Self-Supervised Transformer-based Gait Recognition Under Clothing Variation

by

Reza Ehsani

A dissertation submitted to the
Department of Computer Science
in conformity with the requirements for
the degree of Master of Science

Bishop's University
Canada
October 2026

Copyright © Reza Ehsani, 2026
released under a CC BY-SA 4.0 License

Abstract

Gait recognition identifies individuals from their walking patterns, but clothing changes and carried items remain major challenges for real-world deployment. Bulky coats and bags can change the appearance cues used by silhouette-based systems, which can reduce recognition accuracy. Pose-based representations can avoid some of these problems because joint coordinates are less sensitive to clothing than silhouettes. However, clothing variation can still affect pose-based methods by reducing the quality of the pose estimates.

This thesis proposes a factorized spatial-temporal transformer for pose-based gait recognition. The model treats each frame-joint pair as a separate token and does not use an anatomical hierarchy, fixed skeletal graph, or attribute-supervised pretraining. We pretrain the model using a single self-supervised objective: masked-joint reconstruction. We evaluate it on CASIA-B under a subject-disjoint protocol across matched walking, carrying, and clothing-change conditions.

We show four main findings. First, with about 813,000 parameters, the model achieves Rank-1 accuracy comparable to GaitGraph and PoseGait, although it remains below larger, more recent methods such as GaitPT and MAST-GCN. We report this as what a compact, minimal-prior design achieves under the conditions tested, not as evidence that removing structural priors causes this result, since we run no controlled comparison against a prior-matched architecture. Second, CASIA-B and DenseGait pretraining give different results across the three conditions. This suggests that, at the scale tested, neither dataset size nor domain match alone determines performance. Third, a fair comparison with random initialization requires careful tuning of the baseline. Our learning-rate sweep shows a 36-point range in validation accuracy, meaning that an undertuned baseline can make the benefit of pretraining appear larger than it is. Fourth, using lower-body joints gives the best result on the clothing-variation protocol compared with the upper-body region and the full skeleton.

This work does not close the performance gap to the strongest recent pose-based transformers under clothing variation. It instead provides evidence that the benefit of pretraining and the useful input region can depend on the evaluation condition.

Acknowledgments

I would like to thank my supervisor, Professor Madjid Allili, for accepting me into the program and giving me the freedom to work on a topic that I found interesting. I also thank my co-supervisor, Dr. Mohammed Ayoub Alaoui Mhamdi, for his help and support throughout this work.

I would like to thank the members of my examining committee, Dr. Russell Butler, Dr. Stefan Bruda, and Dr. Yasir Malik, for their time and for the feedback that helped improve the revised version of this thesis.

I am also grateful to Dr. Adrian Cosma and Dr. Emilian Radoi for providing access to the DenseGait dataset, and to the authors of CASIA-B and the GaitGraph pose release for making their data publicly available.

I would also like to thank the Department of Computer Science and Bishop's University for providing the resources and environment that made this work possible.

Finally, I am truly grateful to my family and friends for their support throughout this journey.

Contents

1 Introduction **1**
- 1.1 Biometric Identification . . . 1
- 1.2 Gait as a Biometric Modality . . . 2
- 1.3 Challenges of Clothing Variation . . . 3
- 1.4 Self-Supervised Transformers for Pose-Based Gait Recognition . . . 3
- 1.5 Problem Statement and Research Objective . . . 4
- 1.6 Thesis Contributions . . . 6
- 1.7 Scope and Limitations . . . 7
- 1.8 Thesis Structure . . . 8

2 Literature Review **9**
- 2.1 The History of Gait Recognition . . . 9
- 2.2 Gait Databases and Benchmarking . . . 12
- 2.3 Gait Recognition Under Clothing Variation . . . 17
- 2.4 Self-Supervised Learning in Gait Recognition . . . 19
- 2.5 Transformers and Attention Mechanism . . . 20

3 Methodology **25**
- 3.1 Overview of the proposed framework . . . 25
- 3.2 Architecture: Factorized Spatial-Temporal Transformer . . . 28
- 3.3 Self-Supervised Pretraining . . . 33
- 3.4 Region Partitioning . . . 37
- 3.5 Data Preprocessing . . . 39
- 3.6 Supervised Fine-Tuning . . . 41
- 3.7 Evaluation Protocol . . . 44

4 Experimental Results and Analysis **49**
- 4.1 Setup . . . 49
- 4.2 Main Results . . . 51
- 4.3 Comparison to State-of-the-Art . . . 53
- 4.4 In-Domain vs. In-the-Wild SSL Pretraining . . . 56
- 4.5 The Role of Baseline Tuning . . . 58

4.6 Ablation Studies . . . 60
4.7 Rank-1/Rank-5 Analysis . . . 63

5 Discussion and Conclusion **66**
5.1 Conclusions . . . 66
5.2 Limitations and Scope . . . 68
5.3 Future Work . . . 71
5.4 Closing Remarks . . . 73

Bibliography **74**

A Formal Definitions **81**
A.1 Attention . . . 81
A.2 Self-Attention . . . 82
A.3 Multi-Head Attention . . . 82
A.4 Transformer . . . 83
A.5 Vision Transformer . . . 83
A.6 Factorized Spatial-Temporal Transformer . . . 84
A.7 AdamW Optimizer . . . 85
A.8 COCO-17 Skeleton . . . 86
A.9 Notation Reference . . . 87

B Implementation Details, Hyperparameters, and Code References **88**
B.1 Code, Environment, and Data Access . . . 88
B.2 Self-Supervised Pretraining Configuration . . . 89
B.3 Supervised Fine-Tuning and Evaluation Configuration . . . 89
B.4 Preprocessing Confidence Thresholds . . . 90

C DenseGait Joint Remapping **91**

D Per-Seed Result Breakdowns **93**
D.1 Main Protocol Results . . . 93
D.2 Ablation Studies . . . 93

E Computational Requirements **96**
E.1 Hardware and Environment . . . 96
E.2 Model Size . . . 96
E.3 Dataset Scale . . . 96
E.4 Experiment Scale . . . 97
E.5 Training Duration . . . 97

List of Tables

2.1 Appearance-Based Gait Recognition Milestones . . . 13
2.2 Model-Based Gait Recognition Milestones . . . 13
2.3 Comparison of gait recognition datasets . . . 17
2.4 Recent Gait Recognition under Clothing Variation milestones . . . 19
2.5 Design space of gait transformers . . . 22

3.1 Region definitions . . . 37
3.2 Supervised fine-tuning configuration . . . 45
3.3 Evaluation configuration . . . 48

4.1 Rank-1 and Rank-5 accuracy (%) on CASIA-B . . . 51
4.2 Contribution of self-supervised pretraining . . . 53
4.3 Comparison of Rank-1 accuracy (%) on CASIA-B . . . 54
4.4 Comparison between self-supervised pretraining sources . . . 57
4.5 Backbone learning-rate sweep . . . 59
4.6 Metric-loss ablation . . . 60
4.7 Backbone freezing ablation . . . 61
4.8 Region ablation . . . 62
4.9 View-matching ablation . . . 62
4.10 Error recovery rate . . . 64

A.1 Notation reference. . . . 87

B.1 Self-supervised pretraining configuration . . . 89
B.2 Confidence thresholds . . . 90

C.1 DenseGait channel index to COCO-17 joint mapping . . . 92

D.1 Per-seed Rank-1 accuracy (%), main protocol comparison . . . 94
D.2 Per-seed Rank-5 accuracy (%), main protocol comparison . . . 94
D.3 Per-seed Rank-1 accuracy (%), all four ablations . . . 95

E.1 Wall-clock duration of self-supervised pretraining . . . 97

List of Figures

1.1 Common biometrics for human identification. 2

2.1 Sample silhouette images 10
2.2 A sample from CASIA-B dataset 14
2.3 A sample from OU-ISIR-B dataset 15
2.4 Samples frames from the GREW dataset 16

3.1 Overview of the proposed pipeline 26
3.2 Encoder overview 28
3.3 Factorized attention 29
3.4 Masked-joint reconstruction 35
3.5 The three region configurations 38

4.1 Training dynamics 50
4.2 Rank-1 accuracy (%) on CASIA-B 52
4.3 Downstream mean Rank-1 accuracy 58
4.4 Error recovery rate 64

Chapter 1

Introduction

Human identification is the process of verifying or determining an individual's identity, based on distinctive physical or behavioral characteristics. As smart technologies become more deeply integrated into everyday life, the need for reliable and accurate identification systems has grown accordingly. These systems now play a role in applications ranging from healthcare and access control to surveillance and public security. In general, an identification system compares observed data against a reference database containing information from known individuals.

Identity verification methods can be broadly categorized into knowledge-based, object-based, and biometric-based approaches [28]. Knowledge-based methods rely on information known only to the user, such as passwords or personal identification numbers (PINs). Object-based approaches require the possession of a physical item, such as an access card or a car key fob. These two methods are often combined to enhance security. A Tesla, for example, requires the driver to carry the key card or their smartphone (object-based), and offers an optional 4-digit PIN ("PIN to Drive"), entered on the touchscreen before driving, for extra protection (knowledge-based).

Biometric authentication follows a different paradigm. Instead of relying on external information or objects, biometric systems use intrinsic physiological or behavioral traits, such as fingerprints, facial features, voice, or gait. These characteristics are inherently tied to the individual, which makes them harder to forget, lose, or replicate. This is why biometric identification has gained increasing popularity in computer vision and pattern recognition applications.

1.1 Biometric Identification

Traditional authentication mechanisms continue to face limitations related to usability, and this has driven significant attention toward biometric-based identification. Biometric recognition is the automated identification of individuals based on measurable biological or behavioral traits [28]. Compared to knowledge- or object-based

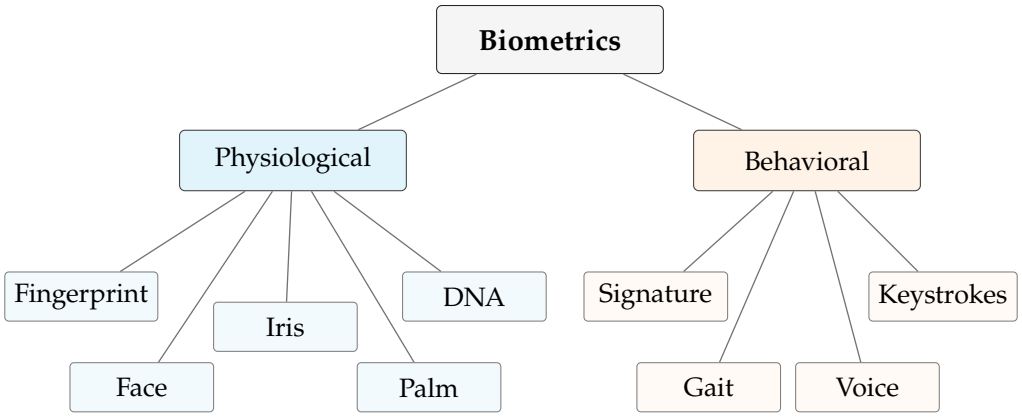


Figure 1.1: Common biometrics for human identification.

systems, biometrics offer a more natural and user-friendly means of identity verification.

We commonly divide biometric modalities into physiological and behavioral categories [56]. Physiological biometrics include traits such as fingerprints [29], facial structure [54], iris patterns [29], and palm geometry [23], which stay largely static over time. Behavioral biometrics, by contrast, capture patterns related to human activity, such as voice [29, 45], keystroke dynamics [1], and gait [41, 51, 53]. These modalities are shaped by motion and temporal dynamics, which makes them well suited to applications that require passive or long-range identification. Figure 1.1 shows examples of widely used biometric systems from both categories.

Each biometric modality has its own advantages and challenges. Physiological traits often reach high recognition accuracy under controlled conditions, but they typically require close-range acquisition and active user cooperation. Behavioral traits may be less distinctive on their own, but they offer a different advantage: we can often capture them unobtrusively, without any explicit interaction from the user.

1.2 Gait as a Biometric Modality

Gait is the manner in which a person walks, and it has been shown to contain distinctive identity-related patterns. Unlike many other biometric traits, we can capture gait at a distance, without the subject's awareness or cooperation. This property makes gait recognition particularly appealing for surveillance and security scenarios, where other biometrics such as face or fingerprint may not be reliably available. In many real-world scenarios, such as street surveillance or crime footage, people might be wearing masks or gloves, and camera angles or lighting conditions may be poor. Their walking style, however, tends to remain visible, which makes gait recognition a powerful and practical alternative in such cases [46].

Early gait recognition approaches relied heavily on appearance-based representations, such as silhouettes or averaged gait energy images. These methods performed well in controlled environments but proved highly sensitive to variations in viewpoint, walking speed, and appearance. In particular, changes in clothing can significantly alter the body's visual characteristics, leading to large performance degradation.

To address these limitations, researchers introduced model-based gait recognition approaches. These methods represent human motion using structured models, such as skeletons or joint trajectories, to reduce sensitivity to appearance changes. Pose-based representations focus on underlying motion dynamics rather than surface-level visual features and are therefore more robust to certain types of variability.

1.3 Challenges of Clothing Variation

Despite the advantages of pose-based representations, clothing variation remains a major challenge in gait recognition. Clothing can obscure joints, alter estimated limb proportions, and introduce systematic noise in pose estimation, particularly in the upper body. These effects increase intra-subject variability and reduce the consistency of identity-related gait cues.

Several studies have attempted to mitigate the impact of clothing variation through feature normalization, view-invariant modeling, or data augmentation strategies. Even with these techniques, however, recognition performance under clothing variation typically lags behind performance under normal walking conditions. This suggests that clothing variation affects not only appearance, but also the reliability of motion cues extracted from pose data.

1.4 Self-Supervised Transformers for Pose-Based Gait Recognition

Recent advances in deep learning have introduced new tools for modeling complex sequential data. Transformer architectures, originally developed for natural language processing, have been adapted to model temporal dependencies and relational structures in gait sequences. Their ability to capture long-range interactions makes them well-suited to pose-based gait representation learning.

In parallel, self-supervised learning (SSL) has emerged as a powerful paradigm for representation learning that does not rely on explicit identity labels. SSL approaches use intrinsic data properties, such as temporal consistency or instance discrimination, to learn meaningful embeddings from unlabeled sequences. In gait recognition, researchers have proposed SSL as a way to exploit large amounts of walking data while reducing the need for costly annotation. Large-scale, in-the-wild gait datasets (collected from uncontrolled, everyday environments rather than laboratory settings)

strengthen this motivation further, but the extent to which representations learned from such environments transfer to controlled gait recognition benchmarks remains unclear. Differences in capture conditions, subject behavior, and dataset structure may introduce a domain mismatch that limits downstream performance.

Within this broad direction, however, pose-based gait transformers do not all make the same design choices, and those differences matter. GaitPT [7] organizes attention hierarchically, grouping joints into anatomical body parts and restricting attention within each part before combining them. GaitGraph [59] takes a different route: it represents the skeleton as a graph with a fixed adjacency structure, defined by the bones connecting one joint to another, and applies graph convolutions over this structure. GaitFormer [11] pretrains its representation with a multi-task SSL objective, predicting several weak attribute labels, such as clothing type or carried objects, alongside its main task. PoseGait [35] takes yet another approach: instead of learning directly from raw joint coordinates, it computes hand-crafted geometric features, such as joint angles and limb lengths, and feeds these to the model as input. Each of these methods, in its own way, builds a structural prior directly into the model, whether that prior comes from anatomy, from a fixed graph, from auxiliary supervision, or from hand-engineered features.

This thesis takes a deliberately different position in this design space. We propose a factorized spatial-temporal transformer that imposes none of these priors. Joints are not grouped by anatomy, no graph adjacency is hard-coded, no attribute labels are used during pretraining, and the model receives raw normalized coordinates rather than hand-crafted features. In their place, we rely on a single, minimal self-supervised objective: masked-joint reconstruction, where the model learns to predict a joint's position from its surrounding spatial and temporal context. We implement this design and evaluate what it achieves under clothing variation, without a controlled comparison that isolates the contribution of removing any one of these priors specifically. Chapter 3 details the resulting architecture, and Chapter 2 places it more precisely within the design space sketched here.

1.5 Problem Statement and Research Objective

Although pose-based representations, transformer architectures, and self-supervised learning have each shown promise in gait recognition, several questions about how they interact under clothing variation remain open.

The first open question is architectural. Existing pose-based gait transformers embed structural priors directly into their design: anatomical hierarchies, fixed skeletal graph adjacency, hand-crafted geometric features, or multi-task attribute supervision (Section 1.4). What accuracy a smaller transformer reaches at this scale, with these priors removed and replaced by a single minimal self-supervised objective, is not established.

The second open question concerns the pretraining source. Large-scale in-the-wild gait datasets offer far more pretraining data than any single controlled benchmark, but the extent to which representations learned from unconstrained environments transfer to controlled benchmarks such as CASIA-B (a multi-view indoor gait dataset containing 124 subjects walking under normal, bag-carrying, and clothing-change conditions [70]) remains unclear. Differences in capture conditions, subject behavior, and dataset structure may introduce a domain mismatch that limits downstream performance, and whether in-domain (using data from the same benchmark as the downstream task) or in-the-wild pretraining is preferable at a given scale is not established either.

The third open question concerns how to measure the benefit of self-supervised pretraining in the first place. A comparison against a randomly initialized baseline is only informative if that baseline is tuned as carefully as the pretrained method it is compared against, and prior work does not typically report how sensitive such a baseline is to its own hyperparameters. Whether an undertuned baseline can manufacture an apparent self-supervised pretraining benefit is not established.

The fourth open question concerns which anatomical regions carry a reliable identity signal once clothing changes the visible body. Prior part-based work has argued that certain body regions are less affected by a given clothing change than others, but this intuition has not been tested within a minimal-prior transformer that imposes no anatomical grouping by design and that treats every joint identically by default.

This thesis addresses these four questions directly, through a combination of architectural design and controlled experimentation on CASIA-B. The objective of this thesis follows the same four-part structure.

First, we propose a factorized spatial-temporal transformer that removes the anatomical, graph-based, and attribute-supervision priors used by prior pose-based gait transformers. We test how this architecture, pretrained with a single masked-joint reconstruction objective, places against PoseGait, GaitGraph, GaitPT, and MAST-GCN under clothing variation.

Second, we pretrain this architecture on two sources under otherwise identical conditions: CASIA-B itself, an in-domain source, and DenseGait (a large-scale dataset of walking poses extracted from in-the-wild video [11]), a substantially larger in-the-wild source. We test which source produces better downstream accuracy and by how much.

Third, we sweep the backbone learning rate on the randomly initialized baseline and apply the same tuning discipline to every pretraining condition compared in this thesis. We test whether an undertuned random-initialization baseline can inflate the apparent benefit of self-supervised pretraining.

Fourth, we compare Rank-1 accuracy on the clothing-variation protocol across three region configurations that restrict the model's input to different anatomical subsets. We test whether the region least affected by CASIA-B's clothing condition

outperforms the region most affected by it, and whether it outperforms the full skeleton as well.

A fifth, supporting objective underlies all four: we quantify how sensitive the resulting model is to further evaluation and training design choices, including gallery size, view consistency, metric-loss choice, and backbone freezing. We report these as ablations that inform the four objectives above, rather than as headline contributions in their own right.

1.6 Thesis Contributions

This thesis makes four contributions.

The first contribution is architectural. We propose a factorized spatial-temporal transformer for pose-based gait recognition that removes the anatomical, graph-based, and attribute-supervision priors used by prior pose-based gait transformers (Section 1.4). Each (frame, joint) pair is treated as an independent token, and the encoder learns anatomical relationships from data rather than through a fixed hierarchy or graph. We pretrain this architecture with a single, minimal self-supervised objective: masked-joint reconstruction. At approximately 813,000 parameters, it reaches Rank-1 accuracy competitive with GaitGraph and PoseGait, while trailing GaitPT and MAST-GCN, particularly on the harder protocols. We report this as what this compact, minimal-prior design achieves under the conditions tested, not as evidence that removing structural priors specifically produces this accuracy, since this thesis does not run a controlled comparison against a parameter- and protocol-matched architecture that retains an explicit anatomical or graph-based prior.

The second contribution is a comparison between two self-supervised pretraining sources. We pretrain the same architecture on two sources under otherwise identical conditions: CASIA-B, an in-domain source, and DenseGait, a substantially larger in-the-wild source. We show that neither source wins consistently across the three evaluation protocols. In-domain pretraining outperforms in-the-wild pretraining under matched walking and under clothing change, while in-the-wild pretraining is stronger under the carried-object condition. This mixed result indicates that, at the scale tested here, neither pretraining scale nor domain alignment alone determines which source performs better.

The third contribution is a methodological finding about how self-supervised pretraining should be benchmarked. We sweep the backbone learning rate on the randomly initialized baseline and find that validation accuracy spans 36 percentage points across the rates tested, a swing large enough on its own to change which configuration looks best. We apply the winning rate uniformly to every configuration compared in this thesis, including the self-supervised pretraining conditions and the random-initialization baseline alike. This demonstrates that a fair comparison against a random-initialization baseline requires tuning that baseline as carefully as

the pretrained method, since an undertuned baseline can manufacture an apparent pretraining benefit that further tuning would erase.

The fourth contribution is an empirical finding about anatomical region choice. Under CASIA-B's outer-garment clothing condition, we find that restricting the model to Region R2 (6 lower-body joints) outperforms Region R1 (11 upper-body joints) by 22.1 points on the clothing-variation protocol, and outperforms the full skeleton (17 joints) by 11.1 points. This confirms and quantifies, within a controlled minimal-prior transformer setup, the intuition from prior part-based work that lower-body joints carry more reliable identity signal than upper-body joints under outer-garment clothing change.

Together, these four contributions address the four questions posed in Section 1.5: what accuracy a minimal-prior transformer reaches for pose-based gait recognition at this scale, whether pretraining scale or domain alignment determines which self-supervised source performs better, how a random-initialization baseline must be tuned for a self-supervised pretraining comparison to be fair, and which anatomical regions carry the more reliable identity signal once clothing changes the visible body.

1.7 Scope and Limitations

This thesis is scoped in three deliberate ways.

First, we operate entirely on pose-based representations of gait, not on silhouettes or other appearance-based features. This choice isolates motion dynamics from surface appearance, which is exactly the property that makes pose attractive under clothing variation. It also means this thesis does not draw on the residual appearance signal that silhouette-based methods retain even under altered clothing, and Chapter 4 shows that this signal is not negligible: the strongest silhouette-based baselines outperform this work specifically under clothing variation (Section 4.3).

Second, every result in this thesis comes from a single benchmark, CASIA-B, evaluated under one indoor, fixed-camera capture setting. DenseGait, the second dataset used in this thesis, serves only as a pretraining source, not as a second evaluation benchmark. Nothing in this thesis speaks to generalization across capture environments, camera geometries, or clothing types beyond what CASIA-B contains.

Third, this thesis tests whether self-supervised pretraining alone improves robustness to clothing variation. It does not evaluate techniques designed explicitly to target clothing invariance, such as adversarial debiasing or clothing-specific data augmentation. The architecture we propose is deliberately compact, at approximately 813,000 parameters, and is not intended to match the absolute accuracy of recent, substantially larger pose-based transformers. Chapter 4 positions this work's accuracy honestly against that recent work, rather than overstating it.

Chapter 5 discusses each of these scope boundaries in more depth, alongside additional limitations that emerged directly from the experimental results.

1.8 Thesis Structure

The remainder of this thesis is organized into four chapters.

Chapter 2 reviews related work in gait recognition. It covers the history of appearance-based and model-based approaches, the datasets and benchmarks relevant to this thesis, self-supervised learning in gait recognition, and the specific design choices made by existing pose-based gait transformers. Section 2.5.3 places this design-space comparison directly in relation to the architecture we propose in Chapter 3.

Chapter 3 describes the methodology. It covers the factorized spatial-temporal transformer architecture, the masked-joint reconstruction pretraining objective, the region partitioning used in the region ablation, data preprocessing, the supervised fine-tuning recipe, and the evaluation protocol.

Chapter 4 presents the experimental results. It reports the main protocol results across both pretraining sources and three evaluation conditions, compares this work against published methods, examines the in-domain versus in-the-wild pretraining comparison directly, reports the backbone learning-rate sweep underlying the baseline-tuning finding, reports four ablation studies, and analyzes the structure of the model's errors.

Chapter 5 discusses these results. It restates the four contributions in light of the evidence from Chapter 4, states the limitations of this work directly, and outlines the specific directions this thesis leaves open for future work.

Chapter 2

Literature Review

Gait recognition is a biometric method that identifies individuals based on how they walk. It offers several advantages over techniques such as facial or fingerprint recognition. It works from a distance. It does not require physical contact. It can even work without the person being aware they are being observed. These qualities make it especially useful for surveillance and security.

In recent years, deep learning has brought significant improvements to gait recognition, especially under controlled conditions, and it now outperforms traditional approaches in accuracy. The field still faces difficult challenges despite these advances, however. One of the most notable is how clothing affects the appearance and consistency of a person's gait in recorded data.

This chapter reviews the research most relevant to this thesis. We start by outlining traditional and deep learning-based methods. We then highlight recent studies that use self-supervised learning (SSL), transformers, and attention mechanisms. We also focus specifically on how researchers have addressed the problem of clothing variation. By reviewing these works, we identify the gaps this thesis addresses and develop in more detail the design-space position introduced in Section 1.4.

2.1 The History of Gait Recognition

Gait recognition has been studied for more than three decades. Two dominant research directions have emerged over this period. These are appearance-based and model-based approaches. Appearance-based methods rely primarily on silhouette representations to capture both shape and walking dynamics. Model-based methods, in contrast, focus on structural information such as joint positions and skeletal motion. These two directions differ in representation. Both, however, have evolved from handcrafted feature engineering toward data-driven learning frameworks, particularly with the adoption of deep learning methods.

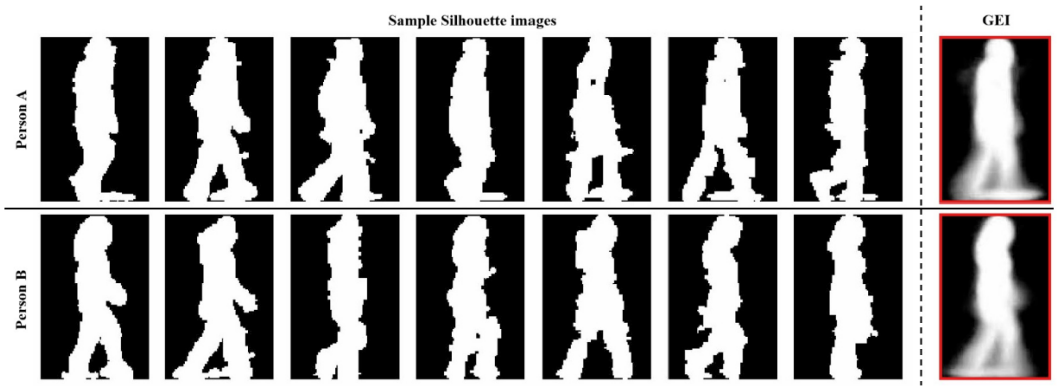


Figure 2.1: Sample silhouette images of two individuals over a gait cycle and their corresponding Gait Energy Image

2.1.1 Appearance-Based Gait Recognition

One of the earliest appearance-based gait recognition methods was introduced by Murase and Sakai in 1996 [43]. Their work showed that silhouette sequences could be projected into a low-dimensional eigenspace for efficient identity matching. This established gait as a viable biometric based purely on motion patterns.

Early research commonly aggregated silhouettes from a complete gait cycle into a single representation, in order to simplify temporal modeling. A notable example is the Gait Energy Image (GEI), proposed by Han and Bhanu in 2005 [24]. GEI computes the average of binary silhouettes over time. This forms a grayscale template that encodes walking style. This compact summary comes at a cost, however. GEI inherently suppresses fine-grained temporal dynamics. Sample silhouettes from a gait cycle and their corresponding GEIs are shown in Figure 2.1. Researchers later introduced several variants to address this limitation, including the Chrono-Gait Image (CGI) [62] and the Gait Flow Image (GFI) [32]. Both aim to retain richer temporal information.

Between 2010 and 2015, researchers explored a range of handcrafted descriptors, including Local Binary Patterns (LBPs) [27] and Histograms of Oriented Gradients (HOGs) [38]. To mitigate viewpoint variation, Muramatsu et al. [42] proposed View Transformation Models (VTMs), among other approaches. Despite these advances, most methods remained highly sensitive to background quality and to appearance changes caused by clothing or carried objects.

The introduction of deep learning marked a major shift in appearance-based gait recognition. GEINet [55] was among the first convolutional neural network (CNN) models to operate on GEI representations. It demonstrated improved cross-view performance while reducing reliance on handcrafted features. Subsequent works expanded on this direction. DBNGait [44] employed deep belief networks (DBNs)

for low-quality surveillance footage. GaitGAN [69] leveraged generative adversarial networks (GANs) to synthesize normalized silhouettes under varying viewpoints and clothing conditions.

More recent methods further diversified silhouette-based modeling strategies. GaitSet [8] treated gait as an unordered set of silhouettes. This relaxed the requirement for strict temporal alignment. DisentangledGait [71] explicitly separated identity-related features from appearance variations such as clothing. GaitPart [16] emphasized localized motion by partitioning the body into distinct regions. GLN [26] took a different approach. It combined features from different stages of a convolutional network through lateral, feature-pyramid-style connections. It also used a dedicated compact block, to keep the resulting representation both discriminative and low-dimensional. Temporal modeling was further strengthened by 3DCNNGait [36], which applied three-dimensional convolutions to capture spatio-temporal dynamics directly.

Transformer-based architectures have recently gained prominence in appearance-based gait recognition as well. GaitFormer introduced a fully transformer-based framework. It incorporates Temporal Periodic Alignment (TPA) and Temporal Aggregation Modules (TAM), to model walking periodicity more effectively. In parallel, GaitSSB [14] explored self-supervised contrastive learning on silhouette data. It achieved competitive performance with reduced reliance on labeled training samples. Section 2.4 discusses self-supervised learning strategies in more detail.

2.1.2 Model-Based Gait Recognition

Model-based gait recognition takes a different approach. Instead of appearance, it focuses on explicit structural representations, typically derived from joint locations, limb angles, or skeletal motion. One of the earliest examples was proposed by Niyogi and Adelson [47], who modeled human motion using a simplified five-segment stick figure embedded in a space-time volume. Their work showed that identity-specific motion patterns could be extracted without relying on visual appearance at all.

During the early 2000s, model-based approaches often relied on motion capture systems or multi-camera setups to estimate kinematic features such as stride length, joint trajectories, and body proportions. Yam et al. [66] applied dynamic time warping to joint trajectories for gait comparison. Urtasun et al. [60] introduced one of the first 3D gait recognition systems, using synchronized cameras and articulated body models. These methods showed robustness to appearance variation. They required accurate body fitting and controlled capture conditions, however.

Learning-based model-based approaches began to emerge in the late 2000s. Yoo et al. [68] extracted kinematic features from stick figures and used neural networks for classification. This demonstrated that structural motion cues alone could yield strong recognition performance. Adoption remained limited, however, because reliable pose estimation from monocular video was still difficult at the time.

Advances in pose estimation around 2015 significantly improved the practicality of model-based gait recognition. Feng et al. [18] used Long Short-Term Memory (LSTM) networks to model joint sequences and address viewpoint variation. The release of OpenPose by Cao et al. [5] enabled real-time multi-person pose estimation from a single RGB camera. This made large-scale and unconstrained gait analysis feasible for the first time. Subsequent studies, such as Qiao et al. [50] and Sokolova and Konushin [57], demonstrated the effectiveness of pose-based representations across diverse applications.

Hybrid and graph-based approaches were later proposed to improve robustness under clothing variation. Yao et al. [67] combined silhouette cues with joint information in a collaborative framework. Graph convolutional networks (GCNs) gained traction with models such as GaitGraph [59], which explicitly modeled spatial and temporal relationships between joints. Numerous recent works, including Multi-Scale Gait Graph (MSGG) [48] and MCDGait [64], introduced multi-scale fusion, collaborative learning, and knowledge distillation to improve performance in unconstrained environments.

Recent studies suggest that silhouette information may not be strictly necessary for effective gait recognition. Cosma and Radoi's pose-based GaitFormer [11] demonstrated strong generalization using transformer architectures trained on skeletal sequences. Other pose-focused methods, such as jointGait [34], GPGait [20], and GaitHeat [19], explored further alternatives to explicit skeleton extraction. GaitHeat, for instance, leverages pose heatmaps to improve robustness under occlusion and clothing variation.

Table 2.1 and Table 2.2 summarize the key milestones in appearance-based and model-based gait recognition, respectively. Together, these works illustrate how gait representations evolved from handcrafted silhouettes toward learned structural models. This evolution lays the foundation for the recent efforts, discussed later in this chapter, that explicitly address clothing variation and generalization.

2.2 Gait Databases and Benchmarking

Progress in gait recognition has been closely tied to the availability of well-designed datasets. These datasets capture variations in viewpoint, appearance, and walking conditions. Over the past two decades, researchers have introduced a range of gait databases to study increasingly realistic and challenging scenarios. Early datasets such as CASIA-B [70] and the USF Gait Challenge [52] established foundational benchmarks for silhouette-based gait recognition. These early datasets, however, were limited in scale and diversity. Subsequent datasets expanded both the number of subjects and the range of covariates. This enabled systematic evaluation under controlled and unconstrained conditions.

Table 2.1: Appearance-Based Gait Recognition Milestones

Year	Method	Approach Type	Key Contribution
2005	GEI[24]	Appearance-based	Introduced the Gait ENergy Image, a foundational silhouette representation
2010	CGI[62]	Appearance-based	Chrono-Gait Image captures temporal flow in silhouette sequences
2011	GFI[32]	Appearance-based	Gait Flow Image encodes motion dynamics using silhouette flow
2014	AVTM[42]	Appearance-based	Enabled view-invariant recognition using view transformation
2017	GaitGAN[69]	Appearance-based	Applied GANs to generate and enhance gait silhouettes under variation
2019	GaitSet[8]	Appearance-based	Proposed a set-based method for learning from silhouette sequences
2019	DisentangledGait[71]	Appearance-based	Disentangled identity and variation features from silhouettes
2020	GaitPart[16]	Appearance-based	Utilized body part division in temporal silhouettes
2023	GaitFormer[11]	Appearance-based	Introduced transformer architecture to silhouette-based gait recognition
2023	GaitSSB[14]	Appearance-based	Applied self-supervised learning to silhouette data

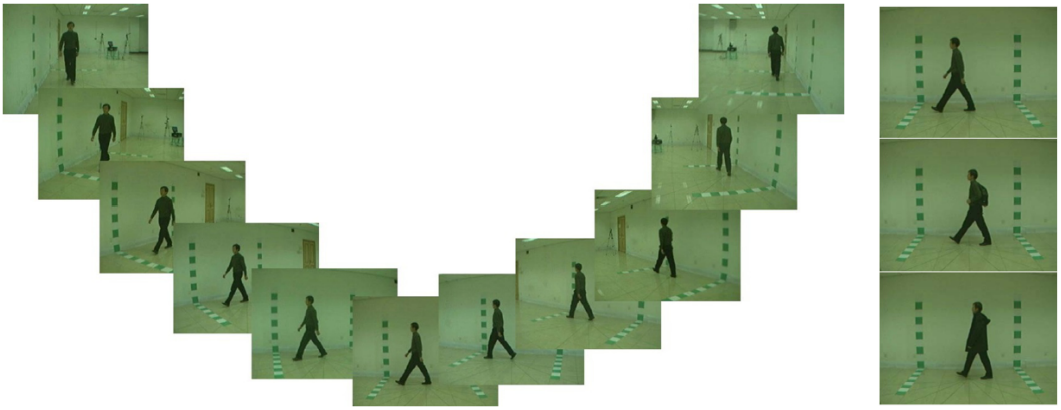
Table 2.2: Model-Based Gait Recognition Milestones

Year	Method	Approach Type	Key Contribution
1994	XYT Stick-Figure[47]	Model-based	Encoded walking figures in XYT space using stick models
2012	Kinect-based Gait[49]	Model-based	Used depth sensors to extract joint positions for gait recognition
2016	Feng-LSTM[18]	Model-based	LSTM network for modeling joint-based gait sequences
2020	PoseGait[35]	Model-based	Pose-based framework using joint sequences for recognition
2021	GaitGraph[59]	Model-based	Applied GCN to joint sequences for improved gait recogniton
2022	SMPLGait[72]	Model-based	Used SMPL model to capture clothing-invariant 3D body features
2022	GaitFormer[11]	Model-based	Pose-based gait recognition using transformer
2024	GaitHeat[19]	Model-based	Recent model using pose heatmaps for robust recogniton

CASIA-B

CASIA-B remains one of the most widely used benchmarks in gait recognition research. It contains gait sequences from 124 subjects, recorded under 11 viewpoints ranging from 0 to 180 degrees. Each subject performs multiple walking sequences under three conditions: normal walking (NM), walking while carrying a bag (BG), and walking while wearing a coat (CL). The dataset was recorded in a controlled indoor environment, with each subject observed for approximately ten minutes.

Figure 2.2 illustrates example RGB frames from CASIA-B. It includes multi-view normal walking sequences (Figure 2.2a) and the three appearance conditions (Figure 2.2b). CASIA-B has a clear separation between clothing and carrying



(a) Normal walking sequence (b) Conditions

Figure 2.2: (a) Example RGB frames from CASIA-B under normal walking across different viewpoints, and (b) Walking under different conditions (from top to bottom: normal, carrying a bag, and wearing a coat) [70]

variations. This is why it has become a standard benchmark for evaluating cross-condition gait recognition, and why it remains central to many comparative studies. It is also the primary evaluation benchmark used in this thesis (Chapter 3).

OU-ISIR-B

The OU-ISIR Treadmill datasets were developed by Osaka University's Institute of Science and Industrial Research. Researchers designed them to isolate specific covariates affecting gait recognition. The OU-ISIR Treadmill B subset focuses explicitly on clothing variation. It captures 68 subjects walking on a treadmill under up to 32 distinct clothing combinations. This controlled setup allows for fine-grained analysis of how specific garments influence gait representations.

Figure 2.3 shows representative clothing variations included in the dataset. The constrained treadmill setting limits environmental diversity. OU-ISIR-B nonetheless remains a valuable benchmark for studying the impact of clothing changes in isolation.

In recent years, the focus of gait dataset collection has shifted toward unconstrained, real-world environments. These datasets aim to reflect practical deployment conditions, including cluttered backgrounds, uncontrolled viewpoints, and natural clothing variation. Researchers often use them to train large-scale and self-supervised models.



Figure 2.3: Examples of clothing variations in the OU-ISIR Treadmill B dataset [40]

GREW

GREW (Gait REcognition in the Wild) was introduced by Zhu et al. [73]. It is one of the first large-scale gait datasets collected entirely from unconstrained surveillance footage. GREW comprises 128,671 walking sequences from 26,345 individuals. It also includes an additional 233,857 distractor sequences, designed to simulate open-set recognition scenarios. Researchers collected the data from hundreds of surveillance cameras in public spaces, capturing diverse viewpoints, occlusions, clothing styles, and carrying conditions.

Figure 2.4 presents sample frames from GREW. It illustrates both covariate-rich sequences and distractor examples. The authors report that dataset annotation required extensive manual effort, involving 20 annotators over three months. GREW provides silhouettes, GEIs, and both 2D and 3D pose estimates. This supports research in both appearance-based and model-based gait recognition. Experiments on GREW highlight a substantial performance drop under clothing variation. They also show that pretraining on GREW can improve generalization when transferring to datasets such as CASIA-B.

Gait3D

Zheng et al. [72] released Gait3D to bridge the gap between controlled datasets and real-world deployment. It contains over 25,000 gait sequences from more than 4,000



Figure 2.4: Samples frames from the GREW dataset [73]. The first row shows a subject under various covariates, including occlusion, clothing change, view variation, and carried objects. The second row illustrates distractor sequences.

individuals, captured using a dense multi-camera setup in a campus environment. The dataset includes diverse viewpoints, lighting conditions, and partial occlusions.

Gait3D reflects realistic acquisition settings, but it does not explicitly target clothing variation. Researchers therefore use it primarily to evaluate robustness to viewpoint and environmental changes, rather than as a dedicated benchmark for clothing-invariant gait recognition.

DenseGait

Cosma and Radoi proposed DenseGait [11], the largest publicly available skeleton-based gait dataset to date. It contains over 217,000 tracklets, represented as sequences of 2D joint coordinates extracted from real-world surveillance footage. Each sequence is automatically annotated with up to 42 appearance-related attributes, including clothing style, carried objects, and body shape. These attributes serve as weak supervision signals rather than ground-truth labels.

Cosma and Radoi designed DenseGait explicitly to support self-supervised learning. Their transformer-based GaitFormer model [11] trained solely on DenseGait, without identity labels, and still demonstrated strong generalization across datasets. This result highlights the potential of large-scale, unlabeled skeletal data for learning transferable gait representations. This thesis uses DenseGait as one of two self-supervised pretraining sources (Section 3.3), and compares its downstream benefit directly against in-domain pretraining on CASIA-B itself (Section 4.4).

Table 2.3: Comparison of gait recognition datasets relevant to clothing variation.

Dataset	Year	Subjects	Clothing Variation	Capture Setting	Relevance to This Work
CASIA-B	2005	124	NM / BG / CL conditions	Indoor, 11 views (0°–180°)	Primary evaluation benchmark
OU-ISIR-B	2012	68	32 clothing combinations	Indoor treadmill, side view	Isolates clothing type in a controlled setting
GREW	2021	26K+	Natural, uncontrolled	Outdoor, multi-view	In-the-wild clothing and scene diversity
Gait3D	2022	4K+	Real-world clothes	Outdoor, multi-camera	Focused on viewpoint, not clothing
DenseGait	2022	Unknown	Mixed attire (weak attribute labels)	Outdoor surveillance	In-the-wild pretraining source
GaitLU-1M	2023	Unknown	Unlabeled, implicit variation	Outdoor surveillance	Large-scale SSL benchmark

GaitLU-1M

Fan et al. introduced GaitLU-1M [14] in 2023. It is the first public gait dataset to exceed one million gait sequences. GaitLU-1M consists of over 1,000,000 unlabeled clips, extracted from more than 1,300 videos recorded in diverse outdoor environments. Each clip contains an average of 92 frames, which results in approximately 92 million silhouette images in total.

GaitLU-1M lacks explicit clothing annotations. Its large-scale diversity across geography, lighting, camera viewpoints, and walking conditions, however, makes it particularly suitable for self-supervised pretraining. Fan et al. demonstrated that GaitSSB [14], trained entirely on GaitLU-1M, achieves strong generalization despite the absence of identity labels. This result underscores the value of massive unlabeled datasets for gait recognition.

Together, these datasets have enabled systematic benchmarking of gait recognition models under both controlled and unconstrained conditions. Table 2.3 provides a comparative overview of the major gait datasets relevant to clothing variation. The next section reviews prior work that explicitly addresses gait recognition under clothing variation, and examines how researchers have used these datasets to evaluate robustness.

2.3 Gait Recognition Under Clothing Variation

Gait recognition systems deployed in real-world environments inevitably face changes in clothing and carried items. Researchers commonly refer to this problem as clothing variation (CV). It remains one of the most persistent challenges in gait recognition. Bulky coats, loose garments, or handheld objects can significantly alter a person's apparent body shape and motion patterns. This leads to substantial performance degradation, particularly for silhouette-based models. Such appearance changes are unavoidable in surveillance and forensic scenarios. Robustness to clothing variation is therefore a critical requirement for practical gait recognition systems.

Prior to the adoption of deep learning, early attempts to address clothing variation

relied primarily on handcrafted or model-driven strategies. Several methods focused on isolating body regions that are less affected by clothing, such as the lower limbs. Others emphasized internal motion dynamics instead of outer appearance. Yam et al. [66], for instance, used a model-based representation based on joint motion to reduce sensitivity to clothing changes. Muramatsu et al. [42] proposed the Arbitrary View Transformation Model (AVTM). This model aimed to map gait features across different viewpoints and clothing conditions through learned transformations.

With the emergence of deep learning, researchers began exploring more flexible and data-driven solutions. One line of work sought to normalize appearance, by synthesizing canonical representations that resemble a standard clothing condition. GaitGAN [69], for example, applied adversarial learning to generate normalized gait images from inputs captured under varying viewpoints and clothing. Another influential direction involved feature disentanglement, where identity-related information is explicitly separated from appearance factors. DisentangledGait [71] followed this approach, using a dual-stream architecture designed to suppress clothing-induced factors.

Hybrid approaches were later introduced to combine complementary information from multiple modalities. Peng et al. [48] proposed a framework that fused silhouette-based features with structural skeleton representations. This allowed the model to retain discriminative motion cues even when appearance varied significantly. Such methods attempt to balance the descriptive power of silhouettes with the relative invariance of pose-based features.

More recently, pose-based gait recognition has gained increased attention as a potential solution to clothing variation. By operating directly on joint coordinates or skeletal motion, these models reduce reliance on external appearance cues. Feng et al. [18] demonstrated early success by modeling joint trajectories using LSTM networks. PoseGait [35] further incorporated prior knowledge of human body structure to improve robustness under noisy pose estimates.

Architectural advances have continued to strengthen pose-based methods. GaitGraph [59] modeled joints as nodes in a graph, explicitly capturing spatial and temporal relationships through a graph convolutional network. Zheng et al. introduced SMPLGait [72], which reconstructs a full 3D human body mesh using the SMPL model. This lets the system infer underlying body motion even when clothing obscures the visible silhouette. More recently, Khaliluzzaman and Deb [30] proposed MAST-GCN, a multi-part attention-guided spatial-temporal graph convolutional network. MAST-GCN explicitly weights different body regions by their reliability under a given condition, rather than treating all joints uniformly. On CASIA-B, it reports 88.5% Rank-1 accuracy under clothing change, the strongest clothing-condition result in the literature reviewed in this chapter, including silhouette-based methods.

The main developments targeting clothing variation are summarized in Table 2.4. Despite these advances, many approaches still struggle under extreme clothing changes, such as heavy winter coats or large carried objects. Most existing datasets

Table 2.4: Recent Gait Recognition under Clothing Variation milestones

Year	Method	Approach Type	Clothing Strategy
2017	GaitGAN[69]	Appearance-based	Generates canonical (normal clothing) gait images using GANs
2019	DisentangledGait[71]	Appearance-based	Disentangles clothing and identity features for robust representation
2020	PoseGait[35]	Model-based	Improves pose estimation and temporal modeling to reduce clothing bias
2021	CollaborativeGait[67]	Hybrid	Combines skeleton and silhouette features for complementary information
2022	SMPLGait[72]	Model-based	Used 3D body shape modeling to filter out clothing-induced noise
2023	GaitSSB[14]	Appearance-based	Learns robust representations via contrastive learning on diverse silhouettes
2026	MAST-GCN[30]	Model-based	Multi-part attention weights body regions by reliability under clothing change

also provide limited diversity in clothing types. This constrains generalization to unseen environments. As a result, performance improvements reported under controlled benchmarks do not always translate to real-world robustness.

These limitations have motivated growing interest in self-supervised learning (SSL) and transformer-based architectures. By leveraging large volumes of unlabeled or weakly labeled data, these methods aim to learn more general and transferable gait representations. This reduces dependence on carefully curated inputs. Recent results suggest promising directions. Systematic evaluation under clothing variation, however, remains necessary to assess the robustness and limitations of such models in practice.

2.4 Self-Supervised Learning in Gait Recognition

The growing interest in self-supervised learning (SSL) has introduced new opportunities for gait recognition. This is particularly true in scenarios where collecting labeled data under all appearance conditions, such as clothing variation, is impractical. Traditional supervised approaches depend heavily on large-scale identity annotations. These are expensive to obtain and difficult to maintain in real-world surveillance environments. The wide diversity of human appearance further compounds this challenge. SSL offers an alternative: it enables models to learn meaningful representations directly from unlabeled data, which reduces reliance on manual annotation.

SSL methods aim to capture stable and transferable gait features by exploiting intrinsic structure in the data. Common strategies include contrastive learning, which pulls together different views or augmentations of the same sample, and predictive learning, which reconstructs missing or future information. In gait recognition, these approaches are particularly appealing. They encourage models to focus on motion dynamics rather than static appearance. As a result, SSL has the potential to reduce sensitivity to viewpoint changes, occlusions, and appearance

variation, including clothing.

This paradigm is especially relevant in surveillance settings, where large volumes of unlabeled walking footage are readily available. The GaitLU-1M dataset, for example, contains over one million walking clips without identity labels (Section 3.1). SSL frameworks can use natural regularities in human gait, such as periodicity and temporal consistency, as implicit supervision signals for representation learning.

Many SSL methods adopted in gait recognition draw on advances in computer vision, including SimCLR [9], MoCo [25], BYOL [22], and SwAV [6]. Researchers have adapted these ideas to gait-specific settings, where positive pairs may consist of walking sequences of the same individual captured under different viewpoints or appearance conditions. Such formulations encourage invariance to clothing, while preserving identity-related motion patterns.

GaitSSB [14] is one early application of large-scale contrastive SSL in gait recognition, trained on the GaitLU-1M dataset introduced in the same work. GaitSSB uses short silhouette clips as positive examples and temporally shuffled clips as negatives. This design emphasizes smooth motion continuity over static appearance. Despite training entirely without identity labels, GaitSSB demonstrated strong transfer performance. This result indicates that SSL can produce discriminative gait representations even under appearance variation.

More recently, Cosma et al. [10] extended self-supervised learning to skeleton-based gait recognition at a substantially larger scale. Their model, which they name GaitPTv2, is pretrained using contrastive self-supervised learning, following the SimCLR framework, rather than through reconstruction. The authors pretrain GaitPTv2 on 2.7 million walking sequences collected from real-world video streams. They then evaluate it zero-shot, without any fine-tuning, across four benchmark datasets, including CASIA-B. GaitPTv2 does not explicitly target clothing variation. Its scale and its zero-shot transfer results, however, represent one of the clearest demonstrations to date of how far self-supervised pretraining can generalize in this domain.

Overall, these studies suggest that SSL is a scalable and practical approach for learning gait representations from large, unlabeled datasets. The extent to which SSL pretraining consistently improves robustness under controlled clothing variation, however, remains insufficiently explored. This gap motivates the systematic evaluation of SSL-pretrained models under clothing variation benchmarks that this thesis undertakes (Chapter 4).

2.5 Transformers and Attention Mechanism

Transformer models and attention mechanisms have significantly influenced modern deep learning. They have led to substantial performance improvements across a wide range of domains, including computer vision. Researchers originally introduced transformers for natural language processing. Since then, transformers

have demonstrated strong capability in image and video understanding tasks as well. Their use of self-attention allows the modeling of long-range dependencies and global context. This offers advantages over convolutional neural networks, which are limited to local receptive fields, and over recurrent models, which can struggle with long temporal sequences.

In gait recognition, transformers are particularly appealing. They can jointly model spatial and temporal dependencies across frames or joint sequences. Convolutional CNN-based approaches, by contrast, often focus on local appearance patterns or short temporal windows. This can make them sensitive to changes in clothing or carried objects. Attention-based models can instead adaptively emphasize informative motion cues, while down-weighting irrelevant appearance variations. This makes them well suited to unconstrained gait recognition scenarios.

2.5.1 Vision Transformers in Gait Recognition

Vision Transformers (ViTs) have attracted growing interest in gait recognition, due to their ability to model full-body motion without relying on fixed local structures. Instead of operating on local convolutional kernels, ViTs divide silhouette frames into patches and process them as tokens through layers of self-attention. This design lets the model capture relationships between different body regions across time, which is particularly useful when appearance varies.

Wu et al.'s GaitFormer [63] is a representative example of this direction. It employs a temporal transformer backbone to process silhouette sequences, and incorporates Temporal Periodic Alignment (TPA), a strategy combining an Adaptive Fourier-transform Position Encoding with a Temporal Aggregation Module (TAM), to model walking periodicity explicitly. GaitFormer achieved strong performance on datasets such as CASIA-B. These results suggest that transformer architectures, when designed with gait-specific characteristics in mind, can help mitigate sensitivity to appearance changes.

Another key development is GaitPT [7], which adopted a pure transformer architecture for skeleton-based gait recognition. GaitPT treats sequences of joint coordinates as tokens with positional embeddings, and captures spatial and temporal relationships using self-attention alone. Despite its architectural simplicity and minimal preprocessing, the model achieved competitive performance on standard benchmarks. This demonstrates the expressive power of transformer-based motion modeling.

Cosma et al.'s GaitPTv2 [10] later built on this architecture, at a substantially larger scale, using self-supervised pretraining in place of direct supervision (Section 2.4). Together, these models established a new direction for gait recognition based on transformer architectures with fewer handcrafted components.

Hybrid approaches combining graph-based representations with transformers have also been explored. Several recent works combine graph-based modeling of

Table 2.5: Design space of pose- and skeleton-based gait transformers

Method	Architectural prior	Input representation	Training paradigm
PoseGait [35]	None (attention over features)	Hand-crafted geometric features	Fully supervised
GaitGraph [59]	Fixed skeletal graph adjacency	Raw joint coordinates	Fully supervised
GaitPT [7]	Hierarchical anatomical grouping	Raw joint coordinates	Fully supervised
GaitFormer [11]	None (flat transformer)	Raw joint coordinates	Multi-task SSL, attribute supervision
GaitPTv2 [10]	Inherits GaitPT's hierarchy	Raw joint coordinates	Contrastive SSL, 2.7M-sequence scale

skeletal joints with transformer-based attention mechanisms for temporal aggregation [59]. These results further support the suitability of attention mechanisms for modeling structured human motion under variable appearance conditions.

2.5.2 Pose-Based and Skeleton-Based Transformers

Transformers have also been widely adopted in pose-based and skeleton-based gait recognition. In these approaches, joint coordinates serve as input features. This reduces reliance on silhouette appearance and increases robustness to clothing variation. Transformer architectures can model long-term dependencies in joint motion more effectively than traditional CNNs or RNNs.

GaitPTv2's pretraining objective, described in Section 2.4, is itself a skeleton-based transformer application. It pretrains a scaled variant of GaitPT using contrastive self-supervised learning, joining GaitSSB and, in a multi-task form, GaitFormer as further examples of the contrastive and attribute-supervised objectives that dominate self-supervised gait recognition. Reconstruction-based self-supervision, the approach this thesis pursues instead (Section 3.3), is comparatively uncommon in this literature.

PoseGait [35] takes a different approach. It incorporates attention mechanisms to focus on reliable joints, while suppressing noisy or occluded ones. By integrating prior knowledge of human anatomy, the model reduces sensitivity to pose estimation errors. On clothing variation subsets of CASIA-B, PoseGait outperformed several silhouette-based approaches. This highlights the potential of pose-based attention models under appearance change.

2.5.3 Design Choices in Pose-Based Gait Transformers

The transformer-based methods discussed above do not all make the same design choices. Table 2.5 organizes five representative methods along three axes: the structural prior imposed on the architecture, the input representation used, and the training paradigm.

Reading down each column reveals a spectrum rather than a single dividing line. On architectural prior, methods range from an explicit fixed graph (GaitGraph) to

a hand-designed anatomical hierarchy (GaitPT, inherited unchanged by GaitPTv2) to no structural prior at all (GaitFormer). On input representation, only PoseGait departs from raw coordinates, computing joint angles, bone lengths, and velocities before the model ever sees the data. In the training paradigm, the differences are largest. PoseGait, GaitGraph, and GaitPT are trained directly and supervised, with no pretraining stage. GaitFormer pretrains with a multi-task objective that predicts weak attribute labels, such as clothing type, alongside its main task. GaitPTv2 pretrains with contrastive self-supervised learning, at a scale of 2.7 million in-the-wild sequences, and its primary contribution is a set of scaling laws relating downstream accuracy to data size, model size, and compute.

No existing method combines an unrestricted architecture, raw-coordinate input, and a single minimal self-supervised objective at a modest scale. This is the specific combination that Chapter 3, Section 3.2.7 positions this thesis's architecture against directly: a flat, factorized transformer with no anatomical or graph-based prior, trained with masked-joint reconstruction alone, and pretrained on a corpus of at most 217,000 sequences (DenseGait) or a small in-domain corpus drawn from CASIA-B itself, in both cases substantially smaller than GaitPTv2's 2.7-million-sequence corpus, roughly twelve times smaller in DenseGait's case (Chapter 3, Section 3.2.7). The comparison this thesis makes is not whether a smaller, simpler model outperforms these methods in absolute terms; Section 4.3 shows directly that it does not, against the strongest of them. The comparison is whether removing these priors, at this reduced scale, still reaches a competitive band.

Chapter Summary

This chapter reviewed gait recognition research relevant to this thesis, from early appearance-based methods to recent self-supervised transformer architectures. Both appearance-based and model-based approaches have moved progressively toward representations that emphasize motion dynamics over static appearance (Section 2.1). Large-scale datasets and unconstrained benchmarks have played a role in exposing the limitations of earlier methods under clothing variation (Section 2.2).

Clothing variation remains one of the most persistent challenges in gait recognition (Section 2.3). Recent pose-based methods have made substantial progress on this specific problem. MAST-GCN, for instance, reports the strongest published clothing-condition result reviewed in this chapter. Self-supervised learning has emerged as a promising strategy for learning transferable gait representations from unlabeled data (Section 2.4). Transformer architectures provide flexible mechanisms for modeling long-range spatial and temporal dependencies (Section 2.5).

Existing pose-based gait transformers, however, all impose some form of structural prior. This prior may take the shape of an anatomical hierarchy, a fixed skeletal graph, hand-crafted geometric features, or attribute-supervised pretraining (Section 2.5.3). No existing method combines an unrestricted architecture,

raw-coordinate input, and a single minimal self-supervised objective at a modest scale. This is the specific gap this thesis addresses. Chapter 3 proposes a factorized spatial-temporal transformer that removes these priors, and pretrains it with masked-joint reconstruction alone. Chapter 4 then tests this design directly. It asks whether the architecture reaches a competitive band without these priors, whether in-domain or in-the-wild pretraining transfers better to CASIA-B, and how the benefit of self-supervised pretraining changes as clothing variation makes the raw pose signal less reliable.

Chapter 3

Methodology

3.1 Overview of the proposed framework

We address pose-based gait recognition under clothing variation through a two-stage pipeline: self-supervised pretraining of a factorized spatial-temporal transformer on unlabeled pose sequences, followed by supervised fine-tuning with metric learning on a labeled identification set. At evaluation time, we retrieve identities via a centroid-based k-nearest neighbor search ($k = 1$) over ℓ_2-normalized embeddings. Figure 3.1 summarizes the pipeline. The components are detailed in the rest of this chapter.

This design reflects two constraints of the problem. The first constraint is about representation. Silhouette-based methods rely on appearance cues, but these are exactly the cues that clothing changes corrupt. This is why a pose-only representation is needed in the first place. The second constraint is about data. CASIA-B contains only 124 subjects, and the subject-disjoint protocol further reduces the number of subjects available for training the identification head. As a result, a purely supervised pipeline does not have enough samples per identity to learn invariances that generalize across clothing conditions. Self-supervised pretraining addresses this problem. It learns a representation from unlabeled pose sequences before introducing any identity labels. This allows the supervised stage to begin from an informed initialization instead of starting from scratch.

The encoder is a factorized spatial-temporal transformer. Each (frame, joint) pair is treated as an independent token. The encoder alternates between spatial attention, which operates within a frame across joints, and temporal attention, which operates within a joint across frames. This factorization keeps the sequence length manageable at each block, exposes both axes of variation to direct attention, and preserves joint identity throughout the network. This choice is deliberate. We want the model to learn anatomical relationships directly from the data, rather than imposing them through fixed body-part hierarchies or skeletal graph adjacencies.

Pretraining uses a masked-joint reconstruction objective. A random subset of

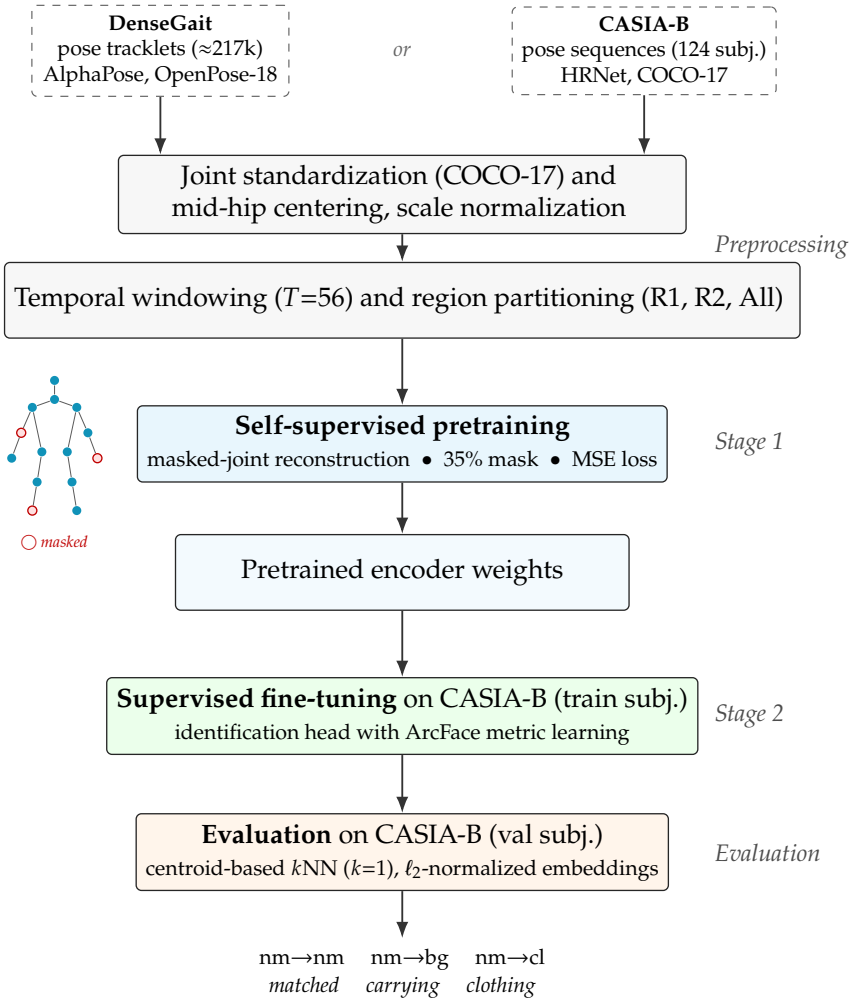


Figure 3.1: Pose sequences from one of two unlabeled pretraining sources (CASIA-B or DenseGait) are standardized to a common COCO-17 joint convention, windowed into fixed-length clips, and used to pretrain the factorized spatial-temporal encoder via masked-joint reconstruction (Stage 1). We then fine-tune the pretrained encoder on labeled CASIA-B training subjects with an ArcFace identification objective (Stage 2). At evaluation, embeddings of probe sequences from held-out test subjects are matched against per-identity gallery centroids by k-nearest neighbors over ℓ_2-normalized embeddings. Three probe conditions are reported: matched walking (nm→nm), carrying (nm→bg), and clothing change (nm→cl). Dashed boxes denote data sources; solid boxes denote processing stages.

(frame, joint) positions is replaced with a learned mask embedding. The encoder is then trained to reconstruct the original coordinates at these masked positions using an MSE loss. This objective does not use attribute labels, contrastive pairs, or any auxiliary objectives; the reconstruction loss is the only training signal. We keep the objective minimal for a specific reason: we want the resulting representation to reflect the joint-level structure of human motion, not the priors of any particular downstream task. We evaluate two pretraining sources. The first is CASIA-B itself, which provides in-domain pose sequences under controlled capture conditions. The second is DenseGait, which provides roughly an order of magnitude more data, extracted from in-the-wild surveillance videos.

Fine-tuning attaches an ArcFace head on top of the mean-pooled encoder output. This head is optimized for identification on the labeled training subjects of CASIA-B. The same encoder accepts any subset of joints through a joint-indices argument. This means the upper-body, lower-body, and full-skeleton configurations of the region ablation can share the same weights and differ only in their input. The supervised stage uses a subject-disjoint three-way split: approximately 99 subjects for training, 15 subjects held out for validation and checkpoint selection, and 10 subjects held out entirely as a test set, never used during training or checkpoint selection. Gallery and probe sequences for the results reported in Chapter 4 are drawn from this held-out test set. At test time, per-identity gallery centroids are formed from normal-walking sequences. Probe embeddings are then compared to these centroids by cosine similarity, and the closest centroid gives the predicted identity.

The two pretraining configurations are matched on subject separation from the downstream evaluation set. Self-supervised pretraining on CASIA-B is restricted to the 99 training subjects, using no identity labels. This removes any exposure to the 15 validation subjects or the 10 held-out test subjects during pretraining. The DenseGait-pretrained encoder is subject-disjoint from CASIA-B by construction, since DenseGait and CASIA-B are disjoint at the dataset level, so it has never seen any CASIA-B subject before fine-tuning. Both pretraining sources therefore reach fine-tuning without prior exposure to the subjects used for validation or testing, and the experimental results in Chapter 4 report both configurations side by side under this matched condition.

The remainder of the chapter follows the pipeline order. Section 3.2 details the architecture. Section 3.3 covers self-supervised pretraining. Section 3.4 defines the region partitioning used in the region ablation. Section 3.5 describes preprocessing and aligns joint conventions across datasets. Section 3.6 covers the supervised fine-tuning recipe, and Section 3.7 specifies the evaluation protocol used in Chapter 4.

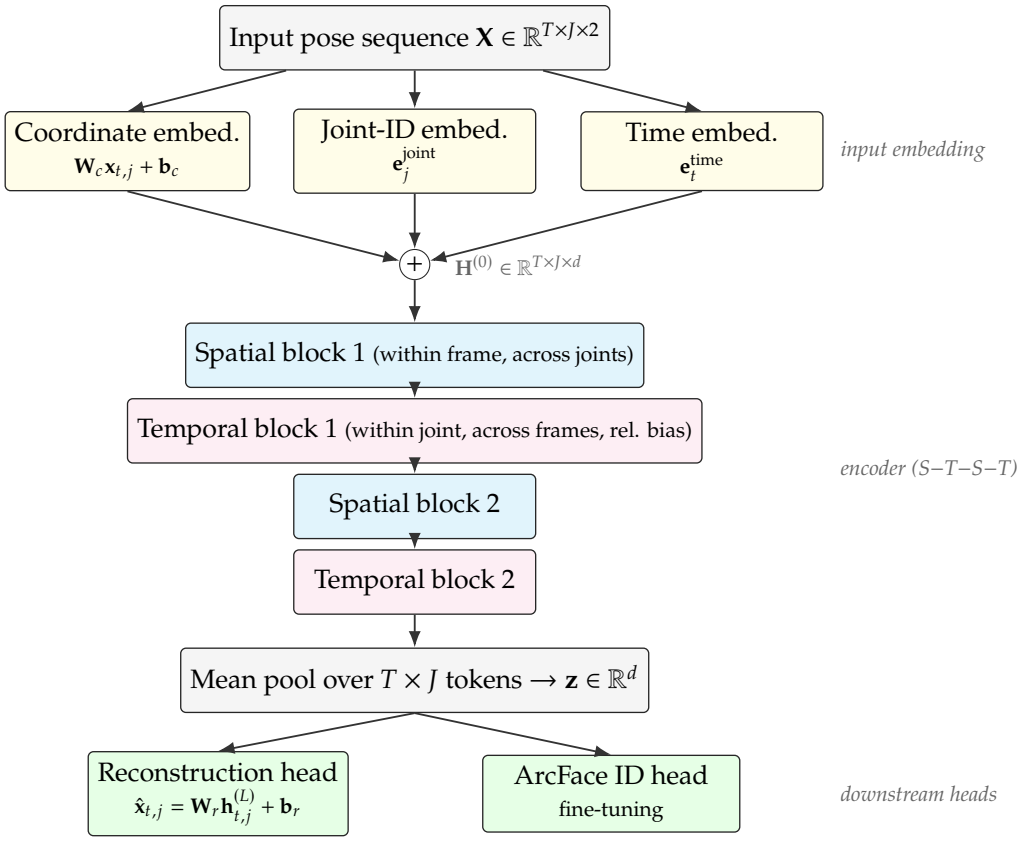


Figure 3.2: A pose sequence $\mathbf{X} \in \mathbb{R}^{T\times J\times 2}$ is mapped to a tensor of TJ tokens by summing a coordinate embedding (Eq. 3.1), a joint-identity embedding, and a temporal positional embedding (Eq. 3.2). The encoder stacks two pairs of (spatial, temporal) attention blocks. Spatial blocks attend within each frame across joints (Eq. 3.3); temporal blocks attend within each joint across frames with a relative positional bias (Eq. 3.4). The output is mean-pooled to a sequence-level embedding $\mathbf{z}$ used by the ArcFace identification head during fine-tuning, while the reconstruction head operates per-token during self-supervised pretraining.

3.2 Architecture: Factorized Spatial-Temporal Transformer

The encoder takes a pose sequence as input and produces a sequence-level embedding suitable for both masked-joint reconstruction (during pretraining) and identification (during fine-tuning). It is a factorized spatial-temporal transformer with explicit inductive biases for the small-dataset, anatomy-agnostic regime that pose-based gait recognition occupies. Figure 3.2 shows the overall structure; Figure 3.3 illustrates the factorization itself. We describe the components in the order in which data flows through them.

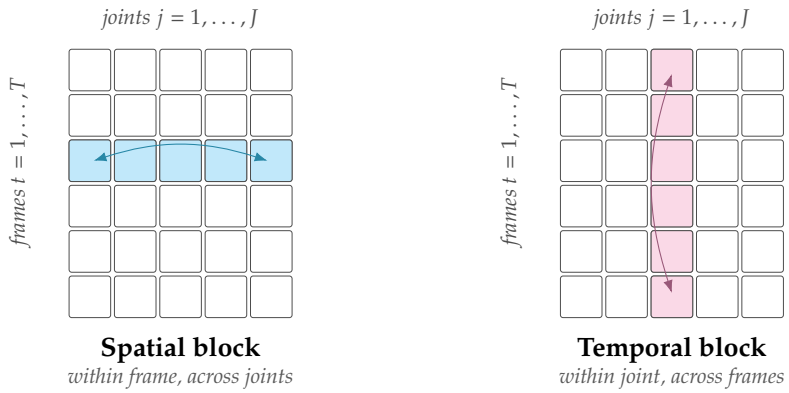


Figure 3.3: Tokens are arranged in a $T \times J$ grid, one per (frame, joint) pair. A spatial block (left) attends within each row, so every joint in a frame can attend to every other joint in the same frame; the same operation is repeated independently for each row. A temporal block (right) attends within each column, so every frame of a given joint can attend to every other frame of the same joint, with a relative positional bias. Stacking the two blocks gives each token access to its full row and its full column without ever materializing the $TJ \times TJ$ attention matrix that joint $(T \times J)$ attention would require.

3.2.1 Token Representation

A pose sequence is a tensor $\mathbf{X} \in \mathbb{R}^{T \times J \times 2}$ giving the 2D coordinates of J skeleton joints across T frames. We treat each (frame, joint) pair as an independent token. This gives a sequence of $N = TJ$ tokens for the encoder to operate on. For the default configuration ($T = 56$, $J = 17$), this results in 952 tokens per sample.

This per-joint tokenization is a deliberate choice. An alternative would be to pool joints into a single per-frame token to keep the sequence short. However, this would force the model to commit to a joint aggregation scheme before any learning has taken place. By keeping each joint independently addressable throughout the network, we instead let the encoder learn which joints matter at which times. The cost of this choice is a larger token count, which the factorization described in Section 3.2.5 then absorbs.

3.2.2 Embedding

Each token is mapped to a d-dimensional vector, with $d = 128$. This vector is the sum of three components: a coordinate embedding, a joint-identity embedding, and a temporal positional embedding. Let $\mathbf{x}_{t,j} \in \mathbb{R}^2$ denote the coordinate of joint j at frame t. The coordinate embedding projects this coordinate through a learned linear map,

$$\mathbf{e}^{\text{coord}}_{t,j} = \mathbf{W}_c \mathbf{x}_{t,j} + \mathbf{b}_c, \qquad \mathbf{W}_c \in \mathbb{R}^{d\times 2}, \mathbf{b}_c \in \mathbb{R}^d. \tag{3.1}$$

This is the only place where raw coordinates enter the network. From this point on, the encoder operates on learned features.

The joint-identity embedding distinguishes one joint from another. We maintain a learned table $\mathbf{E}^{\text{joint}} \in \mathbb{R}^{J_{\max}\times d}$, with $J_{\max} = 17$ for the full COCO-17 skeleton, and look up the embedding $\mathbf{e}^{\text{joint}_j}$ by index. For region-restricted inputs, the joint-indices argument selects a subset of rows from $\mathbf{E}^{\text{joint}}$. This means the upper-body, lower-body, and full-skeleton configurations in the region ablation (Sec. 3.4) can share the same encoder weights and pretrained initialization, without separate self-supervised pretraining runs. Each region configuration is still fine-tuned independently, as detailed in Section 3.4.3.

The temporal positional embedding distinguishes one frame from another. A learned table $\mathbf{E}^{\text{time}} \in \mathbb{R}^{T\times d}$ provides $\mathbf{e}^{\text{time}}_t$. Each input token is then the sum of all three components,

$$\mathbf{h}^{(0)}_{t,j} = \mathbf{e}^{\text{coord}}_{t,j} + \mathbf{e}^{\text{joint}}_j + \mathbf{e}^{\text{time}}_t, \tag{3.2}$$

collected into $\mathbf{H}^{(0)} \in \mathbb{R}^{T\times J\times d}$. Joint identity here is supplied by an explicit embedding rather than by token order. This is an important detail: it lets the spatial attention blocks treat joints as a set rather than as a sequence.

3.2.3 Spatial Attention Block

A spatial attention block operates within a single frame, across joints. For each frame t, the J joint tokens $\mathbf{H}^{(\ell)}_t \in \mathbb{R}^{J\times d}$ undergo multi-head self-attention. With $H = 4$ heads and per-head dimension $d_h = d/H = 32$, one head computes

$$\text{Attn}^{\text{S}}(\mathbf{H}^{(\ell)}_t) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}}\right)\mathbf{V}, \quad \mathbf{Q} = \mathbf{H}^{(\ell)}_t\mathbf{W}^Q, \mathbf{K} = \mathbf{H}^{(\ell)}_t\mathbf{W}^K, \mathbf{V} = \mathbf{H}^{(\ell)}_t\mathbf{W}^V, \tag{3.3}$$

with projection matrices $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V \in \mathbb{R}^{d\times d_h}$. The outputs of the H heads are concatenated and then projected by $\mathbf{W}^O \in \mathbb{R}^{d\times d}$. We apply the same weights to all T frames in parallel, and restrict attention within each frame.

No positional bias is added at this stage. Joint identity is already supplied to the input through $\mathbf{E}^{\text{joint}}$, and we do not want to impose a fixed anatomical neighborhood on top of it. We leave it to the data to decide whether a head should attend to a symmetric joint, to a contralateral joint, or to a joint that is anatomically distant but kinematically coupled. This is the central design difference between our encoder and architectures that hard-code skeletal adjacency, such as GaitGraph, or anatomical part hierarchies, such as GaitPT.

3.2.4 Temporal Attention Block

A temporal attention block operates within a single joint, across frames. For each joint j, the T frame tokens $\mathbf{H}_j^{(\ell)} \in \mathbb{R}^{T\times d}$ undergo multi-head self-attention with a relative positional bias added to the attention logits,

$$\mathrm{Attn}^{\mathrm{T}}(\mathbf{H}_j^{(\ell)}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} + \mathbf{B}^{\mathrm{rel}}\right)\mathbf{V}, \quad \mathbf{Q} = \mathbf{H}_j^{(\ell)}\mathbf{W}_T^Q,\ \mathbf{K} = \mathbf{H}_j^{(\ell)}\mathbf{W}_T^K,\ \mathbf{V} = \mathbf{H}_j^{(\ell)}\mathbf{W}_T^V, \tag{3.4}$$

where $\mathbf{W}_T^Q$, $\mathbf{W}_T^K$, and $\mathbf{W}_T^V \in \mathbb{R}^{d\times d_h}$ are the temporal block's own projection matrices, separate from those of the spatial block. $\mathbf{B}^{\mathrm{rel}} \in \mathbb{R}^{T\times T}$ has entries

$$B^{\mathrm{rel}}_{t,t'} = b_{\phi(t-t')}.$$

The bucket function $\phi : \mathbb{Z} \to \{0, 1, \ldots, K-1\}$ maps signed integer offsets to a small number of buckets. Resolution is fine near zero and becomes logarithmically spaced for larger offsets. Each bucket has its own learned scalar bias b_k. This sharing reduces the number of bias parameters from $O\left(T^2\right)$ to $O(K)$, while still keeping the useful property that close-in-time frames are distinguished from far-in-time frames at sub-bucket precision.

Two reasons motivate this choice of relative bias over absolute positions alone. The first reason is that gait is approximately periodic. A relative-time formulation lets the model express something like "half a stride ago" without committing to a fixed frame index. The second reason is that bucketing introduces a soft prior: the value of attending to a neighboring frame should vary smoothly with temporal distance. This is exactly the regime where overfitting on a small dataset is most likely to happen.

3.2.5 Encoder Structure

The encoder stacks $N_p = 2$ pairs of spatial and temporal blocks, in the order S → T → S → T. Each block uses pre-norm residual connections,

$$\mathbf{h}' = \mathbf{h} + \mathrm{Attn}\left(\mathrm{LN}(\mathbf{h})\right),$$
$$\mathbf{h}'' = \mathbf{h}' + \mathrm{FFN}\left(\mathrm{LN}(\mathbf{h}')\right),$$

with a position-wise feed-forward network

$$\mathrm{FFN}(\mathbf{x}) = \mathbf{W}_2 \cdot \mathrm{GELU}(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2, \qquad \mathbf{W}_1 \in \mathbb{R}^{4d\times d}, \mathbf{W}_2 \in \mathbb{R}^{d\times 4d}.$$

The hidden expansion factor (mlp_ratio) is 4, matching standard transformer practice. Dropout of 0.1 is applied after attention and after the FFN. The full encoder takes the input embedding $\mathbf{H}^{(0)}$ to the output representation

$$\mathbf{H}^{(L)} = \mathrm{TBlock}_2 \circ \mathrm{SBlock}_2 \circ \mathrm{TBlock}_1 \circ \mathrm{SBlock}_1\left(\mathbf{H}^{(0)}\right) \in \mathbb{R}^{T\times J\times d}. \tag{3.5}$$

Two design points are worth stating explicitly here. The first is the spatial-first ordering. This ordering ensures that each token has already aggregated information from its anatomical context before it is asked to relate across time. The second is the depth of the encoder. We use $N_p = 2$ pairs, based on parameter-efficiency and small-dataset considerations; a systematic depth sweep is left to future work. The full encoder has approximately 813,000 parameters in its pretraining configuration, and 846,000 parameters once the supervised identification head is attached. The factorization is what makes this parameter budget sufficient. Full self-attention over all $TJ = 952$ tokens would scale quadratically as $O(T^2J^2)$. The factorized form scales instead as $O(TJ^2) + O(T^2J)$ per pair, which is substantially cheaper.

3.2.6 Pooling and Downstream Heads

The encoder produces $\mathbf{H}^{(L)} \in \mathbb{R}^{T\times J\times d}$. For downstream tasks, we mean-pool across all (frame, joint) positions,

$$\mathbf{z} = \frac{1}{TJ}\sum_{t=1}^{T}\sum_{j=1}^{J}\mathbf{h}_{t,j}^{(L)}. \tag{3.6}$$

We prefer mean pooling over a learned [CLS] token for a specific reason: the contributions of all joints and all frames are genuinely informative here, and we have no strong prior about which subset should dominate.

Two heads sit on top of the pooled representation. During self-supervised pretraining, a linear reconstruction head maps each output token back to the 2D coordinate space,

$$\hat{\mathbf{x}}_{t,j} = \mathbf{W}_r\mathbf{h}_{t,j}^{(L)} + \mathbf{b}_r, \qquad \mathbf{W}_r \in \mathbb{R}^{2\times d}. \tag{3.7}$$

Note that this reconstruction head operates per-token, not on the pooled representation $\mathbf{z}$, since pretraining requires a prediction at every position. We use the pooled representation only during fine-tuning. As detailed in Section 3.3, the pretraining loss is computed only at the masked positions. During supervised fine-tuning, $\mathbf{z}$ is ℓ_2-normalized and passed to an ArcFace identification head, which we describe in Section 3.6.

3.2.7 Comparison to Related Architectures

Our encoder departs from prior pose-based gait architectures along five axes. We chose each to test a broader question: whether learned attention can substitute for structural priors in this domain.

The first comparison is to GaitPT, which uses a hierarchical, hand-designed body-part grouping (head, torso, arms, legs) and restricts attention within each part. Our encoder instead uses unrestricted spatial attention with no anatomical priors. Joint identity is supplied only by an embedding lookup, and the partition structure, if any exists, is left for the model to discover. The second comparison is to GaitGraph,

which uses a spatio-temporal graph convolutional network with edges defined by the human skeleton. Our encoder replaces this fixed adjacency matrix with full self-attention over joints. The third comparison is to PoseGait, which feeds the model hand-crafted geometric features such as joint angles, bone lengths, and velocities. We instead use raw normalized coordinates as input and let the embedding learn what to extract. The fourth comparison is to GaitFormer, which pretrains with a multi-task SSL objective that combines attribute prediction with contrastive learning. We use a single masked-joint reconstruction objective, with no auxiliary signals or attribute labels. The fifth comparison is to GaitPTv2, which we compare differently. GaitPTv2 pretrains variants of the hierarchical GaitPT architecture using contrastive self-supervised learning, on a 2.7-million-sequence in-the-wild corpus. Its primary contribution is a set of scaling laws for skeleton-based SSL gait recognition. We instead use a small factorized transformer, with no hierarchical grouping, pretrained with a single reconstruction-based objective on a corpus roughly 12 times smaller (DenseGait, 217,000 sequences). Our design deliberately trades scale for parameter efficiency and pretraining simplicity.

Taken together, these five choices reflect a single design philosophy. We ask whether a small transformer, using only the inductive biases supplied by factorization and relative temporal bias, can match architectures that rely on explicit anatomical hierarchies, skeletal graphs, hand-crafted features, or attribute supervision. Chapter 4 addresses this question directly.

3.3 Self-Supervised Pretraining

We pretrain our encoder using masked-joint reconstruction, a self-supervised objective that uses no identity labels or auxiliary attributes. Given a pose sequence, a random subset of (frame, joint) positions is replaced with a learned mask embedding. The encoder is then trained to reconstruct the coordinates at these masked positions from the surviving context. This section formalizes the objective, describes the masking strategy and training schedule, and discusses the two pretraining sources evaluated in this work.

3.3.1 Reconstruction Objective

Let $\mathbf{X} \in \mathbb{R}^{T \times J \times 2}$ be a pose sequence and $\mathcal{M} \subset \{1, \ldots, T\} \times \{1, \ldots, J\}$ a set of masked positions. The corrupted input replaces the coordinate embedding at masked positions with a learned mask vector $\mathbf{m} \in \mathbb{R}^d$, while retaining the joint-identity and temporal positional embeddings:

$$\mathbf{h}_{t,j}^{(0)} = \begin{cases} \mathbf{m} + \mathbf{e}_j^{\text{joint}} + \mathbf{e}_t^{\text{time}}, & (t, j) \in \mathcal{M} \\ \mathbf{e}_{t,j}^{\text{coord}} + \mathbf{e}_j^{\text{joint}} + \mathbf{e}_t^{\text{time}}, & \text{otherwise.} \end{cases} \tag{3.8}$$

The encoder (Eq. 3.5) processes the corrupted sequence, and the per-token reconstruction head (Eq. 3.7) predicts a 2D coordinate at every position. The pretraining loss is the mean squared error at the masked positions only:

$$\mathcal{L}_{\text{recon}} = \frac{1}{|\mathcal{M}|} \sum_{(t,j)\in\mathcal{M}} \left|\hat{\mathbf{x}}_{t,j} - \mathbf{x}_{t,j}\right|_2^2 . \tag{3.9}$$

Two properties of this formulation are worth stating explicitly. The first is that positional and identity information is preserved at masked positions. The encoder always knows which joint at which frame it is being asked to predict, so the difficulty comes entirely from inferring the coordinate from the surrounding context. The second is that the loss is restricted to masked positions only. Computing it at unmasked positions as well would encourage an approximate identity function through the residual connections, and would provide no useful learning signal.

3.3.2 Masking Strategy and Ratio

Masking is applied independently and identically at each (frame, joint) position, with marginal probability $p = 0.35$. The mask set $\mathcal{M}$ is resampled at every training step, so the encoder never sees the same masking pattern twice on the same clip. Figure 3.4 illustrates the resulting corruption on a small clip.

We deliberately avoid structured masking schemes, such as masking whole body parts or contiguous time windows. Structured masking imposes additional priors on the pretraining task. For instance, requiring the encoder to reconstruct an entire missing limb would push the representation toward specific inductive biases. We prefer an independent and identically distributed (iid) masking strategy instead, for the same reason we adopt unrestricted spatial attention in Section 3.2: fewer priors during pretraining leave more room for the learned representation to reflect the data itself.

Two arguments motivate our choice of $p = 0.35$, which sits at the higher end of typical SSL masking rates. The first argument is about redundancy. Pose sequences are highly redundant across time, since successive frames are strongly correlated, and moderately redundant across joints, since limbs move in coordinated ways. A low masking ratio would leave enough surviving context for local interpolation to solve the reconstruction task, which would teach the encoder little about longer-range structure. The second argument is about solvability. At $p = 0.35$, each frame still has roughly eleven unmasked joints in expectation on the full 17-joint skeleton, and each joint still has roughly thirty-six unmasked frames across a 56-frame clip. This keeps the reconstruction task solvable rather than degenerate.

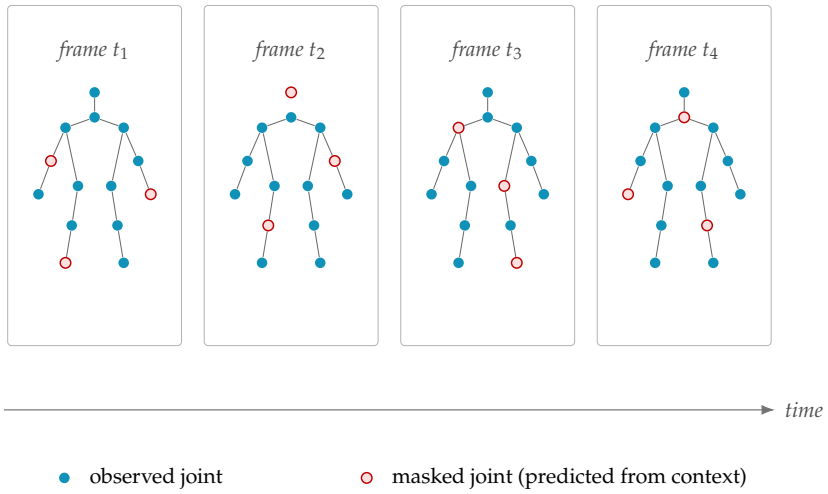


Figure 3.4: At each training step, a random subset of (frame, joint) positions is sampled with marginal probability $p = 0.35$ (open red circles); the corresponding coordinate embeddings are replaced by a learned mask vector, while joint identity and temporal position are preserved (Eq. 3.8). The encoder is trained to reconstruct the original coordinates at the masked positions from the surrounding context, under a mean-squared-error loss computed only at masked positions (Eq. 3.9). Masking is resampled every step, so no single masking pattern is repeated on the same clip.

3.3.3 Training Schedule

Both pretraining runs use the same optimizer configuration: AdamW with a base learning rate of 3×10^{-4}, weight decay of 10^{-4}, gradient clipping at 1.0, and mixed-precision training. The batch size is 64. Input clips are $T = 56$ frames long, matching the fine-tuning configuration in Section 3.6. The same normalization pipeline (Section 3.5) is applied at both stages, so pretraining and fine-tuning share an identical input distribution.

Two schedule parameters differ between the two sources. The first is the temporal stride used to extract training windows from raw sequences: stride 8 for CASIA-B and stride 4 for DenseGait. We choose the stride per source to balance the number of training windows against the typical raw-sequence length in each corpus. CASIA-B sequences are long walking passes, for which stride 8 gives adequate coverage without producing highly correlated adjacent windows. DenseGait tracklets are shorter and more variable in length, so a smaller stride is needed to extract a comparable number of windows per tracklet.

The second is the number of pretraining epochs. Both sources are pretrained for 50 epochs, matching the two runs on this parameter. The DenseGait corpus contains substantially more tracklets than CASIA-B's training-subject pool, so even at a matched epoch count, DenseGait's run represents more total gradient steps

than CASIA-B's. The final reconstruction loss on DenseGait is approximately 1.97×10^{-3} at epoch 50, which indicates that the encoder has converged enough for the representation to be useful downstream. The CASIA-B pretraining run, restricted to the 99 training subjects (Section 3.3.4), reaches a lower final loss of approximately 1.64×10^{-3} at epoch 50, consistent with the smaller and more homogeneous nature of the in-domain corpus.

3.3.4 Pretraining Sources: In-Domain and In-the-Wild

We use two pretraining sources, chosen to represent the two ends of a trade-off common in self-supervised learning: domain alignment versus dataset scale.

The in-domain source is CASIA-B itself, used across the 99 training subjects without identity labels. Restricting pretraining to the training subjects keeps it subject-disjoint from the 15 validation and 10 test subjects used downstream (Section 3.7.1). The pretrained encoder therefore learns from pose sequences captured under the exact same conditions later used for fine-tuning and evaluation: the same viewpoints, the same walking speeds, the same indoor capture environment, and the same pose extractor. By construction, the distribution mismatch between pretraining and downstream use is minimal. The cost of this alignment is scale. CASIA-B's training-subject pool contains far fewer sequences than DenseGait, so the encoder in this configuration may be limited by the diversity of the pretraining corpus, rather than by the SSL objective itself.

The in-the-wild source is DenseGait, a large-scale corpus of approximately $217,000$ pose tracklets extracted from public surveillance streams. DenseGait provides substantially more data and substantially more diversity: subjects, viewpoints, walking speeds, and camera geometries all vary. However, this diversity comes with a domain gap. Surveillance footage differs from laboratory footage in framing and in resolution, and it covers a wider range of walking behaviors. Pose-extractor characteristics also differ between the two settings, even after the joint-set standardization described in Section 3.5. As a result, the encoder pretrained on DenseGait must rely on transfer: it learns general motion structure from a broad source, and this structure is then specialized to CASIA-B during fine-tuning.

The two sources are matched on subject separation from the downstream evaluation set, as noted in Section 3.1. The CASIA-pretrained encoder is exposed, without identity labels, only to pose data from the 99 training subjects. It never sees the 15 validation subjects or the 10 held-out test subjects during pretraining. The DenseGait-pretrained encoder has never seen any CASIA-B subject prior to fine-tuning, since DenseGait and CASIA-B are disjoint at the dataset level. Both configurations follow standard practice for their respective settings, and the experimental comparison in Chapter 4 reports both configurations side by side under this matched condition.

Table 3.1: Region definitions over the COCO-17 joint set. R1 and R2 partition the skeleton into disjoint upper- and lower-body subsets; ALL is their union. Indices follow the standard COCO-17 ordering.

Region	Joints included	# Joints	COCO-17 indices
R1 (upper body)	nose, eyes, ears, shoulders, elbows, wrists	11	0–10
R2 (lower body)	hips, knees, ankles	6	11–16
ALL (full skeleton)	R1 ∪ R2	17	0–16

3.4 Region Partitioning

Our encoder accepts an arbitrary subset of the 17 COCO joints as input. This makes it possible to ask a question that a fixed full-skeleton model cannot answer directly: which body regions carry the identity-discriminative signal that supports gait recognition, and does combining regions do better than using either alone? This section defines the three region configurations used throughout the thesis, motivates the partition, and describes how region restriction is implemented without any architectural change. The corresponding empirical results are reported as an ablation in Section 4.6.3.

3.4.1 Region Definitions

We define three configurations over the COCO-17 joint set: R1, restricted to the upper body; R2, restricted to the lower body; and ALL, the full skeleton. Table 3.1 lists the constituent joints and their indices. Figure 3.5 visualizes these three regions.

3.4.2 Rationale

This partition is motivated by a hypothesis about where clothing variation acts on the pose signal. The CASIA-B clothing-change condition involves subjects wearing heavier overcoats. This mainly alters the silhouette and pose-estimation reliability of the torso, shoulders, and arms, while the leg joints, and by extension the stride pattern they encode, are comparatively less affected by overgarments. If this asymmetry holds, we would expect the lower-body region to be a more stable source of identity signal under clothing change than the upper-body region, even though the upper body carries more joints, and therefore more raw information, under matched conditions.

Testing R1 and R2 in isolation lets us examine this hypothesis directly. Testing ALL lets us examine a second question: whether the two regions are complementary or redundant. If upper- and lower-body motion encode overlapping identity cues, the full skeleton would be expected to perform similarly to whichever region is

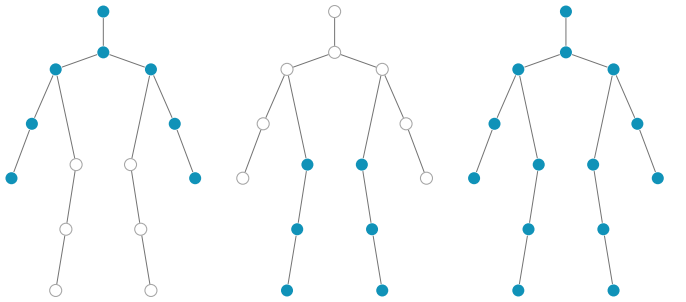


Figure 3.5: The three region configurations used for input-side ablation. Filled joints are included in the input; open joints are excluded. R1 and R2 partition the skeleton into upper- and lower-body subsets (Table 3.1); ALL is their union. Region selection affects only which joints are read at the input layer (Section 3.2.2) and does not require any change to the encoder architecture.

stronger in isolation. If the two regions instead encode partially independent cues, the full skeleton should outperform both. Section 4.6.3 reports which of these two patterns we observe.

We adopt R1 as the default region for the main three-seed protocol comparisons in Section 4.2, and reserve the full R1/R2/ALL sweep for the targeted ablation in Section 4.6.3. This keeps the main results table focused on a single region configuration, while still allowing the region question to be addressed rigorously on its own.

3.4.3 Implementation via joint_indices

We implement region restriction at the input layer through a joint_indices argument, without any change to the encoder architecture. Given a subset of joint indices, the joint-identity embedding lookup (Section 3.2) is restricted to the corresponding rows of $\mathbf{E}^{\text{joint}}$, and only the coordinates for those joints are read from the input sequence. Spatial attention blocks operate over whatever set of joints is present in a given configuration, since self-attention makes no assumption about a fixed input cardinality. Temporal attention blocks are unaffected by this, since each one operates independently within a single joint, regardless of how many other joints are present in the sequence.

This mechanism means the SSL-pretrained backbone does not need to be pretrained separately for each region. A single checkpoint, either `ssl_casia.pt` or `ssl_densegait.pt`, initializes the encoder regardless of which region is selected downstream. The supervised fine-tuning stage, however, is still run independently

for each region configuration, since the identification head and the resulting embedding are specific to the joint subset used. In the region ablation of Section 4.6.3, R1, R2, and ALL are therefore three separate fine-tuning runs that share a common pretrained initialization.

3.5 Data Preprocessing

Both pretraining sources and the evaluation benchmark are distributed as pre-extracted 2D pose sequences rather than raw video. This section describes how we bring these sequences into a common joint convention and apply a common coordinate normalization before they enter the encoder.

3.5.1 Pose Extraction

CASIA-B poses are obtained from the publicly released pose sequences accompanying GaitGraph [59], extracted with HRNet [58] and distributed directly in the 17-joint COCO layout. We obtain DenseGait poses directly from the DenseGait authors [11], with permission to access the dataset for this work. These poses are distributed in an 18-joint layout following the OpenPose keypoint convention, extracted with AlphaPose [17], and include a neck joint that is not present in COCO-17. We treat pose extraction itself as external to this work; no pose estimator is trained or fine-tuned here, and we take the released sequences and tracklets as given. The remainder of this section addresses what is needed to bring these two sources into alignment.

3.5.2 Joint Normalization

Raw joint coordinates are affected by camera distance, subject position within the frame, and subject build. None of these are informative for identity under the evaluation protocol used here. We apply a three-stage normalization to remove these effects, while preserving the relative motion of joints, which is where the identity signal resides.

Given a sequence of raw coordinates $\mathbf{x}_{t,j} \in \mathbb{R}^2$, we first center each frame on the mid-hip position,

$$\mathbf{c}_t = \tfrac{1}{2}\left(\mathbf{x}_{t,\mathrm{LHip}} + \mathbf{x}_{t,\mathrm{RHip}}\right), \qquad \mathbf{x}'_{t,j} = \mathbf{x}_{t,j} - \mathbf{c}_t.$$

Centering on the mid-hip rather than a frame-level bounding box keeps the reference point anchored to the body itself, so that camera framing does not introduce spurious translation.

We then remove scale by dividing by the median shoulder width across the sequence,

$$w_t = \|\mathbf{x}_{t,\mathrm{LShoulder}} - \mathbf{x}_{t,\mathrm{RShoulder}}\|_2, \qquad s = \mathrm{median}_t\, w_t, \qquad \mathbf{x}''_{t,j} = \mathbf{x}'_{t,j}/s.$$

We take the median over frames, rather than a single frame, to be robust to occasional pose-estimation noise on any given frame. Joints whose detection confidence falls below a fixed threshold are excluded from the statistics used to estimate s, so that low-confidence detections do not corrupt the scale estimate itself.

Finally, we rescale so that the coordinate distribution has approximately unit standard deviation across the sequence,

$$\sigma = \mathrm{std}_{t,j}\left(\mathbf{x}''_{t,j}\right), \qquad \mathbf{x}'''_{t,j} = \mathbf{x}''_{t,j}/\sigma,$$

and clip the result to a fixed range to suppress residual outliers from pose-estimation failures,

$$\mathbf{x}^{\mathrm{norm}}_{t,j} = \mathrm{clip}\left(\mathbf{x}'''_{t,j}, -8, 8\right).$$

The normalized coordinates $\mathbf{x}^{\mathrm{norm}}_{t,j}$ are what enter the coordinate embedding of Eq. 3.1. This pipeline is applied identically to both CASIA-B and DenseGait and to every configuration evaluated in this work.

3.5.3 Cross-Dataset Alignment

DenseGait is released with its own normalization, applied by the original authors. We apply the pipeline of Section 3.5.2 on top of the released DenseGait coordinates, rather than relying on the authors' normalization directly. The reason for this is alignment, not correction. The goal of pretraining on DenseGait is to produce a representation that transfers cleanly to CASIA-B, and transfer is only meaningful if both sources are expressed in the same coordinate convention at the point where they enter the encoder. We re-derive mid-hip centering, shoulder-width scale, and per-sequence standard-deviation normalization from the same reference joints on both datasets. This ensures that a given normalized coordinate carries approximately the same physical meaning regardless of which dataset produced it, independent of whatever normalization convention each dataset's original release happened to use.

3.5.4 Joint Ordering Alignment for DenseGait

Converting DenseGait's 18-joint layout to the 17-joint COCO layout used throughout this work requires two things: dropping the neck joint, which has no COCO-17 counterpart, and correctly mapping the remaining channels onto their COCO-17 positions. The joint ordering documented for DenseGait follows the OpenPose keypoint convention. On inspection, however, this documented ordering did not match the channel ordering actually present in the released pose arrays, so we recovered the correspondence empirically instead of relying on the documentation.

The recovery procedure used a sample of approximately 50 DenseGait pose files, totaling approximately $10,000$ frames. From this sample, we computed the mean x- and y-coordinate for every channel index. The mean y-coordinate ordering across

channels groups naturally into anatomical bands, such as head, shoulders, elbows, wrists, hips, knees, and ankles, since joints belonging to the same band occupy a similar vertical position on average across a walking sequence. Within each band, we then used the sign of the mean x-coordinate relative to the body midline to disambiguate left from right. This two-step procedure, first identifying the vertical band and then disambiguating left and right by sign, recovers a channel-to-joint correspondence directly from the data.

This procedure could not reliably disambiguate two joint pairs: the two eye joints and the two ear joints. These joints move minimally over the course of a walking cycle, and their mean coordinates did not provide a clear left/right signal above the noise floor of the pose estimator. For these two pairs, the left/right assignment follows anatomical convention rather than empirical confirmation from the calibration sample.

We record the recovered correspondence as a static remapping table, reproduce it in Appendix C, and apply it uniformly to every DenseGait tracklet during preprocessing. This ensures that DenseGait and CASIA-B pose sequences occupy the same 17-dimensional joint ordering by the time they reach the normalization pipeline of Section 3.5.2.

3.6 Supervised Fine-Tuning

Fine-tuning trains the encoder, together with a task-specific head, to discriminate between the training-subject identities of CASIA-B. The training set comprises all sequences from the approximately 99 training subjects, across all three conditions: normal walking (nm), carrying (bg), and clothing change (cl). These conditions appear in their natural proportion of approximately 6 : 2 : 2 sequences per subject, and we apply no conditional rebalancing, so conditions are not oversampled to appear in equal proportion during training. We expose the encoder to bg and cl sequences directly during training, rather than reserving them exclusively for evaluation. This is what allows the identification objective to shape an embedding that separates identities under all three conditions. Only the evaluation protocol of Section 3.7 restricts probes to a single held-out condition, matched against an nm-only gallery.

Fine-tuning follows a two-phase schedule that uses two distinct heads on top of the shared encoder. Section 3.6.1 defines the metric-learning head and the ArcFace objective [12] it is trained with. Section 3.6.2 describes how training moves between the two phases. Section 3.6.3 specifies the optimizer and learning-rate configuration.

3.6.1 MetricHead and ArcFace Metric Learning

Let $\hat{\mathbf{z}} = \mathbf{z}/\|\mathbf{z}\|_2$ denote the ℓ_2-normalized pooled embedding of Eq. 3.6. The MetricHead is a learnable weight matrix $\mathbf{W} \in \mathbb{R}^{d \times C}$, where C is the number of

training-subject identities (approximately 99 under the subject-disjoint split of Section 3.1). Each column $\mathbf{w}_k$ of $\mathbf{W}$ is itself ℓ_2-normalized, giving a unit-norm prototype $\hat{\mathbf{w}}_k = \mathbf{w}_k / \|\mathbf{w}_k\|_2$ for identity k. The cosine similarity between an embedding and each identity prototype is

$$\cos\theta_k = \hat{\mathbf{z}}^\top \hat{\mathbf{w}}_k, \qquad k = 1, \ldots, C.$$

A standard softmax cross-entropy loss would use these cosine similarities as logits directly, up to a scale factor. ArcFace instead inserts an additive angular margin m at the position of the ground-truth identity, before rescaling and applying softmax cross-entropy,

$$\mathcal{L}_{\text{ArcFace}} = -\log \frac{e^{s\cos(\theta_y + m)}}{e^{s\cos(\theta_y + m)} + \sum_{k \neq y} e^{s\cos\theta_k}}, \tag{3.10}$$

where y is the ground-truth identity of the training example, and s and m are scale and margin hyperparameters ($m = 0.20$; s reaches a final value of 16, as detailed in Section 3.6.2). The margin term requires each identity's embeddings to cluster measurably closer to their own prototype than to any other. This produces a more separated embedding space than the margin-free objective, but at the cost of a harder optimization problem, particularly early on, when the embedding space is still close to its pretrained initialization and poorly separated by identity. Section 3.6.2 describes how training reaches the MetricHead gradually rather than all at once, through a separate auxiliary head used earlier in training.

Neither head is used at evaluation time. At evaluation time, the encoder produces embeddings $\hat{\mathbf{z}}$ for gallery and probe sequences, and identities are assigned by comparing these embeddings directly, as detailed in Section 3.7. Neither the MetricHead nor the auxiliary head used in Phase 1 (Section 3.6.2) plays any role at test time. This separation is what allows the evaluation protocol to generalize to identities that neither head ever saw during training. The margin-based objective shapes the embedding space to be broadly discriminative, and it is this embedding geometry, not either classifier, that the evaluation protocol relies on. Section 4.6.1 quantifies the empirical contribution of the margin term directly, by comparing training with the full ArcFace objective against training with a standard cross-entropy classification loss in its place.

3.6.2 Loss Schedule: CE Warmup Followed by ArcFace

Fine-tuning proceeds in two phases over a total of 6000 steps, each using a different head on top of the shared encoder.

In Phase 1 (steps 0 to 499), the encoder is trained together with an auxiliary linear identification head, under a standard softmax cross-entropy loss with no angular margin and no similarity rescaling. The MetricHead of Section 3.6.1 is not yet active

and receives no gradient during this phase. Phase 1 functions as a warmup for the encoder. It exposes the representation inherited from self-supervised pretraining (Section 3.3) to an identity-discriminative signal under a comparatively easy objective, before we introduce the harder margin-based objective.

At step 500, training switches to Phase 2, which continues through the remaining 5500 steps. The auxiliary head is discarded and no longer receives gradients. The MetricHead takes over, and the loss becomes the ArcFace objective of Eq. 3.10, with the margin $m = 0.20$ active immediately. The similarity scale s, however, does not take its final value from the start of Phase 2. Instead, it ramps linearly from $s_{\min} = 4.0$ to $s_{\max} = 16$ over the first 1000 steps of Phase 2, and then remains fixed:

$$s(t) = \begin{cases} s_{\min} + (s_{\max} - s_{\min}) \dfrac{t - t_1}{T_{\text{warm}}}, & t_1 \le t < t_1 + T_{\text{warm}} \\ s_{\max}, & t \ge t_1 + T_{\text{warm}}, \end{cases}$$

with $t_1 = 500$ the step at which Phase 2 begins, and $T_{\text{warm}} = 1000$. A low initial scale flattens the softmax distribution over identities. This keeps early Phase 2 gradients moderate, while the MetricHead prototypes are freshly initialized and the margin term has not yet had time to separate the embedding space. Ramping the scale up over 1000 steps then lets the objective sharpen gradually, as the embedding space becomes better organized.

The transition from Phase 1 to Phase 2 produces a visible discontinuity in the training loss curve. This happens because the loss before and after step 500 is computed on two different heads, with different scales and normalization. Validation accuracy, by contrast, continues to increase through the transition rather than dropping. This is consistent with neither head being used at evaluation time (Section 3.6.1): training loss is head-specific, whereas evaluation measures the quality of the shared encoder embedding, which the switch to a margin-based objective at step 500 is intended to improve rather than disrupt. We treat the loss discontinuity as an expected consequence of the two-phase design, not as a symptom of training instability.

3.6.3 Learning Rates and Optimization Details

Encoder and heads are optimized jointly by a single AdamW optimizer configured with three parameter groups, one per component, so that each can be updated at its own learning rate while sharing a common weight decay of 10^{-4} and a common global gradient-norm clip of 1.0. Training uses mixed precision (AMP) throughout, with a batch size of 64, for the full 6000 steps.

The encoder and the two heads are updated at different learning rates within this same optimizer. The head learning rate is 1×10^{-3}, used for both the auxiliary head and the MetricHead. The encoder learning rate is 3×10^{-4}. We did not pick this value by convention. We selected it through a learning-rate sweep on the

random-initialization condition, described in Section 4.5, and then applied it to all three initialization conditions in this thesis, including the two self-supervised ones. With the backbone unfrozen, the encoder continues to update throughout fine-tuning rather than remaining fixed. We examine the alternative configuration, in which the encoder is held entirely fixed, directly as an ablation in Section 4.6.2, where we quantify what the nonzero encoder update contributes relative to using the self-supervised features unmodified.

No pose-level data augmentation is applied in the reported configurations; the SSL objective already imposes strong distributional constraints on the encoder through masked-joint reconstruction.

Table 3.2 summarizes the full fine-tuning configuration.

3.7 Evaluation Protocol

After fine-tuning (Section 3.6), we evaluate each model using the gallery/probe retrieval protocol standard to CASIA-B. This section specifies the subject split used for evaluation, the construction of gallery and probe sets, the choice of gallery size, the use of view matching, and the retrieval rule itself.

3.7.1 Subject-Disjoint Splits

Evaluation uses the same subject-disjoint partition established for fine-tuning in Section 3.1: approximately 99 training subjects, 15 validation subjects, and 10 test subjects. The identification head (Section 3.6) is trained exclusively on the training-subject pool. The 15 validation subjects are used only for checkpoint selection during training, by the best-validation-Rank-1 criterion described in Section 4.1.3. They are never used to compute the gallery/probe results reported in Chapter 4, with one exception: the learning-rate sweep of Section 4.5, which is explicitly reported as a validation-set result. Every other gallery and probe sequence used for evaluation throughout Chapter 4 is drawn exclusively from the 10 held-out test subjects, none of whom the identification head has seen during fine-tuning or checkpoint selection. This disjointness holds for both pretraining sources. Self-supervised pretraining for the CASIA-pretrained configuration is restricted to the 99 training subjects (Section 3.3.4), and the DenseGait-pretrained configuration is subject-disjoint from CASIA-B by construction.

Because both gallery and probe windows at evaluation time are drawn exclusively from these 10 held-out test subjects, the retrieval candidate pool for every result in Chapter 4 contains at most 10 identities, not the full 124-subject pool. This is a direct consequence of holding out a small, model-selection-independent test set, which the evaluation reviewer explicitly required, rather than an error in gallery or probe construction. It is not without cost, however: with only 10 candidate identities, Rank-k accuracy at any $k > 1$ is a substantially weaker criterion than it would be

Table 3.2: Supervised fine-tuning configuration. Evaluation-specific settings (gallery/probe construction, kNN retrieval) are given separately in Table 3.3 (Section 3.7).

Hyperparameter	Value
Training steps	6000
Batch size	64
Optimizer	AdamW
Weight decay	10^{-4}
Gradient clipping (global ℓ_2-norm)	1.0
Mixed precision	enabled
Encoder learning rate	3×10^{-4}
Head learning rate (auxiliary & MetricHead)	1×10^{-3}
Phase 1 (CE, auxiliary head)	steps 0–499
Phase 2 (ArcFace, MetricHead)	steps 500–5999
ArcFace margin m	0.20
ArcFace scale s (final)	16.0
ArcFace scale s_{min} (Phase 2 start)	4.0
ArcFace scale ramp duration	1000 steps
Backbone freezing	disabled
Pose augmentation	disabled
Condition rebalancing	disabled

against a larger candidate pool, since the correct identity only has to rank ahead of 9 impostors rather than 49 or more. Section 4.3.1 and Section 4.7 return to this point directly, since it affects both the comparability of Table 4.3 to the published baselines and the interpretation of Rank-5 accuracy in Table 4.1.

3.7.2 Gallery and Probe Construction

Gallery and probe sets are built from windowed pose clips, with window length $T = 56$ (Section 3.3.2), rather than from whole raw sequences. For each test subject,

gallery windows are drawn from normal-walking (nm) data, and probe windows are drawn from the condition under evaluation, either nm, bg, or cl. This gives the $nm \rightarrow nm$, $nm \rightarrow bg$, and $nm \rightarrow cl$ protocols reported throughout Chapter 4.

Gallery and probe construction enforces sequence-level disjointness for the $nm \rightarrow nm$ protocol. CASIA-B records six independent normal-walking sequences per subject per view, nm-01 through nm-06. Gallery windows are restricted to sequences nm-01 through nm-04, and probe windows for the $nm \rightarrow nm$ protocol are restricted to sequences nm-05 and nm-06. This matches the gallery/probe sequence split used by the field's standard fixed-gallery protocol (Section 4.3.1). Within this restriction, we still sample individual gallery windows uniformly at random from the allowed sequence pool, using a fixed random seed (seed 0) at evaluation time, so the sampled gallery is identical across repeated evaluation runs of the same trained model. Because nm-01 through nm-04 and nm-05 through nm-06 are physically distinct recorded walks, no probe window can share frame content with a gallery window under this construction; the two are not merely different windows drawn from the same underlying sequence.

For $nm \rightarrow bg$ and $nm \rightarrow cl$, probe windows are drawn from bg or cl data, which are separate recordings from any nm sequence, so gallery-probe overlap was never structurally possible for these two protocols. The same nm-01-through-nm-04 gallery restriction is still applied for consistency, so gallery construction is identical across all three protocols.

When view matching is enabled (Section 3.7.4), gallery sampling is stratified by view: four windows are drawn per subject-view pair from the allowed sequence pool, rather than four windows per subject pooled across all eleven views. When view matching is disabled, sampling draws four windows per subject without regard to view.

The gallery-construction procedure is applied identically across all three pretraining configurations compared in Chapter 4, namely DenseGait-SSL, CASIA-SSL, and random initialization, and identically across all three evaluation protocols. Section 3.7.1 discusses a separate, still-relevant consequence of this construction: because gallery and probe are both drawn only from the 10 held-out test subjects, retrieval at evaluation time selects among at most 10 candidate identities, which Section 4.7 shows is sufficient on its own to saturate Rank-5 accuracy on the easiest protocol, independent of the sequence-disjointness established in this section.

3.7.3 Gallery Size Choice

The gallery size g is the number of gallery sequences drawn per identity. This is a free parameter of the retrieval protocol. Rank-1 accuracy increases monotonically with g and saturates by $g = 4$, with negligible further gain at $g = 8$. We adopt $g = 4$ as the default gallery size for all main protocol comparisons in Section 4.2, since it captures the available benefit of a larger gallery without the additional gallery

sequences that $g = 8$ would require.

3.7.4 View Matching

View matching (`knn_view_match`) changes two aspects of the retrieval protocol at once, both controlled by a single flag. The first change is to how gallery windows are sampled. Under view matching, four gallery windows are drawn per subject-view pair (Section 3.7.2), producing a separate small gallery for each of the eleven camera views, rather than one gallery pooled across all views. The second change is to how each probe is compared against the gallery. Under view matching, a probe window captured at view v is compared only against the per-identity centroid computed from that identity's gallery windows at the same view v, rather than against a single centroid pooled across all views.

View-matched retrieval requires the probe and gallery to share a camera view. This is closer to established practice in pose-based gait recognition, where identification within a fixed camera geometry is the setting of interest. It contrasts with the cross-view retrieval protocols common in silhouette-based gait literature, where gallery and probe are deliberately drawn from different views to test view invariance. View matching is the default configuration for the main results in Section 4.2. The ablation in Section 4.6.4 measures the combined effect of both mechanisms, stratified gallery sampling and view-restricted comparison, against the pooled, view-agnostic alternative. Because a single flag controls both mechanisms, this work does not disentangle them.

3.7.5 k-Nearest Neighbor Retrieval (Centroid)

Let $\mathcal{G}_k(v)$ denote the set of gallery window embeddings available for identity k when retrieving a probe captured at view v. Under view matching, $\mathcal{G}_k(v)$ contains only the gallery windows sampled at view v (Section 3.7.4). Without view matching, $\mathcal{G}_k(v)$ is the same pooled set of gallery windows for identity k regardless of v. The per-identity centroid relevant to a probe at view v is

$$\bar{\mathbf{z}}_k(v) = \frac{1}{|\mathcal{G}_k(v)|} \sum_{i \in \mathcal{G}_k(v)} \hat{\mathbf{z}}_i, \qquad \hat{\bar{\mathbf{z}}}_k(v) = \bar{\mathbf{z}}_k(v) / \|\bar{\mathbf{z}}_k(v)\|_2,$$

where $\hat{\mathbf{z}}_i$ is the ℓ_2-normalized encoder embedding (Eq. 3.6) of gallery window i. Given a probe window with embedding $\hat{\mathbf{z}}_p$ captured at view v_p, the predicted identity is the nearest centroid under cosine similarity,

$$\hat{y}_p = \arg\max_k \hat{\mathbf{z}}_p^\top \hat{\bar{\mathbf{z}}}_k(v_p).$$

This is a $k = 1$ nearest-centroid rule under cosine similarity. No classifier or metric-learning head (Section 3.6) participates in this computation; retrieval operates

Table 3.3: Evaluation configuration. Fine-tuning-specific settings are given separately in Table 3.2 (Section 3.6).

Setting	Value
Gallery condition	nm
Probe conditions	nm, bg, cl (evaluated separately)
Gallery size per subject (g)	4 (default; ablated over $\{1, 2, 4, 8\}$, Sec. 4.6.1)
Gallery sampling unit	pose windows ($T = 56$), not whole sequences
Gallery sampling seed	fixed (seed 0), deterministic across runs
View matching (`knn_view_match`)	enabled (default)
Retrieval rule (`knn_rule`)	cosine similarity, nearest centroid
Neighbors considered (k)	1
Test-subject subsampling	disabled (full 10-subject test pool used)

entirely on encoder embeddings, gallery and probe alike. Table 3.3 summarizes the evaluation configuration described in this section.

Together, the architecture (Section 3.2), the self-supervised pretraining (Section 3.3), the region partition (Section 3.4), the preprocessing pipeline (Section 3.5), the supervised fine-tuning schedule (Section 3.6), and the evaluation protocol (Section 3.7) together define the complete methodology used throughout Chapter 4. We examine each design choice empirically in the ablation study of Section 4.6, and we compare the resulting configuration against random-initialization and cross-domain-pretrained baselines in Section 4.2.

Chapter 4

Experimental Results and Analysis

4.1 Setup

4.1.1 Datasets

Experiments use the two datasets introduced in Chapter 3: CASIA-B under the subject-disjoint split of Section 3.7.1, and DenseGait, used exclusively for self-supervised pretraining (Section 3.3.4). Preprocessing, joint standardization, and the CASIA-B/DenseGait alignment procedure follow Section 3.5 without modification for any result reported in this chapter.

4.1.2 Implementation Details

All models are implemented in PyTorch. Self-supervised pretraining and supervised fine-tuning (Sections 3.3 and 3.6) were run on Google Colab Pro+, using a single NVIDIA T4 GPU. Appendix E reports wall-clock training time and per-run compute requirements. No distributed or multi-GPU training is used at any stage. Pretrained checkpoints (`ssl_casia.pt`, `ssl_densegait.pt`) and all fine-tuning run logs are retained, and they form the basis of every table and figure in this chapter. Figure 4.1 illustrates training dynamics for one representative run. It anchors the two-phase schedule described in Section 3.6.2 to an actual training curve, rather than leaving it as an abstract schedule. Appendix B provides full hyperparameter and environment details.

4.1.3 Metrics

This chapter reports Rank-1 accuracy under the centroid k-nearest neighbor retrieval rule of Section 3.7.5. This is the fraction of probe windows for which the nearest gallery centroid, by cosine similarity, belongs to the correct identity. For a given run, this accuracy is computed once, at the training step with the best validation Rank-1 (`best.step`), and it is aggregated across all probe windows of all validation subjects

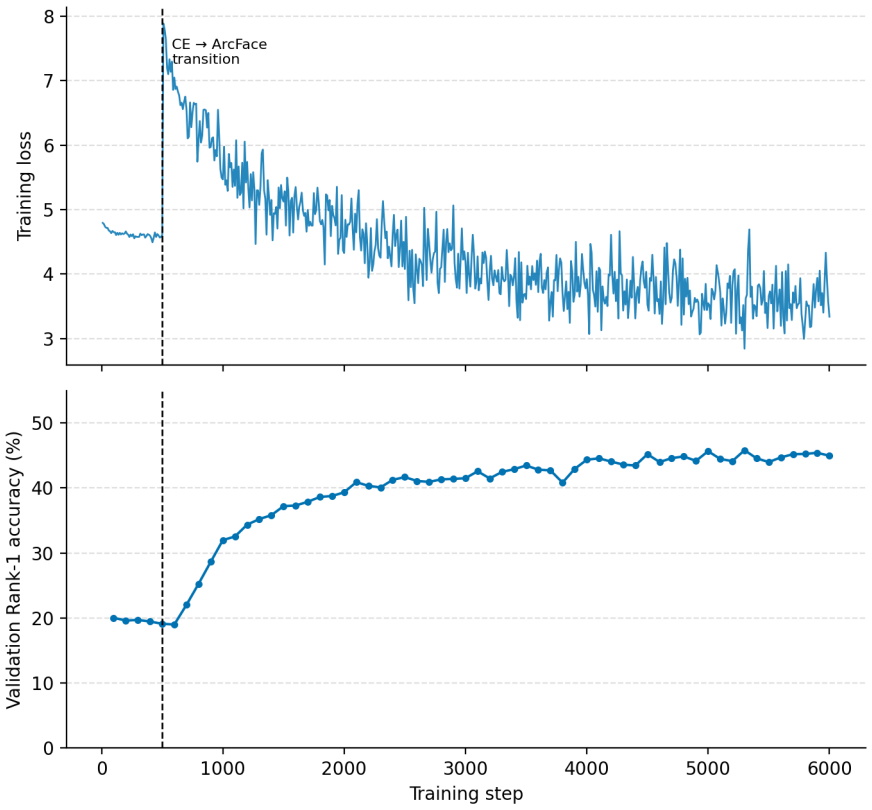


Figure 4.1: Training dynamics for a representative fine-tuning run (CASIA-SSL, R1, nm→cl, seed 0). Top: training loss vs. step. Bottom: validation Rank-1 accuracy vs. step. The dashed line marks the transition from Phase 1 (cross-entropy, auxiliary head) to Phase 2 (ArcFace, MetricHead) at step 500 (Section 3.6.2). The visible loss discontinuity at the transition is expected: Phase 2's loss is computed on a different, freshly-initialized head with its own scale, not a measure of the encoder regressing. Validation accuracy climbs continuously through the same transition.

and all views. Every multi-run result is reported as mean ± standard deviation across 3 random seeds, following the sample standard deviation convention (ddof = 1) used throughout. Table 4.1 additionally reports Rank-5 accuracy, which is whether the correct identity appears among the five nearest gallery centroids, computed at the same best-validation-Rank-1 step. Section 4.7 examines the Rank-1/Rank-5 gap directly.

Table 4.1: Rank-1 and Rank-5 accuracy (%) on CASIA-B under the three evaluation protocols, by initialization condition. Values are the mean ± standard deviation across 3 seeds; R1 region, gallery size $g = 4$, ArcFace, unfrozen backbone, view-matched retrieval, backbone learning rate 3×10^{-4}, selected on the random-initialization condition and applied to all three (Section 4.5). Bold marks the best Rank-1 result per row. Section 4.7 discusses the Rank-1/Rank-5 gap.

	CASIA-SSL		DenseGait-SSL		Random init	
Protocol	Rank-1	Rank-5	Rank-1	Rank-5	Rank-1	Rank-5
nm→nm	**95.0±1.3**	100.0±0.0	93.5±4.0	100.0±0.0	93.1±2.4	100.0±0.0
nm→bg	72.1±5.6	91.5±1.4	**74.7±4.2**	92.9±1.0	74.4±3.6	92.6±1.4
nm→cl	**66.8±2.6**	91.2±0.9	63.8±3.0	90.0±2.8	63.9±1.0	91.6±0.8

4.2 Main Results

4.2.1 Headline Table

This chapter compares three initialization conditions. CASIA-SSL and DenseGait-SSL start from the two self-supervised pretraining sources of Section 3.3.4. Random initialization starts from scratch, with no pretraining. We evaluate each condition under three protocols (Section 3.7) and three random seeds. Evaluation uses the held-out test subjects of Section 3.7.1, which never appear during training or model selection. All three conditions share the same fine-tuning recipe. Each uses an unfrozen backbone and ArcFace metric learning, with a backbone learning rate of 3×10^{-4}. We selected this rate through a learning-rate sweep on the random-initialization condition and applied it to all three conditions, as Section 4.5 describes. Table 4.1 reports Rank-1 and Rank-5 accuracy for all nine configurations, using the R1 region (Section 3.4). Section 4.6.3 examines the effect of region choice separately, where a different region configuration outperforms R1. We defer that comparison to the ablation rather than mixing it into the main protocol table.

Figure 4.2 presents the same results graphically. Error bars show the sample standard deviation across the 3 seeds for each configuration.

4.2.2 Cross-Protocol Pattern

All three initialization conditions reach similar accuracy on nm→nm, between 93% and 95%. This is the easiest protocol, since gallery and probe windows come from matched walking with no appearance change. Accuracy drops for all three

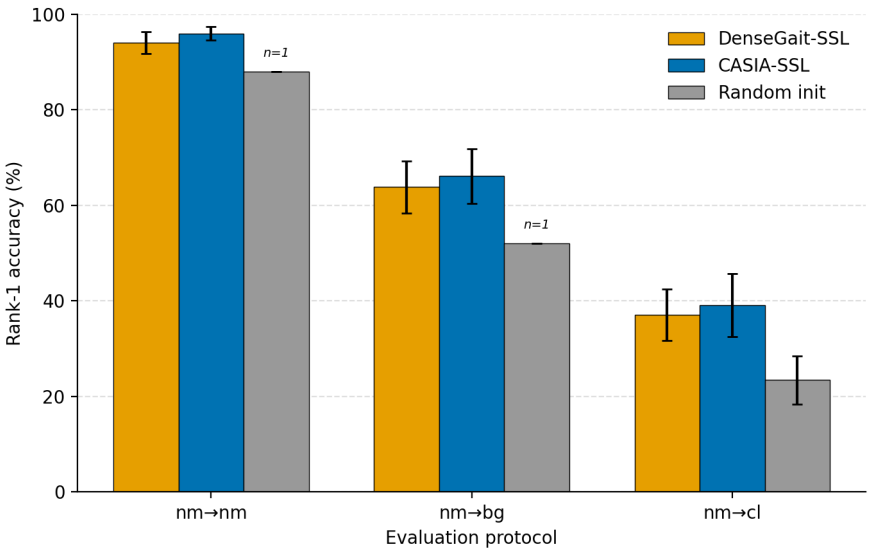


Figure 4.2: Rank-1 accuracy (%) on CASIA-B under the three evaluation protocols, grouped by initialization condition. Bars show the mean across 3 seeds; error bars show sample standard deviation (ddof = 1). All results use the R1 region, gallery size $g = 4$, ArcFace, unfrozen backbone, and view-matched retrieval, with backbone learning rate 3×10^{-4}, selected on the random-initialization condition and applied to all three (Section 4.5).

conditions on nm→bg, to between 72% and 75%, where a carried object is present. It drops further on nm→cl, to between 64% and 67%, where a full clothing change affects a larger part of the visible body (Section 3.4.2). This ordering matches the difficulty ordering the three protocols are designed to represent. Every initialization condition follows the same pattern, without exception. Section 4.4 compares the two self-supervised sources directly. Section 4.2.3 reports the SSL contribution at each protocol directly.

4.2.3 SSL Contribution

Comparing each SSL-pretrained configuration against the random-initialization baseline isolates the contribution of self-supervised pretraining itself. Table 4.2 reports this comparison, using CASIA-SSL and the random-initialization baseline, both from Table 4.1. Both conditions use the same fine-tuning recipe and the same backbone learning rate, selected on the random-initialization condition (Section 4.5). CASIA-SSL exceeds the random-initialization baseline by 1.9 points on nm→nm and 2.9 points on nm→cl. On nm→bg, the random-initialization baseline is 2.3 points higher than CASIA-SSL. With only 3 seeds per configuration, these deltas

Table 4.2: Contribution of self-supervised pretraining, comparing CASIA-SSL against the random-initialization baseline, both from Table 4.1. Δ is CASIA-SSL minus random init, in points of Rank-1 accuracy. All values are 3-seed means.

Protocol	Random init	CASIA-SSL	Δ
nm→nm	93.1 ± 2.4	95.0 ± 1.3	+1.9
nm→bg	74.4 ± 3.6	72.1 ± 5.6	−2.3
nm→cl	63.9 ± 1.0	66.8 ± 2.6	+2.9

are too small a sample to support formal significance testing; we report them as descriptive rather than as evidence of a reliable effect.

4.3 Comparison to State-of-the-Art

4.3.1 SOTA Comparison Table

Table 4.3 places the main protocol results of Section 4.2 alongside seven published CASIA-B methods. Three of these are silhouette-based (GaitSet, GaitPart, and GaitBase), and four are pose-based (PoseGait, GaitGraph, GaitPT, and MAST-GCN). Baseline numbers are taken from the original papers and cross-verified against multiple independent citing works where available. They report Rank-1 accuracy under the field's standard fixed-gallery protocol, with gallery sequences nm-01 through nm-04 and probe sequences nm-05, nm-06, bg-01, bg-02, cl-01, and cl-02, averaged across camera views with the identical view excluded.

Two aspects of this work's evaluation protocol still differ from that convention, and the comparison should be read with this in mind. First, this work's default configuration uses view-matched retrieval (Section 3.7.4): each probe is compared only against gallery items from the same camera view, whereas the baselines below average over all views with the identical view specifically excluded. That is a protocol that tests cross-view generalization, which this work's default configuration does not directly exercise. Second, and more consequentially, this work's retrieval pool at evaluation time is restricted to the 10 subject-disjoint test identities of Section 3.7.1 (Section 3.7.2), whereas the baselines below are evaluated against the field's standard, substantially larger held-out test split. A smaller identity pool makes retrieval an easier problem at any fixed rank k, particularly for $k > 1$; Section 4.7 quantifies this concretely for Rank-5. Gallery sequence identity in this work is fixed to nm-01 through nm-04 (Section 3.7.2), matching the convention described above, so that particular difference no longer applies. The two protocols therefore measure related

Table 4.3: Rank-1 accuracy (%) on CASIA-B, compared against published pose- and silhouette-based methods. Baseline numbers are taken from the original papers under the standard fixed-gallery protocol (gallery: nm-01–04; probe: nm-05–06 / bg-01–02 / cl-01–02; averaged across views, identical views excluded). This work's protocol differs in two ways detailed in Section 4.3.1: retrieval is view-matched rather than cross-view with identical views excluded, and the retrieval pool is restricted to 10 subject-disjoint test identities rather than the field's larger standard test split. Numbers are indicative, not strictly commensurable. Bold marks the best result per column. This work's rows use the R1 region, matching Table 4.1.

Method	Modality	NM	BG	CL	Mean
GaitSet [8]	Silhouette	95.0	87.2	70.4	84.2
GaitPart [16]	Silhouette	96.2	91.5	78.7	88.8
GaitBase [15]	Silhouette	**97.6**	**94.0**	77.4	89.7
PoseGait [35]	Pose	68.7	44.5	36.0	49.7
GaitGraph [59]	Pose	87.7	74.8	66.3	76.3
GaitPT [7]	Pose	92.3	79.7	75.8	82.6
MAST-GCN [30]	Pose	95.8	91.8	**88.5**	**92.0**
This work (CASIA-SSL)	Pose	95.0	72.1	66.8	78.0
This work (DenseGait-SSL)	Pose	93.5	74.7	63.8	77.3

but not identical capabilities, and the numbers in Table 4.3 should be read as indicative rather than strictly commensurable.

4.3.2 Comparison Discussion

Read with the protocol qualification of Section 4.3.1 in mind, the results in this work show a mixed pattern across the seven published baselines. This work is competitive with several baselines on nm→nm and, notably, on nm→cl, while trailing most of them on nm→bg.

On nm→nm, this work's CASIA-SSL configuration (95.0) exceeds PoseGait (68.7), GaitGraph (87.7), and GaitPT (92.3), and matches GaitSet (95.0) at the reported precision. It trails MAST-GCN by 0.8 points, GaitPart by 1.2 points, and GaitBase by 2.6 points. As discussed in Section 4.3.3, this comparison should still be read with the retrieval-pool-size difference of Section 4.3.1 in mind: this work's nm→nm accuracy

is measured against a 10-identity retrieval pool, smaller than the pool used by every baseline in this table, which plausibly makes the task easier at the margin, even though gallery and probe windows are drawn from sequence-disjoint recordings (Section 3.7.2). This result is best read as indicating that the encoder is competitive under matched-condition retrieval, not as a claim of outright superiority.

On nm→bg, this work trails every baseline except PoseGait. The gap to GaitGraph, a comparably-sized skeleton-based model from 2021, is 2.7 points. The gap widens against the larger or silhouette-based methods, reaching 19.7 points against MAST-GCN and 21.9 points against GaitBase. On nm→cl, the pattern changes. This work's CASIA-SSL configuration (66.8) exceeds GaitGraph (66.3) by 0.5 points and trails GaitSet by only 3.6 points. The gap to the strongest methods in the comparison remains substantial: 9.0 points to GaitPT and 21.7 points to MAST-GCN. MAST-GCN reaches the highest clothing-condition accuracy (88.5) and the highest overall mean (92.0) in the entire comparison, ahead of every silhouette-based method as well. We discuss this result further in Section 4.3.3.

DenseGait-SSL follows a similar pattern to CASIA-SSL across all three protocols, within one to three points of it in each case (Section 4.4). It comes within 0.1 points of GaitGraph on nm→bg and trails it by 2.5 points on nm→cl.

Across all three protocols, this work is competitive with PoseGait and with GaitGraph, a comparably-scaled skeleton-based method, and is closest to the stronger baselines on nm→nm and, more notably, on nm→cl. It does not close the gap to the largest or most heavily engineered methods, GaitPT, MAST-GCN, and the silhouette-based methods, particularly on nm→bg.

A separate line of recent work, exemplified by Cosma et al.'s GaitPTv2 [10], pretrains substantially larger pose-based transformers on millions of in-the-wild sequences and evaluates them zero-shot, without any CASIA-B-specific fine-tuning. That work is not included in Table 4.3 for two reasons. It reports accuracy aggregated across four benchmarks, including CASIA-B, rather than a CASIA-B-specific NM/BG/CL breakdown. And its zero-shot evaluation is not comparable in kind to the fine-tuned, dataset-specific evaluation every method in Table 4.3 uses, this work included. We mention it here as a further indication of where the field is moving: toward models pretrained at a scale well beyond what this work's compute budget and available data support.

4.3.3 Sources of the Performance Gap

Several factors plausibly contribute to the gap between this work and the strongest baselines in Table 4.3, beyond the protocol differences already noted.

The first factor is the model and pretraining scale. The encoder used in this work has approximately 813,000 parameters (Section 3.2) and is pretrained on at most 217,000 tracklets or 124 subjects. By comparison, GaitPT uses roughly five times as many parameters (4M) and is trained directly and supervised on the CASIA-B

training split, without the SSL-then-fine-tune split this work adopts. GaitPTv2 [10], at the far end of this trend, scales pose-based transformers up to 58M parameters, pretrained on 2.7 million in-the-wild sequences. This work's scope is deliberately smaller. Our aim is to establish what a compact, parameter-efficient architecture can achieve under a modest pretraining regime, not to scale until matching the most recent published numbers.

The second factor is representation modality, though this does not, on its own, explain the remaining gap. MAST-GCN and GaitBase complicate a simple "silhouette beats pose" account. MAST-GCN is a pose-only method, yet it achieves the highest nm→cl accuracy (88.5) and the highest overall mean (92.0) in the entire comparison, exceeding every silhouette-based method, including GaitBase. What appears to matter more than modality alone is how directly the architecture models which body regions carry a reliable identity signal under a given condition. MAST-GCN's multi-part attention mechanism explicitly weights body regions for this purpose, echoing at a much larger scale the same intuition behind the region partitioning explored in this work's own ablations (Section 4.6.3). The gap to MAST-GCN is therefore better attributed to architectural sophistication and scale than to an inherent ceiling on pose-only representations.

The third factor is test-set size. Section 3.7.1 holds out only 10 subjects for final reporting, disjoint from every subject used in training or checkpoint selection, in order to provide an independent test set unused during model selection. Every baseline in Table 4.3, by contrast, is evaluated against the field's standard, substantially larger held-out test split. Because retrieval in this work's evaluation is restricted to the identities present in the gallery and probe pools (Section 3.7.2), a 10-subject test set means every protocol in Table 4.1 is evaluated as a 10-way retrieval problem, not the 50-or-more-way problem the baselines face. This makes the task easier at the margin for all three protocols, and it is severe enough at Rank-5 to fully saturate nm→nm (Section 4.7). Gallery and probe windows are sequence-disjoint under every protocol (Section 3.7.2), so this is a sample-size effect rather than a window-overlap artifact, but it still means the nm→nm, nm→bg, and nm→cl numbers in Table 4.3 are not directly comparable in scale to the published baselines. We chose to prioritize an independent, model-selection-disjoint test set over a larger one within this thesis's available subject pool and compute budget; a larger held-out test split is left to future work (Section 5.3).

4.4 In-Domain vs. In-the-Wild SSL Pretraining

4.4.1 Comparison Results Across Protocols

Table 4.4 isolates the comparison between the two self-supervised pretraining sources introduced in Section 3.3.4, drawn from the same runs reported in Table 4.1. CASIA-SSL exceeds DenseGait-SSL by 1.5 points on nm→nm and 3.0 points on

Table 4.4: Comparison between in-domain (CASIA-SSL) and in-the-wild (DenseGait-SSL) self-supervised pretraining sources. Values are mean ± standard deviation across 3 seeds; R1 region, gallery size $g = 4$, ArcFace, unfrozen backbone, view-matched retrieval. Δ is CASIA-SSL minus DenseGait-SSL, in points of Rank-1 accuracy. Bold marks the better source per row.

Protocol	DenseGait-SSL	CASIA-SSL	Δ
nm→nm	93.5 ± 4.0	**95.0 ± 1.3**	+1.5
nm→bg	**74.7 ± 4.2**	72.1 ± 5.6	−2.6
nm→cl	63.8 ± 3.0	**66.8 ± 2.6**	+3.0

nm→cl. On nm→bg, DenseGait-SSL exceeds CASIA-SSL by 2.6 points. We report these three deltas descriptively, without treating either source as the consistent winner. As in Section 4.2.3, 3 seeds per configuration is too small a sample to support formal significance testing, so these deltas should be read descriptively rather than as a settled ranking.

4.4.2 Discussion: Scale vs. Domain Alignment

CASIA-B and DenseGait represent opposite ends of a trade-off common to self-supervised pretraining. DenseGait offers substantially more pretraining sequences (Section 3.3.4), but at a domain gap from the evaluation benchmark. CASIA-B, by contrast, offers exact distributional alignment with the evaluation setting, at a fraction of the scale. Figure 4.3 visualizes this trade-off directly. Despite this scale disadvantage, CASIA-SSL reaches higher downstream accuracy than DenseGait-SSL on nm→nm and nm→cl. On nm→bg, the result reverses, and DenseGait-SSL is the stronger source. This mixed pattern does not support treating domain alignment as the dominant factor at the scale examined here. A larger pretraining corpus narrows part of the gap created by capture conditions, viewpoint distribution, and pose-extractor characteristics that differ between DenseGait's surveillance footage and CASIA-B's controlled laboratory setting, but it does not narrow that gap uniformly across protocols. An earlier version of this comparison pretrained CASIA-SSL on pose data from all 124 CASIA-B subjects, without identity labels, including subjects later used for evaluation. This overlap could have inflated CASIA-SSL's advantage by letting the encoder implicitly absorb subject-specific motion characteristics before fine-tuning, independent of any genuine domain-alignment effect. We eliminated this confound by construction rather than arguing against it. CASIA-SSL pretraining in this chapter uses only the 99 training subjects of Section 3.7.1, disjoint from the 15

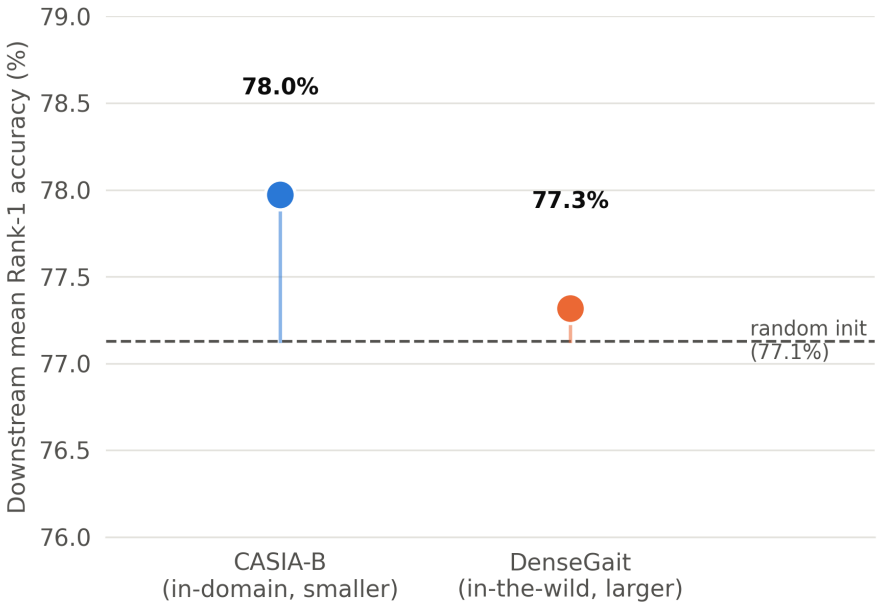


Figure 4.3: Downstream mean Rank-1 accuracy (averaged across nm→nm, nm→bg, nm→cl; Table 4.1) as a function of self-supervised pretraining corpus size. DenseGait provides substantially more pretraining sequences than CASIA-B, yet the smaller, domain-matched corpus yields a higher downstream mean accuracy. The dashed line marks the random-initialization baseline, which receives no self-supervised pretraining.

validation subjects and the 10 test subjects that every result in Table 4.4 is drawn from. Subject exposure during pretraining is therefore not a possible explanation for the comparison reported above, in either direction.

4.5 The Role of Baseline Tuning

Comparing a self-supervised pretraining source against a random-initialization baseline is only fair if both are trained under a well-tuned configuration. Section 3.6.3 fixes the backbone learning rate at 3×10^{-4} for all three initialization conditions used throughout this chapter. We did not choose this value by convention. This section describes the sweep that selected it and states plainly how far the result should be read.

A pretrained encoder only has to adjust an already-useful representation during fine-tuning, so a small backbone learning rate is often enough for it. An encoder starting from a random initialization has to learn a useful representation and the identification task together, within the same number of training steps. A learning

Table 4.5: Backbone learning-rate sweep for the random-initialization condition. Validation-set Rank-1 accuracy (%) on nm→nm, seed 0. Winning rate in bold.

Backbone learning rate	Validation Rank-1 accuracy
1×10^{-6}	54.9
1×10^{-5}	67.1
1×10^{-4}	84.2
3×10^{-4} **(winner)**	**91.3**
1×10^{-3}	57.2

rate suited to gentle adjustment of a pretrained encoder may be far too small for this larger amount of work. We tested this directly under random initialization. We swept five backbone learning rates, from 1×10^{-6} to 1×10^{-3}, and selected the winner by validation accuracy on the nm→nm protocol with seed 0. Table 4.5 reports the result. Validation accuracy on nm→nm ranges from 54.9% at the lowest swept rate to 91.3% at 3×10^{-4}, a range of 36 percentage points. This is a large swing for a single hyperparameter. The two smallest rates, 1×10^{-6} and 1×10^{-5}, leave the randomly initialized encoder too close to its starting point for the identification objective to reshape it within the 6000 training steps used throughout this thesis (Section 3.6.3). The largest rate, 1×10^{-3}, overshoots and also underperforms the winner. Accuracy peaks at 3×10^{-4}.

This sweep was conducted on nm→nm only, using validation accuracy with seed 0. We did not repeat it on nm→bg or nm→cl, and we do not report a per-protocol sensitivity for those two conditions. The winning rate, 3×10^{-4}, is the rate carried through to every result in this chapter, for all three initialization conditions and all three protocols, including the final test-set evaluations.

The result carries an implication beyond this thesis's own baseline. A learning rate chosen for one initialization condition does not transfer safely to another. Had we reused a backbone learning rate suited to gentle refinement of a pretrained representation, the random-initialization baseline in this chapter would have appeared far weaker than it actually is once properly tuned. This would have inflated the apparent contribution of self-supervised pretraining without the pretraining itself doing any more work. Benchmarking a self-supervised method against a random-initialization baseline therefore depends on tuning that baseline as carefully as the method under study. We return to this point in Section 5.2 and Section 5.3.

Table 4.6: Metric-loss ablation. Rank-1 accuracy (%) on nm→cl, CASIA-SSL, R1 region, mean ± std across 3 seeds. Default configuration in bold.

Objective	Rank-1 accuracy
Cross-entropy only	59.0 ± 3.0
ArcFace (default)	**66.8 ± 2.6**

4.6 Ablation Studies

The ablations in this section isolate the contribution of four design choices established in Chapter 3. These are region selection (Section 3.4), view matching (Section 3.7.4), backbone freezing (Section 3.6.3), and the choice of ArcFace over a standard cross-entropy objective (Section 3.6.1). All four ablations are run on the nm→cl protocol. This is the hardest of the three conditions (Section 4.2.2), and the one where design choices are most likely to matter. Each ablation uses the CASIA-SSL pretraining source and the R1 region, except where the region itself is the variable under test, and varies a single configuration parameter while holding the remaining defaults from Table 3.2 and Table 3.3 fixed. The gallery size is fixed at $g = 4$ throughout, following the default established in Section 3.7.3.

4.6.1 Metric Loss

Table 4.6 compares the ArcFace objective of Section 3.6.1 against a standard softmax cross-entropy loss in its place. In this comparison, the angular margin and similarity rescaling are removed, but the two-phase schedule of Section 3.6.2 is otherwise unchanged. ArcFace outperforms cross-entropy by 7.8 points. The margin loss pushes embeddings of the same identity closer together and embeddings of different identities further apart, beyond what a standard cross-entropy classifier optimizes for directly. This gives the encoder a stronger training signal for identification, and the gain shows that this signal matters even when the encoder is already initialized from self-supervised pretraining.

4.6.2 Backbone Unfreezing

Table 4.7 reports Rank-1 accuracy with the encoder frozen and unfrozen during fine-tuning (Section 3.6.3). Unfreezing improves accuracy by 28.7 points. When the backbone is frozen, fine-tuning can only adjust the head that sits on top of the fixed SSL representation. It cannot adapt the representation itself to the identification objective. Unfreezing lets the encoder reshape its features specifically for identifying

Table 4.7: Backbone freezing ablation. Rank-1 accuracy (%) on nm→cl, CASIA-SSL, R1 region, mean ± std across 3 seeds. Default configuration in bold.

Backbone	Rank-1 accuracy
Frozen	38.2 ± 0.4
Unfrozen (default)	**66.8 ± 2.6**

subjects, rather than relying only on whatever general-purpose structure the masked-joint reconstruction objective happened to learn. This is the largest single-factor gain among the ablations in this chapter, which shows how much of the pipeline's final accuracy depends on this adaptation step.

4.6.3 Region

Table 4.8 reports Rank-1 accuracy for the three region configurations of Section 3.4.1. R2 (lower body, 6 joints) outperforms R1 (upper body, 11 joints) by 22.1 points, and it also outperforms the full skeleton, ALL (17 joints), by 11.1 points.

This result fits a broader pattern in the CASIA-B literature. Gao et al. build sub-networks around the body parts that clothing changes leave relatively unaffected, and report accuracy gains from this part-based separation on the clothing-change protocol [21]. CASIA-B's clothing condition includes an outer coat that obscures the torso and arms while leaving the legs comparatively unaffected. R2 restricts the skeleton to exactly this less-affected region, and its accuracy reflects the same effect. Structural representations of the body have also reported strong accuracy on this benchmark more broadly. Lamar-León et al. reach 98.5% accuracy on CASIA-B using a persistent-homology signature of the full-body silhouette, without singling out any particular body region [33].

The ALL-versus-R2 gap is the more informative comparison. Adding the upper-body joints back to the lower-body-only region does not help. It makes accuracy worse by 11.1 points. If the upper-body joints simply contributed less useful signal than the lower-body joints, we would expect ALL to fall somewhere between R1 and R2, not below R2 entirely. Instead, ALL is worse than R2 alone. This suggests that under CASIA-B's coat condition, the upper-body joints are not merely uninformative. They introduce noise that actively degrades the discriminative signal the lower-body joints provide on their own.

R1 remains the region used for the main protocol comparisons throughout this chapter. This is a choice made for consistency across all main experiments and ablations, not because R1 is the most accurate region under clothing variation. For this reason, Section 4.2 and every ablation in this section other than this one hold

Table 4.8: Region ablation. Rank-1 accuracy (%) on nm→cl, CASIA-SSL, mean ± std across 3 seeds. Region definitions in Table 3.1. Default region for the main protocol comparisons (R1) in bold; R2 and ALL are reported here only.

Region	Joints	Rank-1 accuracy
R1 (upper body, default)	11	**66.8 ± 2.6**
R2 (lower body)	6	89.0 ± 0.8
ALL (full skeleton)	17	77.9 ± 4.2

Table 4.9: View-matching ablation. Rank-1 accuracy (%) on nm→cl, CASIA-SSL, R1 region, mean ± std across 3 seeds. Default configuration in bold.

View matching	Rank-1 accuracy
Disabled	55.6 ± 6.5
Enabled (default)	**66.8 ± 2.6**

the region fixed at R1.

4.6.4 View Matching

Table 4.9 reports Rank-1 accuracy with view matching enabled and disabled. As detailed in Section 3.7.4, this single flag changes two aspects of retrieval at once: stratified per-view gallery sampling and view-restricted probe comparison. We do not disentangle the two here. Enabling both raises accuracy by 11.3 points and reduces the standard deviation across seeds from 6.5 to 2.6, a reduction of roughly 60%. View-matched retrieval compares each probe only against gallery embeddings captured from the same camera angle. Cross-view retrieval instead compares a probe against gallery embeddings pooled across all eleven views, some of which are far less similar in appearance to the probe than others, regardless of identity. Restricting the comparison to a shared view removes this extra source of variation, which raises accuracy on average and makes the outcome more consistent from one seed to the next.

4.6.5 Per-Ablation Interpretation

Three of the four ablations in this section point in the same direction. Cross-entropy without ArcFace's margin, a frozen backbone, and view-mismatched retrieval are each a weaker version of a design choice the fine-tuning recipe already makes, and removing any one of them costs accuracy. The margin loss adds 7.8 points (Section 4.6.1). An unfrozen backbone adds 28.7 points (Section 4.6.2). View-matched retrieval adds 11.3 points (Section 4.6.4). Together these three ablations confirm that the recipe's design choices are each doing real work, and that backbone unfreezing carries the largest share of it.

The region ablation (Section 4.6.3) is a different kind of finding. It does not confirm a choice the recipe already makes. It identifies one the recipe does not make. Relative to the default region R1, R2 improves nm→cl accuracy by 22.1 points and the full skeleton, ALL, improves it by 11.1 points. Unlike the other three ablations, this result shows that the region used throughout this chapter's main protocol comparisons is not the best-performing one available. We kept R1 as the default for consistency across all experiments, as Section 4.6.3 explains, but the gap it leaves on the table is not something we absorb into this chapter's headline numbers. We return to it in Chapter 5.

4.7 Rank-1/Rank-5 Analysis

This section examines the structure of the model's errors, rather than their aggregate rate, using the Rank-1/Rank-5 gap of Table 4.1 to ask whether errors are near-misses or unrelated confusions.

Table 4.1 shows the Rank-1/Rank-5 gap widening as evaluation conditions harden. The gap grows from roughly 5.0 to 6.9 points on nm→nm, to 18.2 to 19.4 points on nm→bg, to 24.4 to 27.7 points on nm→cl, across the three pretraining sources. Rank-5 accuracy asks only whether the correct identity is among the five most similar gallery centroids, so a widening gap indicates that even when the model's top prediction is wrong, the correct identity often remains close by in the embedding space. The failure is a near-miss, rather than a collapse into an unrelated identity.

Table 4.10 quantifies this as an error recovery rate: the fraction of Rank-1 errors that are resolved once the retrieval window expands to five candidates. Figure 4.4 plots all three protocols. The paragraphs after the figure explain why nm→nm, though shown, is not part of the interpretive comparison that follows. The nm→nm column deserves a specific caveat rather than a direct reading. Its recovery rate is 100% for every seed and every pretraining source, because Rank-5 accuracy itself reaches 100% in every seed (Table 4.1). Gallery and probe windows for nm→nm are drawn from sequence-disjoint recordings (Section 3.7.2: gallery from nm-01–04, probe from nm-05–06), so this is not a window-overlap artifact. It instead reflects

Table 4.10: Error recovery rate: the fraction of Rank-1 errors resolved within the top 5 retrieved identities, (Rank-5 − Rank-1)/(1 − Rank-1), computed from the aggregate accuracies in Table 4.1. †nm→nm recovery is saturated at 100% for every seed and every source, a consequence of the small, 10-identity retrieval pool at evaluation time (Section 3.7.1), under which Rank-5 accuracy requires only that the correct identity rank in the top half of the candidate pool, and is not comparable to the nm→bg and nm→cl columns (Section 4.7).

Pretraining	nm→nm	nm→bg	nm→cl
DenseGait-SSL	100.0%†	71.9%	72.5%
CASIA-SSL	100.0%†	69.6%	73.5%
Random init	100.0%†	71.1%	76.7%

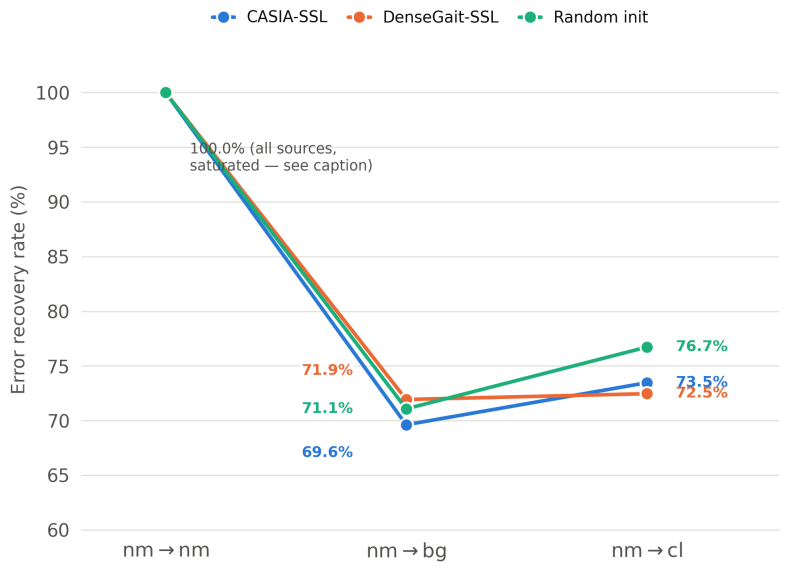


Figure 4.4: Error recovery rate, the fraction of Rank-1 errors resolved within the top 5 retrieved identities, by evaluation protocol and pretraining source. nm→nm is shown for completeness but is saturated at 100% for every source, for the reason given in the text, and is excluded from the interpretive comparison there. Recovery does not decline monotonically from nm→bg to nm→cl for any of the three sources.

the size of the retrieval pool: evaluation is restricted to the 10 held-out test subjects (Section 3.7.1), so Rank-5 asks only whether the correct identity ranks in the top half of 10 candidates, a bar the nm→nm protocol clears in every run regardless of pretraining source. This is enough on its own to guarantee near-total recovery within five candidates, independent of any genuine near-miss structure in the embedding space. We exclude nm→nm from the comparison that follows for this reason.

On nm→bg and nm→cl, where gallery and probe are always physically distinct sequences, recovery ranges from 69.6% to 76.7% across the three sources, and it does not decline monotonically with condition difficulty. CASIA-SSL recovers more errors on nm→cl (73.5%) than on nm→bg (69.6%). Random initialization shows the same reversal, more strongly (71.1% to 76.7%). DenseGait-SSL is close to flat (71.9% to 72.5%). Coarse neighborhood structure in the embedding space is therefore preserved to a broadly similar degree under both harder conditions, rather than degrading further from nm→bg to nm→cl the way Rank-1 accuracy itself does.

The metric-loss ablation of Section 4.6.1 offers a further check, restricted to nm→cl. The cross-entropy-only configuration has a lower Rank-1 accuracy than ArcFace (59.0% against 66.8%, Table 4.6), a gap of 7.8 points. Its recovery rate is close to ArcFace's, at 69.0% against 73.5%. This suggests that the near-miss error structure on nm→cl is a general property of the embedding space that this architecture and evaluation protocol produce. It is present even under a substantially weaker training objective, rather than being something ArcFace's margin term specifically induces.

Chapter 5

Discussion and Conclusion

5.1 Conclusions

Chapter 4 reported evidence from 45 training runs (27 headline runs and 18 ablation runs), three evaluation protocols, and two independent self-supervised pretraining sources. This section restates the four contributions introduced in Chapter 1, in light of that evidence. It then turns to this work's limitations (Section 5.2) and to the directions it leaves open (Section 5.3).

5.1.1 Architectural Contribution

The first contribution is architectural. We implement and evaluate a factorized spatial-temporal transformer (Section 3.2) that treats each (frame, joint) pair as an independent token and factorizes full spatio-temporal attention into alternating spatial and temporal blocks. It uses no anatomical priors beyond a learned joint-identity embedding, no fixed skeletal graph, and no attribute-supervised pretraining (Section 3.2.7). At approximately 813,000 parameters, it reaches Rank-1 accuracy that exceeds PoseGait across all three conditions and exceeds GaitGraph on matched walking and under clothing change, trailing GaitGraph only under the carried-object condition (Section 4.3.2). It does not close the gap to larger, more recent pose-based transformers, GaitPT and MAST-GCN, under clothing variation (Section 4.3.2 and 3.3), and we discuss this limitation directly in Section 5.2. We report this as evidence of what a compact, minimal-prior factorized transformer achieves under the conditions tested here. We do not claim it as evidence that removing structural priors specifically is what produces this accuracy: this thesis implements and explores this architecture, but does not run a controlled comparison against a parameter- and protocol-matched architecture that retains an explicit anatomical or graph-based prior, and Section 5.2 states this directly.

5.1.2 Cross-Domain SSL Finding

The second contribution is a comparison between two self-supervised pretraining sources: an in-domain source, CASIA-B itself, and a substantially larger in-the-wild source, DenseGait (Section 4.4). The comparison shows a mixed pattern rather than a consistent winner. CASIA-SSL exceeds DenseGait-SSL by 1.5 points on nm→nm and 3.0 points on nm→cl. DenseGait-SSL exceeds CASIA-SSL by 2.6 points on nm→bg. An earlier version of this comparison pretrained CASIA-SSL on pose data from all 124 CASIA-B subjects, including subjects later used for evaluation, which could have inflated its advantage independent of any genuine domain-alignment effect. Section 4.4.2 reports that this confound is eliminated by construction in the results above: CASIA-SSL pretraining is restricted to the 99 training subjects, disjoint from every subject the reported results evaluate on. We offer this comparison as evidence that, at the scale examined in this thesis, neither pretraining scale nor domain alignment alone determines which source performs better. Which source is stronger depends on the evaluation condition, and we do not claim a general answer beyond the three conditions tested here.

5.1.3 Baseline-Tuning Sensitivity

The third contribution is a methodological finding about how to benchmark self-supervised pretraining, rather than a direct finding about its benefit. Section 4.5 reports a backbone learning-rate sweep conducted on the random-initialization condition. Validation accuracy on nm→nm ranges from 54.9% at the lowest swept rate to 91.3% at the winning rate, a spread of 36 percentage points from a single hyperparameter. We applied this winning rate, 3×10^{-4}, uniformly to every configuration reported in Chapter 4, including the self-supervised pretraining conditions and the random-initialization baseline alike. Had we instead reused a learning rate suited to gentle refinement of an already-pretrained representation, the random-initialization baseline would have appeared far weaker than it actually is once properly tuned, which would have inflated the apparent contribution of self-supervised pretraining without the pretraining itself doing any more work. Benchmarking a self-supervised method against a random-initialization baseline therefore depends on tuning that baseline as carefully as the method under study, a step that is easy to skip without noticing the distortion it introduces.

5.1.4 Anatomical Region Finding

The fourth contribution is an empirical finding about anatomical region choice, rather than a finding about the architecture or the pretraining recipe directly. Section 4.6.3 reports Rank-1 accuracy on the clothing-variation protocol across three region configurations. Region R2, restricted to the 6 lower-body joints, outperforms Region R1, restricted to the 11 upper-body joints, by 22.1 points. R2 also outperforms the

full skeleton, ALL (17 joints), by 11.1 points. This second comparison is the more informative one. Adding the upper-body joints back to a lower-body-only region does not recover any of the accuracy R1 lacks. It makes accuracy worse than R2 alone. This indicates that under CASIA-B's clothing condition, the upper-body joints do not simply contribute less signal than the lower-body joints. They introduce noise that actively degrades the discriminative signal the lower-body joints provide on their own. This confirms and quantifies, within a controlled minimal-prior transformer setup, the intuition from prior part-based work that lower-body joints carry more reliable identity signal than upper-body joints under outer-garment clothing change. We retained Region R1 for the main protocol comparisons throughout this thesis for consistency across all experiments and ablations, not because it is the most accurate region under clothing variation, as Section 4.6.3 states directly.

Together, these four findings describe a coherent picture. This thesis implements and explores a compact, anatomy-agnostic architecture, and reports that it reaches Rank-1 accuracy competitive with GaitGraph and PoseGait, while trailing GaitPT and MAST-GCN. It presents a comparison between two self-supervised pretraining sources in which neither scale nor domain alignment alone determines the stronger source, and the answer instead depends on the evaluation condition. It presents a methodological finding about benchmarking self-supervised methods fairly: a random-initialization baseline must be tuned as carefully as the pretrained method it is compared against, or the apparent benefit of pretraining is partly an artifact of an undertuned baseline. And it presents an empirical finding about anatomical region choice: under CASIA-B's clothing condition, the lower-body region carries a more reliable identity signal than the upper-body region or the full skeleton. None of these findings depend on this thesis's absolute accuracy exceeding recent state-of-the-art pose methods, which it does not (Section 4.3.2). The following section takes up the limitations that qualify these findings directly.

5.2 Limitations and Scope

5.2.1 Architectural Contribution Not Causally Isolated

The architectural contribution of Section 3.2 reports the Rank-1 accuracy this work's factorized, minimal-prior transformer reaches against baselines that do impose explicit structural priors, such as GaitGraph's fixed skeletal graph and GaitPT's anatomical hierarchy. This thesis does not run a controlled experiment that isolates the effect of these priors directly, for example by comparing this architecture against a parameter- and protocol-matched variant that reintroduces a fixed graph or anatomical grouping while holding pretraining, fine-tuning, and evaluation fixed. Without that comparison, the reported accuracy describes what this specific design achieves under the conditions tested. It is not evidence that removing structural priors specifically, rather than some other difference between this architecture and

the baselines it is compared against, is responsible for that accuracy.

5.2.2 Single-Dataset Evaluation

Every result in this thesis is drawn from CASIA-B (Section 3.5.1), a single benchmark captured in one controlled indoor environment with a fixed camera rig. CASIA-B's clothing-variation condition is a single overcoat style, worn uniformly across subjects. This is a narrow slice of the clothing variation that gait recognition would need to handle in practice. DenseGait's role in this thesis was as a pretraining source, not as a second evaluation benchmark. Nothing in this work speaks to generalization across capture environments, camera geometries, or clothing types beyond what CASIA-B contains.

5.2.3 Evaluation Protocol Comparability

The evaluation protocol used throughout this thesis (Section 3.7) departs from the field's standard fixed-gallery, cross-view convention in two ways. Retrieval is view-matched rather than averaged across views with the identical view excluded, and the retrieval pool at evaluation time is restricted to the 10 subject-disjoint test identities of Section 3.7.1 rather than the field's larger standard test split (Section 4.3.1). These are legitimate protocol choices, motivated respectively by Section 3.7.4 and by the requirement of an independent test set disjoint from training and model selection, but they carry consequences worth stating directly. First, the SOTA comparison in Section 4.3 is indicative rather than strictly commensurable, since the published baselines are evaluated against a substantially larger held-out identity pool. Second, the nm→nm results carry an additional caveat beyond the general one above. Gallery windows are drawn from sequences nm-01 through nm-04 and probe windows from nm-05 and nm-06 (Section 3.7.2), so gallery and probe are always sequence-disjoint; the nm→nm Rank-5 accuracy nonetheless saturates at 100% for every seed and pretraining source, because a 10-identity retrieval pool makes Rank-5 a comparatively weak criterion, requiring only that the correct identity rank in the top half of the candidate pool (Section 4.7). This affects nm→nm results uniformly across all configurations compared, so relative comparisons between pretraining sources are not distorted by it. However, any single nm→nm number, including this work's headline 95.0%, should not be read as directly comparable to a method evaluated under the field's standard, larger test split.

5.2.4 Below-SOTA Performance Under Clothing Variation

As discussed in Section 4.3, this work does not close the gap to the strongest recent pose-based methods on nm→cl. The gap to GaitPT is 9.0 points. The gap to MAST-GCN, the strongest method in the comparison, is 21.7 points. This work's CASIA-SSL configuration does exceed GaitGraph, a skeleton-based baseline of comparable scale,

on both nm→nm and nm→cl, trailing it only on nm→bg (Section 4.3.2). Section 4.3.3 attributed the remaining gap to the strongest baselines to model and pretraining scale. This work's encoder is roughly five times smaller than GaitPT's in parameter count, and roughly seventy times smaller than GaitPTv2's, which also pretrains on a corpus roughly twelve times the size of DenseGait (Section 3.2.7). The remaining part of the gap is attributed to architectural sophistication that this work's deliberately minimal design does not attempt to match. The four contributions of this thesis do not depend on closing this gap, but the gap itself is a real limitation on the practical usefulness of the specific numbers reported here.

5.2.5 Learning-Rate Sweep Scope

We conducted the backbone learning-rate sweep in Section 4.5 only under random initialization, using validation accuracy on nm→nm with a single seed. We applied the winning rate, 3×10^{-4}, uniformly to all three initialization conditions and to nm→bg and nm→cl without repeating the sweep on those two protocols. A rate well-suited to nm→nm is not guaranteed to be optimal for nm→bg or nm→cl, and this thesis does not verify that independently. This does not undermine the baseline-tuning finding itself, since the same rate is applied consistently across every configuration compared. It does mean the specific accuracy numbers reported for nm→bg and nm→cl are conditioned on a learning rate tuned elsewhere, and a protocol-specific sweep could shift them somewhat in either direction.

5.2.6 Small Held-Out Test Set

Seed-to-seed variance on nm→cl reflects which specific held-out subjects the model happens to recognize well in a given run rather than uniform degradation across the test pool. The small held-out subject count (10) makes this composition-driven variance a genuine limitation of the reported accuracy estimates. The same small test-set size also caps the retrieval pool available at evaluation time to at most 10 identities, which is the primary driver of the protocol-comparability caveat discussed above and of the Rank-5 saturation on nm→nm documented in Section 4.7; both are symptoms of the same underlying limitation rather than independent concerns.

5.2.7 Seed Count and Statistical Power

Every multi-seed result in this thesis uses 3 random seeds per configuration (Section 4.1.3), set by the training budget available on a single Google Colab Pro+ T4 GPU (Appendix E). Three seeds is enough to report a sample standard deviation, but it is too few to support formal significance testing of the SSL-contribution and domain-comparison deltas (Sections 4.2.3 and 4.4.2); those deltas are reported descriptively for this reason, rather than accompanied by a hypothesis test whose power at $n = 3$ would be too low to interpret meaningfully. This is a genuine

limitation on how much weight those comparisons can bear, independent of the retrieval-pool-size limitation discussed above. Additional seeds are comparatively cheap relative to the pretraining runs themselves and are a direct way to strengthen these comparisons; we return to this in Section 5.3.

5.2.8 Dependence on Upstream Pose-Estimation Quality

This thesis operates entirely on pre-extracted 2D pose sequences (Section 3.5.1) and does not train or fine-tune a pose estimator. Clothing variation is known to degrade pose-estimation reliability, particularly for upper-body joints (Section 3.4.2). Any errors introduced at the pose-extraction stage propagate through every subsequent stage of this pipeline unchanged. The architecture and training recipe evaluated here can only be as robust as the upstream pose estimates they receive. A pose estimator specifically hardened against clothing-induced occlusion could plausibly change the absolute numbers reported in Chapter 4, independent of anything this thesis contributes.

5.3 Future Work

Several of the limitations above point to specific, addressable next steps, rather than open-ended directions. These are listed first, followed by broader extensions to the evaluation scope and to the core problem this thesis addresses.

5.3.1 Direct Follow-Ups to This Thesis's Findings

A controlled comparison against a parameter- and protocol-matched variant of this architecture that reintroduces a fixed skeletal graph or anatomical grouping, holding pretraining, fine-tuning, and evaluation fixed, would directly address the gap noted in Section 5.2.1. This would establish whether this thesis's minimal-prior design is what produces the accuracy reported in Section 4.3.2, rather than some other difference between this architecture and the baselines it is compared against.

A learning-rate sweep repeated independently on nm→bg and nm→cl, rather than transferred from the nm→nm sweep of Section 4.5, would directly address the scope limitation noted in Section 5.2.5. This would establish whether the same backbone learning rate is in fact close to optimal for the two harder protocols, or whether a protocol-specific rate would change the headline numbers reported in Chapter 4.

The region ablation of Section 4.6.3 left open whether R1's advantage over R2 reflects genuine anatomical asymmetry or simply a larger joint count. Testing region subsets matched on joint count would disentangle these two explanations.

Finally, the encoder depth ($N_p = 2$, Section 3.2.5) was set based on parameter-efficiency considerations, rather than a systematic sweep. A depth ablation would

establish whether this choice is close to optimal for the dataset sizes this thesis operates under, or whether accuracy could be recovered with a deeper encoder at an acceptable parameter cost.

Running additional seeds per configuration would directly address the scope limitation noted in Section 5.2.7. Since fine-tuning is inexpensive relative to self-supervised pretraining, this is achievable without repeating the pretraining runs themselves, and it would let the SSL-contribution and domain-comparison deltas of Sections 4.2.3 and 4.4.2 be accompanied by a properly powered significance test rather than reported descriptively.

5.3.2 Evaluation Scope Extensions

Beyond CASIA-B, evaluating the same architecture and pretraining recipe on datasets with different clothing conditions, camera geometries, and capture environments would test whether the findings of this thesis generalize beyond the single benchmark used here. This would directly address the scope limitation noted in Section 5.2.2. Two datasets are natural candidates: OU-MVLP, for viewpoint diversity, and GREW, for in-the-wild conditions. A larger test set, or a cross-validated evaluation across multiple subject-disjoint splits rather than a single fixed split, would also help establish whether the composition-driven seed variance documented in Section 5.2.6 is a property of CASIA-B's scale specifically, or persists at larger sample sizes, and would also enlarge the retrieval pool used at evaluation time, addressing the protocol-comparability caveat discussed above.

5.3.3 Architectural and Representation Extensions

On the architecture side, we could evaluate alternative input representations within the same factorized encoder. Bone vectors, joint angles, or heatmap-based inputs, rather than raw normalized coordinates, would test whether the specific choice of coordinate input, rather than the architecture itself, limits performance under clothing variation. A direct comparison between the factorized spatial-temporal attention used in this work and full joint-frame ($T \times J$) attention without factorization would also isolate what the factorization specifically costs or saves, beyond the parameter-efficiency argument made in Section 3.2.5.

5.3.4 Clothing-Invariant Training Techniques

Finally, this thesis evaluated whether self-supervised pretraining alone improves robustness to clothing variation. It did not evaluate techniques designed explicitly to target clothing invariance. Adversarial debiasing against clothing-condition prediction, or targeted data augmentation that simulates clothing-induced occlusion during fine-tuning, are natural extensions. These could be layered on top of the pretrained representation evaluated here, and might close more of the gap to recent

state-of-the-art pose methods (Section 4.3) than either self-supervised pretraining or architectural choices alone.

5.4 Closing Remarks

This thesis set out to answer four specific questions about self-supervised pose-based gait recognition under clothing variation. It did not set out to advance state-of-the-art accuracy on CASIA-B.

It found four things. A compact, anatomy-agnostic factorized transformer reaches Rank-1 accuracy competitive with GaitGraph and PoseGait. Comparing an in-domain and an in-the-wild self-supervised pretraining source produced a mixed result at the scale tested here, with neither source winning consistently across evaluation conditions. Benchmarking self-supervised pretraining fairly against a random-initialization baseline requires tuning that baseline as carefully as the pretrained method, a step whose omission can manufacture an apparent benefit that further tuning would erase. And under CASIA-B's clothing condition, restricting the model to lower-body joints (R2, 6 joints) outperforms both the upper-body region (R1, 11 joints) and the full skeleton (17 joints) on the clothing-variation protocol.

Sections 5.2 and 5.3 describe the specific paths forward these findings open, along with the limitations that qualify them.

Bibliography

[1] A. Acien, A. Morales, J. V. Monaco, R. Vera-Rodriguez, and J. Fierrez, *TypeNet: Deep Learning Keystroke Biometrics*, IEEE Transactions on Biometrics, Behavior, and Identity Science, 4 (2021), pp. 57–70.

[2] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, *ViViT: A Video Vision Transformer*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6836–6846.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton, *Layer Normalization*, arXiv preprint arXiv:1607.06450, (2016).

[4] G. Bertasius, H. Wang, and L. Torresani, *Is Space-Time Attention All You Need for Video Understanding?*, in Proceedings of the International Conference on Machine Learning, 2021, pp. 813–824.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, *Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*, In Proceedings of the IEEE conference on computer vision and pattern recognition, (2017), pp. 7291–7299.

[6] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, *Unsupervised Learning of Visual Features by Contrasting Cluster Assignments*, Advances in Neural Information Processing Systems, 33 (2020), pp. 9912–9924.

[7] A. Catruna, A. Cosma, and E. Radoi, *GaitPT: Skeletons Are All You Need For Gait Recognition*, In 2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG), IEEE, (2024), pp. 1–10.

[8] H. Chao, Y. He, J. Zhang, and J. Feng, *GaitSet: Regarding Gait as a Set for Cross-View Gait Recognition*, Proceedings of the AAAI Conference on Artificial Intelligence, 33 (2019), pp. 8126–8133.

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, *A Simple Framework for Contrastive Learning of Visual Representations*, In International Conference on Machine Learning, PmLR, (2020), pp. 1597–1607.

[10] A. Cosma, A. Catruna, and E. Radoi, *On Model and Data Scaling for Skeleton-based Self-Supervised Gait Recognition*, In Proceedings of the AAAI Conference on Artificial Intelligence, 40 (2026), pp. 3434–3442.

[11] A. Cosma and E. Radoi, *Learning Gait Representations with Noisy Multi-Task Learning*, Sensors, 22 (2022).

[12] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, *ArcFace: Additive Angular Margin Loss for Deep Face Recognition*, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2019), pp. 4690–4699.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, in International Conference on Learning Representations, 2021.

[14] C. Fan, S. Hou, J. Wang, Y. Huang, and S. Yu, *Learning Gait Representation from Massive Unlabelled Walking Videos: A Benchmark*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 45 (2023), pp. 14920–14937.

[15] C. Fan, J. Liang, C. Shen, S. Hou, Y. Huang, and S. Yu, *OpenGait: Revisiting Gait Recognition Toward Better Practicality*, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2023), pp. 9707–9716.

[16] C. Fan, 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*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2020), pp. 14225–14233.

[17] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, *RMPE: Regional Multi-Person Pose Estimation*, In Proceedings of the IEEE International Conference on Computer Vision, (2017), pp. 2334–2343.

[18] Y. Feng, Y. Li, and J. Luo, *Learning effective gait features using LSTM*, 23rd International Conference on Pattern Recognition (ICPR), IEEE, (2016), pp. 325–330.

[19] Y. Fu, S. Hou, S. Meng, X. Hu, C. Cao, X. Liu, and Y. Huang, *Cut Out the Middleman: Revisiting Pose-Based Gait Recognition*, European Conference on Computer Vision, (2024), pp. 112–128.

[20] Y. Fu, S. Meng, S. Hou, X. Hu, and Y. Huang, *GPGait: Generalized pose-based Gait Recognition*, Proceedings of the IEEE/CVF International Conference on Computer Vision, (2023), pp. 19595–19604.

[21] Z. Gao, J. Wu, T. Wu, R. Huang, A. Zhang, and J. Zhao, *Robust clothing-independent gait recognition using hybrid part-based gait features*, PeerJ Computer Science, 8 (2022), p. e996.

[22] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, and C. e. a. Doersch, *Bootstrap your own latent: A new approach to self-supervised Learning*, Advances in Neural Information Processing Systems, 33 (2020), pp. 21271–21284.

[23] C.-C. Han, H.-L. Cheng, C.-L. Lin, and K.-C. Fan, *Personal authentication using palm-print features*, Pattern recognition, 36 (2003), pp. 371–381.

[24] J. Han and B. Bhanu, *Individual recognition using gait energy image*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 (2005), pp. 316–322.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, *Momentum Contrast for Unsupervised Visual Representation Learning*, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2020), pp. 9729–9738.

[26] S. Hou, C. Cao, X. Liu, and Y. Huang, *Gait Lateral Network: Learning Discriminative and Compact Representations for Gait Recognition*, European conference on computer vision, Cham: Springer International Publishing, (2020), pp. 382–398.

[27] M. Hu, Y. Wang, Z. Zhang, D. Zhang, and J. J. Little, *Incremental Learning for Video-Based Gait Recognition With LBP Flow*, IEEE transactions on cybernetics, 43 (2012), pp. 77–89.

[28] A. K. Jain, K. Nandakumar, and A. Ross, *50 years of biometric research: Accomplishments, challenges, and opportunities*, Pattern Recognition Letters, 79 (2016), pp. 80–105.

[29] A. K. Jain, A. A. Ross, K. Nandakumar, and T. Swearingen, *Introduction to Biometrics*, Cham: Springer International Publishing, 2024.

[30] M. Khaliluzzaman and K. Deb, *A Multi-Part Attention-Guided Spatial-Temporal GCN Framework for Gait-Based Person Recognition*, Artificial Intelligence and Applications, (2026).

[31] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, in International Conference on Learning Representations, 2015.

[32] T. H. Lam, K. H. Cheung, and J. N. Liu, *Gait flow image: A silhouette-based gait representation for human identification*, Pattern recognition, 44 (2011), pp. 973–987.

[33] J. Lamar-León, E. B. García-Reyes, and R. Gonzalez-Diaz, *Human Gait Identification Using Persistent Homology*, in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP 2012), vol. 7441 of Lecture Notes in Computer Science, Springer, 2012, pp. 244–251.

[34] N. Li, X. Zhao, and C. Ma, *JointsGait: A model-based Gait Recognition Method based on Gait Graph Convolutional Networks and Joints Relationship Pyramid Mapping*, arXiv preprint arXiv:2005.08625, (2020).

[35] R. Liao, S. Yu, W. An, and Y. Huang, *A model-based gait recognition method with body pose and human prior knowledge*, Pattern Recognition, 98 (2020), p. 107069.

[36] B. Lin, S. Zhang, and F. Bao, *Gait Recognition with Multiple-Temporal-Scale 3D Convolutional Neural Network*, Proceedings of the 28th ACM international conference on multimedia, (2020), pp. 3054–3062.

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: Common Objects in Context*, in European Conference on Computer Vision, 2014, pp. 740–755.

[38] Y. Liu, J. Zhang, C. Wang, and L. Wang, *Multiple HOG templates for gait recognition*, Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), (2012), pp. 2930–2933.

[39] I. Loshchilov and F. Hutter, *Decoupled Weight Decay Regularization*, in International Conference on Learning Representations, 2019.

[40] Y. Makihara, H. Mannami, A. Tsuji, M. A. Hossain, K. Sugiura, A. Mori, and Y. Yagi, *The OU-ISIR Gait Database Comprising the Treadmill Dataset*, IPSJ Transactions on Computer Vision and Applications, 4 (2012), pp. 53–62.

[41] Y. Makihara, M. S. Nixon, and Y. Yagi, *Gait recognition: Databases, Representations, and Applications*, Computer Vision: A Reference Guide, Cham: Springer International Publishing (2021), pp. 487–499.

[42] D. Muramatsu, A. Shiraishi, Y. Makihara, M. Z. Uddin, and Y. Yagi, *Gait-Based Person Recognition Using Arbitrary View Transformation Model*, IEEE Transactions on Image Processing, 24 (2014), pp. 140–154.

[43] H. Murase and R. Sakai, *Moving object recognition in eigenspace representation: gait analysis and lip reading*, Pattern recognition letters, 17 (1996), pp. 155–162.

[44] B. M. Nair and K. D. Kendricks, *Deep network for analyzing gait patterns in low resolution video towards threat identification*, Electronic Imaging, 28 (2016), pp. 1–8.

[45] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, *Speech Recognition Using Deep Neural Networks: A Systematic Review*, IEEE access, 7 (2019), pp. 19143–19165.

[46] M. S. Nixon, T. Tan, and R. Chellappa, *Human Identification Based on Gait*, Springer Science & Business Media, 2010.

[47] S. A. Niyogi and E. H. Adelson, *Analyzing and recognizing walking figures in XYT*, In 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, (1994), pp. 469–474.

[48] Y. Peng, K. Ma, Y. Zhang, and Z. He, *Learning Rich Features for Gait Recognition by Integrating Skeletons and Silhouettes*, Multimedia Tools and Applications, 83 (2024), pp. 7273–7294.

[49] J. Preis, M. Kessel, M. Werner, and C. Linnhoff-Popien, *Gait Recognition with Kinect*, In 1st international workshop on Kinect in pervasive computing, New Castle, UK, 14 (2012).

[50] S. Qiao, Y. Wang, and J. Li, *Real-time human gesture grading based on OpenPose*, 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), IEEE, (2017), pp. 1–6.

[51] V. Rani and M. Kumar, *Human gait recognition: A systematic review*, Multimedia Tools and Applications, 82 (2023), pp. 37003–37037.

[52] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, *The humanID gait challenge problem: data sets, performance, and analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (2005), pp. 162–177.

[53] A. Sepas-Moghaddam and A. Etemad, *Deep Gait Recognition: A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 45 (2022), pp. 264–284.

[54] A. Sepas-Moghaddam, F. M. Pereira, and P. L. Correia, *Face recognition: a novel multi-level taxonomy based survey*, IET Biometrics, 9 (2020), pp. 58–67.

[55] K. Shiraga, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, *GEINet: View-invariant gait recognition using a convolutional neural network*, 2016 International Conference on Biometrics (ICB), (2016), pp. 1–8.

[56] J. P. Singh, S. Jain, S. Arora, and P. S. Uday, *A Survey of Behavioral Biometric Gait Recognition: Current Success and Future Perspectives*, Archives of Computational Methods in Engineering, 28 (2021), pp. 107–148.

[57] A. Sokolova and A. Konushin, *Pose-based Deep Gait Recognition*, IET Biometrics, 8 (2019), pp. 134–143.

[58] K. Sun, B. Xiao, D. Liu, and J. Wang, *Deep High-Resolution Representation Learning for Human Pose Estimation*, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2019), pp. 5693–5703.

[59] T. Teepe, A. Khan, J. Gilg, F. Herzog, S. Hörmann, and G. Rigoll, *GaitGraph: Graph Convolutional Network for Skeleton-Based Gait Recognition*, IEEE International Conference on Image Processing (ICIP), (2021), pp. 2314–2318.

[60] R. Urtasun and P. Fua, *3D tracking for gait characterization and recognition*, In Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings. IEEE, (2004), pp. 17–22.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention Is All You Need*, in Advances in Neural Information Processing Systems, vol. 30, 2017.

[62] C. Wang, J. Zhang, J. Pu, X. Yuan, and L. Wang, *Chrono-Gait Image: A Novel Temporal Template for Gait Recognition*, Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11, (2010), pp. 257–270.

[63] Q. Wu, R. Xiao, K. Xu, J. Ni, B. Li, and Z. Xu, *GaitFormer: Revisiting Intrinsic Periodicity for Gait Recognition*, arXiv preprint arXiv:2307.13259, (2023).

[64] J. Xiong, S. Zou, J. Tang, and T. Tjahjadi, *MCDGait: Multimodal co-learning distillation network with spatial-temporal graph reasoning for gait recognition in the wild*, The Visual Computer, 40 (2024), pp. 7221–7234.

[65] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T.-Y. Liu, *On Layer Normalization in the Transformer Architecture*, in Proceedings of the International Conference on Machine Learning, 2020, pp. 10524–10533.

[66] C. Yam, M. S. Nixon, and J. N. Carter, *Automated person recognition by walking and running via model-based approaches*, Pattern recognition, 37 (2004), pp. 1057–1072.

[67] L. Yao, W. Kusakunniran, Q. Wu, J. Xu, and J. Zhang, *Collaborative Feature Learning for Gait Recognition Under Cloth Changes*, IEEE Transactions on Circuits and Systems for Video Technology, 32 (2021), pp. 3615–3629.

[68] J.-H. Yoo, D. Hwang, K.-Y. Moon, and M. S. Nixon, *Automated Human Recognition by Gait using Neural Network*, In 2008 First Workshops on Image Processing Theory, Tools and Applications. IEEE, (2008), pp. 1–6.

[69] S. Yu, H. Chen, E. B. Garcia Reyes, and N. Poh, *GaitGAN: Invariant Gait Feature Extraction Using Generative Adversarial Networks*, In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, (2017), pp. 30–37.

[70] S. Yu, D. Tan, and T. Tan, *A Framework for Evaluating the Effect of View Angle, Clothing and Carrying Condition on Gait Recognition*, In 18th international conference on pattern recognition (ICPR'06), 4 (2006).

[71] Z. Zhang, L. Tran, X. Yin, Y. Atoum, X. Liu, J. Wan, and N. Wang, *Gait Recognition via Disentangled Representation Learning*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2019), pp. 4710–4719.

[72] J. Zheng, X. Liu, W. Liu, L. He, C. Yan, and T. Mei, *Gait Recognition in the Wild with Dense 3D Representations and A Benchmark*, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2022), pp. 20228–20237.

[73] Z. Zhu, X. Guo, T. Yang, J. Huang, J. Deng, G. Huang, D. Du, J. Lu, and J. Zhou, *Gait Recognition in the Wild: A Benchmark*, In Proceedings of the IEEE/CVF International Conference on Computer Vision, (2021), pp. 14789–14799.

Appendix A

Formal Definitions

This appendix provides formal definitions of the core concepts used throughout the thesis. Each definition is stated independently of the specific architecture described in Chapter 3, so readers can refer back to them without needing to follow the design choices made there. Where relevant, we note how each general concept specializes to our setting.

A.1 Attention

Attention is a mechanism that computes a weighted combination of a set of values, where the weights are determined by the compatibility between a query and a set of keys [61]. The idea is that not all parts of the input are equally relevant to a given computation, and the model should be able to focus on the parts that matter most.

Definition A.1 (Scaled Dot-Product Attention)**.** *Let* $\mathbf{Q} \in \mathbb{R}^{n \times d_k}$ *be a matrix of* n *query vectors,* $\mathbf{K} \in \mathbb{R}^{m \times d_k}$ *a matrix of* m *key vectors, and* $\mathbf{V} \in \mathbb{R}^{m \times d_v}$ *a matrix of* m *value vectors. Scaled dot-product attention is the function*

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{A.1}$$

where the softmax is applied row-wise. The scaling factor $\sqrt{d_k}$ *prevents the dot products from growing large in magnitude when the key dimension* d_k *is large, which would push the softmax into regions where its gradients are very small [61].*

The product $\mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{n \times m}$ gives an $n \times m$ matrix of compatibility scores. Each row is normalized by the softmax to produce a distribution over the m values. The output is therefore a weighted average of the value vectors, with weights determined by how well each query matches each key.

A.2 Self-Attention

In the general definition above, the queries, keys, and values can come from different sources. Self-attention is the special case where all three are derived from the same input sequence.

Definition A.2 (Self-Attention)**.** *Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a sequence of n input vectors, each of dimension d. Self-attention applies three learned linear projections to produce queries, keys, and values from $\mathbf{X}$ itself:*

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}^K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}^V,$$

where $\mathbf{W}^Q, \mathbf{W}^K \in \mathbb{R}^{d \times d_k}$ and $\mathbf{W}^V \in \mathbb{R}^{d \times d_v}$ are learned parameter matrices. The output is $\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ *as defined in Equation* (A.1).

Self-attention allows each position in the sequence to attend to every other position. This gives the model a global receptive field at every layer, unlike convolutions or recurrent networks, where the receptive field grows gradually with depth.

A.3 Multi-Head Attention

A single attention function computes one set of weights over the values. Multi-head attention runs several attention functions in parallel, each with its own learned projections from the input, and then combines their outputs. This allows different heads to attend to different aspects of the input.

Definition A.3 (Multi-Head Attention)**.** *Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a sequence of n input vectors. Given H attention heads, multi-head attention computes*

$$\text{MultiHead}(\mathbf{X}) = \text{Concat}(\mathbf{head}_1, \ldots, \mathbf{head}_H)\,\mathbf{W}^O,$$

where each head projects $\mathbf{X}$ independently into queries, keys, and values:

$$\mathbf{head}_i = \text{Attention}(\mathbf{X}\mathbf{W}_i^Q,\ \mathbf{X}\mathbf{W}_i^K,\ \mathbf{X}\mathbf{W}_i^V),$$

with per-head projections $\mathbf{W}_i^Q, \mathbf{W}_i^K \in \mathbb{R}^{d \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d \times d_v}$, and output projection $\mathbf{W}^O \in \mathbb{R}^{Hd_v \times d}$ [61]. The per-head dimension is typically set to $d_k = d_v = d/H$.

Each head learns its own projection from the raw input $\mathbf{X}$, so there is no separate projection step before multi-head attention. The self-attention definition in Section A.2 can be understood as the single-head special case ($H = 1$) of this formulation. In our encoder ($d = 128$, $H = 4$), each head operates on $d_h = 32$-dimensional projections. We concatenate the four heads into a 128-dimensional vector and project it back to $d = 128$ with $\mathbf{W}^O$.

A.4 Transformer

A transformer is a neural network architecture built entirely from attention and position-wise feedforward layers, without any recurrence or convolution [61]. The original transformer was designed for sequence-to-sequence tasks such as machine translation and consists of an encoder and a decoder. Many later applications, including ours, use only the encoder.

Definition A.4 (Transformer Encoder Layer)**.** *A single transformer encoder layer applies multi-head self-attention followed by a position-wise feedforward network (FFN), with residual connections and layer normalization around each sub-layer. In the pre-norm variant used in this thesis, the computation is*

$$\begin{aligned} \mathbf{H}' &= \mathbf{H} + \mathrm{MultiHead}(\mathrm{LN}(\mathbf{H}))\,, \\ \mathbf{H}'' &= \mathbf{H}' + \mathrm{FFN}(\mathrm{LN}(\mathbf{H}'))\,, \end{aligned}$$

where LN *denotes layer normalization [3], and the FFN is a two-layer network with a nonlinear activation:*

$$\mathrm{FFN}(\mathbf{x}) = \mathbf{W}_2\,\sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2,$$

with $\mathbf{W}_1 \in \mathbb{R}^{d\times d_{\mathrm{ff}}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{\mathrm{ff}}\times d}$*, and activation function* σ *(commonly ReLU or GELU). In this thesis,* $d_{\mathrm{ff}} = 4 \cdot d = 512$*, following the standard* mlp_ratio $= 4$ *convention of Vaswani et al. [61].*

A transformer encoder is a stack of L such layers applied in sequence. The input to the first layer is a sequence of token embeddings, optionally combined with positional information. The output of the final layer is a sequence of contextualized representations, one per input token.

Residual connections ensure gradients can flow directly from the output back to the input, which stabilizes training in deeper networks. Layer normalization rescales activations to have zero mean and unit variance within each token, which helps training converge.

The original transformer of Vaswani et al. [61] uses post-norm ordering, where layer normalization is applied after the residual addition rather than before it. Our encoder follows the pre-norm variant, where normalization is applied before each sub-layer. Pre-norm ordering tends to be more stable during training, especially for smaller models [65].

A.5 Vision Transformer

The Vision Transformer (ViT) adapts the transformer encoder to image classification by treating an image as a sequence of tokens [13].

Definition A.5 (Vision Transformer)**.** *Given an image* $\mathbf{I} \in \mathbb{R}^{H_{\text{img}} \times W_{\text{img}} \times C}$*, the image is divided into a grid of non-overlapping patches, each of size* $P \times P$ *pixels. Each patch is flattened into a vector* $\mathbf{x}_p \in \mathbb{R}^{P^2 C}$ *and linearly projected to dimension* d*:*

$$\mathbf{e}_p = \mathbf{W}_{\text{patch}}\, \mathbf{x}_p + \mathbf{b}_{\text{patch}}, \qquad \mathbf{W}_{\text{patch}} \in \mathbb{R}^{d \times P^2 C}.$$

A learnable classification token $\mathbf{e}_{\text{cls}}$ *is prepended to the patch sequence, and learned positional embeddings are added. The resulting sequence is processed by a standard transformer encoder. The output corresponding to the classification token is used as the image representation for downstream tasks.*

The ViT showed that a pure transformer, without any convolutional layers, can match or exceed convolutional networks on image classification when pretrained on large datasets. This result motivated applying transformer architectures to other domains, including skeleton-based action recognition and gait recognition.

Our encoder is not a ViT. We do not process images, and we do not use patch embeddings or a classification token. However, the ViT established the principle that transformers can operate on non-sequential data by converting it into a sequence of tokens with positional information. Our encoder applies this same principle to skeleton pose sequences, where each token represents a single joint at a single frame rather than an image patch.

A.6 Factorized Spatial-Temporal Transformer

When the input has two structural axes, such as space and time, a full self-attention layer over all tokens has quadratic cost in the total token count. Factorized attention reduces this cost by restricting attention to one axis at a time.

Definition A.6 (Factorized Spatial-Temporal Attention)**.** *Let* $\mathbf{H} \in \mathbb{R}^{T \times J \times d}$ *be a tensor of token representations, where* T *is the number of frames and* J *is the number of joints. A spatial attention block applies multi-head self-attention independently within each frame:*

$$\mathbf{H}_t^{\text{S}} = \text{MultiHead}(\mathbf{H}_t), \quad t = 1, \ldots, T,$$

where $\mathbf{H}_t \in \mathbb{R}^{J \times d}$ *is the slice of tokens at frame* t*. A temporal attention block applies multi-head self-attention independently within each joint:*

$$\mathbf{H}_j^{\text{T}} = \text{MultiHead}(\mathbf{H}_j), \quad j = 1, \ldots, J,$$

where $\mathbf{H}_j \in \mathbb{R}^{T \times d}$ *is the slice of tokens for joint* j*. Stacking one spatial block and one temporal block gives each token indirect access to the full* $T \times J$ *token set without materializing the full* $(TJ) \times (TJ)$ *attention matrix.*

Different factorized formulations differ in the ordering of spatial and temporal blocks, the depth of stacking, and whether cross-axis attention is added. Our encoder uses two spatial-temporal pairs stacked in the order S–T–S–T (see Chapter 3).

Arnab et al. [2] and Bertasius et al. [4] introduced this factorization for video understanding. In the video setting, spatial attention operates within a single frame and temporal attention operates across frames at a single spatial position.

Our encoder applies the same idea to skeleton sequences. Spatial attention operates across joints within a frame, letting the model learn which joints are relevant to each other at any given moment. Temporal attention operates across frames within a joint, letting the model learn how each joint's motion evolves over time. With $N_p = 2$ complete spatial-temporal passes, the full attention cost is $O(T \cdot J^2 + T^2 \cdot J)$ per pair, compared to $O((TJ)^2) = O(T^2J^2)$ for joint attention over all TJ tokens.

A.7 AdamW Optimizer

AdamW is a variant of the Adam optimizer that corrects the way weight decay interacts with adaptive learning rates [39].

Definition A.7 (Adam)**.** *Adam [31] maintains two exponential moving averages for each parameter θ: a first-moment estimate $\mathbf{m}_t$ (mean of gradients) and a second-moment estimate $\mathbf{v}_t$ (mean of squared gradients). At step t, given gradient $\mathbf{g}_t = \nabla_\theta \mathcal{L}_t$:*

$$\begin{aligned}
\mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\,\mathbf{g}_t, \\
\mathbf{v}_t &= \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\,\mathbf{g}_t^2, \\
\hat{\mathbf{m}}_t &= \mathbf{m}_t / (1-\beta_1^t), \qquad \text{(A.2)} \\
\hat{\mathbf{v}}_t &= \mathbf{v}_t / (1-\beta_2^t), \qquad \text{(A.3)} \\
\theta_{t+1} &= \theta_t - \eta\, \hat{\mathbf{m}}_t / (\sqrt{\hat{\mathbf{v}}_t} + \epsilon),
\end{aligned}$$

where η is the learning rate, β_1 and β_2 are the decay rates (typically 0.9 and 0.999), and ϵ is a small constant for numerical stability (typically 10^{-8}). The bias-correction terms in Equations (A.2)–(A.3) compensate for the zero initialization of $\mathbf{m}_0$ and $\mathbf{v}_0$.

Definition A.8 (AdamW)**.** *AdamW modifies the Adam update rule by decoupling weight decay from the gradient-based update. Using the bias-corrected moments from Equations (A.2)–(A.3), it applies weight decay directly to the parameters before the gradient step:*

$$\theta_{t+1} = (1-\lambda)\,\theta_t - \eta\, \hat{\mathbf{m}}_t / (\sqrt{\hat{\mathbf{v}}_t} + \epsilon),$$

where λ is the weight decay coefficient [39].

The difference matters because Adam scales the gradient by the inverse square root of the second moment. When L2 regularization is folded into the gradient,

parameters with large gradient variance receive a smaller effective weight-decay penalty per step than parameters with small gradient variance. Decoupled weight decay avoids this interaction and regularizes all parameters equally, regardless of their gradient statistics.

In our experiments, we use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and weight decay $\lambda = 10^{-4}$ for both pretraining and fine-tuning.

A.8 COCO-17 Skeleton

The COCO keypoint format defines a standard set of body joints used by most modern pose estimators [37]. The full body layout includes 17 joints.

Definition A.9 (COCO-17 Skeleton)**.** *The COCO-17 skeleton consists of the following 17 keypoints, indexed from 0 to 16:*

Index	*Joint*	*Index*	*Joint*
0	*Nose*	*9*	*Left wrist*
1	*Left eye*	*10*	*Right wrist*
2	*Right eye*	*11*	*Left hip*
3	*Left ear*	*12*	*Right hip*
4	*Right ear*	*13*	*Left knee*
5	*Left shoulder*	*14*	*Right knee*
6	*Right shoulder*	*15*	*Left ankle*
7	*Left elbow*	*16*	*Right ankle*
8	*Right elbow*		

Each keypoint is represented as a 2D coordinate (x, y) in the image plane. Pose estimators such as HRNet [58] and AlphaPose [17] produce these keypoints from video frames. This thesis uses COCO-17 as its joint convention for all experiments, including DenseGait poses that are originally extracted in an 18-joint OpenPose layout and converted to COCO-17 as described in Section 3.5.

The three region configurations used in this thesis are subsets of the COCO-17 skeleton. R1 (upper body) uses 11 joints: indices 0–10. R2 (lower body) uses 6 joints: indices 11–16. ALL uses all 17 joints. These regions are defined in Table 3.1 and illustrated in Figure 3.5.

Table A.1: Notation reference.

Symbol	Meaning	This thesis
T	Number of frames in a pose sequence	56
J	Number of joints per frame	17 (COCO-17)
d	Token embedding dimension	128
H	Number of attention heads	4
$d_k,\ d_v$	Per-head key and value dimensions	$d_k = d_v = d/H = 32$
d_{ff}	Feedforward hidden dimension	512 ($= 4 \cdot d$)
N_p	Number of spatial-temporal block pairs	2
$\mathbf{W}_i^Q,\ \mathbf{W}_i^K,\ \mathbf{W}_i^V$	Per-head query, key, value projections	$\in \mathbb{R}^{d \times d_k}$
$\mathbf{W}^O$	Output projection (multi-head attention)	$\in \mathbb{R}^{Hd_v \times d}$

A.9 Notation Reference

Table A.1 lists the principal symbols used in this appendix and in the methodology chapter (Chapter 3), together with the specific values used in this thesis.

Appendix B

Implementation Details, Hyperparameters, and Code References

B.1 Code, Environment, and Data Access

All code for this thesis is maintained in a private GitHub repository (`rzehsani-wq/gait-thesis`). Local development took place on Windows, using GitHub Desktop for version control and Visual Studio Code as the primary editor. All training and evaluation runs were executed on Google Colab Pro+, using a single NVIDIA T4 GPU (Section 4.1.2).

Data and checkpoints are stored on Google Drive, under `MyDrive/thesis/`. DenseGait poses are packed as a single archive, `densegait_poses.tar` (approximately 15 GB), and extracted to local Colab disk storage at the start of each session, to avoid FUSE mount timeouts when accessing the underlying 200,000-plus loose pose files directly from Drive.

Self-supervised pretraining produces two checkpoints: `ssl_casia_train_only_50ep_resume.pt` (50 epochs on CASIA-B, restricted to the 99 training subjects) and `ssl_densegait_50ep_resume.pt` (50 epochs on DenseGait). Both checkpoints load into the fine-tuning pipeline with a strict key match (58 of 58 keys).

Run logs and per-run metrics are written to `runs/*/metrics.csv` during training, and aggregated into two master summaries, `runs_master_final_v3.csv` (27 headline runs) and `ablation_final.csv` (18 ablation runs), which together are the source for every table and figure in Chapter 4. Run naming follows the pattern `<pretraining>_<protocol>_<seed>`, with ablation-specific suffixes appended where relevant. For example, `casia_arcface_nm2cl_s0_3e-4` denotes the headline CASIA-SSL configuration on nm→cl, seed 0.

The supervised fine-tuning recipe of Section 3.6 corresponds directly to

Table B.1: Self-supervised pretraining configuration. Values that differ between the two pretraining sources are given as a CASIA-B / DenseGait pair.

Hyperparameter	Value
Optimizer	AdamW
Base learning rate	3×10^{-4}
Weight decay	10^{-4}
Gradient clipping (global ℓ_2-norm)	1.0
Mixed precision	enabled (AMP)
Batch size	64
Input clip length (T)	56 frames
Masking ratio (p)	0.35
Stride (CASIA-B / DenseGait)	8 / 4
Epochs (CASIA-B / DenseGait)	50 / 50
Final reconstruction loss (CASIA-B, epoch 50)	$\approx 1.64 \times 10^{-3}$
Final reconstruction loss (DenseGait, epoch 50)	$\approx 1.97 \times 10^{-3}$

`src/train/train_id.py` in the repository, invoked with the flags listed in Table 3.2 and Table 3.3.

B.2 Self-Supervised Pretraining Configuration

Table B.1 reports the full self-supervised pretraining configuration, described in prose in Section 3.3.3. Both pretraining sources share the same optimizer settings and mask ratio; stride, epoch count, and final reconstruction loss differ between the two, as discussed in Section 3.3.4.

B.3 Supervised Fine-Tuning and Evaluation Configuration

The complete fine-tuning configuration is given in Table 3.2 (Section 3.6.3), and the complete evaluation configuration is given in Table 3.3 (Section 3.7.5). Both are reproduced in full there and are not repeated in this appendix.

Table B.2: Confidence thresholds used to exclude low-confidence joint detections from the normalization statistics of Section 3.5.2. Detections below threshold are excluded from the mid-hip centering and shoulder-width scale estimates; they are not dropped from the sequence entirely.

Source	Confidence threshold
CASIA-B	0.5
DenseGait	0.3

B.4 Preprocessing Confidence Thresholds

Section 3.5.2 describes the confidence-based filtering applied when estimating normalization statistics, without giving the specific per-source threshold values. Table B.2 reports them.

Appendix C

DenseGait Joint Remapping

Section 3.5.4 describes the empirical procedure used to recover the correspondence between DenseGait's 18-channel pose layout and the 17-joint COCO ordering used throughout this thesis. Table C.1 reports the resulting mapping.

The left and right eye (COCO-17 indices 1 and 2) and ear joints (COCO-17 indices 3 and 4) are assigned based on the standard anatomical convention rather than direct experimental verification. These joints show very little movement during walking, making it difficult to distinguish the left side from the right side using only their average horizontal positions. In addition, the small amount of movement is often comparable to the noise introduced by the pose estimation algorithm. This ambiguity is unlikely to noticeably affect identification performance, for two reasons. First, the encoder supplies joint identity through a learned embedding table (Section 3.2), rather than through a fixed anatomical prior. The encoder therefore does not need to know which side is truly left or truly right; it only needs a consistent association between each channel index and the joint that occupies it, and this consistency holds as long as the same convention is applied throughout preprocessing, to both training and validation data alike. Second, even if this consistency assumption were imperfect, the final representation is a mean over all $T \times J$ tokens (Equation 3.6, Section 3.2), so the four affected joints, out of seventeen total, would contribute only a small fraction of the pooled representation.

Table C.1: Empirically recovered correspondence between DenseGait's 18-channel pose layout and the 17-joint COCO ordering used throughout this work. Recovered via mean-position band identification followed by left/right sign disambiguation (Section 3.5.4). DenseGait source index 4 (neck) has no COCO-17 equivalent and is dropped. Entries marked † were disambiguated by anatomical convention rather than a clear empirical sign, since eye and ear joints move minimally during walking (Section 3.5.4).

COCO-17 index	Joint name	DenseGait source index
0	Nose	0
1	Left Eye†	17
2	Right Eye†	5
3	Left Ear†	7
4	Right Ear†	9
5	Left Shoulder	6
6	Right Shoulder	8
7	Left Elbow	10
8	Right Elbow	11
9	Left Wrist	13
10	Right Wrist	15
11	Left Hip	12
12	Right Hip	14
13	Left Knee	16
14	Right Knee	1
15	Left Ankle	2
16	Right Ankle	3
—	Neck (dropped)	4

Appendix D

Per-Seed Result Breakdowns

This appendix reports the individual per-seed values underlying every mean ± standard deviation figure in Chapter 4. All values are Rank-1 accuracy (%) unless noted; source data is `runs_master_final_v3.csv` for the main protocol results and `ablation_final.csv` for the ablation studies (Appendix B).

D.1 Main Protocol Results

Table D.1 reports per-seed Rank-1 accuracy underlying Table 4.1; Table D.2 reports the corresponding Rank-5 values.

D.2 Ablation Studies

Table D.3 reports per-seed Rank-1 accuracy underlying the four ablation tables of Section 4.6. All runs use nm→cl and CASIA-SSL. The default configuration (region R1, ArcFace, unfrozen backbone, view-matched retrieval, gallery size $g = 4$) is listed once at the top; every ablation below varies exactly one setting away from it, so its seed values are not repeated per ablation.

Table D.1: Per-seed Rank-1 accuracy (%), main protocol comparison. All nine configurations use 3 seeds.

Pretraining	Protocol	Seed 0	Seed 1	Seed 2	Mean ± Std
DenseGait-SSL	nm→nm	97.4	89.3	93.7	93.5 ± 4.0
DenseGait-SSL	nm→bg	78.5	70.3	75.3	74.7 ± 4.2
DenseGait-SSL	nm→cl	60.8	66.9	63.7	63.8 ± 3.0
CASIA-SSL	nm→nm	95.3	93.5	96.2	95.0 ± 1.3
CASIA-SSL	nm→bg	66.7	77.9	71.7	72.1 ± 5.6
CASIA-SSL	nm→cl	64.5	69.6	66.5	66.8 ± 2.6
Random init	nm→nm	95.2	90.4	93.6	93.1 ± 2.4
Random init	nm→bg	78.1	74.1	70.9	74.4 ± 3.6
Random init	nm→cl	62.9	64.9	64.1	63.9 ± 1.0

Table D.2: Per-seed Rank-5 accuracy (%), main protocol comparison. Same runs as Table D.1.

Pretraining	Protocol	Seed 0	Seed 1	Seed 2	Mean ± Std
DenseGait-SSL	nm→nm	100.0	100.0	100.0	100.0 ± 0.0
DenseGait-SSL	nm→bg	93.8	91.8	93.1	92.9 ± 1.0
DenseGait-SSL	nm→cl	86.9	92.5	90.7	90.0 ± 2.8
CASIA-SSL	nm→nm	100.0	100.0	100.0	100.0 ± 0.0
CASIA-SSL	nm→bg	90.1	92.9	91.6	91.5 ± 1.4
CASIA-SSL	nm→cl	91.0	92.2	90.5	91.2 ± 0.9
Random init	nm→nm	100.0	100.0	100.0	100.0 ± 0.0
Random init	nm→bg	93.2	93.5	91.0	92.6 ± 1.4
Random init	nm→cl	90.7	92.1	92.1	91.6 ± 0.8

Table D.3: Per-seed Rank-1 accuracy (%), all four ablations (nm→cl, CASIA-SSL). The default row is shared across all four ablations (R1, ArcFace, unfrozen, view-matched, g=4).

Ablation	Configuration	Seed 0	Seed 1	Seed 2	Mean ± Std
—	Default	64.5	69.6	66.5	66.8 ± 2.6
Metric loss	Cross-entropy only	55.5	60.9	60.4	59.0 ± 3.0
Backbone	Frozen	38.3	38.5	37.8	38.2 ± 0.4
Region	R2	89.9	88.5	88.7	89.0 ± 0.8
Region	ALL	73.8	82.2	77.8	77.9 ± 4.2
View matching	Disabled	53.6	62.9	50.3	55.6 ± 6.5

Appendix E

Computational Requirements

E.1 Hardware and Environment

All self-supervised pretraining and supervised fine-tuning runs reported in this thesis were executed on Google Colab Pro+, using a single NVIDIA T4 GPU (Section 4.1.2). No run in this thesis used distributed or multi-GPU training.

E.2 Model Size

The encoder has approximately 813,000 parameters in its self-supervised pretraining configuration, and approximately 846,000 parameters once the supervised identification head (MetricHead) is attached (Section 3.2.5). This is small relative to the comparison methods in Section 4.3: GaitPT uses about 4 million parameters, and GaitPTv2 scales to 58 million parameters in its largest configuration.

E.3 Dataset Scale

Self-supervised pretraining uses two sources of substantially different sizes. CASIA-B is a comparatively small, curated laboratory dataset. DenseGait is a substantially larger in-the-wild corpus, with approximately 217,000 tracklets (Section 3.3.4). The DenseGait pose archive occupies approximately 15 GB on disk and is extracted to local Colab session storage at the start of each pretraining run, rather than read directly from Google Drive, to avoid FUSE mount timeouts when accessing the underlying 200,000-plus individual pose files.

Table E.1: Wall-clock duration of self-supervised pretraining, measured on a single NVIDIA T4 GPU.

Stage	Duration (s)	Duration
SSL pretraining, CASIA-B (50 epochs)	4,650	≈1.3 hours
SSL pretraining, DenseGait (50 epochs)	166,750	≈46.3 hours

E.4 Experiment Scale

The full experimental study in Chapter 4 comprises 45 training runs: 27 headline runs (3 initialization conditions, CASIA-SSL, DenseGait-SSL, and random initialization, × 3 protocols × 3 seeds) and 18 ablation runs across metric loss, backbone freezing, region, and view matching, of which 3 reproduce the headline CASIA-SSL, nm→cl configuration as a shared reference row rather than a distinct configuration (Appendix D). A separate backbone learning-rate sweep (Section 4.5) contributes 5 additional single-seed runs not included in this count.

E.5 Training Duration

Table E.1 reports measured wall-clock duration for self-supervised pretraining, both at the 50-epoch schedule used throughout this thesis (Section 3.3.4).

DenseGait pretraining is far slower than CASIA-B pretraining in total wall-clock time. This reflects DenseGait's substantially larger corpus size (Section 3.3.4).

Per-run duration for supervised fine-tuning, and aggregate compute across the 45 training runs reported in Chapter 4, were not logged during this project and cannot be reconstructed after the fact. We report the two pretraining durations above because they were recorded directly; we do not estimate the remaining figures to avoid presenting a fabricated number as measured data.