pooling operates by taking the maximum value across frames. A future direction would be to explore the Set Transformer [67] architecture as an alternative to Set Pooling, considering advantages in handling set-structured data. The Set Transformer employs a self-attention mechanism to process every element in an input set, which enables it to encode pairwise and higher-order interactions among elements of the set. Moreover, Set Transformer contains parameterized and learnable components. On the other hand, Set Pooling processes elements independently using simpler pooling operations. Regarding our study, SkeletonGait is the top-performing method, and SwinGait also demonstrates strong capabilities, which is the second best performing approach. By considering the complementary strengths observed when combining different modalities, a valuable future work would be to introduce the Skeleton Map modality as a new information branch within the SwinGait architecture.

## REFERENCES

1. Seely, R. D., S. Samangoeei, M. Lee, J. N. Carter and M. S. Nixon, “The University of Southampton Multi-Biometric Tunnel and Introducing a Novel 3D Gait Dataset”, *2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems*, pp. 1–6, Washington, DC, USA, 2008.
2. Shen, C., S. Yu, J. Wang, G. Q. Huang and L. Wang, “A Comprehensive Survey on Deep Gait Recognition: Algorithms, Datasets, and Challenges”, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, Vol. 7, No. 2, pp. 270–292, 2025.
3. Wang, C., J. Zhang, L. Wang, J. Pu and X. Yuan, “Human Identification Using Temporal Information Preserving Gait Template”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 11, pp. 2164–2176, 2012.
4. Zhang, Y., Y. Huang, S. Yu and L. Wang, “Cross-View Gait Recognition by Discriminative Feature Learning”, *IEEE Transactions on Image Processing*, Vol. 29, pp. 1001–1015, 2020.
5. Jain, A. K., P. Flynn and A. A. Ross, *Handbook of Biometrics*, Springer-Verlag, Berlin, Heidelberg, 2007.
6. Wan, C., L. Wang and V. V. Phoha, “A Survey on Gait Recognition”, *ACM Computing Surveys (CSUR)*, Vol. 51, No. 5, pp. 1–35, 2018.
7. Zhao, G., G. Liu, H. Li and M. Pietikainen, “3D Gait Recognition Using Multiple Cameras”, *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 529–534, Southampton, 2006.
8. Bolle, R., S. Pankanti and A. K. Jain, *Biometrics, Personal Identification in Networked Society: Personal Identification in Networked Society*, Kluwer Academic

Publishers, USA, 1st edn., 1998.

9. Wang, L., T. Tan, H. Ning and W. Hu, “Silhouette analysis-based gait recognition for human identification”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, pp. 1505–1518, 2003.
10. Liao, R., C. Cao, E. B. Garcia, S. Yu and Y. Huang, “Pose-Based Temporal-Spatial Network (PTSN) for Gait Recognition with Carrying and Clothing Variations”, *Chinese Conference on Biometric Recognition*, pp. 474–483, Springer, Cham, 2017.
11. Yamauchi, K., B. Bhanu and H. Saito, “Recognition of Walking Humans in 3D: Initial Results”, *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 45–52, Miami, FL, USA, 2009.
12. Shiraga, K., Y. Makihara, D. Muramatsu, T. Echigo and Y. Yagi, “GEINet: View-Invariant Gait Recognition Using a Convolutional Neural Network”, *International Conference on Biometrics (ICB)*, pp. 1–8, Halmstad, Sweden, 2016.
13. Chao, H., Y. He, J. Zhang and J. Feng, “GaitSet: Regarding Gait as a Set for Cross-View Gait Recognition”, *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 8126–8133, Honolulu, Hawaii, USA, 2019.
14. Shen, C., F. Chao, W. Wu, R. Wang, G. Q. Huang and S. Yu, “LidarGait: Benchmarking 3D Gait Recognition with Point Clouds”, *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1054–1063, Vancouver, BC, Canada, 2023.
15. Fan, C., J. Ma, D. Jin, C. Shen and S. Yu, “SkeletonGait: Gait Recognition Using Skeleton Maps”, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, No. 2, pp. 1662–1669, 2024.
16. Han, J. and B. Bhanu, “Individual Recognition Using Gait Energy Image”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 2, pp.

316–322, 2006.

17. Sarkar, S., P. Phillips, Z. Liu, I. Vega, P. Grother and K. Bowyer, “The HumanID Gait Challenge Problem: Datasets, Performance, and Analysis”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 2, pp. 162–177, 2005.
18. Li, J., Y. Zhang, Y. Zeng, C. Ye, W. Xu, X. Ben, F.-Y. Wang and J. Zhang, “Rethinking Appearance-Based Deep Gait Recognition: Reviews, Analysis, and Insights From Gait Recognition Evolution”, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 36, No. 6, pp. 9777–9797, 2025.
19. Liao, R., Z. Li, S. S. Bhattacharyya and G. York, “PoseMapGait: A Model-Based Gait Recognition Method with Pose Estimation Maps and Graph Convolutional Networks”, *Neurocomputing*, Vol. 501, No. C, p. 514–528, 2022.
20. Ariyanto, G. and M. S. Nixon, “Model-Based 3D Gait Biometrics”, *2011 International Joint Conference on Biometrics (IJCB)*, pp. 1–7, Washington, DC, USA, 2011.
21. Zheng, J., X. Liu, W. Liu, L. He, C. Yan and T. Mei, “Gait Recognition in the Wild with Dense 3D Representations and A Benchmark”, *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20196–20205, New Orleans, LA, USA, 2022.
22. Wang, Z., S. Hou, M. Zhang, X. Liu, C. Cao and Y. Huang, “GaitParsing: Human Semantic Parsing for Gait Recognition”, *IEEE Transactions on Multimedia*, Vol. 26, pp. 4736–4748, 2024.
23. Jin, D., C. Fan, W. Chen and S. Yu, “Exploring More from Multiple Gait Modalities for Human Identification”, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39, No. 4, pp. 4120–4128, 2025.

24. Zu, Z., X. Guo, T. Yang, J. Huang, J. Deng, G. Huang, D. Du, J. Lu and J. Zhou, “Gait Recognition in the Wild: A Benchmark”, *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14769–14779, Montreal, QC, Canada, 2021.
25. Yu, S., D. Tan and T. Tan, “A Framework for Evaluating the Effect of View Angle, Clothing and Carrying Condition on Gait Recognition”, *18th International Conference on Pattern Recognition (ICPR’06)*, Vol. 4, pp. 441–444, 2006.
26. Takemura, N., Y. Makihara, D. Muramatsu, T. Echigo and Y. Yagi, “Multi-View Large Population Gait Dataset and Its Performance Evaluation for Cross-View Gait Recognition”, *IPSJ Transactions on Computer Vision and Applications*, Vol. 10, No. 1, 2018.
27. Fan, C., J. Liang, C. Shen, S. Hou, Y. Huang and S. Yu, “OpenGait: Revisiting Gait Recognition Towards Better Practicality”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9707–9716, Vancouver, BC, Canada, 2023.
28. Fan, C., S. Hou, Y. Huang and S. Yu, “Exploring Deep Models for Practical Gait Recognition”, *arXiv Preprint arXiv:2303.03301*, 2024.
29. Fan, C., Y. Peng, C. Cao, X. Liu, S. Hou, J. Chi, Y. Huang, Q. Li and Z. He, “GaitPart: Temporal Part-Based Model for Gait Recognition”, *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14213–14221, Seattle, WA, USA, 2020.
30. Lin, B., S. Zhang and X. Yu, “Gait Recognition via Effective Global-Local Feature Representation and Local Temporal Aggregation”, *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14628–14636, Montreal, QC, Canada, 2021.

31. Huang, X., D. Zhu, H. Wang, X. Wang, B. Yang, B. He, W. Liu and B. Feng, “Context-Sensitive Temporal Feature Learning for Gait Recognition”, *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12889–12898, Montreal, QC, Canada, 2021.
32. Zheng, J., X. Liu, X. Gu, Y. Sun, C. Gan, J. Zhang, W. Liu and C. Yan, “Gait Recognition in the Wild with Multi-Hop Temporal Switch”, *arXiv Preprint arXiv:2209.00355*, 2022.
33. Teepe, T., J. Gilg, F. Herzog, S. Hörmann and G. Rigoll, “Towards a Deeper Understanding of Skeleton-based Gait Recognition”, *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1568–1576, New Orleans, LA, USA, 2022.
34. Fu, Y., S. Meng, S. Hou, X. Hu and Y. Huang, “GPGait: Generalized Pose-Based Gait Recognition”, *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19538–19547, Paris, France, 2023.
35. Ma, K., Y. Fu, D. Zheng, C. Cao, X. Hu and Y. Huang, “Dynamic Aggregated Network for Gait Recognition”, *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22076–22085, Vancouver, BC, Canada, 2023.
36. Dou, H., P. Zhang, W. Su, Y. Yu, Y. Lin and X. Li, “GaitGCI: Generative Counterfactual Intervention for Gait Recognition”, *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5578–5588, Vancouver, BC, Canada, 2023.
37. Wang, L., B. Liu, F. Liang and B. Wang, “Hierarchical Spatio-Temporal Representation Learning for Gait Recognition”, *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19582–19592, Paris, France, 2023.
38. Loper, M., N. Mahmood, J. Romero, G. Pons-Moll and M. J. Black, “SMPL: a

- Skinned Multi-Person Linear Model”, *ACM Transactions on Graphics*, Vol. 34, No. 6, pp. 1–16, 2015.
39. Zhu, H., W. Zheng, Z. Zheng and R. Nevatia, “GaitRef: Gait Recognition with Refined Sequential Skeletons”, *arXiv Preprint arXiv:2304.07916*, 2023.
  40. Wei, S., Z. Chen, F. Wei, S. Z. Yang and C. Wang, “DyGait: Gait Recognition Network Based on Skeleton Dynamic Features”, *IEEE Access*, Vol. 12, pp. 189535–189546, 2024.
  41. Huang, P., Y. Peng, S. Hou, C. Cao, X. Liu, Z. He and Y. Huang, “Occluded Gait Recognition with Mixture of Experts: An Action Detection Perspective”, *Computer Vision – ECCV 2024*, pp. 380–397, Springer Nature Switzerland, Cham, 2025.
  42. Zheng, J., X. Liu, B. Zhang, C. Yan, J. Zhang, W. Liu and Y. Zhang, “It Takes Two: Accurate Gait Recognition in the Wild via Cross-granularity Alignment”, *Proceedings of the 32nd ACM International Conference on Multimedia*, p. 8786–8794, New York, NY, USA, 2024.
  43. Zheng, W., H. Zhu, Z. Zheng and R. Nevatia, “GaitSTR: Gait Recognition with Sequential Two-Stream Refinement”, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, Vol. 6, No. 4, pp. 528–538, 2024.
  44. Wu, Z., C. Zhang, H. Xu, P. Jiao and H. Wang, “DAGait: Generalized Skeleton-Guided Data Alignment for Gait Recognition”, *arXiv Preprint arXiv:2503.18830*, 2025.
  45. Liu, J., Y. Ke, T. Zhou, Y. Qiu and C. Wang, “GaitRGA: Gait Recognition Based on Relation-Aware Global Attention”, *Sensors*, Vol. 25, No. 8, 2025.
  46. Dong, Y., C. Yu, R. Ha, Y. Shi, Y. Ma, L. Xu, Y. Fu and J. Wang, “HybridGait: a Benchmark for Spatial-Temporal Cloth-Changing Gait Recognition with Hy-

- brid Explorations”, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, No. 2, pp. 1600–1608, 2024.
47. Wang, Z.-Y., J. Liu, J. Chen and R. Chellappa, “VM-Gait: Multi-Modal 3D Representation Based on Virtual Marker for Gait Recognition”, *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 5326–5335, Tucson, AZ, USA, 2025.
  48. Zheng, J., X. Liu, S. Wang, L. Wang, C. Yan and W. Liu, “Parsing is All You Need for Accurate Gait Recognition in the Wild”, *Proceedings of the 31st ACM International Conference on Multimedia*, p. 116–124, Ottawa, ON, Canada, 2023.
  49. Ma, K., Y. Fu, C. Cao, S. Hou, Y. Huang and D. Zheng, “Learning Visual Prompt for Gait Recognition”, *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 593–603, Seattle, WA, USA, 2024.
  50. Wang, Z., S. Hou, M. Zhang, X. Liu, C. Cao, Y. Huang, P. Li and S. Xu, “QAGait: Revisit Gait Recognition from a Quality Perspective”, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, No. 6, pp. 5785–5793, 2024.
  51. Xu, H., C. Zhang, Z. Wu, P. Jiao and H. Wang, “PSGait: Gait Recognition Using Parsing Skeleton”, *arXiv Preprint arXiv:2503.12047*, 2025.
  52. He, K., X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition”, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.
  53. Fu, Y., Y. Wei, Y. Zhou, H. Shi, G. Huang, X. Wang, Z. Yao and T. Huang, “Horizontal Pyramid Matching for Person Re-Identification”, *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 8295–8302, Honolulu, Hawaii, USA, 2019.
  54. Luo, H., Y. Gu, X. Liao, S. Lai and W. Jiang, “Bag of Tricks and a Strong Baseline

- for Deep Person Re-Identification”, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1487–1495, Long Beach, CA, USA, 2019.
55. Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, *arXiv Preprint arXiv:2010.11929*, 2021.
  56. Liu, Z., Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”, *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, Montreal, QC, Canada, 2021.
  57. Cao, Z., G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 1, pp. 172–186, 2021.
  58. Tan, D., K. Huang, S. Yu and T. Tan, “Efficient Night Gait Recognition Based on Template Matching”, *Proceedings of the 18th International Conference on Pattern Recognition*, Vol. 3, p. 1000–1003, Hong Kong, China, 2006.
  59. Iwashita, Y., R. Baba, K. Ogawara and R. Kurazume, “Person Identification from Spatio-Temporal 3D Gait”, *International Conference on Emerging Security Technologies (EST)*, pp. 30–35, Canterbury, UK, 2010.
  60. Benedek, C., B. Galai, B. Nagy and Z. Janko, “Lidar-Based Gait Analysis and Activity Recognition in a 4D Surveillance System”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 28, No. 1, p. 101–113, 2018.
  61. Zhang, Z., L. Tran, F. Liu and X. Liu, “On Learning Disentangled Representations for Gait Recognition”, *IEEE Transactions on Pattern Analysis and Machine*

*Intelligence*, Vol. 44, No. 1, pp. 345–360, 2022.

62. Yamada, H., J. Ahn, O. M. Mozos, Y. Iwashita and R. K. and, “Gait-Based Person Identification Using 3D LiDAR and Long Short-Term Memory Deep Networks”, *Advanced Robotics*, Vol. 34, No. 18, pp. 1201–1211, 2020.
63. Li, X., Y. Makihara, C. Xu and Y. Yagi, “Multi-View Large Population Gait Database With Human Meshes and Its Performance Evaluation”, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, Vol. 4, No. 2, pp. 234–248, 2022.
64. Peng, Y., C. Cao and Z. He, “Occluded Gait Recognition”, *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Gold Coast, Australia, 2023.
65. Li, W., S. Hou, C. Zhang, C. Cao, X. Liu, Y. Huang and Y. Zhao, “An In-Depth Exploration of Person Re-Identification and Gait Recognition in Cloth-Changing Conditions”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13824–13833, Vancouver, BC, Canada, 2023.
66. Zaheer, M., S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov and A. J. Smola, “Deep Sets”, *Advances in Neural Information Processing Systems*, Vol. 30, p. 3394–3404, Red Hook, NY, USA, 2017.
67. Lee, J., Y. Lee, J. Kim, A. Kosiorek, S. Choi and Y. W. Teh, “Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks”, *Proceedings of the 36th International Conference on Machine Learning*, pp. 3744–3753, Long Beach, CA, USA, 2019.
68. Sun, Y., Q. Bao, W. Liu, Y. Fu, M. J. Black and T. Mei, “Monocular, One-stage, Regression of Multiple 3D People”, *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11159–11168, Montreal, QC, Canada, 2021.

69. Fu, Y., Y. Wei, Y. Zhou, H. Shi, G. Huang, X. Wang, Z. Yao and T. Huang, “Horizontal Pyramid Matching for Person Re-Identification”, *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 8295–8302, Honolulu, Hawaii, USA, 2019.
70. Erdem, D. B. and B. Gökberk, “Analysis of Viewpoint Dependency in Gait Recognition”, *2024 32nd Signal Processing and Communications Applications Conference (SIU)*, p. 1–4, 2024.
71. Tiñini Alvarez, I. R. and G. Sahonero-Alvarez, “Cross-View Gait Recognition Based on U-Net”, *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, Glasgow, UK, 2020.
72. Zhang, C., X.-P. Chen, G.-Q. Han and X.-J. Liu, “Spatial Transformer Network on Skeleton-Based Gait Recognition”, *Expert Systems*, Vol. 40, No. 6, p. 13244, 2023.
73. Teepe, T., A. Khan, J. Gilg, F. Herzog, S. Hörmann and G. Rigoll, “Gaitgraph: Graph Convolutional Network for Skeleton-Based Gait Recognition”, *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 2314–2318, Anchorage, AK, USA, 2021.
74. Liang, J., C. Fan, S. Hou, C. Shen, Y. Huang and S. Yu, “GaitEdge: Beyond Plain End-to-End Gait Recognition for Better Practicality”, *Computer Vision – ECCV 2022*, pp. 375–390, Tel Aviv, Israel, 2022.