GaitContour: Efficient Gait Recognition based on a Contour-Pose Representation

Yuxiang Guo, Anshul Shah, Jiang Liu, Ayush Gupta,
Rama Chellappa, Cheng Peng
Johns Hopkins University, Baltimore, MD, USA
`{yguo87, ashah95, jiangliu, agupt120, rchella4, cpeng26 }@jhu.edu`

Abstract

Gait recognition holds the promise to robustly identify subjects based on walking patterns instead of appearance information. In recent years, this field has been dominated by learning methods based on two input formats: silhouette images and sparse keypoints. Compared to image-based approaches, keypoint-based methods can achieve significantly higher efficiency due to their sparsity. However, sparsity also results in information loss, thereby reducing performance. In this work, we propose a novel, keypoint-based Contour-Pose representation, which compactly encodes both body shape and parts information. We further propose a local-to-global architecture, called GaitContour, to leverage this novel representation and efficiently compute subject embedding in two stages. The first stage consists of a local transformer that extracts features from five different body regions. The second stage then aggregates the regional features to estimate a global human gait representation. Such a design significantly reduces the complexity of the attention operation and improves both efficiency and performance. Through large scale experiments, Gait-Contour is shown to perform significantly better than previous keypoint-based methods. Furthermore, the Contour-Pose representation also achieves new SoTA performances on fusion-based gait recognition methods.

1. Introduction

Unconstrained biometric identification, especially in outdoor and long-range situations, has been a longstanding challenge [36, 39, 51, 52]. While RGB-based face and body recognition systems focus on learning *spatially* discriminative features, real-world effects like challenging viewpoints, low face resolution, changing appearances (*e.g.*, clothes and glasses), *etc.*, can significantly affect model performance [27, 28, 47]. Gait analysis employs an alternative modality for human recognition by learning discriminative

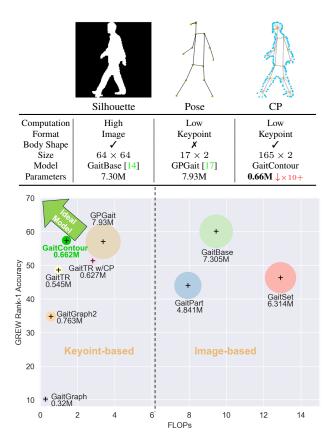|	Silhouette	Pose	CP
Computation	High	Low	Low
Format	Image	Keypoint	Keypoint
Body Shape	✓	✗	✓
Size	64×64	17×2	165×2
Model	GaitBase [14]	GPGait [17]	GaitContour
Parameters	7.30M	7.93M	**0.66M** ↓×10+

Figure 1. Comparison of our proposed Contour-Pose and other gait representations. The size of bubbles denotes the number of parameters. GaitTR w/CP represents extracting Contour-Pose(CP) feature through GaitTR. GaitContour achieves a good balance between efficiency and accuracy.

features extracted from human walking patterns. It can be more robust in challenging, unconstrained situations, where color-space information is unreliable due to turbulence, and has been deployed in many applications including human authentication [4], health [11], crime analysis [20], *etc.*

Research on gait analysis has a long history [37, 39]. The more recent developments in this field are based on deep learning methods, where the format of inputs to the neural

network can be roughly categorized in two ways: silhouette images and keypoints. To extract discriminative features from these two types of data, models of different complexities are applied. Previous works mainly focus on the performance differences from a modality perspective, such as silhouette, keypoints, or multi-modal. This study analyzes the **models' efficiency and effectiveness tradeoff** by using different representations, which is crucial for assessing the implementation cost in real-life applications.

Gait image sequences, which capture human motion as a series of 2D binary silhouettes or skeleton maps, are typically processed by large CNN models. Relying on dense representations, these models require significant computations to generate effective features, which often lead to a latency of 80-100ms. Such cost makes image-based gait recognition less favorable compared to face and body recognition, which takes around 10-20ms, despite its advantages in privacy preservation and performance.

Pose keypoints are predefined semantic points extracted from human images and magnitudes smaller in input size than images. Such smaller dimensionality offers several benefits, including smaller models, faster processing, and smaller template sizes (the identification vectors generated by the model), However, the reduced accuracy of keypoint-based models is a major bottleneck. If keypoint-based models can achieve competitive performance, they can enable broader downstream applications, e.g., gait recognition in low-power, efficient systems with real-time demands.

Considering the advantages and disadvantages of the two gait representations, we pose this question: can we use slightly denser keypoints to represent human movement and improve keypoint-based models' performance? Based on the success of the image-based models, we argue that body shape also plays a significant role in gait recognition. As such, it can be beneficial to extract more keypoints around the contour of a human to complement semantic human poses. To this end, we propose a novel gait representation called Contour-Pose, as illustrated in Fig. 1. We compactly represent the human body shape using a series of contour points around the silhouette, *e.g.*, approximated by Teh-Chin algorithm [43]. However, naively using contour points cannot achieve good performance due to the inconsistent correspondence and ordering between frames. To address this issue, we draw inspiration from the blend skinning process in building the Skinned Multi-Person Linear Model (SMPL) [31], where the final skin vertex locations are highly correlated to the joint centers. We use pose keypoints as anchors to select relevant contour points. Specifically, for every pose keypoint, we select a few close contour points to form a connected graph in a clockwise fashion, simulating the SMPL skinning process in a 2D style. Together, this contour-pose representation compactly represents the semantic regions of the human body and its shape.

Compared to the typical silhouette, our Contour-Pose representation is an order of magnitude smaller in dimension.

When used in place of conventional pose keypoints, Contour-Pose can already improve the performances of prior keypoint-based gait recognition models [41, 42, 49]. However, these models are designed for sparse pose keypoints, and incur higher computational costs due to the larger amount of point inputs in Contour-Pose and the quadratic complexity of Transformers, which are commonly used in motion analysis [3, 35, 49]. Our analysis reveals that a significant portion of the computation in a Transformer is dedicated to relationships that are not usually pertinent, *e.g.*, there is little correlation between the contour points surrounding the head and legs.

In light of this consideration, we propose GaitContour, a Transformer-based method that is designed in a local-to-global fashion to maximize performance and efficiency. GaitContour operates in two stages: a Local Contour-Pose Transformer (Local-CPT), and a Global Pose-Feature Transformer (Global-PFT). As contour points are defined with respect to keypoints, we propose a Local-CPT to extract local features of the specific regions using shared weights. This design reduces the number of model parameters and allows for sharing the general low-level features. Local-CPT's outputs are aggregated to form global keypoint features. The Global-PFT focuses on this sparse set of global keypoint features to generate human IDs.

As shown in Fig. 1, GaitContour achieves a significantly better efficiency-performance trade-off based on richer information and tailored architecture design, compared to previous keypoint-based models [17]. Furthermore, we can leverage Contour-Pose in place of keypoints for image-keypoint-fusion modeling, similar to the setup of Skeleton-Gait++ [15], and achieve new State-of-The-Art results. This further demonstrates the effectiveness of our proposed representation. In summary, our contributions are as follows:

1. We propose a novel gait representation, called Contour-Pose, which augments pose keypoints with contour points extracted from silhouettes; this representation contains rich information, is compact in size, and can improve current keypoint-based gait recognition methods.
2. We propose a novel gait recognition method, called GaitContour, which leverages Contour-Pose and a Transformer-based design; GaitContour processes Contour-Pose in a local-to-global fashion, which maximizes efficiency.
3. We evaluate our novel gait representation and recognition method over several large-scale datasets, and find significant performance and efficiency improvements compared to previous SOTA methods both in keypoint-only and fusion scenarios.

2. Related Works

2.1. Gait Representation

In past decades, researchers have used a variety of representations to capture human gait motion, including RGB images [26, 50], binary masks/silhouettes [8, 16, 30], optical flow images [12, 23, 24], 2D skeleton/pose keypoints [17, 41, 42, 49], and gait-oriented templates, like Gait Energy Image(GEI) [21], Gait History Image [5], *etc*. In recent years, novel gait representations have also emerged, in the form of Li-DAR pointcloud [38], 3D mesh [26, 51], and event stream cameras [46]; but these representations are difficult to compute, and consequently, datasets are limited in scale. The current datasets and methods are mainly based on *images* and *keypoints*, which are the focus of this work.

2.2. Keypoint-based Gait Recognition

Keypoint-based methods are typically designed to predict identities based on 2D pose keypoints across many frames. These methods utilize semantic positions, *e.g.*, knees and wrists, along with their spatial and temporal connections as inputs. By distilling images into semantically meaningful points, keypoint-based methods can be more robust to noisy factors such as different clothing and self-occlusion. GaitGraph [42] and its successor GaitGraph2 [41] treat pose keypoints as a graph and employ a Graph Convolutional Network (GCN) to extract features. GaitTR [49] and GaitMixer [34] capture the global temporal and spatial relationship through a transformer-like [44] architecture. GPGait [17] is based on a Part-Aware Graph Convolutional Network (PAGCN) to explore the keypoint representation under cross-domain settings. [18] applies physics-augmented autoencoder to distill physics knowledge into gait recognition. Even though keypoints perform well in other motion-related tasks such as action recognition [48], they have lower performance in gait recognition compared to image-based methods; but they are generally much faster and more efficient because of sparser inputs.

2.3. Image and Fusion-based Gait Recognition

The human silhouette is a mainstream representation used in current image-based recognition methods, typically based on Convolutional Neural Networks (CNNs). Specifically, GaitGL [30] improves the quality of embeddings further by aggregating both local and global descriptors to capture local details and contextual relations. Recent works have explored different backbones, *e.g.*, GaitBase [14], DeepGaitV2 [13], to achieve higher performance across diverse datasets, especially in outdoor unconstrained scenarios, *e.g.*, GREW [51]. Recently, SkeletonGait [15] transferred the keypoints into a skeleton map, so that a large feature extraction model can be employed, achieving higher performance. Comparatively, image-based meth-

ods take longer to process, e.g. 102ms/sequence for Deep-GaitV2 [13], as they perform convolution operations at every pixel of the sequence, and consist of many layers.

Given the unique characteristics inherent to each modality, several works focus on fusing the inputs. Castro et al. [7] fuse depth map, gray image and optical flow through CNNs to perform gait recognition. DME [19] incorporates RGB images and silhouettes, enriching the feature space expression. SMPLGait [51] integrates 2D and 3D modalities, i.e. silhouette and human mesh, in feature spaces, which have been shown to improve performance. Bifusion [33] and MMGaitFormer [10] fuse silhouette and pose information, employing concatenation and cross attention respectively. These methods primarily concentrate on fusing the modalities in feature spaces, as the extraction of different modalities involves multiple isolated backbones.

3. Method

3.1. Contour-Pose

Inspired by previous works that attempt to combine the body shape and pose in the feature space [10, 33], we look at a more principled approach to extract body shape features without any neural networks to design keypoint-based method with more representation fidelity. Ideally, this novel representation should have the following properties:

- *Information Preservation*: The representation should have minimal information loss. i.e., the original signals can be restored from the representation.
- *Compactness*: The representation should be concise, such that its downstream processing is efficient.
- *Temporal Consistency*: The representation should have consistent ordering across frames.

Pose keypoints are very compact, but do not contain enough information to reconstruct the silhouettes; silhouettes contain more information, but are not as compact. To this end, we propose **Contour-Pose**, which compresses a silhouette to a series of contour points to augment pose keypoints. The contour points at sufficient density preserve most information in a mask, while being compact. The process to produce Contour-Pose is demonstrated in Fig. 3, and is defined formally next.

Suppose we have a subject's silhouette S and pose P across T frames:

$$S = [s_t \in \mathbb{R}^{H \times W}]_{t=1}^T, P = [p_t \in \mathbb{R}^{V \times 2}]_{t=1}^T, \quad (1)$$

where H, W and V represent frame height, width, and the number of keypoints. For each frame, pose keypoints consist of V nodes $[(x_1, y_1), ...(x_V, y_V)]$ in 2D, and E edges between them, which are commonly defined [42, 49].

We hypothesize that a significant portion of the useful information in s_t lies on edges, as similar ideas have been examined before [2, 29, 45]. Instead of using an edge image
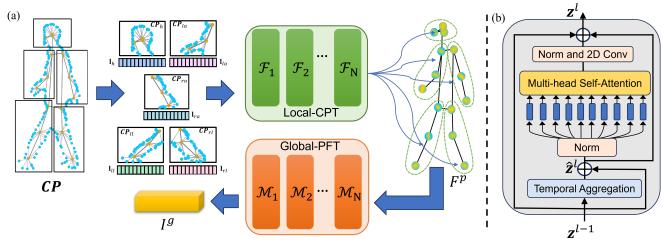
Figure 2. (a) **An Overview of GaitContour**. The Contour-Pose is partitioned into five regions, i.e. head, left arm, right arm, left leg, and right leg. Local-CPT extracts features from each region separately. GaitContour combines these local features into an identity embedding through a Global Pose-Feature Transformer. This local-to-global design enhances both efficiency and effectiveness for GaitContour. (b) **The structure of the Temporal Transformer Layer**. It extracts the spatiotemporal correlation between each point, serving as a basic block for Local-CPT and Global-PFT.
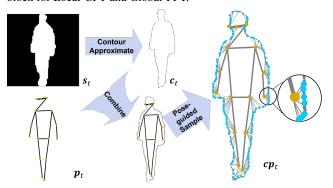

Figure 3. The construction of Contour-Pose. The pose is combined with the contour points sampled from the silhouette edge. In particular, contour points are sampled based on their distances from neighborhood poses. As shown in the zoomed area, Contour-Pose is the n nearby contour points from each pose with connections.

as previous works [29], we sample the points on it, leading to a more compact representation to describe the body shape. To this end, we employ popular **contour approximation** methods [6, 43] to 1). compute the silhouette edge points, and 2). approximate the edge with a lower number of contour points c_t. In practice, 300+ points are extracted on the contour of s_t.

Although the approximated contour compactly contains the silhouette information, in practice, we find that contour points alone do not yield good performance with current keypoint-based gait recognition methods [42, 49]. This is likely due to the lack of temporal consistency in contour point approximation, where the points in neighboring frames do not have the same semantic meaning and may arbitrarily shift based on the approximation algorithm. This consistency helps to build a stable graph to describe the walking pattern.

To establish consistent connections among frames, we draw inspiration from the blending skin process in

SMPL [31], where the skin vertices are controlled by the joint positions. We treat the contour and pose points as analogous to skin and joints in 2D, leveraging pose keypoints as a **semantic guide** to refine contour point selection. For each pose point in p_t, we select n nearby contour points to form a connected graph. To ensure consistency, we arrange the contour points associated with each pose keypoint in clockwise order. The merged contour and pose keypoints, named Contour-Pose, are defined as follows:

$$CP = [cp_t \in \mathbb{R}^{(V \times n + V) \times 2}]_{t=1}^T. \qquad (2)$$

As demonstrated in Fig. 3, Contour-Pose combines the semantic information expressed in poses and the body shape information expressed by silhouettes. By directly applying Contour-Pose on current keypoint-based methods, *e.g.*, GaitTR [49], we can already observe significant improvements in gait recognition performances without any architectural modification, as shown in Table 4 (d-e). For more details on the construction of Contour-Pose, please refer to the supplemental material.

3.2. GaitContour

Contour-Pose already improves the baseline gait recognition performance. We note that Transformer-based methods like GaitTR [49] compute attention in $\mathcal{O}(n^2)$ complexity with respect to the input points. This leads to significantly more operations, as Contour-Pose has $V \times n$ additional points on top of pose keypoints. We propose **GaitContour**, an efficient transformer-based gait recognition model developed for Contour-Pose. As shown in Fig. 2 (a), GaitContour first computes local features in five defined regions with a Local Contour-Pose Transformer (Local-CPT); it then aggregates local features and computes a global representation with a Global Pose-Feature Transformer (Global-PFT). Such a local-to-global design allows each transformer to focus on relevant points and features, thereby significantly

improving efficiency and performance. Both Local-CPT and Global-PFT are built using the Temporal Transformer Layer.

3.2.1 Temporal Transformer Layer

To compute the spatiotemporal correlation between each point, we utilize a Temporal Transformer Layer (TTL) as shown in Fig. 2 (b). It captures the points' movement with time through Temporal Aggregation (TA) and uses a self-attention mechanism to extract the relations between each point. Each layer's operations can be described and formulated as follows:

$$\begin{aligned} \hat{\mathbf{z}}^l &= \mathbf{z}^{l-1} + \text{TA}(\mathbf{z}^{l-1}), \\ \mathbf{z}^l &= \text{BN}(\text{Conv}(\text{MHA}(\text{BN}(\hat{\mathbf{z}}^l)))) + \hat{\mathbf{z}}^l + \mathbf{z}^{l-1}, \end{aligned} \quad (3)$$

where $\hat{\mathbf{z}}^l, \mathbf{z}^l \in \mathbb{R}^{T \times J \times C}$ are the output features of TA and layer l, respectively; T, J, C denotes the frame number, the number of points/tokens and the channel dimension. We follow [49] to use convolutions along the time axis to reason about temporal information in the TA module. This temporal transformer structure is commonly used [3, 35, 49], and is shown to be effective for gait analysis.

3.2.2 Local Contour-Pose Transformer

Contour points in $\boldsymbol{CP}$ are defined in relationship to pose keypoints and represent more detailed body-shape information; To leverage this relationship, we propose to first compute local features based on contour points and keypoints. It stands to reason that local details, *e.g.*, contour shapes in the left foot, bear a minor correlation to contour points surrounding the head. Guided by this hypothesis, we define five body regions based on keypoints - head, left arm, right arm, left leg, and right leg, defined as:

$$\boldsymbol{CP}_r \in \mathbb{R}^{T \times (3+3n) \times 2}, \forall r \in \{h, la, ra, ll, rl\}, \quad (4)$$

each region contains 3 keypoints and each keypoint is associated with n contour points, a total of $(3 + 3n)$ points.

A Local Contour-Pose Transformer (Local-CPT) is a transformer architecture built upon TTL. Denoting every TTL block within Local-CPT to be $\mathcal{F}_i$, the forward process of Local-CPT can be described as follows:

$$F_r = \mathcal{F}_N \circ \mathcal{F}_{N-1} \circ ... \mathcal{F}_2(\mathcal{F}_1(\gamma(\boldsymbol{CP}_r)) \oplus \mathbf{l}_r)), \quad (5)$$

$$F_r^p = AvgPool(F_r, n+1), \quad (6)$$

where $\circ$ represents operate in series, $\mathbf{l}_r \in \mathbb{R}^{1 \times C_1}$ is a learnable regional embedding indicating the current Contour-Pose region, $F_r \in \mathbb{R}^{T \times (3+3n) \times C_N}$ is the output feature vector for every region, C_i is the channel size of $\mathcal{F}_i$ and γ is a sinusoidal embedding function. The embedding function

Table 1. The statistics of four large-scale datasets

Dataset	Id	Seq	Frames/Seq	Distractor
OUMVLP	10,307	288,696	24.88	✗
GREW	26,345	128,671	109.92	✓
Gait3D	4,000	25,309	129.57	✗
SUSTech	1050	25,216	91.61	✗
CCPG	200	16,282	107.14	✗
BRIAR	1,216	84,913	1943.25	✓

allows us to map the 2D positions to a higher dimensional Fourier space which has been shown to help in improved learning [32, 40]. Note that we use a single Local-CPT to process all regional Contour-Pose components, and provide $\mathbf{l}_r$ after $\mathcal{F}_1$ as the region indicator. The features in F_r are then averaged pooled with $(n+1)$ stride to $F_r^p \in \mathbb{R}^{T \times 3 \times C_N}$ to condense the point-wise dimension. Based on this design, every layer in Local-CPT has a complexity of $\mathcal{O}(\frac{n^2}{5})$ by focusing on local parts.

3.2.3 Global Pose-Feature Transformer

Once the regional features are computed, a Global Pose-Feature Transformer (Global-PFT) is used to compute a global ID representation. Similar to Local-CPT, Global-PFT is built on TTL, where each layer is denoted as $\mathcal{M}_i$. The forward process for Global-PFT can defined as follows:

$$I^g = AvgPool(\mathcal{M}_X \circ \mathcal{M}_{X-1} \circ ... \mathcal{M}_1(F^p), V), \quad (7)$$

$$F^p = F_h^p \oplus F_{la}^p \oplus F_{ra}^p \oplus F_{ll}^p \oplus F_{rl}^p, \quad (8)$$

where $F^p \in \mathbb{R}^{T \times 15 \times C_N}$ is the concatenated feature of regional features in F_r^p; Global-PFT takes F^p and to obtain the subject's identification embedding in $I^g \in \mathbb{R}^{1 \times S_X}$, where S_i is the channel size of $\mathcal{M}_i$.

During training, triplet loss [22] is applied to maximize the distance of representations from different identities and minimize the ones from the same identity.

4. Experiments

4.1. Datasets and Metrics

We evaluate our method on several large scale datasets, i.e. OUMVLP [1], CCPG [25], SUSTech1K [38], GREW [52], Gait3D [51] and BRIAR [9]. The first three are constrained datasets and the rest are unconstrained datasets to assess the performance of challenging real-world scenarios. The statistics of these datasets are in Table 1. One highlight of BRIAR is that its video clips are of much longer duration, at around 1900 frames per sequence, compared to around 110 frames per sequence for other gait datasets. BRIAR also has 70 sequences per identity, compared to datasets such as OUMVLP, which has 30 sequences per identity.

Table 2. Quantitative comparison of **keypoint-based** gait recognition methods across six large scale datasets. The best performers are colored, and the second best methods are underlined. GaitTR has a similar compute cost but a lower performance. GaitContour outperforms GPGait with a smaller template size and compute cost. The Ver represents TAR@FAR=10^{-3}.

Method	Params(M)	FLOPs(G)	Template size	Testing Datasets											
				OUMVLP	GREW		Gait3D		BRIAR			SUSTech1K		CCPG Full	
				R1	R1	R5	R1	R5	R1	R5	Ver	R1	R5	R1	mAP
GaitGraph2 [41]	0.76	0.58	1 × 384	62.1	33.5	-	11.1	-	6.8	16.2	1.5	18.6	40.2	5.0	2.4
GaitTR [49]	0.55	0.98	1 × 256	39.8	48.6	65.5	7.2	16.4	42.2	68.7	30.5	30.8	56.0	24.3	9.7
GPGait [17]	7.93	3.38	19 × 256	59.1	57.0	68.5	22.4	35.9	38.0	56.6	32.0	47.4	70.6	54.7	25.8
GaitContour(Ours)	0.66	1.41	1 × 256	60.8	57.4	72.9	25.3	41.3	55.2	74.6	40.0	55.5	72.3	57.8	27.1

This abundance of temporal information allows us to explore trade-offs in real applications, *e.g.*, when image-based methods become too computationally expensive on inputs with a large number of frames.

Evaluation Metric For OUMVLP, Gait3D, SUSTech1K and GREW, *rank retrieval* is employed to evaluate gait recognition performance. For the BRIAR dataset, we also measure performance using Receiver-Operating Characteristics (ROC) curves. For different gait recognition algorithms, Additionally, we compare the number of parameters, Floating Point Operations (FLOPs), and output template size to understand their efficiency in performance and storage. Note that, we only account for the parameters and FLOPs of the encoder. The implementation details are provided in the supplementary material section.

4.2. Quantitative Evaluation

As summarized in Table 2, we compare GaitContour against other SOTA keypoint-based gait recognition algorithms. All evaluation results other than those for BRIAR come from the respective original papers. GaitContour performs significantly better compared to current keypoint-based methods across most datasets. In particular, GaitTR [49] and GaitContour both use a Transformer-based architecture. Despite the 10X larger input size for Contour-Pose, Gait-Contour is similar to GaitTR [49] in model size, FLOPs, and template size, while achieving much better performances across all datasets. Compared to GPGait [17], which is 12/19X larger in model/template size, GaitContour achieves better results. This demonstrates both the rich information in our novel input representation and the efficiency of processing it with GaitContour.

4.3. Ablation Study

GaitContour architecture design. In this work, we proposed three key design concepts: *regional embedding*, *a shared Local-CPT for different regions*, and *sinusoidal embedding*. As shown in Table 3, we perform several ablation studies to confirm their usefulness. We find that regional embedding and sinusoidal embedding improve performance by 1.5% and 1.9% respectively; when both techniques are used together, we achieve a 3.9% improvement

Table 3. Ablation study on region embedding (Region), a shared Local-CPT (Shared CPT), and sinusoidal embedding (Sin). The results are shown using rank retrieval, mean Average Precision(mAP) and mean Inverse Negative Penalty (mINP) with model size. Results are based on Gait3D [51].

Region	Shared CPT	Sin	Rank-1	Rank-5	mAP	mINP	Param(M)
	✓		21.42	37.23	16.38	7.45	0.56
	✓	✓	23.32	40.74	17.48	7.65	0.56
✓		✓	22.92	39.24	17.05	8.30	0.66
✓		✓	23.12	40.84	17.42	7.61	1.78
✓	✓	✓	25.32	41.34	18.62	8.34	0.66

Table 4. Comparison among different keypoint-based inputs and Contour-Pose applied on various keypoint-based methods. Contour-Pose$_{NA}$ stands for a Contour-Pose configuration with no clock-wise arrangement.

Index	Method	Representation	FLOPs(G)	Gait3D
(a)	GaitTR [49]	Pose (17)	0.98	7.2
(b)	GaitTR [49]	Contour (112)	1.81	4.5
(c)	GaitTR [49]	Contour-Pose (165)	2.83	19.3
(d)	GaitGraph [42]	Contour-Pose (165)	1.97	8.7
(e)	GaitGraph2 [41]	Contour-Pose (165)	8.37	16.2
(f)	GaitContour	Contour-Pose$_{NA}$ (165)	1.41	13.9
(g)	GaitContour	Contour-Pose (165)	1.41	25.3

Table 5. Comparison under the small template size or model size.

Model	Parameters(M)	FLOPs(G)	Gait3D	GREW
GaitGL [30]	11.19	58.55	29.7	47.3
GaitBase [14]	7.30	9.45	64.6	60.1
GaitGL$_{tiny}$	0.77	1.45	12.2	17.4
GaitBase$_{tiny}$	0.72	1.62	18.2	3.6
GaitContour	0.66	1.41	25.3	57.4

(a) Comparison under comparable model size

Model	Template size	Gait3D	GREW
GaitBase [14]	16 × 256	64.6	60.1
GaitBase$_{squeeze}$	1 × 256	50.5	40.7
GaitContour	1 × 256	25.3	57.4

(b) Comparison under models with the same template size

overall. Furthermore, if we use five individual transformers, with the same structure as Local-CPT, the overall performance degrades by 2.2% and the model size goes up by 1.12M parameters, 269% of a shared Local-CPT. Using a shared transformer to process all regional information allows more augmentation on the input side and an overall more performant model.
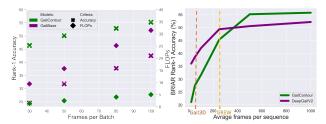
The effectiveness of Contour-Pose and temporal consis-

tency. In Table 4, we explore different keypoint-based gait representations to show the effectiveness of Contour-Pose. GaitTR [49] is used as the method to benchmark different inputs based on their performances on the Gait3D dataset. The vanilla GaitTR (a) achieves an improvement of 7.2% with pose keypoints. One direct approach to leverage silhouette information is to use contour points directly; to this end, we directly sample 112 contour points to construct a directed graph, where each contour point has two edge connections to the front and back contour points. This representation (b) leads to a decrease in performance when applied to GaitTR. If we use a pose-guided sampling scheme to construct Contour-Pose instead, we see a significant accuracy improvement on GaitTR from 7.2% to 19.3% (c). We believe that this improvement comes from an established order based on pose keypoints across frames, i.e., temporal consistency. We further demonstrate the importance of point ordering in Contour-Pose by examining the performance of Contour-Pose$_{NA}$, where the contour points are not sorted in a clockwise order. This leads to a significant decrease in performance from 25.3% to 13.9% on GaitContour (f).

The effect of template size. Template size is the subject identity embedding size produced by the gait recognition model. Larger templates can contain more information but are more expensive to store. In real applications, a vast number of identity embeddings needs to be stored and compared, making the template size a key consideration. As shown in Table 5, we examine the performance of GaitBase [14] with an equivalent template size to GaitContour. We note that the GaitBase [14] backbone is unchanged; instead, only the output template size is reduced from 16×256 to 1×256, noted as GaitBase$_{squeeze}$. Despite its significantly larger model size, GaitBase's performance drops from 64.6% to 50.5% on Gait3D and 60.1% to 40.7% on GREW with a constrained template size. On the GREW dataset, GaitContour even achieves 16.7% better performance compared to GaitBase with the same template size. This demonstrates the necessity for image-based methods to have a large template size to achieve good performance, and the efficiency of GaitContour, an effective way to concentrate sequences to a small feature size.

The effect of model size on performance. The size and computational cost of a model are also crucial aspects for real deployment. Larger models are better function approximators, but are more costly to train and infer; this is a non-trivial issue particularly in gait recognition when multi-frame inputs can be very high in dimensionality. Image-based methods generally construct larger models, as shown in Table 6. If we reduce the model size for these image-based models, as shown in Table 5, to be comparable to GaitContour, their performances significantly degrade. To maintain the original backbone structures, we only reduce the channel numbers. Interestingly, GaitBase's

performance dropped from 60.1% to 3.6% on the GREW dataset, demonstrating these large-scale architectures cannot be straightforwardly reduced to achieve a balance between efficiency and performance.

(a) The changes of performance and FLOPs with different **temporal windows**. GaitContour is significantly more efficient and has better performance.

(b) The changes of performance with different **temporal diversity**. GaitContour needs more temporal diversity during training to achieve good performance.

Figure 4. The effect of temporal information during training. Results are evaluated on the BRIAR dataset.

4.4. Fusion-based Comparison with Contour-Pose

We further analyze the utility of Contour-Pose in fusion-based gait recognition models. Fusion-based gait recognition [15, 33] leverages both silhouette and key-point representations to improve recognition performances. To this end, we propose a fusion method called Contour-Pose++, which uses a large backbone model in similar fashion to SkeletonGait++ [15]. Specifically, we map Contour-Pose to a 2D skeletonmap, extracting features from the skeletonmap and silhouette image using two CNN-based networks. These features are fused in a large backbone, i.e. DeepGaitV2. Please refer to the supplemental material for more specific architecture design.

As shown in Table 6, Contour-Pose++ consistently outperforms other image or fusion-based methods on five benchmark datasets. Notably, even though SkeletonGait++ includes a silhouette branch incorporating body shape information with pose keypoints, Contour-Pose++ yields significant improvements, e.g., 3% and 2% on Gait3D and SUSTech1. This demonstrates the effectiveness of Contour-Pose in encoding discriminative gait features.

4.5. Analysis on Temporal Information

In principle, gait recognition explores temporal patterns to perform biometrics; however, this aspect of gait recognition has seldom been analyzed due to the frame number limitation in popular datasets. As we demonstrate in Fig. 4a, for image-based and keypoint-based methods, models trained with a larger temporal window size can obtain better performance. This is particularly useful if the dataset has long-duration sequences, *e.g.*, in BRIAR. Training with more frames requires more computation, especially when the model itself is already large. As we showed in Fig. 4a, large models like GaitBase not only have more FLOPs than

7

Table 6. Quantitative comparison of **image and fusion-based** gait recognition methods across six large scale datasets. The best performers are colored, and the second best methods are underlined. The Ver represents TAR@FAR=10^{-3}.

Method	Params(M)	FLOPs(G)	Template size	OUMVLP	GREW		Gait3D		BRIAR			SUSTech1K		CCPG Full	
				R1	R1	R5	R1	R5	R1	R5	Ver	R1	R5	R1	mAP
GaitSet [8]	6.31	12.91	62 × 256	87.1	46.3	63.6	36.7	58.3	39.4	60.2	30.5	65.0	84.8	77.7	46.4
GaitPart [16]	4.84	7.93	30 × 128	88.5	44.0	60.7	28.2	47.6	41.4	61.5	32.3	59.2	80.8	77.8	45.5
GaitBase [14]	7.30	9.45	16 × 256	90.8	60.1	-	64.6	74.7	42.4	63.5	32.2	76.1	89.4	-	-
GaitGL [30]	11.19	58.55	64 × 256	89.7	47.3	64.4	29.7	48.5	52.0	70.7	39.8	63.1	82.8	69.1	27.0
DeepGaitV2 [13]	13.20	569.00	16 × 256	91.9	77.7	88.9	74.4	88.0	52.2	70.2	46.5	77.4	90.2	90.3	62.0
BiFusion [33]	7.56	8.26	16 × 256	89.9	45.5	64.5	30.8	49.9	48.5	63.4	36.5	62.1	83.4	77.5	46.7
SkeletonGait++ [15]	13.27	91.79	16 × 256	-	85.8	92.6	77.6	89.4	57.9	76.6	51.5	81.3	95.5	90.1	63.6
Contour-Pose++	13.27	91.79	16 × 256	-	86.1	93.4	79.6	89.8	59.7	77.6	53.1	83.3	95.8	92.1	67.1

GaitContour, but a steeper rate at which their FLOPs increase given more frames. In fact, GaitContour uses fewer FLOPs with a 100-frame input compared to a 30-frame to GaitBase. The performance gap between GaitContour and GaitBase also increases as the temporal window size goes down, likely because GaitBase overfits more on smaller details due to its dense inputs and large model.

From Table 2 and Table 6, we observe a smaller performance gap between keypoint-based and image-based methods on the BRIAR dataset compared to other datasets. This is a key difference between the BRIAR dataset and other benchmark sets with few frames. The abundance of temporal information in the BRIAR dataset enhances keypoint-based method performance. Keypoint-based methods can only observe compressed spatial information, making it challenging to extract discriminative subject features. Therefore, more temporal diversity is required during the training phase to build robust subject features. The substantial volume of temporal information in the BRIAR dataset allows keypoint-based methods to extract finer differences between subjects, achieving even higher performance than some image-based methods. The performance gap observed in GREW and Gait3D further supports this assumption. We conduct experiments on the BRIAR dataset to validate this finding. As shown in Fig. 4b, when less temporal diversity is employed during training, the performance of GaitContour drops more significantly than DeepGaitV2, demonstrating that the keypoint-based methods need more temporal diversity during training. Previous public datasets, such as GREW, have limited temporal diversity—only 13.3% of that in BRIAR—leading to the lower performance of keypoint-based methods.

4.6. Discussion

Limitation and Prospect Regarding the potential drawbacks of Contour-Pose, occlusion and limited field of view can result in imperfect silhouettes, particularly in uncontrolled environments. Such imperfections can pose challenges during training, as boundaries in adjacent frames may vary significantly. Currently, we apply point-wise aug-

mentation to mitigate this problem, as we did for silhouettes. We believe GaitContour still holds great potential. While there remains a performance gap between GaitContour and image-based methods, GaitContour demonstrates effective performance when there is abundant temporal diversity during training. Furthermore, due to its lightweight and efficient nature, GaitContour can serve as a plug-in module for any image-based method. Based on previous works [10, 33], the combination of keypoint-based and image-based methods consistently improves the overall performance.

5. Conclusion

In this work, we propose a novel gait representation called Contour-Pose and a gait recognition model, GaitContour, that leverages the advantages of Contour-Pose to achieve significant improvements in performance and efficiency. Contour-Pose uses a pose-guided sampling process on a silhouette, which approximates contour points from silhouette edges and sample points based on distances from pose keypoints. This representation efficiently preserves information from both silhouette and pose keypoints, and is temporally consistent. We can observe significant performance improvements when Contour-Pose is applied to various keypoint-based recognition models. We further develop GaitContour, which is tailored to analyze Contour-Pose. GaitContour contains two components: Local-CPT, and Gloabl-PFT. Local-CPT analyzes Contour-Pose at five different local regions and aggregates the outputs to a sparse global feature. Global-PFT then generates a subject identity embedding based on this global feature. Compared to a conventional Transformer, this local-to-global design significantly improves the model efficiency. Our experiments show that GaitContour is on par with SOTA image-based methods on a practical dataset, while maintaining efficiency in model size, template size, and FLOPs to that of keypoint-based methods, thereby making gait recognition much more practical for real applications.

6. Acknowledgement

References

[1] An, W., Yu, S., Makihara, Y., Wu, X., Xu, C., Yu, Y., Liao, R., Yagi, Y.: Performance evaluation of model-based gait on multi-view very large population database with pose sequences. IEEE transactions on biometrics, behavior, and identity science **2**(4), 421–430 (2020) 5

[2] Baumberg, A., Hogg, D.: Learning flexible models from image sequences. In: Computer Vision—ECCV'94: Third European Conference on Computer Vision Stockholm, Sweden, May 2–6, 1994 Proceedings, Volume I 3. pp. 297–308. Springer (1994) 3

[3] Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V.: Attention augmented convolutional networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3286–3295 (2019) 2, 5

[4] Benedek, C., Gálai, B., Nagy, B., Jankó, Z.: 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 (2016) 1

[5] Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Transactions on pattern analysis and machine intelligence **23**(3), 257–267 (2001) 3

[6] Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000) 4

[7] Castro, F.M., Marin-Jimenez, M.J., Guil, N., Pérez de la Blanca, N.: Multimodal feature fusion for cnn-based gait recognition: an empirical comparison. Neural Computing and Applications **32**, 14173–14193 (2020) 3

[8] Chao, H., He, Y., Zhang, J., Feng, J.: Gaitset: Regarding gait as a set for cross-view gait recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 8126–8133 (2019) 3, 8

[9] Cornett, D., Brogan, J., Barber, N., Aykac, D., Baird, S., Burchfield, N., Dukes, C., Duncan, A., Ferrell, R., Goddard, J., et al.: Expanding accurate person recognition to new altitudes and ranges: The briar dataset. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 593–602 (2023) 5

[10] Cui, Y., Kang, Y.: Multi-modal gait recognition via effective spatial-temporal feature fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17949–17957 (2023) 3, 8

[11] Del Din, S., Elshehabi, M., Galna, B., Hobert, M.A., Warmerdam, E., Suenkel, U., Brockmann, K., Metzger, F., Hansen, C., Berg, D., et al.: Gait analysis

with wearables predicts conversion to parkinson disease. Annals of neurology **86**(3), 357–367 (2019) 1

[12] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (December 2015) 3

[13] Fan, C., Hou, S., Huang, Y., Yu, S.: Exploring deep models for practical gait recognition. arXiv preprint arXiv:2303.03301 (2023) 3, 8

[14] Fan, C., Liang, J., Shen, C., Hou, S., Huang, Y., Yu, S.: Opengait: Revisiting gait recognition towards better practicality. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9707–9716 (2023) 1, 3, 6, 7, 8

[15] Fan, C., Ma, J., Jin, D., Shen, C., Yu, S.: Skeletongait: Gait recognition using skeleton maps. arXiv preprint arXiv:2311.13444 (2023) 2, 3, 7, 8

[16] Fan, C., Peng, Y., Cao, C., Liu, X., Hou, S., Chi, J., Huang, Y., Li, Q., He, Z.: Gaitpart: Temporal part-based model for gait recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14225–14233 (2020) 3, 8

[17] Fu, Y., Meng, S., Hou, S., Hu, X., Huang, Y.: Gpgait: Generalized pose-based gait recognition. arXiv preprint arXiv:2303.05234 (2023) 1, 2, 3, 6

[18] Guo, H., Ji, Q.: Physics-augmented autoencoder for 3d skeleton-based gait recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19627–19638 (2023) 3

[19] Guo, Y., Peng, C., Lau, C.P., Chellappa, R.: Multimodal human authentication using silhouettes, gait and rgb. In: 2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG). pp. 1–7. IEEE (2023) 3

[20] Hadid, A., Ghahramani, M., Kellokumpu, V., Pietikäinen, M., Bustard, J., Nixon, M.: Can gait biometrics be spoofed? In: Proceedings of the 21st international conference on pattern recognition (ICPR2012). pp. 3280–3283. IEEE (2012) 1

[21] Han, J., Bhanu, B.: Individual recognition using gait energy image. IEEE transactions on pattern analysis and machine intelligence **28**(2), 316–322 (2005) 3

[22] Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings (2015), http://arxiv.org/abs/1412.6622 5

[23] Hui, T.W., Tang, X., Loy, C.C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8981–8989 (2018) 3

[24] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2462–2470 (2017) 3

[25] Li, W., Hou, S., Zhang, C., Cao, C., Liu, X., Huang, Y., Zhao, Y.: An in-depth exploration of person re-identification and gait recognition in cloth-changing conditions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13824–13833 (2023) 5

[26] Li, X., Makihara, Y., Xu, C., Yagi, Y., Yu, S., Ren, M.: End-to-end model-based gait recognition. In: Proceedings of the Asian conference on computer vision (2020) 3

[27] Li, Y., Vasconcelos, N.: Repair: Removing representation bias by dataset resampling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9572–9581 (2019) 1

[28] Li, Y., Li, Y., Vasconcelos, N.: Resound: Towards action recognition without representation bias. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 513–528 (2018) 1

[29] Liang, J., Fan, C., Hou, S., Shen, C., Huang, Y., Yu, S.: Gaitedge: Beyond plain end-to-end gait recognition for better practicality. In: European Conference on Computer Vision. pp. 375–390. Springer (2022) 3, 4

[30] Lin, B., Zhang, S., Yu, X.: Gait recognition via effective global-local feature representation and local temporal aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14648–14656 (2021) 3, 6, 8

[31] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 851–866 (2023) 2, 4

[32] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 5

[33] Peng, Y., Ma, K., Zhang, Y., He, Z.: Learning rich features for gait recognition by integrating skeletons and silhouettes. Multimedia Tools and Applications pp. 1–22 (2023) 3, 7, 8

[34] Pinyoanuntapong, E., Ali, A., Wang, P., Lee, M., Chen, C.: Gaitmixer: skeleton-based gait representation learning via wide-spectrum multi-axial mixer.

In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023) 3

[35] Plizzari, C., Cannici, M., Matteucci, M.: Spatial temporal transformer network for skeleton-based action recognition. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part III. pp. 694–701. Springer (2021) 2, 5

[36] Sepas-Moghaddam, A., Etemad, A.: Deep gait recognition: A survey. IEEE transactions on pattern analysis and machine intelligence 45(1), 264–284 (2022) 1

[37] Sepas-Moghaddam, A., Etemad, A.: Deep gait recognition: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 45(1), 264–284 (2023). https://doi.org/10.1109/TPAMI.2022.3151865 1

[38] Shen, C., Fan, C., Wu, W., Wang, R., Huang, G.Q., Yu, S.: Lidargait: Benchmarking 3d gait recognition with point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1054–1063 (2023) 3, 5

[39] Shen, C., Yu, S., Wang, J., Huang, G.Q., Wang, L.: A comprehensive survey on deep gait recognition: algorithms, datasets and challenges. arXiv preprint arXiv:2206.13732 (2022) 1

[40] Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems 33, 7537–7547 (2020) 5

[41] Teepe, T., Gilg, J., Herzog, F., Hörmann, S., Rigoll, G.: Towards a deeper understanding of skeleton-based gait recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1569–1577 (2022) 2, 3, 6

[42] Teepe, T., Khan, A., Gilg, J., Herzog, F., Hörmann, S., Rigoll, G.: Gaitgraph: Graph convolutional network for skeleton-based gait recognition. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 2314–2318. IEEE (2021) 2, 3, 4, 6

[43] Teh, C.H., Chin, R.T.: On the detection of dominant points on digital curves. IEEE Transactions on pattern analysis and machine intelligence 11(8), 859–872 (1989) 2, 4

[44] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017) 3

[45] Wang, L., Tan, T., Ning, H., Hu, W.: Silhouette analysis-based gait recognition for human identification. IEEE transactions on pattern analysis and machine intelligence 25(12), 1505–1518 (2003) 3

[46] Wang, Y., Du, B., Shen, Y., Wu, K., Zhao, G., Sun, J., Wen, H.: Ev-gait: Event-based robust gait recognition using dynamic vision sensors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6358–6367 (2019) 3

[47] Weinzaepfel, P., Rogez, G.: Mimetics: Towards understanding human actions out of context. International Journal of Computer Vision 129(5), 1675–1690 (2021) 1

[48] Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018) 3

[49] Zhang, C., Chen, X.P., Han, G.Q., Liu, X.J.: Spatial transformer network on skeleton-based gait recognition. Expert Systems p. e13244 (2023) 2, 3, 4, 5, 6, 7

[50] Zhang, Z., Tran, L., Liu, F., Liu, X.: On learning disentangled representations for gait recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 44(1), 345–360 (2020) 3

[51] Zheng, J., Liu, X., Liu, W., He, L., Yan, C., Mei, T.: Gait recognition in the wild with dense 3d representations and a benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20228–20237 (2022) 1, 3, 5, 6

[52] Zhu, Z., Guo, X., Yang, T., Huang, J., Deng, J., Huang, G., Du, D., Lu, J., Zhou, J.: Gait recognition in the wild: A benchmark. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 14789–14799 (2021) 1, 5