Gait Transformer: End-to-End Transformer Backbone for Gait Recognition

Saihui Hou[1], **Wenpeng Lang**[1], **Jilong Wang**[2,3], **Yan Huang**[3], **Liang Wang**[3], **Yongzhen Huang**[1,4*]

[1]Beijing Normal University
[2]University of Science and Technology of China
[3]Institute of Automation, Chinese Academy of Sciences
[4]WATRIX.AI

Abstract

Gait recognition has emerged as a promising biometric technique for long-distance and non-intrusive human identification. While Transformers have revolutionized vision tasks, their adaptation to gait recognition remains underexplored due to domain-specific challenges such as sparse silhouette modality, spatial-temporal dynamics, fine-grained motion cues, and limited training data. In this paper, we propose **Gait Transformer (GaT)**, an end-to-end Transformer backbone specifically tailored for silhouette-based gait recognition. GaT introduces three key components: (1) a hybrid patch embedding module that combines convolutional stems with group-batch normalization to enhance structural preservation; (2) a decomposed token mixer that explicitly models both short-range and long-range dependencies across spatial-temporal dimensions; and (3) a hybrid positional encoding strategy that integrates absolute, relative, and rotary embeddings to support efficient training under data scarcity. **Without relying on any pretraining**, GaT achieves state-of-the-art performance on Gait3D, GREW, and CCGR-MINI.

1 Introduction

Gait recognition, which aims to identify individuals based on walking patterns, has gained increasing attention due to its unique advantages in long-distance, non-invasive scenarios. Recent advances in Vision Transformers (ViT) (Dosovitskiy et al. 2021) and Swin Transformers (SwinT) (Liu et al. 2021b) have significantly advanced computer vision tasks (Carion et al. 2020; Strudel et al. 2021). However, their direct application as **backbones** for gait recognition remains underexplored and non-trivial.

Prior studies have investigated incorporating transformers into gait recognition pipelines. For example, GaitViT (Mogan et al. 2022) applies a vanilla ViT to Gait Energy Images (GEI) (Han and Bhanu 2005), but performs poorly due to modality mismatch and inadequate temporal modeling. SwinGait (Fan et al. 2023a) improves performance by introducing window-based attention into the later stages of a CNN backbone (*i.e.*, DeepGaitV2-P3D), but suffers from several limitations: (1) heavy reliance on shallow convolutional layers (up to 13 layers on Gait3D (Zheng et al. 2022) and GREW (Zhu et al. 2021)[1]); (2) dependency on CNN pretraining and weight transfer, complicating the training process; (3) use of bilinear interpolation to connect convolutional and self-attention layers, compromising architectural elegance; (4) lack of explicit modeling of long-range dependencies, which self-attention mechanisms are inherently suited to handle (Vaswani et al. 2017). These limitations underscore the absence of a unified end-to-end Transformer-based backbone specifically designed for gait recognition.

Method	GaitViT	SwinGait	GaT
Input			
Input Type	Template	Sequence	Sequence
Conv Stem	None	Deep	Shallow
Positional Encoding	Global	Local	Global+Local
No Pretrain	✓	✗	✓
No Interpolation	✓	✗	✓
Spatial Modeling	✓	✓	✓
Temporal Modeling	✗	✓	✓
Short-Range Modeling	N/A	✓	✓
Long-Range Modeling	N/A	✗	✓

Figure 1: Comparison between our GaT with GaitViT (Mogan et al. 2022) and SwinGait (Fan et al. 2023a).

We argue that directly applying ViT or SwinT to gait tasks leads to unsatisfactory performance due to four domain-specific challenges: (1) **Unique Modality**: Silhouettes are sparse, binary, and structurally driven, making them fundamentally different from natural images; (2) **Spatial-Temporal Dynamics**: Gait recognition requires capturing both spatial configurations and their temporal evolution; (3) **Fine-Grained Motion**: Subtle limb movements encode essential identity cues; (4) **Data Scarcity**: Gait datasets are relatively small (Zheng et al. 2022; Zhu et al. 2021), leading to overfitting and limited generalization[2].

[1]We count the convolutional layers in the main branch of residual blocks and exclude 1×1 convolutional layers used for channel transformation (He et al. 2016; Fan et al. 2023a).

[2]Transformer attention layers lack the inductive biases inherent to CNNs. Consequently, Transformer backbones generally **demand more training data** compared to CNNs (Liu et al. 2021a).

*Corresponding Author

To address these challenges, we propose **Gait Transformer (GaT)**, a unified backbone explicitly designed for silhouette-based gait recognition. It consists of three key components: (1) **Patch Embedding**: A lightweight convolutional stem precedes the token projection layer to suppress segmentation noise and better preserve structural details. Group Normalization (Wu and He 2018) is incorporated alongside Batch Normalization (Ioffe and Szegedy 2015) to improve robustness against noisy inputs. (2) **Token Mixer**: This module is decomposed into Short-Range and Long-Range Mixers. The Short-Range Mixer captures local spatial-temporal patterns, while the Long-Range Mixer models global dependencies across the sequence. (3) **Positional Encoding**: A hybrid strategy that integrates 2D learnable global positional encoding with a combination of 3D Rotary Positional Embedding (Su et al. 2024) and 3D Learnable Relative Position Bias (Liu et al. 2022) within localized attention, introducing strong inductive priors for efficient learning from limited data.

Compared to GaitViT (Mogan et al. 2022), GaT jointly captures spatial and temporal variations from silhouette sequences, leading to substantial accuracy improvements. Compared to SwinGait (Fan et al. 2023a), GaT provides the following advantages: (1) reduced reliance on convolutional layers; (2) enables fully end-to-end training from scratch; (3) eliminates the need for interpolation and downsampling bridges; (4) explicitly models long-range dependencies; (5) achieves superior performance on multiple benchmarks.

In summary, our contributions are threefold:

- We systematically identify the key challenges in adapting general-purpose Transformer architectures to gait recognition, including modality mismatch, complex spatial-temporal dynamics, subtle motion patterns, and the scarcity of labeled training data.
- We propose **Gait Transformer**, a specialized Transformer backbone with customized designs for patch embedding, token interaction, and positional encoding, tailored specifically for gait analysis.
- We establish new state-of-the-art results on the challenging Gait3D, GREW, and CCGR-MINI benchmarks, while *enabling fully end-to-end training without reliance on pretrained CNNs*.

Why We Need Transformer Backbones for Gait Recognition? Transformers have become the dominant backbone across a wide range of vision tasks, serving as the foundation for both large-scale pretraining and generative modeling. However, their potential remains largely untapped in gait recognition, where CNN-based architectures still prevail. This work takes a pioneering step toward addressing this gap. Compared to CNNs, Transformers offer several key advantages: (1) **Dynamic Modeling**: the ability to adaptively capture spatial-temporal motion patterns; (2) **Long-Range Dependency Modeling**: effective aggregation of global contextual information; (3) **Scalability and Transferability**: flexible adaptation to various model sizes and domains. We demonstrate that Transformer backbones can serve as a powerful foundation for gait recognition, extracting more discriminative features through their dynamic and long-range modeling capabilities. Our findings open up new directions for Transformer-based gait pretraining, generative modeling, and broader applications in motion analysis.

2 Related Work

Gait Recognition

Existing gait recognition methods can be broadly categorized into appearance-based and model-based approaches, where the choice of input modality determines the corresponding modeling strategy.

Appearance-based methods primarily rely on binary silhouettes, which are widely adopted due to their ability to suppress color and texture information. GaitSet (Chao et al. 2019) pioneered the paradigm of treating silhouette sequences as unordered sets, inspiring a range of follow-up works that utilize 2D CNNs for gait representation learning (Hou et al. 2020; Fan et al. 2023b). More recently, 3D CNN-based methods have become the mainstream paradigm, offering improved performance by more effectively modeling spatial-temporal dependencies (Fan et al. 2023a; Ma et al. 2024; Xiong et al. 2024).

In contrast, model-based methods mostly extract identity features from pose sequences, which provide explicit structural and motion cues while being inherently robust to appearance variations. Sparse pose sequences can be processed using Graph Convolutional Networks (GCNs) to model dependencies (Teepe et al. 2021; Fu et al. 2023), or transformed into heatmap representations to enable CNN-based learning (Fan et al. 2024).

Recent research has explored alternative modalities for gait recognition, including RGB-based (Ye et al. 2024; Jin et al. 2025), point cloud-based (Shen et al. 2023, 2025), and parsing-based representations (Wang et al. 2023d; Zheng et al. 2023). However, silhouette-based approaches remain the *mainstream* paradigm in both academic research and practical applications (Fan et al. 2025), and therefore serve as the primary focus of this work.

Transformers in Gait Field

The success of Transformer architectures has revolutionized both the natural language processing (Vaswani et al. 2017) and computer vision communities (Dosovitskiy et al. 2021; Liu et al. 2021b), demonstrating remarkable capability in large-scale multi-modal learning (Liu et al. 2023, 2024; Bai et al. 2025).

In the domain of silhouette-based gait recognition, beyond early attempts such as GaitViT (Mogan et al. 2022) and SwinGait (Fan et al. 2023a), recent methods have increasingly integrated self-attention mechanisms into their architectures. For example, DANet (Ma et al. 2023) introduces a self-attention module to dynamically select representative local motion patterns and aggregate them into robust global motion representations. VPNet (Ma et al. 2024) combines self-attention with visual prompts to adaptively handle various covariates such as clothing and carrying conditions. GaitMoE (Huang et al. 2024) customizes a self-attention mechanism specifically for occluded gait recognition, enhancing robustness under partial observation.

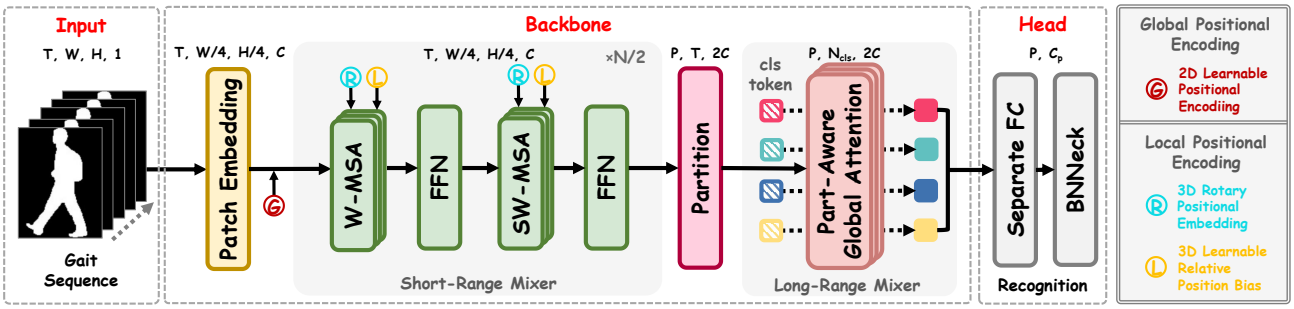

Figure 2: Overview of GaT. T, H, W, and C denote the temporal, height, width, and channel dimensions, respectively. N, N_{cls}, and P refer to the number of mixer blocks, class tokens, and body parts. C_{p} is the final part-level feature dimension. We show two *successive* Short-Range Mixer blocks: the first uses window-based attention (W-MSA), and the second applies a shifted-window strategy (SW-MSA). The stack maintains spatial resolution and doubles the embedding to $2C$ in the last two blocks (residuals omitted). The Long-Range Mixer aggregates global features with learnable part-level class tokens as queries.

Nevertheless, most of these methods apply self-attention only at the final stage of the network, treating it as an auxiliary enhancement rather than a fundamental design component. To date, few efforts have been made to develop a Transformer-based backbone tailored for processing silhouette sequences, leaving the full potential of self-attention in gait modeling largely underexplored.

3 Our Approach

In this section, we introduce **Gait Transformer (GaT)**, an end-to-end Transformer-based backbone specifically tailored for gait recognition from silhouette sequences. As illustrated in Figure 2, the architecture consists of three key components: (1) **Patch Embedding**, which converts raw silhouette frames into visual tokens; (2) **Token Mixer**, which models spatial and temporal dependencies across both local and global contexts; and (3) **Positional Encoding**, which injects motion dynamics and structural priors into the token sequence to enhance spatial-temporal awareness.

Patch Embedding: Structure-Aware Tokenization

The patch embedding module aims to transform raw silhouette sequences into high-quality token representations. Unlike RGB images, silhouettes inherently suppress variations in color and texture caused by clothing or background clutter. However, they are typically obtained via upstream preprocessing pipelines (*e.g.*, detection, tracking, segmentation), which often introduce noise such as shape distortion, missing limbs, or residual background artifacts. Moreover, due to the relatively small scale of existing gait datasets, it is challenging to learn discriminative embeddings using a single linear projection as in the vanilla ViT (Dosovitskiy et al. 2021). To address these challenges, we introduce two key designs to enhance the tokenization process: a lightweight convolutional front-end and a mixed normalization strategy.

Convolutional Stem We adopt a shallow stack of convolutional layers, consisting of an initial convolution followed by two residual blocks, as a front-end to suppress noise in silhouette frames. This design enhances local structure awareness and provides more stable inputs for subsequent Transformer layers. Unlike prior uses of convolution in Transformer backbones, which mainly aim to improve training stability (Xiao et al. 2021), we leverage convolution explicitly to mitigate silhouette-specific artifacts and address the *dumb patch* issue (Fan et al. 2023a), where patches may contain all-zero or all-one values. Furthermore, in contrast to SwinGait, which employs up to 13 convolutional layers, our convolutional stem is much shallower (5 convolutions followed by an additional one for token projection) and, importantly, *does not require any pretraining*.

Group-Batch Normalization Most existing gait recognition methods incorporate Batch Normalization (BN) within convolutional layers to improve inter-sample discriminability by computing statistics across a batch. However, BN is highly sensitive to appearance shifts (Pan et al. 2018), making it vulnerable to corrupted statistics in the presence of severe occlusions or segmentation errors. To mitigate this issue, we explore normalization strategies that operate at the individual sample level, *i.e.*, Instance Normalization (IN) (Ulyanov, Vedaldi, and Lempitsky 2016) which has been widely used in the visual domain (Huang and Belongie 2017; Nam and Kim 2018). Nevertheless, due to the inherently noisy nature of silhouette inputs, IN, which normalizes each channel independently, may overly amplify noise and reduce feature robustness. To strike a balance between stability and structural preservation, we adopt Group Normalization (GN) (Wu and He 2018), which normalizes features within groups of channels, thereby enhancing robustness while retaining fine-grained spatial details.

Specifically, we integrate GN into the residual blocks of our convolutional stem, combining it with BN in the first normalization layer of main branch (illustrated in the appendix). Compared to prior works that explore multi-normalization strategies (Pan et al. 2018), our approach introduces two distinct considerations: (1) we explicitly favor GN over IN based on the specific properties of silhouette-based gait recognition, and (2) our hybrid normalization is applied only in the shallow convolution stem, while the main body of the network remains a pure transformer architecture.

Summary At the end of patch embedding, which incorporates the above gait-specific designs (the final residual block applies a spatial stride of $(2, 2)$ to reduce GPU mem-

ory consumption and the detailed configuration is present in Table 1), each silhouette frame is partitioned into 2×2 patches. Formally, given an input tensor of shape $X_{\text{in}} \in \mathbb{R}^{T\times H_{\text{in}}\times W_{\text{in}}\times C_{\text{in}}}$ ($H_{\text{in}}=64, W_{\text{in}}=48, C_{\text{in}}=1$), the patch embedding operation produces an output $X_{\text{pe}} \in \mathbb{R}^{T\times H\times W\times C}$ with ($H=16, W=12$), where each frame is represented by 16×12 tokens.

Token Mixer: Hierarchical Modeling

To capture discriminative gait patterns, it is essential to model spatial appearance and temporal dynamics in a unified and structured manner. Furthermore, gait recognition requires fine-grained modeling of subtle motions (*e.g.*, head tilt, foot swing) that arise in localized spatial regions and evolve over time. To this end, we divide the token mixer into two specialized modules: a **Short-Range Mixer** for local motion patterns and a **Long-Range Mixer** for global sequence semantics.

Short-Range Mixer The Short-Range Mixer captures fine-grained local motion via localized spatial-temporal attention. It is implemented using a 3D window-based self-attention mechanism (Liu et al. 2022). Specifically, we partition the input token grid along the spatial and temporal dimensions into non-overlapping 3D windows (*i.e.*, cubes) and perform self-attention independently within each window.

Formally, given a token cube $X_{\text{cube}} \in \mathbb{R}^{M_{\text{d}}\times M_{\text{h}}\times M_{\text{w}}\times C}$ inside a local window, the attention is computed as

$$\begin{aligned}\text{Attention}(Q^{(\text{s})},K^{(\text{s})},V^{(\text{s})}) &= \text{Softmax}(\tau Q^{(\text{s})}_{\text{rot}} K^{(\text{s})}_{\text{rot}} + B)V \\ Q^{(\text{s})}_{\text{rot}}, K^{(\text{s})}_{\text{rot}} &= \text{RoPE}(\text{Norm}(Q^{(\text{s})})),\text{RoPE}(\text{Norm}(K^{(\text{s})}))\end{aligned} \tag{1}$$

where $(Q^{(\text{s})}, K^{(\text{s})}, V^{(\text{s})})$ are obtained through linear projections of X_{cube} for queries/keys/values. $(M_{\text{d}}, M_{\text{h}}, M_{\text{w}})$ denote the window size along the temporal, height, and width dimensions, respectively. $\text{Norm}(\cdot)$ is L_2 normalization applied to token features, and τ is a learnable temperature initialized with 64 to increase the variance of the attention matrix. $\text{RoPE}(\cdot)$ and B represent the positional encoding that will be discussed in the next Section 3. To enable cross-window communication, we adopt the *shifted-window* strategy along both spatial and temporal dimensions (Liu et al. 2022). Given a window size of $(M_{\text{d}}, M_{\text{h}}, M_{\text{w}})$, the corresponding shift is set to $(\lfloor M_{\text{d}}/2 \rfloor, \lfloor M_{\text{h}}/2 \rfloor, \lfloor M_{\text{w}}/2 \rfloor)$. Additionally, *in the last two blocks, we increase the embedding dimension to $2C$ by doubling the number of attention heads, and introduce a linear projection on the residual branch to ensure dimensional alignment.*

Compared with standard 3D window attention (Liu et al. 2022), our Short-Range Mixer introduces two key modifications:

- A hybrid positional encoding strategy that integrates the parameter-free 3D Rotary Positional Embedding with a 3D Learnable Relative Position Bias B, enhancing spatiotemporal awareness while maintaining computational efficiency.
- The removal of patch merging operations during the token mixing stage, which preserves full spatial resolution and helps retain fine-grained details crucial for gait recognition.

In addition, a two-layer feed-forward network is appended after each window attention block. To mitigate overfitting due to the limited scale of gait data, we set the expansion ratio of the hidden dimension to 2.0.

Long-Range Mixer The Long-Range Mixer is designed to aggregate high-level and cycle-level semantics from the entire sequence. One of the key advantages of self-attention over convolution is its ability to model long-range dependencies, particularly desirable for gait recognition where identity cues are distributed throughout the whole walking sequence. However, directly applying global attention over all frame patches is computationally expensive. To address this, we propose a **part-aware global attention** mechanism that explicitly models long-term dependencies at the part level.

Formally, we use $X \in \mathbb{R}^{T\times H\times W\times 2C}$ to denote the features output by the stack of Short-Range Mixers. We first partition each frame along the vertical axis into P horizontal parts (*e.g.*, head, torso, legs), and apply mean/max pooling within each region to reduce token redundancy. This gives us a compact representation:

$$X^{\text{t}}_{\text{p}} = \text{AvgP}(X_{\text{t,h}_{\text{p}}:\text{h}_{\text{p+1}},:,:}) + \text{MaxP}(X_{\text{t,h}_{\text{p}}:\text{h}_{\text{p+1}},:,:}) \tag{2}$$

where p indexes the part and t indexes the frame, AvgP and MaxP denotes the mean and max pooling.

For each part p, we introduce a set of learnable **part-level class tokens** $CLS^{(\text{g})}_{\text{p}}$, and perform temporal self-attention to aggregate features across the sequence:

$$\mathbf{Z}_{\text{p}} = \text{Attention}(CLS^{(\text{g})}_{\text{p}}, K^{(\text{g})}_{\text{p}}, V^{(\text{g})}_{\text{p}}) \tag{3}$$

where the key/value pairs $(K^{(\text{g})}_{\text{p}}, V^{(\text{g})}_{\text{p}})$ are the linear projections of X^{t}_{p}. The attention computation follows Equation (1), with three differences: (i) Queries are replaced by learnable class tokens; (ii) No positional encoding is used, as position information has already been injected in prior short-range stages, and the goal here is global aggregation; (iii) We use a single Long-Range Mixer block for GaT without a subsequent feed-forward layer after the attention module.

Summary Our core backbone consists of short-range and long-range token mixers that transform the input silhouette sequence into feature representations of shape $P\times N_{\text{cls}}\times 2C$, where P denotes the number of body parts and N_{cls} the number of class tokens per part. To obtain final representations, we apply max pooling across the class tokens and employ a classification head similar to that in (Fan et al. 2023a; Ma et al. 2024). Specifically, the classification head includes a linear projection layer, followed by Batch Normalization (BN), and an identity classification layer (used only during training) for each part. The features before BN are used for computing the triplet loss, while those after BN are used for the cross-entropy loss. During inference, we compute the average feature distance across all parts to measure the similarity between probe and gallery.

Positional Encoding: Spatial-Temporal Priors

Unlike convolutional operations, which naturally encode positional relationships through fixed kernel structures, self-attention mechanisms are inherently permutation-invariant

and require explicit positional encoding to model structural priors. For gait recognition, positional information is critical: spatial dependencies (*e.g.*, head-to-shoulder motion) and temporal continuity (*e.g.*, walking cycle rhythm) both contribute essential cues for identity discrimination.

However, standard learnable positional embeddings often require large-scale datasets to generalize well (Dosovitskiy et al. 2021), an assumption that does not hold for current gait benchmarks. To address this, we design a hybrid positional encoding strategy, combining learnable global positional embeddings and lightweight relative positional cues within local windows.

Global Positional Encoding We inject learnable global positional encoding immediately after the patch embedding and before the token mixer. In particular, *we restrict global positional encoding to the spatial domain only*, dubbed as Global2D. This choice is based on the observation that, during training, fixed-length sequences (*e.g.*, 30 frames) are sampled, whereas test sequences exhibit variable lengths. Hence, we define a learnable embedding $\mathbf{E} \in \mathbb{R}^{H \times W \times C}$ such that each token $\mathbf{X}_{\text{pe}}^{\text{t,h,w}}$ is updated by:

$$\mathbf{X}_{\text{pe}}^{\text{t,h,w}} \leftarrow \mathbf{X}_{\text{pe}}^{\text{t,h,w}} + \mathbf{E}_{\text{h,w}}, \tag{4}$$

where t is the temporal index, (h, w) denotes the spatial location, and C is the embedding dimension.

Local Positional Encoding In the Short-Range Mixer, we further enhance the model's sensitivity to spatial-temporal order using local positional encoding within each 3D window. Specifically, we combine two complementary techniques: 3D Rotary Positional Embedding (RoPE3D) (Su et al. 2024) and Learnable Relative Position Bias (Relative3D) (Liu et al. 2022).

RoPE3D encodes relative positions by rotating the query and key vectors in the attention calculation using sinusoidal functions. Specifically, given query $\mathbf{q}_\text{i}$ and key $\mathbf{k}_\text{j}$ for tokens i and j in a flattened local window, RoPE modifies the dot-product attention as:

$$\langle \text{RoPE}(\mathbf{q}_\text{i}), \text{RoPE}(\mathbf{k}_\text{j}) \rangle = \langle \mathbf{q}_\text{i}^{\text{rot}}, \mathbf{k}_\text{j}^{\text{rot}} \rangle, \tag{5}$$

where the vectors $\mathbf{q}^{\text{rot}}, \mathbf{k}^{\text{rot}}$ are generated by applying a dimension-wise rotary transformation that encodes their positions in space and time. RoPE introduces no learnable parameters, yet allows attention weights to respond to directional and sequential shifts.

Relative3D is further added to complement RoPE3D by capturing fine-grained, asymmetric, or dataset-specific patterns. For a pair of tokens i and j in a window, we calculate their relative offset $\Delta t, \Delta h, \Delta w$ along the spatial-temporal dimensions and retrieve a scalar bias $\mathbf{B}_{\Delta\text{t},\Delta\text{h},\Delta\text{w}}$:

$$\text{Attention}_{\text{i,j}} \leftarrow \text{Attention}_{\text{i,j}} + \mathbf{B}_{\Delta\text{t},\Delta\text{h},\Delta\text{w}}. \tag{6}$$

Together, RoPE3D provides generalizable inductive structure, while the learnable bias in Relative3D adapts to the statistical characteristics of gait sequences. We empirically find that this hybrid design improves both convergence and final accuracy compared to using either method alone.

Dataset	Convolutional Stem		Short-Range Mixer			Long-Range Mixer	
	Blocks	*Strides*	*Window*	*Heads*	*Blocks*	*Heads*	*ClsToken*
Gait3D	$[C64, R128, R256]$	[1, 1, 2]	$4\times4\times4$	8	10	16	16
GREW					14		
CCGR-MINI					10		32

Table 1: Backbone settings. (1) **Convolutional Stem**: $C64$ denotes a 3D convolutioinal layer with 64 output channels; $R128$/$R256$ are 3D residual blocks with 128/256 channels. *Strides* refer to spatial strides. (2) **Short-Range Mixer**: *Window* is the self-attention cube size over temporal, height, and width dimensions; *Heads* is the attention heads; *Blocks* is the number of token mixer blocks. (3) **Long-Range Mixer**: *Heads* denotes the attention heads; *ClsToken* is the number of class tokens per part.

Summary Positional encoding is essential to the success of Transformer architectures. However, most existing designs, particularly learnable embeddings, assume access to large-scale pretraining data. Our hybrid design addresses this limitation by injecting inductive structure into the backbone, making the model more robust to gait data scarcity. We believe this design substantially bridges the gap between Transformer architectures and real-world gait datasets.

4 Experiments

Settings

Datasets We conduct our experiments on three large-scale gait datasets: Gait3D (Zheng et al. 2022), GREW (Zhu et al. 2021), and CCGR-MINI (Zou et al. 2024). Gait3D and GREW are collected in unconstrained real-world environments, where variations such as camera viewpoints, clothing changes, carrying conditions, lighting, and occlusions pose significant challenges for gait modeling. CCGR-MINI represents the most comprehensive collection of data captured under controlled conditions, featuring 33 viewpoints and 53 covariates per subject. For all datasets, we follow the official protocols for training and evaluation in all our experiments.

Implementations The backbone configurations for each dataset are summarized in Table 1. Following the convolutional stem, an additional 3D convolutional layer with kernel size $[3, 3, 3]$, stride $[1, 2, 2]$, and padding $[1, 1, 1]$ is applied to complete the tokenization stage, resulting in 16×12 tokens per frame. The embedding dimension is uniformly set to 288 across all datasets[3]. In the Short-Range Mixer, we do not perform any spatial downsampling. In the Long-Range Mixer, the number of body parts is fixed to 16. More details are provided in the appendix and code will be released.

Performance Comparison

Gait3D and GREW The performance comparisons on Gait3D and GREW are presented in Table 2 and Table 3, respectively. To facilitate a clearer understanding, we categorize existing methods into three groups based on the primary types of layers utilized in their backbone architectures:

[3]The embedding dimension must be divisible by 6 to satisfy the structural constraints of 3D RoPE (Su et al. 2024), and also by the number of attention heads (*e.g.*, 8 in the Short-Range Mixer) to facilitate the stack of multi-head self-attention.

Method	Backbone (*main layer*)	End-to -End	Gait3D R1	Gait3D mAP
GaitSet (Chao et al. 2019)	2D Conv	√	36.7	30.0
GaitPart (Fan et al. 2020)			28.2	21.6
GaitBase (Fan et al. 2023b)			64.6	55.3
DeepGaitV2-2D (Fan et al. 2023a)			68.2	60.4
GaitGL (Lin, Zhang, and Yu 2021)	3D/P3D Conv	√	29.7	22.3
GaitGCI (Dou et al. 2023)			50.3	39.5
DANet (Ma et al. 2023)			48.0	/
DyGait (Wang et al. 2023c)			66.3	56.4
HSTL (Wang et al. 2023a)			61.3	55.5
DeepGaitV2-3D (Fan et al. 2023a)			72.8	63.9
DeepGaitV2-P3D (Fan et al. 2023a)			74.4	65.8
VPNet (Ma et al. 2024)			<u>75.4</u>	/
CLTD (Xiong et al. 2024)			69.7	/
GaitMoE (Huang et al. 2024)			73.7	66.2
SwinGait (Fan et al. 2023a)	Transformer	×	75.0	<u>67.2</u>
GaT (**Ours**)		√	**77.1**	**69.2**

Table 2: Comparison on Gait3D. The results are reported in rank-1 (R1, %) and mean Average Precision (mAP, %). The best result is marked in **bold** and the second best is marked with <u>underline</u>.

2D Convolutional (2D Conv), 3D/Pseudo-3D Convolutional (3D/P3D Conv), and Transformer-based models.

From the comparison results, several key observations can be made:

(1) **Convolution-based backbones remain dominant**. Most existing methods still rely heavily on either 2D or 3D/P3D convolutional layers for feature extraction. In contrast, transformer-based architectures remain relatively underexplored in the gait recognition field.

(2) **GaT establishes new state-of-the-art results**. Our proposed GaT consistently outperforms all existing convolutional and transformer-based methods across both benchmarks. This highlights the superior effectiveness of GaT's architecture in handling in-the-wild gait data.

(3) **End-to-end training without pretraining**. Unlike SwinGait, which requires pretraining and weight transfer from DeepGaitV2, GaT achieves strong performance without relying on any external pretraining. It is trained in an end-to-end manner, further demonstrating its efficiency and practicality.

CCGR-MINI As a recent benchmark, CCGR-MINI is specifically designed to evaluate gait recognition performance under diverse and challenging real-world conditions. As previously noted, it includes 33 camera viewpoints and 53 covariates, covering a broad spectrum of variations in appearance, view angle, and walking behavior. Due to the novelty and difficulty of this dataset, only a limited number of methods have been evaluated on it to date. We summarize the available comparative results in Table 4. Despite the increased complexity of CCGR-MINI, our proposed GaT achieves state-of-the-art performance on this benchmark, demonstrating strong generalization ability and robustness across diverse scenarios.

Ablation Study

Ablation Study on Patch Embdding In Table 5, we conduct an ablation study on Patch Embedding to investigate its impact on the overall performance.

Method	Backbone (*main layer*)	End-to -End	GREW R1	GREW R5
GaitSet (Chao et al. 2019)	2D Conv	√	48.4	63.6
GaitPart (Fan et al. 2020)			47.6	60.7
GaitBase (Fan et al. 2023b)			60.1	75.5
DeepGaitV2-2D (Fan et al. 2023a)			68.6	82.0
GaitGL (Lin, Zhang, and Yu 2021)	3D/P3D Conv	√	47.3	63.6
GaitGCI (Dou et al. 2023)			68.5	80.8
DyGait (Wang et al. 2023c)			71.4	83.2
HSTL (Wang et al. 2023a)			62.7	76.6
DeepGaitV2-3D (Fan et al. 2023a)			79.4	88.9
DeepGaitV2-P3D (Fan et al. 2023a)			77.7	87.9
VPNet (Ma et al. 2024)			<u>80.0</u>	<u>89.4</u>
CLTD (Xiong et al. 2024)			78.0	87.8
GaitMoE (Huang et al. 2024)			79.6	89.1
SwinGait (Fan et al. 2023a)	Transformer	×	79.3	88.9
GaT (**Ours**)		√	**82.6**	**90.6**

Table 3: Comparison on GREW. The results are reported in rank-1 (R1, %) and rank-5 (R5, %). The best result is marked in **bold** and the second best is marked with <u>underline</u>.

Method	Backbone (*main layer*)	CCGR-MINI R1	CCGR-MINI mAP
GaitSet (Chao et al. 2019)	2D Conv	13.8	15.4
GaitPart (Fan et al. 2020)		8.0	10.1
GaitBase (Fan et al. 2023b)		27.0	24.9
GaitGL (Lin, Zhang, and Yu 2021)	3D/P3D Conv	17.5	18.1
DeepGaitV2-P3D (Fan et al. 2023a)		<u>39.4</u>	<u>36.0</u>
GaT (**Ours**)	Transformer	**41.1**	**37.9**

Table 4: Comparison on CCGR-MINI. The results are reported in rank-1 (R1, %) and mean Average Precision (mAP, %). The best result is marked in **bold** and the second best is marked with <u>underline</u>.

In the first part of the table, we explore different architectural designs for convolutional stem. The result in the first row shows that employing a lightweight stack of convolutional layers is helpful for generating high-quality token embeddings from silhouette inputs. From the second to fourth rows, we evaluate several alternative configurations to examine the trade-off between performance and convolutional complexity. Our goal is to minimize reliance on convolutional operations, thereby preserving the transformer-dominant nature of the backbone. The results suggest that minimal convolutional preprocessing can influence the downstream transformer's ability to model gait features effectively. This also opens up opportunities for further research on how to better integrate or replace convolutional stems in gait recognition architectures.

In the second part of Table 5, we compare different normalization strategies applied within the convolutional stem. The results show that Group-Batch Normalization (GBN) achieves the best performance when dealing with domain-specific binary silhouettes. GBN effectively balances the stability of Batch Normalization with the fine-grained feature modeling of Group Normalization, making it a reasonable and effective choice for our setting.

Ablation Study on Token Mixer We conduct a detailed ablation study on the design of Short-Range and Long-Range Mixers in GaT, focusing on its key hyper-parameters.

Convolutional Stem	Normalization Type	Gait3D	
		R1	mAP
-	-	74.1	65.5
$[C64]$	-	74.9	66.3
$[C64, R128]$	Group-Batch Norm	75.9	67.7
$[C64, R128, R256]$	Group-Batch Norm	**77.1**	**69.2**
$[C64, R64, R128, R256]$	Group-Batch Norm	76.9	69.1
$[C64, R128, R256]$	Batch Norm	76.6	69.1
$[C64, R128, R256]$	Instance-Batch Norm	76.0	68.5
$[C64, R128, R256]$	Group-Batch Norm	**77.1**	**69.2**

Table 5: Ablation study on Patch Embedding. R1 (%) for rank-1 accuracy and mAP (%) for mean Average Precision.

Short-Range Mixer		Long-Range Mixer		Gait3D	
$\#Blocks$	$\#Head$	$\#ClsToken$	$\#Head$	R1	mAP
10	8	16	16	**77.1**	**69.2**
8	8	16	16	76.9	68.7
12	8	16	16	75.9	68.9
10	4	16	16	76.1	68.5
10	16	16	16	76.5	69.1
10	8	8	16	76.6	69.2
10	8	32	16	76.3	68.7
10	8	16	8	76.6	69.1
10	8	16	32	76.3	68.8

Table 6: Ablation study on Token Mixer. R1 (%) for rank-1 accuracy and mAP (%) for mean Average Precision.

The results are summarized in Table 6.

Specifically, we examine the impact of the following components: For the Short-Range Mixer, we vary the number of stacked blocks and attention heads to investigate how local spatial aggregation affects performance. For the Long-Range Mixer, we analyze the effect of changing the number of class tokens and the number of heads in global attention layers. The experimental results demonstrate that GaT maintains robust and competitive performance across a broad range of hyper-parameter settings. This suggests that the architecture is not overly sensitive to specific configurations, making it both stable and easy to finetune in practice.

Ablation Study on Positional Encoding In Table 7, we investigate the effects of different combinations of positional encoding strategies, which are critical to GaT in modeling spatial-temporal dependencies from silhouette sequences.

Specifically, we analyze both global and local positional encoding components. (1) For Global Positional Encoding, we find that incorporating 3D learnable positional encoding (Global3D) leads to a significant drop in performance. This degradation is likely attributed to the discrepancy between the sampling strategies used during training (*i.e.*, limited-length sequences) and those during inference (*e.g.*, variable-length sequences). (2) For Local Positional Encoding, we experiment with two approaches: 3D Rotary Positional Encoding (RoPE3D) and 3D Learnable Relative Position Bias (Relative3D). Both methods individually yield competitive performance, validating their ability to capture local spatial and temporal structure within short-range attention windows. Interestingly, when these two encodings are used in combination, we observe further improvement in accuracy.

Global Positional Encoding	Local Positional Encoding	Gait3D	
		R1	mAP
Global2D	RoPE3D+Relative3D	**77.1**	**69.2**
-	RoPE3D+Relative3D	74.8	66.0
Global3D	RoPE3D+Relative3D	69.2	60.8
Global2D	-	75.1	66.4
Global2D	RoPE3D	76.4	68.9
Global2D	Relative3D	75.8	68.3

Table 7: Ablation study on Positional Encoding. R1 (%) for rank-1 accuracy and mAP (%) for mean Average Precision.

Discussion

While GaT demonstrates competitive performance across multiple benchmarks, there are still some limitations that suggest potential directions for future improvement.

First, the current architecture relies on a convolutional stem for early-stage feature extraction. Although we have minimized the use of convolutional layers to preserve the transformer-dominant design, this dependency introduces an implicit inductive bias and partially deviates from a fully token-based processing pipeline. Future work could explore convolution-free tokenization strategies or leverage more flexible patch embedding techniques to further reduce architectural dependency on convolutional priors.

Second, the model performance is sensitive to the choice of positional encoding. As shown in our ablation studies, the use of suboptimal positional encoding strategies could significantly degrade performance. This highlights the need for more robust and generalizable positional encoding mechanisms. In future work, we aim to investigate flexible positional encoding schemes or dynamic positional priors that can adapt to varying sequence lengths and view variations in unconstrained gait recognition scenarios.

Overall, addressing these limitations may lead to even more generalizable and lightweight gait recognition models, and we believe that the GaT framework offers a solid foundation for such future advancements.

5 Conclusion

In this paper, we propose GaT, a novel Transformer-based framework tailored for unconstrained gait recognition. By introducing a hierarchical token mixing architecture that integrates both short-range and long-range dependencies, GaT effectively captures fine-grained motion patterns and global structural cues from silhouette sequences. We further enhance the model with a lightweight convolutional stem for efficient token generation and employ hybrid positional encoding strategies to provide essential spatial-temporal priors. Comprehensive experiments on three challenging benchmarks demonstrate that GaT achieves state-of-the-art performance, surpassing both convolution-based and existing transformer-based methods. We believe that GaT provides a strong foundation for future work in transformer-based gait recognition, and paves the way for more robust, scalable, and real-world applicable gait analysis systems.

Acknowledgments

This work is jointly supported by National Natural Science Foundation of China (62206022, 62276025, 62476027) and the Fundamental Research Funds for the Central Universities (2253200026).

References

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*, 213–229.

Chao, H.; He, Y.; Zhang, J.; and Feng, J. 2019. GaitSet: Regarding gait as a set for cross-view gait recognition. In *AAAI*, volume 33, 8126–8133.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Dou, H.; Zhang, P.; Su, W.; Yu, Y.; Lin, Y.; and Li, X. 2023. Gaitgci: Generative counterfactual intervention for gait recognition. In *CVPR*, 5578–5588.

Fan, C.; Hou, S.; Huang, Y.; and Yu, S. 2023a. Exploring deep models for practical gait recognition. *arXiv preprint arXiv:2303.03301*.

Fan, C.; Hou, S.; Liang, J.; Shen, C.; Ma, J.; Jin, D.; Huang, Y.; and Yu, S. 2025. OpenGait: A Comprehensive Benchmark Study for Gait Recognition towards Better Practicality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Fan, C.; Liang, J.; Shen, C.; Hou, S.; Huang, Y.; and Yu, S. 2023b. OpenGait: Revisiting Gait Recognition Towards Better Practicality. In *CVPR*, 9707–9716.

Fan, C.; Ma, J.; Jin, D.; Shen, C.; and Yu, S. 2024. SkeletonGait: Gait Recognition Using Skeleton Maps. In *AAAI*, volume 38, 1662–1669.

Fan, C.; Peng, Y.; Cao, C.; Liu, X.; Hou, S.; Chi, J.; Huang, Y.; Li, Q.; and He, Z. 2020. GaitPart: Temporal Part-Based Model for Gait Recognition. In *CVPR*, 14225–14233.

Fu, Y.; Meng, S.; Hou, S.; Hu, X.; and Huang, Y. 2023. GPGait: Generalized Pose-based Gait Recognition. In *ICCV*.

Han, J.; and Bhanu, B. 2005. Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2): 316–322.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hou, S.; Cao, C.; Liu, X.; and Huang, Y. 2020. Gait Lateral Network: Learning Discriminative and Compact Representations for Gait Recognition. In *ECCV*.

Huang, P.; Peng, Y.; Hou, S.; Cao, C.; Liu, X.; He, Z.; and Huang, Y. 2024. Occluded Gait Recognition with Mixture of Experts: An Action Detection Perspective. In *ECCV*, 380–397.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 1501–1510.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, volume 37, 448–456.

Jin, D.; Fan, C.; Ma, J.; Zhou, J.; Chen, W.; and Yu, S. 2025. On Denoising Walking Videos for Gait Recognition. In *CVPR*, 12347–12357.

Lin, B.; Zhang, S.; and Yu, X. 2021. Gait Recognition via Effective Global-Local Feature Representation and Local Temporal Aggregation. In *ICCV*, 14648–14656.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved baselines with visual instruction tuning. In *CVPR*, 26296–26306.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. *NeurIPS*, 34892–34916.

Liu, Y.; Sangineto, E.; Bi, W.; Sebe, N.; Lepri, B.; and De Nadai, M. 2021a. Efficient Training of Visual Transformers with Small Datasets. In *NeurIPS*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; and Hu, H. 2022. Video swin transformer. In *CVPR*, 3202–3211.

Luo, H.; Gu, Y.; Liao, X.; Lai, S.; and Jiang, W. 2019. Bag of tricks and a strong baseline for deep person re-identification. In *CVPR Workshops*, 0–0.

Ma, K.; Fu, Y.; Cao, C.; Hou, S.; Huang, Y.; and Zheng, D. 2024. Learning Visual Prompt for Gait Recognition. In *CVPR*, 593–603.

Ma, K.; Fu, Y.; Zheng, D.; Cao, C.; Hu, X.; and Huang, Y. 2023. Dynamic Aggregated Network for Gait Recognition. In *CVPR*, 22076–22085.

Mogan, J. N.; Lee, C. P.; Lim, K. M.; and Muthu, K. S. 2022. Gait-vit: Gait recognition with vision transformer. *Sensors*, 22(19): 7362.

Nam, H.; and Kim, H.-E. 2018. Batch-instance normalization for adaptively style-invariant neural networks. *NeurIPS*, 31.

Pan, X.; Luo, P.; Shi, J.; and Tang, X. 2018. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 464–479.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 5533–5541.

Shen, C.; Fan, C.; Wu, W.; Wang, R.; Huang, G. Q.; and Yu, S. 2023. LidarGait: Benchmarking 3D Gait Recognition With Point Clouds. In *CVPR*, 1054–1063.

Shen, C.; Wang, R.; Duan, L.; and Yu, S. 2025. LidarGait++: Learning Local Features and Size Awareness from LiDAR Point Clouds for 3D Gait Recognition. In *CVPR*, 6627–6636.

Strudel, R.; Garcia, R.; Laptev, I.; and Schmid, C. 2021. Segmenter: Transformer for semantic segmentation. In *ICCV*, 7262–7272.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Teepe, T.; Khan, A.; Gilg, J.; Herzog, F.; Hörmann, S.; and Rigoll, G. 2021. GaitGraph: Graph Convolutional Network for Skeleton-Based Gait Recognition. In *ICIP*, 2314–2318.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 4489–4497.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Wang, L.; Liu, B.; Liang, F.; and Wang, B. 2023a. Hierarchical Spatio-Temporal Representation Learning for Gait Recognition. In *ICCV*.

Wang, L.; Ma, Y.; Luan, P.; Yao, W.; Li, C.; and Liu, B. 2023b. HiH: A Multi-modal Hierarchy in Hierarchy Network for Unconstrained Gait Recognition. *arXiv preprint arXiv:2311.11210*.

Wang, M.; Guo, X.; Lin, B.; Yang, T.; Zhu, Z.; Li, L.; Zhang, S.; and Yu, X. 2023c. DyGait: Exploiting Dynamic Representations for High-performance Gait Recognition. In *ICCV*.

Wang, Z.; Hou, S.; Zhang, M.; Liu, X.; Cao, C.; and Huang, Y. 2023d. GaitParsing: Human semantic parsing for gait recognition. *IEEE Transactions on Multimedia*, 26: 4736–4748.

Wu, Y.; and He, K. 2018. Group normalization. In *ECCV*, 3–19.

Xiao, T.; Singh, M.; Mintun, E.; Darrell, T.; Dollár, P.; and Girshick, R. 2021. Early convolutions help transformers see better. *NeurIPS*, 34: 30392–30400.

Xiong, H.; Feng, B.; Wang, X.; and Liu, W. 2024. Causality-inspired discriminative feature learning in triple domains for gait recognition. In *ECCV*, 251–270.

Ye, D.; Fan, C.; Ma, J.; Liu, X.; and Yu, S. 2024. BigGait: Learning Gait Representation You Want by Large Vision Models. In *CVPR*, 200–210.

Zheng, J.; Liu, X.; Liu, W.; He, L.; Yan, C.; and Mei, T. 2022. Gait Recognition in the Wild with Dense 3D Representations and A Benchmark. In *CVPR*, 20228–20237.

Zheng, J.; Liu, X.; Wang, S.; Wang, L.; Yan, C.; and Liu, W. 2023. Parsing is all you need for accurate gait recognition in the wild. In *ACM MM*, 116–124.

Zhu, Z.; Guo, X.; Yang, T.; Huang, J.; Deng, J.; Huang, G.; Du, D.; Lu, J.; and Zhou, J. 2021. Gait Recognition in the Wild: A Benchmark. In *ICCV*, 14789–14799.

Zou, S.; Fan, C.; Xiong, J.; Shen, C.; Yu, S.; and Tang, J. 2024. Cross-covariate gait recognition: A benchmark. In *AAAI*, volume 38, 7855–7863.