Physics-Augmented Autoencoder for 3D Skeleton-Based Gait Recognition

Hongji Guo and Qiang Ji
Rensselaer Polytechnic Institute, Troy, NY 12180
{guoh11, jiq}@rpi.edu

Abstract

In this paper, we introduce physics-augmented autoencoder (PAA) framework for 3D skeleton-based human gait recognition. Specifically, we construct the autoencoder with a graph-convolution-based encoder and a physics-based decoder. The encoder takes the skeleton sequence as input and produces the generalized positions and forces of each joint, which are taken by the decoder to reconstruct the input skeleton based on the Lagrangian dynamics. In this way, the intermediate representations are physically plausible and discriminative. During the inference, the decoder is discarded and a RNN-based classifier takes the output of the encoder for gait recognition. We evaluated our proposed method on three benchmark datasets including Gait3D, GREW, and KinectGait. Our method achieves state-of-the-art performance for 3D skeleton-based gait recognition. Furthermore, extensive ablation studies show that our method generalizes better and is more robust with small-scale training data by incorporating the physics knowledge. We also validated the physical plausibility of the intermediate representations by making force predictions on real data with physical annotations.

1. Introduction

In general, gait refers to the walking style of a person. It is a unique biometric pattern varying from person to person. Gait recognition aims at identifying humans based on their gaits. It has many important applications such as security surveillance [2], smart homes [58], and healthcare [43]. Extensive work have been focusing on 2D-based gait recognition such as using silhouettes [11], gait energy image [32] and 2D human skeleton [49]. However, 2D-based gait recognition suffers from occlusion and changing of view angles. The occlusion caused by the clothes and stuffs being carried introduce irrelevant information which lead to degradation of the performance. And 2D-based methods may not generalize well if the training data and query input are from different view angles. Besides, human gait is a dynamic process taking place in the 3D space. 2D-based gait recognition methods rely on appearance features, which ignore the underlying physical dynamics that are crucial for model robustness and generalization. So modeling 3D is intuitive and effective for the understanding of gait. On the other hand, 2D-based methods may require large amount of data for training, which is impractical for many real cases.

To address these issues, we study 3D skeleton-based gait recognition in this paper, with a focus on modeling the physical dynamics of human gait. The objective is to leverage the widely applicable physics laws to improve the model generalization, robustness, and interpretability. By incorporating the physics knowledge into the model, we aim to obtain physical features of human gait so that they can be used for recognition. Specifically, we model the human joint positions and forces in the generalized coordinates. With the joint positions and the forces applied onto them, the gait dynamics can be captured in a compact and precise way. Another advantage of modeling physics is that the model can generalize better since the embedded physics laws are universal for different subjects.

To encode the physical representations of human gait, we adopt an autoencoder [20, 61] architecture. Specifically, we construct a spatial-temporal graph convolution network as the encoder to better adapt the skeleton topology of the input. The output of the graph convolution network is fed into two networks to predict the joint positions and forces in the generalized coordinates, which are treated as the intermediate representations of the autoencoder. Then the decoder takes the generalized joint positions and forces to reconstruct the input skeleton sequence based on the Lagrangian dynamics.
dynamics [62]. Indeed, the decoder is a differentiable solver that embeds the physical constraints. In this way, the intermediate representations should capture the physical dynamics otherwise the decoder cannot reconstruct the input gait sequence correctly.

Different from appearance-based features that are extracted by purely data-driven models, the intermediate representations of PAA are the fundamental parameters that dominate the human gait process. Thus, we expect them to be more generalizable and robust for gait recognition. With these physical representations, we use a recurrent neural network as the classifier to perform the gait recognition.

For the training, the autoencoder and the classifier are optimized by the reconstruction loss and gait recognition loss respectively. Specifically, we first train the autoencoder with the reconstruct loss and then jointly train the classifier with autoencoder. During the inference, only the encoder and classifier are kept for gait recognition. We evaluated the proposed method on three benchmark datasets including Gait3D [65], GREW [66], and KinectGait [1]. To demonstrate the effectiveness of physics modeling, we applied the proposed PAA on data with physical annotations and compare with the ground truth.

In summary, the main contributions of this paper are:

- We propose a physics-augmented autoencoder for 3D skeleton-based gait recognition that models the underlying physical dynamics of human gait. The physical representations are learned by the autoencoder in an unsupervised manner.
- By incorporating the physics modeling, the obtained physical features are more compact and precise. We verify the physical plausibility of the encoded representations by applying PAA on real data with physical annotations and compare with the ground truth.
- The proposed PAA achieves state-of-the-art performance on three benchmark datasets. We also demonstrate that PAA generalizes better and is more data-efficient by extensive ablation studies.

2. Related Work

2.1. Skeleton-based gait recognition

As the development of accurate pose estimation algorithms [3, 46, 5, 40] and affordable depth sensors such as Microsoft Kinect [64, 17], skeleton-based gait recognition is attracting more attention because of its efficient representation and robustness under occlusion. In this paper, we focus on 3D instead of 2D gait recognition such as [33, 54, 26, 12, 53]. Here, we review the 3D skeleton-based gait recognition approaches.

Early work focused on hand-crafted features, which are lightweight and human-orientated. Andersson et al. [1] extracted anthropometric gait features and performs the classification by a KNN classifier. Sun et al. [45] used the lengths of some specific skeletons as static features and the angles of swing limbs as dynamic features to construct a view-invariant walking model for skeleton-based gait recognition. Yang et al. [60] proposed a method for extracting relative distance features and anthropometric features for robust gait recognition. To capture the dynamics of the human gait, Khamsemanan et al. [28] proposed a model-based technique using posture-based features, which are composed of displacements of all joints between adjacent frames in the body-centered coordinates.

Recently, deep learning based methods become dominant. Huynh-The et al. [25] extracted spatiotemporal feature with a convolutional network to perform the skeleton-based gait recognition. Liu et al. [36] introduced a method using skeleton gait energy image (SkeGEI), relative distance and angle (DA) as features. Then a convolutional neural network is used to capture the spatial relationship and a LSTM is used to model the temporal dependencies. To address the occlusion problem and make the model view-invariant, Choi et al. [6] proposed a method that minimizes the influence of noisy patterns and ensure the frame-level discriminative power. Through a two-state linear matching process, the high-quality frame-level scores are used for classification by a weighted majority voting scheme. Further, Hasan et al. [18] used stacked autoencoder to learn the discriminant view-invariant gait representations to adapt the variations in view. The encoded features and other spatiotemporal features are combined to be classified by a recurrent neural network. For skeleton-based gait recognition, the data is in the graph format, which is intuitively to be modeled by graph-based model to effectively capture the spatial and temporal dependencies. In this paper, we transfer some graph convolution methods [50, 49] from 2D to 3D skeleton-based gait recognition. However, neither models using hand-crafted features nor purely data-driven methods are satisfied for recognition accuracy, robustness, and generalization. To address these challenges, we combine the domain physics knowledge and the data to improve the robustness and generalization of the model.

2.2. Physics-informed neural networks

Recently, physics knowledge has been introduced into neural networks such as physical priors [37] and constraints [21]. The objective is to utilize the domain knowledge to help the tasks especially when the amount of training data is limited, or better generalization and interpretability are desired. It has been applied for many important real tasks such as weather forecast [27], turbulent flow prediction [55], and seismic response modeling [63].
Figure 2: **Overall framework of Physics-Augmented Autoencoder (PAA).** The input of the model is a skeleton sequence. Firstly, a graph-convolution-based encoder takes the input and predicts the joint positions and forces in the generalized coordinates. Then, the physics-based decoder takes the generalized positions and forces to reconstruct the input skeleton sequence based on the Lagrangian dynamics. During the inference, the decoder is discarded and a RNN-based classifier takes the concatenation of the generalized joint positions and forces to perform gait recognition.

Majority of existing methods incorporate physics engines and solvers into the network to achieve the physics modeling. In an energy-conserving system with specific physical parameters, Lagrangian dynamics and Hamiltonian mechanics are exploited to model the positions, momentum, and other physical parameters [7, 51]. To encode the physics knowledge into a machine learning framework with low annotation cost, many work adopt the autoencoder architecture [61, 48, 38, 15, 8] since the model can be learned in an unsupervised manner by reconstructing the input. For human gait modeling specifically, Takeishi et al. [47] constructs a variational autoencoder (VAE) with a physics engine as decoder for human gait synthesis. However, the hidden states of the VAE are not well specified, which makes it difficult to connect these states with the real physical world. Different from existing work, our autoencoder specifically models the generalized joint positions and their corresponding forces with an additional task-oriented branch for gait recognition. The physics modeling is achieved through a differentiable physics engine [10, 23, 19, 41, 13, 9]. By jointly train the model for both physics modeling and the downstream task, the learned intermediate representations of the autoencoder are both interpretable and discriminative for gait recognition.

### 3. Method

In this section, we first give the overall framework of our proposed physics-augmented autoencoder (PAA) in Sec. 3.1. Then we introduce the graph-convolution-based encoder and physics-based decoder in Sec. 3.3 and Sec. 3.4 respectively. The details of the classifier is provided in Sec. 3.5. Finally, we discuss the training and evaluation procedures in Sec. 3.6.

#### 3.1. Overall framework

An overall framework of PAA is shown in Figure 2. The input of the model is a 3D skeleton sequence, which is composed of the human joint coordinates of each frame. The input skeleton sequence is fed into the graph-convolution-based encoder to generate the joint positions and their corresponding forces in the generalized coordinates. Then, the physics-based decoder is used to reconstruct the input skeleton sequence. By training the autoencoder in this way, the encoder can generate physical representations well. During the inference, the decoder is discarded. We concatenate the generalized joint positions and forces to form the feature of each frame. A recurrent neural network is used to perform the gait recognition.

#### 3.2. Preliminaries

The input of the model is a human skeleton sequence. Graphs are constructed to represent the human skeleton. At each time, the human skeleton is represented by a graph \( G = (V, E) \), where \( V = \{v_1, ..., v_N\} \) is the node set of \( N \) joints and \( E \) is the edge set of bones. \( E \) is described by an adjacency matrix \( A \in \mathbb{R}^{N \times N} \), where \( A_{i,j} = 1 \) denotes there is connection between \( v_i \) and \( v_j \), and \( A_{i,j} = 0 \) means no connections. The graph \( G \) is undirected so \( A \) is symm-
ric.
Denote the input as $X \in \mathbb{R}^{T \times N \times C}$, where $T$ is the length of the input sequence and $C$ is the dimension of node features. Then $X_t \in \mathbb{R}^{N \times C}$ is the gait feature at time $t$.

3.3. Graph-convolution-based encoder

To encode the generalized joint positions and forces from the input skeleton sequence, we construct a spatial-temporal graph convolution network (ST-GCN) [59]. The ST-GCN is composed of a series of ST-GCN blocks. Each block contains a spatial graph convolution followed by a temporal graph convolution, which alternatingly extracts spatial and temporal features. The spatial convolution allows the information flow within each frame and the temporal convolution models the dynamics along the time dimension. The last ST-GCN block is connected to two fully-connected networks to output the generalized joint positions and forces of each frame, which are used as the input of the decoder.

Spatial-temporal graph convolution. Given the input feature $X$, the $k$-th ST-GCN block performs the following update at time $t$:

$$X_t^{(k+1)} = \sigma(\Lambda^{-\frac{1}{2}} \tilde{A} \Lambda^{-\frac{1}{2}} X_t^{(k)} W^{(k)})$$

where $\tilde{A} = A + I$ is the adjacency matrix with self-loops, $\Lambda = \sum_j(A_{ij} + I_{ij})$ is diagonal degree matrix of $A$ and $\sigma(\cdot)$ is the sigmoid activation function. $W^{(k)}$ is the weight matrix. By performing the spatial graph convolution, the features are aggregated spatially by the neighbors. Multiple graph convolution blocks are stacked to make up the main body of the encoder.

Following the spatial convolution, the temporal convolution is achieved by a standard $1 \times \Gamma$ convolution along the time dimension of the feature. By going through a series of ST-GCN blocks, the features exchanged their information so that their physical dependencies can be well captured in the following procedures.

Position and force encoding. After going through $K$ blocks of spatial-temporal convolution, the updated features are mapped to the physical representations including the joint positions and their corresponding forces in the generalized coordinates. Denote the encoded joint positions and forces as $q \in \mathbb{R}^{T \times D}$ and $f \in \mathbb{R}^{T \times D}$, where $D$ is the total degree-of-freedom of all joints. These physical representations are used for reconstruction by the decoder and also for gait recognition by the classifier.

3.4. Physics-based decoder

Given the encoded generalized joint positions and forces, the goal of the decoder is to reconstruct the input skeleton sequence. To ensure the intermediate representations of the autoencoder is the desired physical positions and forces, we construct a differentiable physics solver as the decoder.

**Body parameters fitting.** To solve for the generalized positions and forces sequentially, we need basic body parameters to build the physical system of the decoder for the reconstruction. Given the input skeleton sequence, we match it to a pre-defined human model [42] to obtain basic body parameters such as approximated mass and bone length. The fitting is achieved by keypoint matching followed by a fully-connected neural network.

**Generalized coordinates.** Different from Cartesian coordinates, generalized coordinates are a set of parameters that represent the state of a system in a configuration space [14]. These parameters uniquely define the state of the system. In this work, we model the human joint system in the generalized coordinates for a compact and precise representation.

**Physics-augmented decoding.** The physics-based decoder reconstructs the skeleton sequence in an online manner [56] based on the Lagrangian dynamics. It takes the current position $q_t$, velocity $\dot{q}_t$, control forces $f$ and inertial properties $\mu$ as the input. And it returns the position and velocity at the next time step, $q_{t+1}$ and $\dot{q}_{t+1}$:

$$D(q_t, \dot{q}_t, f, \mu) = [q_{t+1}, \dot{q}_{t+1}]$$

$D$ denotes the decoder. The position prediction is by taking the simple integration as $q_{t+1} = q_t + \Delta t \dot{q}_t$, where $\Delta t$ is the discritized time interval.

To solve $\dot{q}_{t+1}$, the decoder solves the Lagrangian dynamic equation in the generalized coordinates:

$$M(q_t, \mu) \dot{q}_{t+1} = M(q_t, \mu) \dot{q}_t - \Delta t (c(q_t, \dot{q}_t, \mu) - f) + J^T(q_t) \tau$$

where $M$ is the mass matrix, $c$ is the Coriolis and gravitational force, and $\tau$ is the contact force in the generalized coordinate system with contact Jacobian matrix $J$. $\tau$ can be obtained by solving the linear complementarity problem (LCP):

$$\begin{align*}
\text{find } & \tau, v_{t+1} \\
\text{such that } & \tau > 0, v_{t+1} > 0, \tau^T v_{t+1} = 0
\end{align*}$$

The velocity $v_{t+1}$ can be written as a linear function of $\tau$:

$$v_{t+1} = J \dot{q}_{t+1} = JM^{-1}(M \dot{q}_t - \Delta t (c - f) + J^T \tau) = A \tau + b$$

where $A = JM^{-1}J^T$ and $b = J(\dot{q}_t + \Delta t M^{-1}(f - c))$. Then the LCP procedure is formulated as a function that maps $(A, b)$ to the contact force $f$:

$$f_{LCP}(A(q_t, \mu), b(q_t, \dot{q}_t, f, \mu)) = \tau$$

The details of the optimization process can be found in [56]. By recursively solving the Lagrangian dynamic equation,
we obtain the predicted generalized position of each joint at every time step. Then, a neural network re-scales the generalized positions to the input scale and maps them to Cartesian coordinates, which is the output of the decoder.

The output joint positions are used to compute the reconstruction loss to train the autoencoder. By constructing the decoder as a differentiable physics solver, the intermediate representations of the autoencoder are constrained to be physically plausible otherwise the decoder is not able to make reconstruction correctly.

### 3.5. Classifier

The goal of the classifier is to identify people based on the encoded generalized position and force sequences. We adopt a recurrent neural network (RNN) as the classifier. At each frame, we concatenate the joint positions \( q_t \in \mathbb{R}^D \) and forces \( f_t \in \mathbb{R}^D \) to form the feature vector \( W_t \in \mathbb{R}^{2D} \). The formed feature sequence is fed into the RNN to recognize the gait by performing the pooling of the output features. The details of the RNN architecture and parameters are available in the supplementary.

### 3.6. Training and evaluation

#### Training.

To efficiently train the model to capture the physics and for accurate gait recognition. We adopt a two-stage training strategy. At first stage, we train the autoencoder without gait recognition. Given one input sequence \( X \), the PAA outputs the reconstructed skeleton sequence \( X' \). The loss function of the first stage is the mean squared error of the input skeleton sequence reconstruction:

\[
L_{\text{reconstruct}} = \text{MSE}(X, X') \quad (7)
\]

In the second stage, we jointly train the classifier and the autoencoder. Deonote the classification output as \( P' \in \mathbb{R}^C \), where \( C \) is the total number of subjects. The loss function of the second stage can be written as:

\[
L = L_{\text{gait}} + \lambda L_{\text{reconstruct}} \quad (8)
\]

where \( L_{\text{gait}} \) is the triplet loss or cross-entropy loss of prediction \( P' \) and the groundtruth \( P \), \( \lambda \) is a hyper-parameter that measures the weight of reconstruction loss. The training process is summarized in Algorithm 1. In practice, \( \lambda \) is small since the main task of the model is gait recognition and the autoencoder is pre-trained in the first stage. An ablation study of the balance is shown in Section 4.4.

#### Inference.

The decoder is used to constrain the encoder to generate physical representations during the training, so it is discarded during the inference. Given a query input, the encoder generates the physical representations, and then the RNN-based classifier takes the representations to perform the gait recognition.

### Algorithm 1 Training

**Input:** \( D = \{X_k \in \mathbb{R}^{T \times N \times C}, y_k\}_{k=1}^K \) - training data  
**Output:** \( \Theta_e, \Theta_c \) - parameters of encoder and classifier  

**Training stage 1**

1. for \( k = 1 \) to \( K \) do
2. Generate \( q_k, f_k \) by the encoder
3. Sequentially predict \( q_k' \) based on Eq. (3)
4. Re-scale \( q_k' \) to \( X_k' \)
5. Update \( \Theta_e \) by minimizing \( L_{\text{reconstruct}} \)
6. end for

**Training stage 2**

7. for \( k = 1 \) to \( K \) do
8. Generate \( q_k, f_k \) by encoder
9. Make prediction \( P' \)
10. Repeat line 3 to line 4
11. Update \( \Theta_e, \Theta_c \) by minimizing \( L \) in Eq. (8)
12. end for
13. return \( \Theta_e, \Theta_c \)

### 4. Experiments

#### 4.1. Datasets

Gait3D [65] is a large-scale gait dataset with 4,000 subjects and 25,309 sequences extracted from 39 cameras in an unconstrained indoor scene. It provides 3D human meshes recovered from video frames, which provides 3D pose and shape of human bodies. In this paper, we use the 3D pose from Gait3D. Following the settings in [65], we select 18,940, 1000, and 5369 sequences for training, validation, and testing respectively.

GREW [66] is a large-scale gait dataset captured in real-world environments. It contains 26,345 identities and 128K sequences with rich attributes for unconstrained gait recognition. In this paper, we use the 3D pose of the GREW estimated by [5].

KinectGait [1] is a relative large-scale skeleton-based human gait dataset captured by Microsoft Kinect V1. There are totally 164 subjects with 5 sequences for each subject. The people walked in a semi-circular path in front of the Kinect sensor when recording the data and the sensor followed the people using a spinning dish.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train Set</th>
<th>Test Set</th>
<th>Batch</th>
<th>Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gait3D</td>
<td>3000 18940</td>
<td>1000 6369</td>
<td>128</td>
<td>400</td>
</tr>
<tr>
<td>GREW</td>
<td>20000 102887</td>
<td>6600 24000</td>
<td>128</td>
<td>600</td>
</tr>
<tr>
<td>K-Gait</td>
<td>164 656</td>
<td>164 166</td>
<td>4</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 1: Experimental settings of each dataset.

#### 4.2. Implementation Details

**Settings.** We implemented the proposed framework in PyTorch [39]. All the models were trained using the Adam
Table 2: Experiment results on Gait3D and GREW. Our PAA outperforms SOTA methods on both datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Gait3D</th>
<th>GREW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-1</td>
<td>R-5</td>
</tr>
<tr>
<td>PoseGait [34]</td>
<td>26.12</td>
<td>39.79</td>
</tr>
<tr>
<td>GaitGraph [50]</td>
<td>31.71</td>
<td>48.50</td>
</tr>
<tr>
<td>GaitGraph2 [49]</td>
<td>33.20</td>
<td>49.62</td>
</tr>
<tr>
<td>PAA (ours)</td>
<td><strong>38.92</strong></td>
<td><strong>59.08</strong></td>
</tr>
</tbody>
</table>

Table 3: Experiment results on KinectGait. Our proposed PAA achieves SOTA performance against other methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Rank-1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN [1]</td>
<td>87.70</td>
</tr>
<tr>
<td>Dynamic LSTM [31]</td>
<td>96.56</td>
</tr>
<tr>
<td>RDF [60]</td>
<td>95.4</td>
</tr>
<tr>
<td>Posture [28]</td>
<td>97.00</td>
</tr>
<tr>
<td>CNN-LSTM [36]</td>
<td>97.79</td>
</tr>
<tr>
<td>PAA (ours)</td>
<td><strong>98.42</strong></td>
</tr>
</tbody>
</table>

4.3. Main results and comparison

Gait recognition. The experiment results on Gait3D and GREW are shown in Table 2. We report the Rank-1 and Rank-5 recognition rates for comparison. On both datasets, we achieve state-of-the-art performance. We also conducted the experiments on KinectGait, whose gait 3D pose were obtained from depth sensors instead of pose estimation algorithms. The experiment results are shown in Table 3. With more accurate 3D pose and fewer subjects, the performance is much better. Our proposed PAA also achieves SOTA performance comparing with other methods that rely on hand-crafted features or purely data-driven models.

Qualitative skeleton reconstruction results. We visualize the skeletons reconstructed by the decoder after each training stage. The skeletons from Gait3D and GREW are shown in Figure 3 and Figure 4 respectively. After training stage 1, the PAA can well reconstruct the input skeleton since only the reconstruction loss is adopted in this stage. The skeleton from training stage 2 has a relative reconstruction error but is also quite close to the ground-truth skeleton. Thus, the joint training in the second training stage does not degenerate the physical modeling much when performing the gait recognition. Otherwise, the decoder cannot reconstruct the input skeleton based on the incorrect physical representations. More qualitative results and failure cases are available in the supplementary.

4.4. Ablation studies

Decoder types. To demonstrate the effectiveness of the physics-based decoder, we construct multiple autoencoders with different types decoders and compare them with the physics-based decoder. Specifically, we evaluated multilayer perceptron (MLP), recurrent neural network (RNN), long short-term memory network (LSTM), and graph con-
volution network (GCN). The experiment results are shown in Table 4. By adopting the physics-based decoder, our model achieves better performance than other types of decoders, which demonstrates the effectiveness of the physics modeling in the decoder.

**Encoder types.** To further understand the properties of the PAA, we also studied different types of encoders. Specifically, we replace the graph convolution network with MLP, RNN, and LSTM. The comparison is shown in Table 5. From the results, the GCN-based encoder gives the best performance.

**Balance between gait recognition loss and reconstruction loss.** In the second training stage, the training loss function is composed of a gait recognition loss for classification and a mean squared error loss for skeleton reconstruction. We aim to maximize the recognition rate while optimizing the autoencoder for better physical representation encoding. By tuning the hyperparameter $\lambda$ in $\mathcal{L}$, we visualize the recognition rate in Figure 5. We empirically select $\lambda = 0.3$ since it leads to best performance on both Gait3D and GREW datasets.

**Training strategies.** In order to better encode the physical representations, we first pre-train the autoencoder without classifier using only the reconstruct loss. Then we jointly train the gait recognizer and the autoencoder with the cross-entropy loss and the reconstruction loss. The pre-training strategy makes the joint training converge faster. Comparing with training the whole model from scratch with the total loss function $\mathcal{L}$, the performances also improve on Gait3D (36.12%→38.92%) and GREW (37.35%→38.71%).

**Training the model with small-scale data.** In many real situations, the amount of training data is limited. Purely data-driven methods may encounter large performance decay. To demonstrate the physics modeling improves the data-efficiency, we reduce the amount of training data from 100% to 20% and make a comparison. We repeated the experiment of each setting for five times and computed the average performance. The experiment results on Gait3D and GREW are plotted in Figure 6. By comparison, our proposed PAA is more robust compared with other methods when the amount of training data is limited.

**Generalization.** Our model generates physical representations guided by the physics laws, which is widely applicable for different subjects. Thus, we expect better generalization performance compared with purely data-driven approaches. To test the generalization capability of our model, we performed the cross-view and cross-carrying experiments on GREW dataset. We divide the dataset based on the meta annotations of GREW so that training data and testing data are from different views (cross-view), or with different carryings (cross-carrying). The experiment results are shown in Table 6. Under both settings, our PAA outperforms the SOTA methods, which demonstrates that it generalizes better.

**Robustness.** In real situations, people may carry stuffs and be blocked by unrelated obstacles. To test the robustness of our model under these conditions, we simulate the occlusions by adding random Gaussian noise following the procedures in [6]. Specifically, we add noise to different parts of human body to test the model, including upper body, lower body, and whole body. We conducted the experiments on Gait3D dataset. The experiment results are shown in Table 7. Compared with other SOTA methods, our proposed PAA gives more robust performance with the noisy input.

**Number of joints.** The physics modeling of PAA is based on the 3D coordinates of joints. The number of joints can affect the physics modeling and further the gait recognition performance. To study the impact, we varied the number of joints for our model. The experiment results on Gait3D are shown in Table 8. In general, more joints bring better performance since the physics modeling can be improved with

---

**Figure 5:** Balance between the gait recognition loss and reconstruction loss. We plot the recognition rates on Gait3D and GREW with respect to different $\lambda$. $\lambda = 0.3$ performs best on both datasets.

**Figure 6:** Experiment results with small-scale training data on Gait3D and GREW. Our proposed PAA is more data-efficient with limited training data.

<table>
<thead>
<tr>
<th>Method</th>
<th>X-View (%)</th>
<th>X-Carrying (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoseGait [34]</td>
<td>20.87</td>
<td>20.16</td>
</tr>
<tr>
<td>GaitGraph [50]</td>
<td>28.10</td>
<td>27.33</td>
</tr>
<tr>
<td>GaitGraph2 [49]</td>
<td>28.61</td>
<td>28.04</td>
</tr>
<tr>
<td>PAA (ours)</td>
<td><strong>35.26</strong></td>
<td><strong>36.03</strong></td>
</tr>
</tbody>
</table>

Table 6: Generalization results. X-View denotes cross-view and X-carrying denotes cross-carrying.
Table 7: Experiment results under occlusions on Gait3D. The results are reported as Rank-1 recognition rate (%). “No” denotes without occlusions. By adding noise to different body parts, our proposed PAA stay robust relatively.

<table>
<thead>
<tr>
<th>Method</th>
<th>No</th>
<th>Upper</th>
<th>Lower</th>
<th>Whole</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoseGait [34]</td>
<td>26.12</td>
<td>24.65</td>
<td>21.93</td>
<td>20.65</td>
</tr>
<tr>
<td>GaitGraph [50]</td>
<td>31.71</td>
<td>27.48</td>
<td>25.74</td>
<td>23.91</td>
</tr>
<tr>
<td>GaitGraph2 [49]</td>
<td>33.20</td>
<td>30.55</td>
<td>28.61</td>
<td>26.10</td>
</tr>
<tr>
<td>PAA (ours)</td>
<td><strong>38.92</strong></td>
<td><strong>37.25</strong></td>
<td><strong>33.40</strong></td>
<td><strong>32.06</strong></td>
</tr>
</tbody>
</table>

Table 8: Ablation study of number of joints on Gait3D.

<table>
<thead>
<tr>
<th># of joints</th>
<th>14</th>
<th>20</th>
<th>24</th>
<th>37</th>
<th>54</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank-1</td>
<td>34.86</td>
<td>37.12</td>
<td>38.92</td>
<td>39.83</td>
<td>40.91</td>
</tr>
<tr>
<td>Rank-5</td>
<td>50.16</td>
<td>54.23</td>
<td>59.08</td>
<td>59.48</td>
<td><strong>62.52</strong></td>
</tr>
</tbody>
</table>

Table 9: Comparison of adding physics supervision. With a few additional data with force annotation, the performance of PAA can be improved.

<table>
<thead>
<tr>
<th>Method</th>
<th>Gait3D</th>
<th>GREW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-1</td>
<td>R-5</td>
</tr>
<tr>
<td>PAA</td>
<td>38.92</td>
<td>59.08</td>
</tr>
<tr>
<td>PAA+Forces</td>
<td>39.82</td>
<td>61.22</td>
</tr>
</tbody>
</table>

4.6. Computation efficiency and model complexity

Computation cost is an important concern especially for real-world applications. Here we make a comparison of the number of model parameters and computation cost in Table 10. Compared with other methods, our model is more compact and efficient. Since our decoder is a differentiable physical solver that does not need to be updated, our model size does not increase much comparing with other graph-convolution-based methods [50, 49]. During the inference, the decoder is discarded so reconstruction is not needed. We adopt a RNN-based recognizer that sequentially processes the input, which may lowers the inference speed. To alleviate the impact of classifier, we may study more efficient sequence processing models such as Transformer [52, 16].

4.7. Comparison of 2D and 3D gait recognition

Although 3D skeleton is view-invariant and robust under occlusion, it loses some appearance information such as the shape of human, which can provides important information for gait recognition. So there is a trade-off using either 2D or 3D input. Compared with 2D gait data, 3D skeletons seem to be more difficult to obtain. This is true to some extent. But for the datasets we use, the 3D poses from Gait3D and GREW are obtained by pose estimation.
<table>
<thead>
<tr>
<th>Method</th>
<th>Input</th>
<th>R-1 (%)</th>
<th>R-5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEINet [44]</td>
<td>2D Silhouette (88 × 128)</td>
<td>7.00</td>
<td>16.30</td>
</tr>
<tr>
<td>GaitSet [4]</td>
<td></td>
<td>42.60</td>
<td>63.10</td>
</tr>
<tr>
<td>GLN [22]</td>
<td></td>
<td>42.20</td>
<td>64.50</td>
</tr>
<tr>
<td>GaitGL [35]</td>
<td>(88 × 128)</td>
<td>23.50</td>
<td>38.30</td>
</tr>
<tr>
<td>CSTL [24]</td>
<td></td>
<td>12.20</td>
<td>21.70</td>
</tr>
<tr>
<td>SMPLGait [65]</td>
<td></td>
<td>53.20</td>
<td>71.00</td>
</tr>
</tbody>
</table>

Table 11: Comparison of 2D-based and 3D-based gait recognition methods on Gait3D.

5. Conclusion, Limitations, and Future Work

Conclusion. In this paper, we introduce physics-augmented autoencoder (PAA) for 3D skeleton-based gait recognition. By combining a graph-convolution-based encoder and a physics-based decoder, the model learns discriminative physical representations, which are fed into a RNN-based classifier for gait recognition. Our method achieves state-of-the-art performance on Gait3D, GREW, and KinectGait. With physics modeling, our method generalizes better and is more robust and data-efficient.

Limitations. Our proposed method relies on physics modeling, which requires 3D skeleton input. Sometimes, 2D pose data is easier to obtain. So we may study how we can conduct the physics modeling on 2D data.

Future work. In this work, we adopt the Lagrangian dynamics for the physics-based decoder. Other physical modeling approaches are also feasible such as Hamiltonian mechanics. We will evaluate and compare different modeling methods in the future. And we may extend the proposed framework on other related tasks such as skeleton-based human action recognition.
References


