GaitCloud: Leveraging Spatial-temporal Information for LiDAR-base Gait Recognition with A True-3D Gait Representation

Shaoxiong Zhang, Hiromitsu Awano, Takashi Sato
Kyoto University
zhang.shaoxiong.86c@st.kyoto-u.ac.jp

Abstract

Gait recognition using point clouds captured by LiDAR (Light Detection And Ranging) sensors offers better adaptability to variations in walking conditions compared to camera-based methods, due to the precise spatial information captured. However, existing methods typically project the point clouds into a sequence of 2D depth images extended along the time dimension and adopt gait recognition networks optimized for camera-based approaches. This planar projection compromises the integrity of the 3D coordinates (length, width, and depth) and results in severe silhouette deformations with varied observation viewpoints, similar to the camera-based methods.

To better utilize the spatial information in gait point clouds, we propose a true 3D gait representation using efficient point cloud voxelization, termed ***GaitCloud****. Additionally, we explore the unique nature of LiDAR-captured point clouds and present two improved modules adapted to our method, called Layer Encoder (LE) and Horizontal Convolutional Pooling (HCP). Evaluation results using the open-access gait dataset SUSTech1K show that our method outperforms the state-of-the-art, achieving recognition accuracies of 93.1% and 89.2% in cross-view and variance experiments, respectively. These results demonstrate that 3D gait representation based on point cloud voxelization more effectively utilizes spatial information than depth images, offering new possibilities for high-performance LiDAR-based gait recognition. The source code is available at https://github.com/seagrgz/GaitCloud-master.git.*

1. Introduction

Gait, or the walking pattern of humans, has been studied for many years as an essential biometric feature for person identification and motion analysis due to its uniqueness and immutability [15, 16, 27]. In recent years, significant progress [9, 20, 26, 28–30] has been made in camera-based gait recognition, thanks to the successful application of Convolutional Neural Networks (CNNs) in computer vision [18]. This progress enables the unique advantages of gait, such as in-range and non-contact observation, distinguishing it from conventional biometrics like facial recognition and fingerprinting.

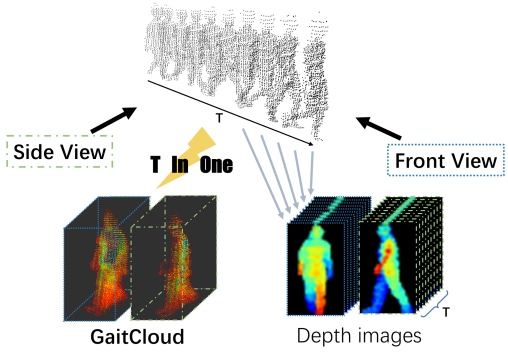


Figure 1. GaitCloud vs depth images. Examples of gait representation for handling gait point clouds captured from different views. Both GaitCloud and depth images represent a gait sequence expended in the time dimension.

A general vision-based gait recognition workflow takes a sequence of temporally continuous gait images as input, as shown in Figure 2a. The recognition network consists of a 2D encoder and a temporal fusion module to extract spatial and temporal features, along with a classifier to predict identities. Some works [12, 26, 31] compress the temporal gait information into a single image by creating a gait template from all the images in the gait sequence. Other studies [20, 30] stack the gait sequence along the temporal dimension to extract geometric and temporal features simultaneously using a 3D CNN.

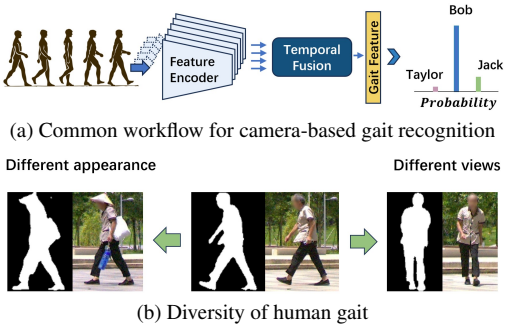
Nevertheless, vision-based gait recognition is susceptible to intra-individual variations due to the ambiguity of individual gait characteristics. Specifically, the silhouettes of the same individual can exhibit significant deformation when captured from different camera viewpoints or influenced by clothing and belongings such as bags and umbrellas, as shown in Figure 2b. Although vari-

ous studies have attempted to mitigate the impact of these variations—such as constructing skeletons [37, 38] or 3D meshes [8, 19, 36, 39] to refine silhouette-based feature extraction, or employing more sophisticated encoding techniques [4, 9, 29, 33, 34]—significant challenges remain in gait recognition using camera-based networks. The fundamental disparity between 2D images and the richly detailed 3D spatial information of the real world continues to hinder the performance of camera-based gait recognition, especially when confronted with substantial deviations from normal gait patterns or heavy occlusions.

With the impressive success of LiDAR (Light Detection and Ranging) in high-accuracy object detection, the potential for gait analysis using sparse point clouds has garnered attention. Additionally, its unique privacy-preserving capability and the independence on object color, texture, and luminance sets it apart from conventional video cameras, which are also in-range sensors. Benedek *et al*. [2, 3, 11] pioneered the first LiDAR-based gait recognition using depth-silhouettes constructed by point cloud planar projection. This success, followed by others [1, 25, 32], demonstrates that the human gait captured by a single LiDAR sensor contains discriminative features of an individual, even with extremely low vertical resolution.

However, all existing studies on LiDAR-based gait recognition use a sequence of depth silhouette images as the model input due to its high interoperability with well-developed camera-based methods. Direct features of human gait, such as the variations in the three-dimensional angles of the leg joints, along with implicit features, such as the height and stride of an individual, are blurred during the planar projection used in this type of gait representation. Moreover, pixel positions and numerical features are encoded differently in the forward propagation of a convolutional network. This discrepancy in encoding hinders the permutability of spatial dimensions—height, width, and depth—each of which equally holds spatial information. As a result, this information loss may reduce the model's adaptability to samples observed from varying viewpoints. Additionally, the imbalance between the vertical and horizontal resolutions in LiDAR-captured point clouds has not been thoroughly investigated.

In this paper, we propose GaitCloud, a simple yet efficient 3D gait representation based on point cloud voxelization. As shown in Figure 1, GaitCloud creates a density distribution of gait sequences over a gait cycle by integrating voxel values from each frame. This compact representation addresses issues of large data volume, low information density, and weak correspondence between points in adjacent frames. As the recognition model, we use 3D residual blocks [14] as the basic encoder blocks to extract spatial-temporal features for metric learning and classification. In addition, we propose a Layer Encoder (LE) block as the input layer to address the resolution gap between vertical and horizontal directions in LiDAR-captured point clouds. We also introduce Horizontal Convolutional Pooling (HCP) based on the Horizontal Pyramid Pooling [10] module to fit our three-dimensional feature map provided by the encoder.



(a) Common workflow for camera-based gait recognition

(b) Diversity of human gait

Figure 2. Illustration of (a) camera-based gait recognition and (b) diversity of human gait. A feature encoder and a time fusion module take gait representation as input and extract gait-related features along spatial and temporal dimensions. This appearance-based gait recognition workflow faces difficulties due to the diversity of human gait.

Our contribution can be summarized as follows: (1) We first introduce voxelization for constructing a gait representation. By utilizing a 3D array to directly represent spatial points, we preserve original spatial information to the greatest extent possible, while maintaining the permutability of the three-dimensional coordinates. (2) We demonstrate the robustness of the voxelized gait representation to viewpoint variations, presenting a LiDAR-captured gait voxelization workflow and corresponding encoder modules. (3) We perform cross-experiments on various gait samples using a large-scale open-access dataset, SUSTech1K [25], surpassing the method proposed in the same paper.

2. Related Works

Gait Recognition. Recent studies on gait recognition can be primarily categorized based on the gait capture modality: camera-based or LiDAR-based. Some works have also explored gait captured by depth cameras [17, 23], which provide images with depth information. However, their gait recognition performance is often limited due to the lower resolution compared to video cameras and the inaccurate spatial information compared to LiDAR sensors.

Camera-based Methods. Most camera-based methods employ gait silhouette images as their model input due to their efficiency and simplicity. Early investigations [26, 31] achieved high cross-view accuracy using a simple CNN-based network, with a 2D gait representation known as Gait Energy Image (GEI) [12]. The GEI calculates pixel-wise energy distribution along the temporal dimension within a gait sequence, revealing potential independence between

individual gait features and the temporal order of the sequence. Recent silhouette-based works [4, 9, 29, 34] successfully adapt their models to appearance variations caused by viewpoints and attributes such as bags and clothing, focusing on dynamic changes in moving body parts. Model-based methods [8,19,36–39] reconstruct 3D representations such as skeletons or 3D meshes from images captured by cameras. Notably, [36, 39] leverage body shape information from the Skinned Multi-Person Linear Model (SMPL) to refine the silhouette encoding, resulting in a more robust model against gait variations. Some studies [13] experimented with utilizing spatial information extracted from LiDAR-captured point clouds, employing a point-wise transformer [6] as an attention mechanism to enhance the adaptability of silhouette-based approaches to gait variations. Although these attempts achieved limited accuracy on diverse gait data captured in a real-world scenario, comparative experiments among camera-based, LiDAR-based, and fusion-based methods indicate that purely LiDAR-based approaches may potentially achieve higher accuracy than image-based methods.

LiDAR-based Methods. Current LiDAR-based gait recognition methods [1, 25, 32] heavily rely on techniques originally developed for camera-based ones, involving the projection of 3D point clouds to construct depth silhouette images. Ahn *et al*. [1, 32] proposed CNN-LSTM-based models focusing on handling variations in LiDAR-captured gait point clouds. Despite attempts to address viewpoint and distance variations by rotating point clouds before planar projection and employing a multi-branch fusion network, these methods achieved limited accuracy due to silhouette image deformations caused by the loss of overlapping parts during projection. Recently, the first benchmark for LiDAR-based gait recognition was established with the large-scale open-access dataset SUSTech1K [25]. This study outperformed state-of-the-art (SOTA) methods at the time, both camera-based and LiDAR-based, in investigating gait variations under different walking conditions (clothing, carrying objects, *etc*.) and changing viewpoints, identified as variance and cross-view experiments. Despite achieving success with a large cohort (1050 individuals), the study [25] continued to use gait representations based on depth silhouette images, suggesting potential for further improvement in recognition accuracy.

3. Gait Recognition with GaitCloud

3.1. Problem Setting

The primary objective of gait recognition is to distinguish individuals based on their walking patterns. Specifically, given a gait $\mathcal{S} = \{S_j^i | i = 1, 2..., N; j = 1, 2, ..., M_i\}$, where $\mathcal{S}$ is the j-th gait sequence for i-th individual. A group of gait sequences $\mathcal{S}_p$ serves as probes whose identities

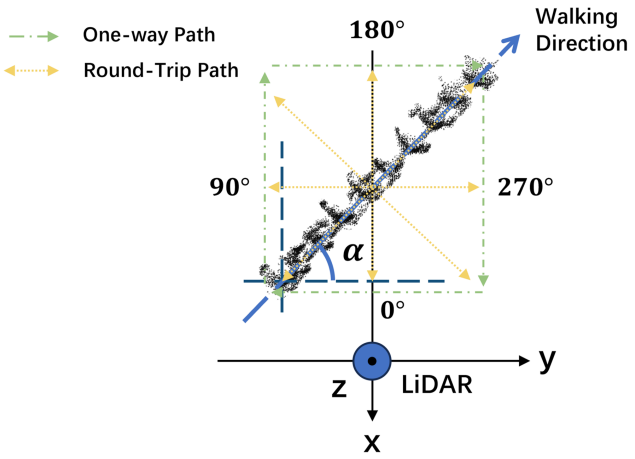


Figure 3. The coordinate system in SUSTech1K [25]. Yellow dotted lines represent round-trip paths with attribute names 000°, 045°, 090°, 135°, 180°, 225°, 270°, and 315°. Green dash-dotted lines represent the one-way paths with attribute names ∗000°, ∗090°, ∗180°, and ∗270°. α is the offset angle of the gait sequence.

are to be predicted by referencing a gait sequence gallery $\mathcal{S}_g$. To achieve this, we define the problem of LiDAR-based gait recognition into three steps: (1) a data processing function $\mathfrak{J}$ transforms raw sequences S into gait samples $\mathcal{V}$, which serve as inputs for the gait recognition network; (2) a feature encoder $\mathfrak{E}$ computes embeddings $\mathcal{G}$ from the gait samples $\mathcal{V}$, and (3) a distance function $\mathfrak{K}$ calculates the distance $\mathcal{D}$ between the embeddings of the probe $\mathcal{G}_p$ and the gallery $\mathcal{G}_g$, which can be formulated as:

$$\mathcal{D}_{g,p} = \mathfrak{K}(\mathcal{G}_g, {\mathcal{G}_p}^T). \tag{1}$$

The identity prediction is determined by the k-nearest neighbors of $\mathcal{G}_p$ in the embedding space of $\mathcal{G}_g$. The entire workflow can be formulated as follows:

$$\mathcal{D}_{g,p} = \mathfrak{K}(\mathfrak{E}(\mathfrak{J}(\mathcal{S}_g)), \mathfrak{E}(\mathfrak{J}(\mathcal{S}_p))^T). \tag{2}$$

We propose GaitCloud, a true 3D gait representation, along with an enhanced model tailored to the unique properties of LiDAR point clouds. We use the open-access dataset SUSTech1K [25] for training and inference, following the protocol outlined in the same study.Figure 3 depicts the coordinate system used in SUSTech1K. Here, α denotes the angle relative to the positive x direction for each sequence.

3.2. 3D Gait Representation

The objective of constructing a 3D gait representation is to transform the raw gait sequences $\mathcal{S}$ with a variable number of points into a fixed-size 3D array $\mathcal{V} \in Z^{w \times l \times h}$, where w, l, h represents the width, length, and height of voxelized gait samples, respectively. Figure 4 sketches the construction process of the proposed gait cloud representation.

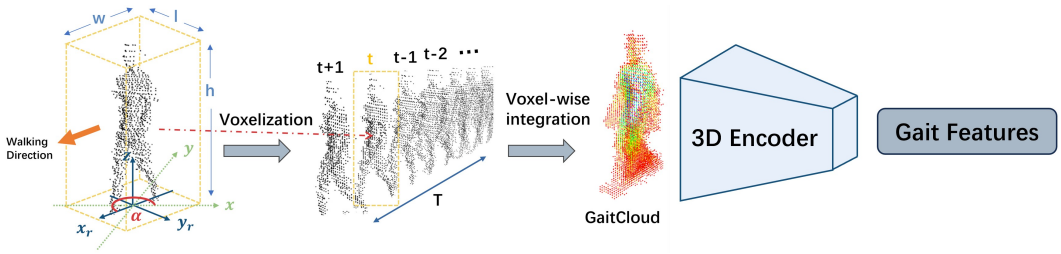


Figure 4. Workflow for constructing a voxelized gait sample (GaitCloud) from a LiDAR-captured gait sequence. GaitCloud conducts voxelization for each frame in the gait sequence with a coordinate system aligned with the walking direction. Voxelized frames are overlapped along time dimension by calculating voxel-wise summation.

3.2.1 Frame-wise Voxelization Protocol

For a raw gait sequence $\mathcal{S} \in R^{L\times P\times 3}$, where L is the sequence length, P is the number of points in one frame, and the last dimension represents the coordinate, x, y, z. We aim to rotate sequences from various viewpoints to face the same direction. To achieve this alignment of the walking direction of each sequence with the positive x-axis, we calculate the rotation angle α using Equations (3) and (4):

$$\alpha = \begin{cases} \alpha' & \text{if} \quad x_n - x_o > 0 \\ \alpha' + \pi & \text{if} \quad x_n - x_o < 0 \text{ and } \mathcal{S} \in \text{View}_{90} \\ \alpha' - \pi & \text{if} \quad x_n - x_o < 0 \text{ and } \mathcal{S} \notin \text{View}_{90} \end{cases} \tag{3}$$

$$\alpha' = \arctan(\frac{y_n - y_0}{x_n - x_0}) \tag{4}$$

Here, $(x_o, y_o) = \text{mean}(x, y) \in \mathcal{S}_o$, and $(x_n, y_n) = \text{mean}(x, y) \in \mathcal{S}_n$ where $\mathcal{S}_o$, $\mathcal{S}_n$ denote the o-th and n-th frames, respectively, with $n\,(\neq o)$ is any integer between o and L. We sample several combinations of o and n within the sequence to compute the average as the offset angle α.

We rotate frames F_l in the sequence using the 3D rotation matrix:

$$[x_r, y_r, z] = [x, y, z] \cdot \begin{bmatrix} \cos(-\alpha) & -\sin(-\alpha) & 0 \\ \sin(-\alpha) & \cos(-\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

where x_r and y_r denote the coordinates after rotation. The z coordinate remains unchanged as the rotation is performed along the z-axis.

For the second step of frame-wise processing, we conduct voxelization on the rotated frames F'_l. To achieve a high-quality gait cloud representation, we select (x_c, y_c, z_{min}) as the voxelization centroid. x_c and y_c are computed as the average coordinates of points in the chest area, which remain relatively stable under various walking conditions, while z_{min} represents the minimum z value indicating the ground level. Then, the voxelization of a single frame can be formed as:

$$\begin{cases} I_x = \frac{x_j - x_c}{x_{\text{res}}} + \frac{w}{2} - 1 \\ I_y = \frac{-y_c}{y_{\text{res}}} + \frac{l}{2} - 1 \\ I_z = \frac{z_j - z_c}{z_{\text{res}}} \end{cases}, \tag{6}$$

where $0 \le I_x, I_y, I_z < w, l, h$ and $[I_x, I_y, I_z]$ is the index of the voxelized point corresponding to the original point $[x_j, y_j, z_j]$ in the rotated frame F'_l, the points whose indices fall outside the voxelization boundary (w, l, h) are ignored. x_{res},y_{res},z_{res} are the resolutions of voxelization in the respective dimensions. During the voxelization process, each index of the voxelized points is assigned a placeholder with an arbitrary fixed value, serving as a counter for the representation of point density distribution which will be explained next.

3.2.2 Temporal Fusion

To preserve gait diversity in the voxelized gait samples and ensure consistency in comparative experiments, we employ random temporal cropping on each sequence to extract a continuous gait sequence of fixed length T. Specifically, for each sequence $\mathcal{S} \in R^{L\times P\times 3}$, if the sequence length $L > T$, we simply extract a segment of length T from $\mathcal{S}$. For sequences $\mathcal{S}$ shorter than T, we append $\mathcal{S}$ to itself until its length exceeds T, and then apply the same cropping process.

Once the frame-wise voxelization and temporal cropping are completed, the raw sequence $\mathcal{S}$ is transformed into a sequence of voxelized frames $V \in Z^{T\times w\times l\times h}$, where T is the sequence length. The final voxelized gait representation $\mathcal{V}$ is obtained by summing voxel-wise along the sequence as shown in Figure 4. This can be formulated as:

$$\mathcal{V}_{x,y,z} = \sum_{t=1}^{T} V_{t,(x,y,z)}. \tag{7}$$

3.3. Model Structure

Building upon the residual network backbone designed for a depth silhouette image input [25], we design a model

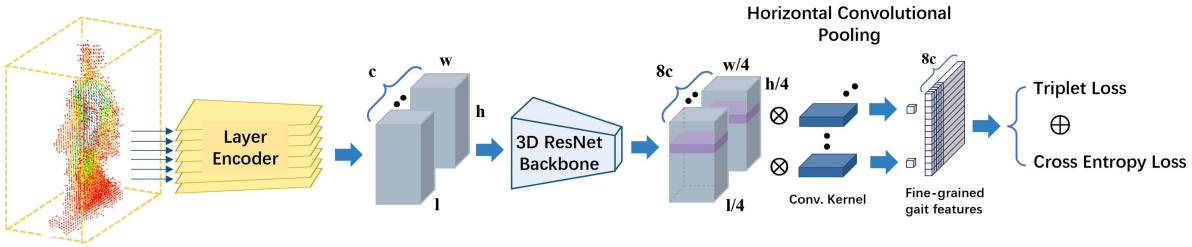
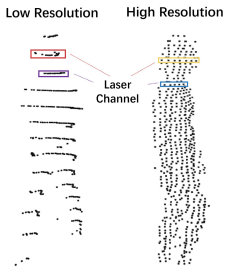


Figure 5. Overall structure of our gait recognition network. We implement a layer-wise 2D encoder LE to extract dense horizontal features. A 3D residual network is used to filter the fine-grained spatial gait features. We use an HCP module instead of an HPP to extract potential position features within each part of the 3D feature maps.

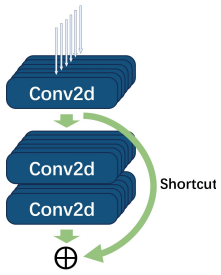
structure tailored for GaitCloud. Specifically, we remove the set pooling branch, as temporal feature representation is inherently embedded with spatial features. In addition, we extended the original ResNet9 architecture [14] to accommodate the 3D array input, incorporating two simple modifications: a Layer Encoder (LE) and Horizontal Convolutional Pooling (HCP). The overall model structure is shown in Figure 5. The model takes the GaitCloud representation $\mathcal{V}$ a 3D array as input. The LE is concatenated with a 3D encoder consisting of four stacked residual blocks, which filter fine-grained gait features. HCP serves as a bottleneck network to extract in-plane features and reduce the dimensionality of the feature maps. In the output stage, the embeddings of the input gait, used for metric learning and inference, pass through a single-layer fully connected network. Classification features are computed on an extended branch via BNNeck [22] from these embeddings.

Layer Encoder. Figure 6a demonstrates the disparity between the vertical and horizontal resolutions typically observed in LiDAR-captured point clouds. This characteristic results in points captured by the same laser channel being naturally grouped into a layered structure, leading to a higher correlation among points within the same layer. To address this challenge, we designed LE to extract 2D features within each layer of the 3D input, serving as the input layer for the entire model. As shown in Figure 6b, the LE comprises a 3D convolutional layer followed by a 3D residual block where the kernel height across all layers is set to 1. This module transforms the input into high-dimensional intra-layer features, effectively mitigating information imbalances among different dimensions.

3D Encoder Backbone. We employ a 3D residual network as the backbone of the 3D encoder, building on the successful approach of LidarGait [25]. Specifically, we concatenate four residual blocks, each with the same structure as ResNet9. The 3D encoder processes the encoded input samples provided by the input layer and converts them into high-dimensional feature maps. These maps are expected to contain fine-grained features that can effectively differentiate individual gaits, accommodating both small inter-



(a) Hierarchical properties of gait point clouds.



(b) Detailed structure of LE.

Figure 6. (a) Hierarchical properties of gait point clouds. Points within a square box are captured by the same laser channel. (b) Detailed structure of Layer Encoder (LE). The 3D input is sliced vertically at the input and assembled back into a 3D array.

subject differences and large intra-subject differences.

Horizontal Convolutional Pooling. To enhance the physical interpretability of distance computations between embeddings, we adopt Horizontal Pyramid Pooling (HPP) [10], which is an effective module originally designed for person re-identification to refine channel-wise features and reduce the dimensionality. It extracts channel-wise local features at multiple scales by summing the results of mean and max operations within segments of the feature map, which are vertically divided into different numbers of bins. In this study, the raw feature maps include an additional dimension compared to the image-based methods, where spatial information is conveyed by voxel positions rather than the voxel values themselves. Capitalizing on this distinction, we implement the original HPP by replacing the mean operation with a 1-layer convolution to calculate weighted sums instead of means, aiming to extract potential positional features within the feature maps.

3.4. Training and Inference

The triple margin loss [24], which is commonly used in metric learning, depends on the distance between the misclassified gallery-probe pairs, but the classification relies on the absolute distance between different classes. That means

Model	Input	Probe Attributes (Rank-1 accuracy)								Overall	Views
		Normal	Bag	Clothing	Carrying	Umbrella	Uniform	Occlusion	Night	$Rank1$	$Rank1$
GaitBase [8]	Silhouettes	81.46	77.48	49.60	75.77	75.55	76.66	81.40	25.92	76.12	64.53
LidarGait (Reprod.)	Depth Images	84.45	82.84	64.58	81.11	48.66	70.06	87.93	83.84	77.99	83.05
LidarGait [25]		**91.80**	88.64	74.56	89.03	67.50	80.86	**94.53**	90.41	86.77	87.24
Ours	GaitCloud	89.82	**90.3**	**79.17**	**89.65**	**85.79**	**89.02**	94.39	**90.62**	**89.20**	**93.10**

Table 1. Rank-1 accuracy on SUFTech1K for variance and cross-view experiments. LidarGait (Reprod.) indicates the results observed from our reproduction. The last column shows the average accuracy in the cross-view experiment, while the other columns show the accuracy in the variance experiment. The Normal attribute represents additional normal samples for the probe set with the identifier 01-nm.

the triplet margin loss might become excessively small, potentially slowing the weight updates even when the model is not fully trained. To address this issue, we adopt a combined loss strategy [21]. This approach integrates the cross-entropy loss to complement and reinforce the triplet margin loss. Specifically, the cross-entropy loss is scaled using a predefined weight to align its scale with that of the triplet margin loss, following the approach outlined in [25]. During training, the triplet margin loss is computed independently for each segment of the embeddings, and the mean of these segment losses is taken as the final loss value [5].

During inference, Euclidean distances are computed separately for each segment of the embeddings, and the mean distance is evaluated to make predictions, analogous to the approach used in loss calculation.

4. Experiments

We verified the robustness of our approach across diverse human gait using the comprehensive attributes available in SUSTech1K [25], including Normal, Bag, Clothing, Carrying, Occlusion, Night, Uniform, Umbrella, and Views. Specifically, the Normal attribute is further subdivided into two: sequences labeled '00-nm' and '01-nm,' representing variations in gait during inference. Our experiments were divided into two groups: the cross-view experiment, focusing on the attribute of Views, and the variance experiment, examining other attributes identified as condition variations.

4.1. Evaluation Protocol

SUSTech1K comprises gait data from a total of 1050 identities, each characterized by one normal attribute and at least one varied attribute, captured from 12 viewpoints (round-trip paths: $000°$, $045°$, $090°$, $135°$, $180°$, $225°$, $270°$, $315°$, unidirectional paths: $*000°$, $*090°$, $*180°$, $*270°$), as shown in Figure 3. Following the dataset partition outlined in the benchmark, we allocated 250 identities with 6010 sequences for training and reserved the remaining 800 identities with 19213 sequences for testing. The gait sequences in the test set are categorized into gallery and probe sets based on their attributes, depending on the experiments conducted. Inference is executed after each epoch (750 iterations in our setup), computing Rank-1 accuracy across all trials. The model is considered to reach the best performance on which epoch the highest mean accuracy across all attributes is observed. We used 3D expanded GaitBase [8] without Set Pooling [5] as the baseline model structure. LE is implemented as an input encoder, replacing the single-layer CNN in the baseline model.

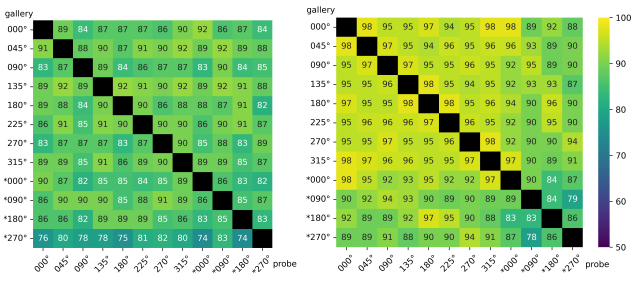
4.2. Implementation Details

During the construction of the GaitCloud representation, we set the dimensions of w, l, and h of the gait cloud $\mathcal{V} \in Z^{w \times l \times h}$ to 40, 40, and 64, respectively. The resolution across all three dimensions is 0.03125 m, resulting in the voxel sizes of 1.25 m, 1.25 m, and 2 m for width, length, and height, respectively. The placeholder voxel value is 1, and the length T of the voxelized gait sequences $V \in Z^{T \times w \times l \times h}$ to construct a gait sample is set to 20.

We use the SGD optimizer with an initial learning rate of 0.1, a momentum of 0.9, and a weight decay of 0.0005 across all experiments. The learning rate is decreased by a factor of 0.1 at 40% and 70% of the entire training process. The total iteration number is set to 75000, equivalent to 100 epochs with a training batch size of 8. For inference, we use a batch size of 64 to balance time and memory consumption. We store the gait sample in the data type of Uint8 and run models in bfloat16 to accelerate the training and inference.

4.3. Comparative Results

We compare the recognition accuracy of our method with the SOTA LiDAR-based method, LidarGait [25], and the camera-based method, GaitBase [8], in variance and cross-view experiments. Point-based methods such as PointNet [6] and PointTransformer [35] are excluded due to their inadequate performance in gait recognition tasks involving large intra-class variations. In the variance experiment, the gallery set comprises all samples from the Normal subset, while the varied samples are grouped by their attributes as the probe set. In the cross-view experiment, each view is alternatively used as the gallery set, with the remaining views serving as separate probe sets. Notably, the average accuracy of cross-view experiments is calculated due to balanced sample numbers across all views, and



(a) LidarGait (b) GaitCloud (ours)

Figure 7. Cross-view performance comparison between LidarGait and GaitCloud. Rank-1 accuracy is evaluated for all trials except those with the same gallery and probe. "*" indicate the views from one-way paths.

different galleries are used to test the probe sets.

Variance Experiment. According to the evaluation results from the variance experiment, detailed in Table 1, our GaitCloud achieves SOTA results across most conditions, particularly excelling in the umbrella subset, which exhibits the most pronounced shape variations compared to normal samples. This underscores GaitCloud's superior robustness to significant appearance changes compared to planar projection-based gait representations. The lowest recognition accuracy is observed with the clothing attribute, maintaining a gap of approximately 5% compared to the Umbrella subset. This challenge primarily stems from the difficulty in eliminating the global interference from clothing by averaging vertical segments of embeddings. A slight decrease in accuracy was observed for the Normal and Occlusion attributes compared to results reported by LidarGait. We attribute these to the imbalanced data distribution of the training set, detailed in **SuppMat:Sect.1.2**. These imbalances lead the model to prioritize adapting to attributes with larger sample populations, thereby marginally improving overall accuracy across the entire training set while slightly compromising accuracy for minor attributes.

Cross-view Experiment. Figure 7 presents results from the cross-view experiment comparing LidarGait with our proposal. The evaluation shows that our method achieves extremely high accuracy for groups featuring round-trip paths. While certain groups (notably $*090°$ and $*270°$) exhibit negligibly lower accuracy compared to LidarGait, we consistently achieve around 90% accuracy in the majority of groups containing one-way paths.

Data Volume of Gait Representation GaitCloud and a depth image sequence of length 10, commonly used in image-based gait recognition, exhibit comparable data sizes. The data size of a depth image sequence scales with the number of frames used in a sample, whereas that of GaitCloud remains constant. Detailed comparisons are provided in **SuppMat:Sect.2**.

Model	Input	Variance	View
LidarGait [8]	Depth Images	86.77	87.24
Baseline	GaitCloud	89.19	92.30
Baseline+LE		88.99	92.73
Baseline+LE+HCP		89.20	93.10
Baseline	GaitCloud (non-rotated)	76.97	70.95
Baseline+LE		84.71	83.78
Baseline+LE+HCP		89.71	90.71
3D-LidarGait	GaitCloud (temporal)	89.37	93.17

Table 2. Rank-1 accuracy across different network structures and pretreatments.

5. Ablation study

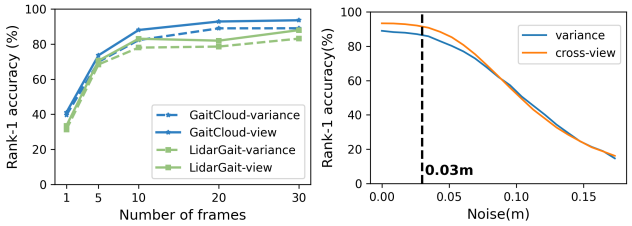
5.1. Effectiveness of 3D Gait representation

To verify the effectiveness of the 3D Gait representation proposed for achieving high-performance gait recognition, we compare the accuracy of both variance and cross-view experiments using rotated, non-rotated, and time-expanded GaitCloud. The same feature encoder described in Section 3.3 is used in all groups. For the time-expanded GaitCloud, we reconstruct LidarGait using 3D encoder layers to adapt to the three-dimensional input.

As shown in Table 2, the baseline model with rotated gait input shows a notable accuracy improvement of +12.22% in the variance experiment and +21.35% in the cross-view experiment. For the model with the proposed modules implemented, which achieves comparable accuracy in the variance experiment with rotated and non-rotated input, an accuracy improvement of +2.39% is observed in the cross-view experiment. We attribute these improvements to the elimination of the need for the model to adjust to view differences among samples with the same label when using rotated point clouds. Instead, variations manifest as horizontal incompleteness in the point clouds, to which CNN-based models are known to exhibit greater robustness [7]. 3D-LidarGait outperforms the original version and even our proposal in variance and cross-view experiments due to its spatio-temporal completeness. However, the extension of the 3D representation in the temporal dimension significantly increases the computational complexity, which is about 7 times higher than that of GaitCloud. A detailed comparison is available in **SuppMat:Sect.2.1**.

5.2. Effectiveness of Proposed Modules

To evaluate the effectiveness of model modifications tailored to the unique characteristics of LiDAR-captured point clouds, we sequentially compared the impact of each proposed module, namely LE and HCP, on recognition accuracy in both variance and cross-view experiments. As



(a) Frame number. (b) Measurement error.

Figure 8. Recognition accuracy using gait samples with (a) varying number of frames and (2) adding Gaussian noise with different standard deviations.

shown in Table 2 and mentioned in Section 5.1, different network structures present indistinguishable recognition accuracy in both variance and cross-view experiments with rotated gait samples. However, when using non-rotated samples, the model incorporating all proposed modules achieves higher performance in both experiments. This indicates that the positional encoding properties of LE and HCP enable the model to better adapt to variations in viewpoint, although it still lags behind the groups using gait rotation in the cross-view experiment.

According to the experiments, we found the triple margin loss of the baseline model suffers from severe divergence during training, and even fails to converge with the initial learning rate when reducing the number of residual blocks. This issue was mitigated by adding LE and HCP. In the follow-up experiments with the base learning rate set to 0.05, the accuracy remains stable until the number of residual blocks is reduced to 1. At this point, the model with the proposed modules achieves 4.95% better accuracy than the baseline in the cross-view experiment. This indicates that adopting LE and HCP models helps to balance the parameters across training samples with substantial variation. Detailed results can be found in **SuppMat:Sect.2.3**.

5.3. Impact of Frame Number and Measurement Precision

The raw sequence length used for constructing the GaitCloud representation is 20, whereas for LidarGait, it is 10. Given this distinction, we evaluated the recognition accuracy using different gait sequence lengths of 1, 5, 10, 20, and 30. Additionally, we evaluate the robustness of the proposed framework to point-wise variation, which may arise from LiDAR measurement errors or practical non-reproducibility, by adding Gaussian noise with different standard deviations. Figure 8 shows the evaluation results. The recognition accuracy of both GaitCloud and LidarGait improves as the number of frames increases. Regardless of the number of frames, GaitCloud consistently outperforms LidarGait in both variance and cross-view experiments. Furthermore, GaitCloud is highly robust to blurred LiDAR measurements up to 0.03 m. GaitCloud representations with varying numbers of frames are shown in **SuppMat:Sect.1.3**

6. Discussion

Ethical Discussion. The SUSTech1K gait dataset used in this study is open-access and available upon signing a dataset release agreement. All subjects associated with the dataset have been informed of and have consented to the collection, processing, use, and sharing of data for research purposes. LiDAR-captured gait point clouds are considered unlikely to pose privacy concerns since they are nearly unrecognizable to the naked eye. Furthermore, the integration of the temporal dimension in the GaitCloud representation further obscures any contour information within the gait that could be directly recognizable by a human.

Limitation. The gait data in SUSTech1K is acquired using a 128-beam LiDAR, providing dense point clouds over a long range. This ensures that the quality of the gait representation used in LidarGait does not significantly degrade within a short range of 5 to 10 meters. However, for gait samples captured from a longer distance or by a low-resolution LiDAR sensor with 16 or 32 beams, the accuracy of LiDAR-based recognition may suffer significantly due to point density variations, similar to the challenges faced with low-resolution images in camera-based gait recognition.

7. Conclusion

In this paper, we introduced a novel true 3D gait representation, GaitCloud, for LiDAR-based gait recognition, replacing the conventional method that creates 2D depth images. This is the first study to explore an efficient voxelization protocol for gait point clouds with sparse information, reducing the data size of 3D gait samples to a level comparable to that of depth image sequences. Additionally, we analyzed the unique properties of LiDAR-captured gait point clouds and proposed two encoder modules, Layer Encoder (LE) and Horizontal Convolutional Pooling (HCP), to utilize the rich temporal-spatial information in GaitCloud.

The combination of GaitCloud with the proposed modules achieved the highest accuracy on the open-access LiDAR-based gait dataset, SUSTech1K [25]. GaitCloud demonstrated significant improvements in adapting to the diversity of human gait in both variance and cross-view experiments. The use of point cloud rotation to compensate for viewpoint variations, either alone or in conjunction with proposed modules, yielded outstanding results across most attributes, particularly the Umbrella subset where LidarGait struggled. This demonstrated the feasibility of using a true 3D gait representation for high-accuracy recognition and proved the effectiveness of the proposed modules in enhancing the performance of gait recognition models.

References

[1] Jeongho Ahn, Kazuto Nakashima, Koki Yoshino, Yumi Iwashita, and Ryo Kurazume. 2V-Gait: Gait recognition using 3D LiDAR robust to changes in walking direction and measurement distance. In *Proc. IEEE/SICE International Symposium on System Integration (SII)*, pages 602–607, 2022. 2, 3

[2] Csaba Benedek, Bence Gálai, Balázs Nagy, and Zsolt Jankó. Lidar-based gait analysis and activity recognition in a 4D surveillance system. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(1):101–113, 2018. 2

[3] Csaba Benedek, Bálazs Nagy, Bence Gálai, and Zsolt Jankó. Lidar-based gait analysis in people tracking and 4D visualization. In *Proc. European Signal Processing Conference (EUSIPCO)*, pages 1138–1142, 2015. 2

[4] Tianrui Chai, Annan Li, Shaoxiong Zhang, Zilong Li, and Yunhong Wang. Lagrange motion analysis and view embeddings for improved gait recognition. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 20217–20226, 2022. 2, 3

[5] Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng. Gaitset: regarding gait as a set for cross-view gait recognition. In *Proc. the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. 6

[6] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 77–85, 2017. 3, 6

[7] Valentin Delchevalerie, Adrien Bibal, Benoît Frénay, and Alexandre Mayer. Achieving rotational invariance with bessel-convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 28772–28783, 2021. 7

[8] Chao Fan, Junhao Liang, Chuanfu Shen, Saihui Hou, Yongzhen Huang, and Shiqi Yu. OpenGait: Revisiting gait recognition towards better practicality. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 9707–9716, June 2023. 2, 3, 6, 7

[9] Chao Fan, Yunjie Peng, Chunshui Cao, Xu Liu, Saihui Hou, Jiannan Chi, Yongzhen Huang, Qing Li, and Zhiqiang He. Gaitpart: Temporal part-based model for gait recognition. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 14213–14221, 2020. 1, 2, 3

[10] Yang Fu, Yunchao Wei, Yuqian Zhou, Honghui Shi, Gao Huang, Xinchao Wang, Zhiqiang Yao, and Thomas Huang. Horizontal pyramid matching for person re-identification. *Proc. AAAI Conference on Artificial Intelligence*, 33(01):8295–8302, Jul. 2019. 2, 5

[11] Bence Gálai and Csaba Benedek. Feature selection for lidar-based gait recognition. In *Proc. International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM)*, pages 1–5, 2015. 2

[12] J. Han and Bir Bhanu. Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):316–322, 2006. 1, 2

[13] Xiao Han, Peishan Cong, Lan Xu, Jingya Wang, Jingyi Yu, and Yuexin Ma. LiCamGait: Gait recognition in the wild by using LiDAR and camera multi-modal visual sensors, 2022. 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 770–778, 2016. 2, 5

[15] Fabian Horst, Sebastian Lapuschkin, Wojciech Samek, Klaus-Robert Müller, and Wolfgang I. Schöllhorn. Explaining the unique nature of individual gait patterns with deep learning. *Scientific Reports*, 9(1), feb 2019. 1

[16] F. Horst, M. Mildner, and W.I. Schöllhorn. One-year persistence of individual gait patterns identified in a follow-up study – a call for individualised diagnose and therapy. *Gait & Posture*, 58:476–480, 2017. 1

[17] Wonjin Kim and Yanggon Kim. Human body model using multiple depth camera for gait analysis. In *Proc. IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 70–75, 2018. 2

[18] Yann LeCun, Koray Kavukcuoglu, and Clement Farabet. Convolutional networks and applications in vision. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 253–256, 2010. 1

[19] Xiang Li, Yasushi Makihara, Chi Xu, and Yasushi Yagi. End-to-end model-based gait recognition using synchronized multi-view pose constraint. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. Worksh. (ICCVW)*, pages 4089–4098, 2021. 2, 3

[20] Beibei Lin, Shunli Zhang, and Feng Bao. Gait recognition with multiple-temporal-scale 3D convolutional neural network. In *Proc. the 28th ACM International Conference on Multimedia*, MM '20, page 3054–3062, New York, NY, USA, 2020. 1

[21] Beibei Lin, Shunli Zhang, and Xin Yu. Gait recognition via effective global-local feature representation and local temporal aggregation. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 14628–14636, 2021. 6

[22] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. Worksh. (CVPRW)*, pages 1487–1495, 2019. 5

[23] Johannes Preis, Moritz Kessel, Martin Werner, and Claudia Linnhoff-Popien. Gait recognition with kinect. In *Proc. Workshop on Kinect in Pervasive Computing*, 01 2012. 2

[24] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 815–823, 2015. 5

[25] Chuanfu Shen, Fan Chao, Wei Wu, Rui Wang, George Q. Huang, and Shiqi Yu. LidarGait: Benchmarking 3D gait recognition with point clouds. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 1054–1063, 2023. 2, 3, 4, 5, 6, 8

[26] Kohei Shiraga, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. GEINet: View-invariant gait recognition using a convolutional neural network. In *Proc. International Conference on Biometrics (ICB)*, pages 1–8, 2016. 1, 2

[27] Erik B. Simonsen and Tine Alkjær. The variability problem of normal human walking. *Medical Engineering & Physics*, 34(2):219–224, 2012. 1

[28] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1505–1518, 2003. 1

[29] Ming Wang, Xianda Guo, Beibei Lin, Tian Yang, Zheng Zhu, Lincheng Li, Shunli Zhang, and Xin Yu. DyGait: Exploiting dynamic representations for high-performance gait recognition. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 13378–13387, 2023. 1, 2, 3

[30] Thomas Wolf, Mohammadreza Babaee, and Gerhard Rigoll. Multi-view gait recognition using 3D convolutional neural networks. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 4165–4169, 2016. 1

[31] Zifeng Wu, Yongzhen Huang, Liang Wang, Xiaogang Wang, and Tieniu Tan. A comprehensive study on cross-view gait based human identification with deep CNNs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):209–226, 2017. 1, 2

[32] Hiroyuki Yamada, Jeongho Ahn, Oscar Martinez Mozos, Yumi Iwashita, and Ryo Kurazume. Gait-based person identification using 3D LiDAR and long short-term memory deep networks. *Advanced Robotics*, 34(18):1201–1211, 2020. 2, 3

[33] Shaoxiong Zhang, Yunhong Wang, and Annan Li. Cross-view gait recognition with deep universal linear embeddings. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 9091–9100, 2021. 2

[34] Yuqi Zhang, Yongzhen Huang, Shiqi Yu, and Liang Wang. Cross-view gait recognition by discriminative feature learning. *IEEE Transactions on Image Processing*, 29:1001–1015, 2020. 2, 3

[35] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 16239–16248, 2021. 6

[36] Jinkai Zheng, Xinchen Liu, Wu Liu, Lingxiao He, Chenggang Yan, and Tao Mei. Gait recognition in the wild with dense 3D representations and a benchmark. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 20196–20205, 2022. 2, 3

[37] Zhu Zheng, Guo Xianda, Yang Tian, Huang Junjie, Deng Jiankang, Huang Guan, Du Dalong, Lu Jiwen, and Zhou Jie. Gait recognition in the wild: A benchmark. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 14769–14779, 2021. 2, 3

[38] Haidong Zhu, Wanrong Zheng, Zhaoheng Zheng, and Ram Nevatia. GaitRef: Gait recognition with refined sequential skeletons. In *2023 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10, 2023. 2, 3

[39] Haidong Zhu, Zhaoheng Zheng, and Ram Nevatia. Gait recognition using 3-d human body shape inference. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 909–918, 2023. 2, 3