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 discared 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.

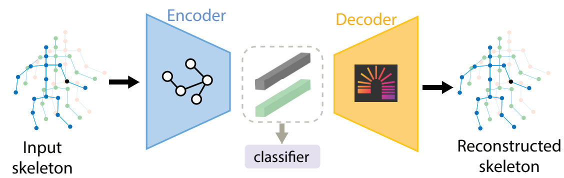


Figure 1: **Illustration of physics-augmented autoencoder (PAA).** It contains a graph-convolution-based encoder and a physics-based decoder to generate physical representations, which are fed into a classifier for gait recognition.

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 [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 to verify the force prediction. We also conducted extensive ablation studies for the model components, generalization, and robustness.

In summary, the main contributions of this paper are:

- We propose a physics-augmented autoenoder 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].

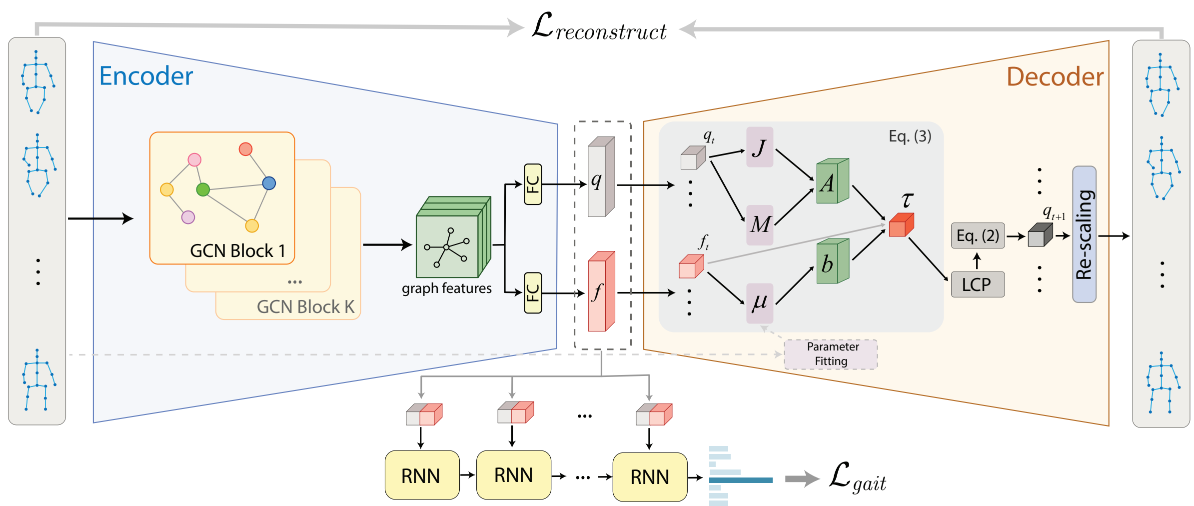


Figure 2: **Overall framework of Physics-Augmented Autoencoder (PAA).** The input of the model is a skeleton sequence. Firstly, a graph-convolution-based encoder takes the input and predicts the joint positions and forces in the generalized coordinates. Then, the physics-based decoder takes the generalized 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 genearlized coordinates. At the same time, the input skeleton sequence is fed into a fitting module to regress basic human body parameters, which are used in the decoder. Given the generalized joint positions and forces, a physics solver plays as the decoder 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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, ..., v_N\}$ is the node set of N joints and $\mathcal{E}$ is the edge set of bones. $\mathcal{E}$ is described by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where $\mathbf{A}_{i,j} = 1$ denotes there is connection between v_i and v_j, and $\mathbf{A}_{i,j} = 0$ means no connections. The graph $\mathcal{G}$ is undirected so $\mathbf{A}$ is symmetric.

Denote the input as $\mathbf{X} \in \mathbb{R}^{T\times N\times C}$, where T is the length of the input sequence and C is the dimension of node features. Then $\mathbf{X}_t \in \mathbb{R}^{N\times C}$ 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 extract 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 $\mathbf{X}$, the k-th ST-GCN block performs the following update at time t:

$$\mathbf{X}_t^{(k+1)} = \sigma\big(\Lambda^{-\frac{1}{2}}\tilde{\mathbf{A}}\Lambda^{-\frac{1}{2}}\mathbf{X}_t^{(k)}W^{(k)}\big) \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, $\Lambda = \sum_j(\mathbf{A}_{ij} + \mathbf{I}_{ij})$ is diagonal degree matrix of $\mathbf{A}$ 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 throught 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 $\boldsymbol{q} \in \mathbb{R}^{T\times D}$ and $\boldsymbol{f} \in \mathbb{R}^{T\times D}$, 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 $\boldsymbol{q}_t$, velocity $\dot{\boldsymbol{q}}_t$, control forces $\boldsymbol{f}$ and inertial properties $\boldsymbol{\mu}$ as the input. And it returns the position and velocity at the next time step, $\boldsymbol{q}_{t+1}$ and $\dot{\boldsymbol{q}}_{t+1}$:

$$D(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \boldsymbol{f}, \boldsymbol{\mu}) = [\boldsymbol{q}_{t+1}, \dot{\boldsymbol{q}}_{t+1}] \tag{2}$$

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

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

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{q}_t, \boldsymbol{\mu})\dot{\boldsymbol{q}}_{t+1} = {} & \boldsymbol{M}(\boldsymbol{q}_t, \boldsymbol{\mu})\dot{\boldsymbol{q}}_t - \Delta t(\boldsymbol{c}(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \boldsymbol{\mu}) - \boldsymbol{f}) \\ & + \boldsymbol{J}^T(\boldsymbol{q}_t)\boldsymbol{\tau}\end{aligned} \tag{3}$$

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

$$\begin{aligned}&\text{find } \boldsymbol{\tau}, \boldsymbol{v}_{t+1} \\ &\text{such that } \boldsymbol{\tau} > 0, \boldsymbol{v}_{t+1} > 0, \boldsymbol{\tau}^T\boldsymbol{v}_{t+1} = 0\end{aligned} \tag{4}$$

The volocity $\boldsymbol{v}_{t+1}$ can be written as a linear function of $\boldsymbol{\tau}$:

$$\begin{aligned}\boldsymbol{v}_{t+1} &= \boldsymbol{J}\dot{\boldsymbol{q}}_{t+1} = \boldsymbol{J}\boldsymbol{M}^{-1}(\boldsymbol{M}\dot{\boldsymbol{q}}_t - \Delta t(\boldsymbol{c} - \boldsymbol{f}) + \boldsymbol{J}^T\boldsymbol{\tau}) \\ &= \boldsymbol{A}\boldsymbol{\tau} + \boldsymbol{b}\end{aligned} \tag{5}$$

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

$$f_{LCP}(\boldsymbol{A}(\boldsymbol{q}_t, \boldsymbol{\mu}), \boldsymbol{b}(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \boldsymbol{f}, \boldsymbol{\mu})) = \boldsymbol{\tau} \tag{6}$$

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 $\mathbf{q}_t \in \mathbb{R}^D$ and forces $\mathbf{f}_t \in \mathbb{R}^D$ to form the feature vector $\mathbf{W}_t \in \mathbb{R}^{2D}$. 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 $\mathbf{X}$, the PAA outputs the reconstructed skeleton sequence $\mathbf{X}'$. The loss function of the first stage is the mean squared error of the input skeleton sequence reconstruction:

$$\mathcal{L}_{reconstruct} = MSE(\mathbf{X}, \mathbf{X}') \tag{7}$$

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

$$\mathcal{L} = \mathcal{L}_{gait} + \lambda \mathcal{L}_{reconstruct} \tag{8}$$

where $\mathcal{L}_{gait}$ is the triplet loss or cross-entropy loss of prediction $\mathbf{P}'$ and the groundtruth $\mathbf{P}$, λ is a hyper-parameter that measures the weight of reconstruction loss. The training process is summarized in Algorithm 1. In practice, λ 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: $\mathcal{D} = \{\boldsymbol{X}_k \in \mathbb{R}^{T\times N\times C}, y_k\}_{k=1}^K$ - training data
Output: Θ_e, Θ_c - parameters of encoder and classifier

Training stage 1
1: **for** $k = 1$ to K **do**
2: Generate $\mathbf{q}_k$, $\mathbf{f}_k$ by the encoder
3: Sequentially predict $\mathbf{q}'_k$ based on Eq. (3)
4: Re-scale $\mathbf{q}'_k$ to $\mathbf{X}'_k$
5: Update Θ_e by minimizing $\mathcal{L}_{reconstruct}$
6: **end for**

Training stage 2
7: **for** $k = 1$ to K **do**
8: Generate $\mathbf{q}_k$, $\mathbf{f}_k$ by encoder
9: Make prediction $\mathbf{P}'$
10: Repeat line 3 to line 4
11: Update Θ_e, Θ_c by minimizing $\mathcal{L}$ in Eq. (8)
12: **end for**
13: **return** Θ_e, Θ_c

4. Experiments

4.1. Datasets

Gait3D [65] is a large-scale gait dataset with 4,000 subjects and 25309 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 18940, 1000, and 5369 sequences for training, validation, and testing respectively.

GREW [66] is a large-scale gait dataset captured in real-world enviroments. It contains 26345 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.

Dataset	Train Set		Test Set		Batch	Epochs
	#Id	#Seq	#Id	#Seq		
Gait3D	3000	18940	1000	6369	128	400
GREW	20000	102887	6000	24000	128	600
K-Gait	164	656	164	166	4	20

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

Method	Gait3D		GREW	
	R-1	R-5	R-1	R-5
PoseGait [34]	26.12	39.79	24.59	35.44
GaitGraph [50]	31.71	48.50	30.22	46.85
GaitGraph2 [49]	33.20	49.62	31.05	47.22
PAA (ours)	**38.92**	**59.08**	**38.71**	**62.07**

Table 2: **Experiment results on Gait3D and GREW.** Our PAA outperforms SOTA methods on both datasets.

Method	Rank-1 (%)
KNN [1]	87.70
Dynamic LSTM [31]	96.56
RDF [60]	95.4
Posture [28]	97.00
CNN-LSTM [36]	97.79
PAA (ours)	**98.42**

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

optimizer [29] with a learning rate of 0.01. For the input size, the number of joints of Gait3D is 24 following the SMPL format. The number of joints of GREW is 14. The number of joints of KinectGait is 20. The training-testing split is provided in Table 1. We made comparison with PoseGait [34], GaitGraph [50], and GaitGraph2 [49]. All the experiments on Gait3D and GREW are conducted based on the official released source code. The input size is modified for each dataset correspondingly.

Decoder. For the physics-based decoder, we adopt the nimblephysics [56], which is a differentiable physical solver that can be integrated into PyTorch for backpropagation and the model is trained in an end-to-end manner. It does not contains trainable parameters.

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.

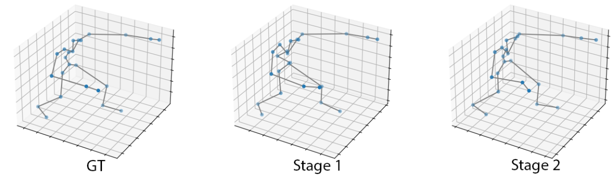


Figure 3: Reconstructed Gait3D skeletons after each training stage. GT denotes the ground-truth skeleton.

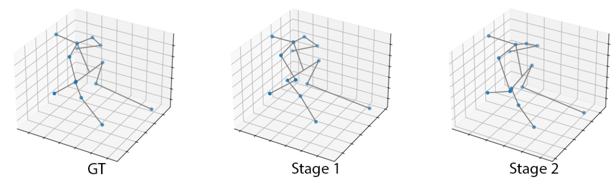


Figure 4: Reconstructed GREW skeletons from different stages. GT denotes the ground-truth skeleton (input).

Decoder type	Gait3D (%)	GREW (%)
MLP	21.67	20.48
RNN	23.80	22.92
LSTM	24.59	24.26
GCN	29.20	28.05
Physics-based	**38.92**	**38.71**

Table 4: **Results with different types of decoders.** We replace the decoder with different types of networks. The physics-based decoder gives the best performance.

Encoder type	Gait3D (%)	GREW (%)
MLP	25.19	24.61
RNN	30.02	29.80
LSTM	30.74	31.09
GCN	**38.92**	**38.71**

Table 5: **Results with different types of encoders.** We replace the encoder with different types of networks. The GCN-based encoder gives the best performance.

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 multi-layer perceptron (MLP), recurrent neural network (RNN), long short-term memory network (LSTM), and graph con-

(a) Gait3D (b) GREW

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

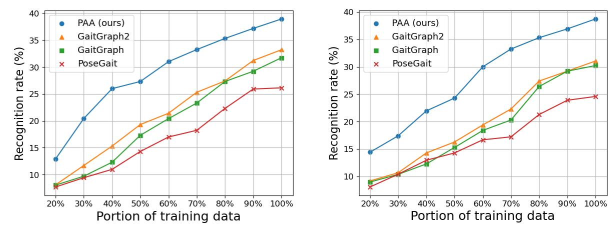
volution network (GCN). The experiment results are shown in Table 4. By adopting the physics-based decoder, our model achieves better performance than other types of decoders, which demonstrates the effectiveness of the physics modeling in the decoder.

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 are 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 λ 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 comparing with other methods when the amount of training data is limited.



(a) Gait3D (b) GREW

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

Method	X-View (%)	X-Carrying (%)
PoseGait [34]	20.87	20.16
GaitGraph [50]	28.10	27.33
GaitGraph2 [49]	28.61	28.04
PAA (ours)	**35.26**	**36.03**

Table 6: **Generalization results.** X-View denotes cross-view and X-carrying denotes cross-carrying.

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 and 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

Method	No	Upper	Lower	Whole
PoseGait [34]	26.12	24.65	21.93	20.65
GaitGraph [50]	31.71	27.48	25.74	23.91
GaitGraph2 [49]	33.20	30.55	28.61	26.10
PAA (ours)	**38.92**	**37.25**	**33.40**	**32.06**

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.

f # of joints	14	20	24	37	54
Rank-1	34.86	37.12	38.92	39.83	**40.91**
Rank-5	50.16	54.23	59.08	59.48	**62.52**

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

Method	Gait3D		GREW	
	R-1	R-5	R-1	R-5
PAA	38.92	59.08	38.71	62.07
PAA+Forces	39.82	61.22	40.15	62.13

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

more detailed skeleton structure.

Adding physics supervision. To make fair comparison, we only use skeleton sequences as the training data in the main experiments. Here we show the PAA can be improved by leveraging a few physical annotations such as measured forces. Specifically, we incorporate the force annotations of 50 subjects from [30] by adding a force prediction supervision in the loss function:

$$\mathcal{L}_{force} = MSE(\boldsymbol{f}, \boldsymbol{f}_{GT}) \tag{9}$$

where $\boldsymbol{f}_{GT}$ is the ground-truth forces. The experiment results are shown in Table 9. With the supervision of ground-truth force annotations, the recognition rate is improved. Thus, it is practical and effective to incorporate a few physical annotations to obtain better performance in real-world deployment.

4.5. Physical plausibility of intermediate representations

By adopting a physics-based decoder, we aim to encode the intermediate representations as generalized positions and the forces of the human joints. To verify the correctness of the prediction from the encoder, we evaluate PAA on a public dataset of human locomotion [30]. The dataset provides motion capture of humans as well as the physical force annotations. We compare the generalized forces of joints with the force annotations of the dataset in a gait cycle. Specifically, we show the annotated ground

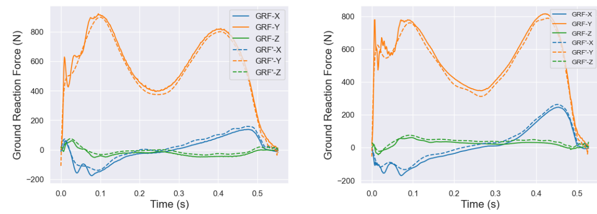


Figure 7: **Ground reaction force (GRF) prediction.** Here we plot the ground reaction forces from two sequences to verify the physical plausibility. GRF' (dashed line) denotes the ground reaction force predicted by the encoder.

Method	# of Parameters	FLOPs
PoseGait [34]	7.74M	0.08G
GaitGraph [50]	320K	0.28G
GaitGraph2 [49]	255K	0.19G
PAA (ours)	225K	0.12G

Table 10: **Comparison of model complexity and computational efficiency.** Our proposed PAA has less model parameters and low FLOPs.

reaction forces and the predicted ones in Figure 7. Comparing with the ground-truth forces, our proposed PAA makes reasonable force predictions that are close to the real physical annotations, which demonstrates that PAA can generate physically plausible representations by the encoder. More visualization and failure cases are available in the supplementary.

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

Method	Input	R-1 (%)	R-5 (%)
GEINet [44]	2D Silhouette (88 × 128)	7.00	16.30
GaitSet [4]		42.60	63.10
GaitPart [11]		29.90	50.60
GLN [22]		42.20	64.50
GaitGL [35]		23.50	38.30
CSTL [24]		12.20	21.70
SMPLGait [65]		**53.20**	**71.00**
GEINet [44]	2D Silhouette (44 × 64)	5.40	14.20
GaitSet [4]		36.70	58.30
GaitPart [11]		28.20	47.60
GLN [22]		31.40	52.90
GaitGL [35]		29.70	48.50
CSTL [24]		11.70	19.20
SMPLGait [65]		**46.30**	**64.50**
PoseGait [34]	3D Skeleton (24 × 3)	26.12	39.79
GaitGraph [50]		31.71	48.50
GaitGraph2 [49]		33.20	49.62
PAA (ours)		**38.92**	**59.08**

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

algorithms from 2D. So basically we start from 2D. Assuming the 3D skeleton is provided, we only compare with 3D-based methods in this paper. To have a full overview, we make performance comparisons on Gait3D and GREW in Table 11 and Table 12 respectively. Compared with state-of-the-art 2D-based gait recognition methods, the performance of our proposed PAA is not as good as them. This may due to following reasons: (1) The input 3D skeleton is not accurate or ambiguous due to the pose estimation algorithms. We notice there are some low-quality skeletons such as the ones visualized in Figure 8. As some of the 3D skeletons are inaccurate or physically implausible, the gait recognition may suffer due to the aggregated error from the beginning. On the other hand, the skeletons of KinectGait dataset are captured by the Microsoft Kinect sensors (depth sensors), they have much better accuracy and quality than the skeletons of Gait3D and GREW. Thus, we see a very high recognition rate (98.26%) on KinectGait. (2) We simply adopt a RNN as the classifier for recognition. It may not well capture the spatial-temporal dependencies among the gait sequences, we may study more advanced sequential models such as Transformer. (3) Although the underlying physical representations are essential and discriminative, they may lose some intra-dependencies such as the relationship between the upper body and lower body. To verify this, we tested different features from the PAA. Specifically, we tested the graph convolution features from the encoder as well as the ensemble features of graph convolution features and the physical features. The experiment results are shown in Table 13.

Method	Input	R-1 (%)	R-5 (%)
GEINet [44]	2D Silhouette (64 × 64)	6.82	13.42
TS-CNN [57]		13.55	24.55
GaitPart [11]		44.01	60.68
GaitSet [4]		**46.28**	**63.58**
PoseGait [34]	3D Skeleton (14 × 3)	24.59	35.44
GaitGraph [50]		30.22	46.85
GaitGraph2 [49]		31.05	47.22
PAA (ours)		**38.71**	**62.07**

Table 12: **Comparison of 2D-based and 3D-based gait recognition methods on GREW.**

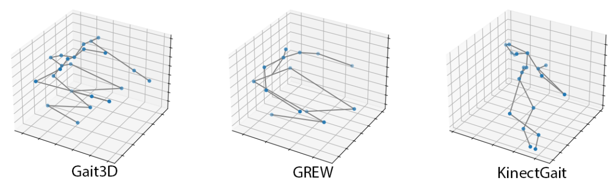


Figure 8: **Low-quality skeletons in the datasets.** Some inaccurate skeletons in the dataset may cause downgrade of the performance.

Feature	Gait3D		GREW	
	R-1	R-5	R-1	R-5
PAA-GCN	30.20	48.20	31.97	49.82
PAA-Physics	38.92	59.08	38.71	62.07
PAA-GCN+Physics	**39.78**	**59.69**	**39.97**	**62.75**

Table 13: **Comparison of features from PAA.** GCN features from the PAA can improve the physical features.

5. Conclusion, Limitations, and Future Work

Conclusion. In this paper, we introduce physics-augmented autoencoder (PAA) for 3D skeleton-based gait recognition. By combing 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 require 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

[1] Virginia Andersson and Ricardo Araujo. Person identification using anthropometric and gait data from kinect sensor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015. 2, 5, 6

[2] Imed Bouchrika. A survey of using biometrics for smart visual surveillance: Gait recognition. In *Surveillance in Action*, pages 3–23. Springer, 2018. 1

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019. 2

[4] Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng. Gaitset: Regarding gait as a set for cross-view gait recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8126–8133, 2019. 9

[5] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7035–7043, 2017. 2, 5

[6] Seokeon Choi, Jonghee Kim, Wonjun Kim, and Changick Kim. Skeleton-based gait recognition via robust frame-level matching. *IEEE Transactions on Information Forensics and Security*, 14(10):2577–2592, 2019. 2, 7

[7] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020. 3

[8] Zijun Cui, Chenyi Kuang, Tian Gao, Kartik Talamadupula, and Qiang Ji. Biomechanics-guided facial action unit detection through force modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8694–8703, 2023. 3

[9] Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in neural information processing systems*, 31, 2018. 3

[10] Jonas Degrave, Michiel Hermans, Joni Dambre, et al. A differentiable physics engine for deep learning in robotics. *Frontiers in neurorobotics*, page 6, 2019. 3

[11] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14225–14233, 2020. 1, 9

[12] Shuo Gao, Jing Yun, Yumeng Zhao, and Limin Liu. Gait-d: Skeleton-based gait feature decomposition for gait recognition. *IET Computer Vision*, 16(2):111–125, 2022. 2

[13] Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. Add: Analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020. 3

[14] Jerry Ginsberg. *Engineering dynamics*, volume 10. Cambridge University Press, 2008. 4

[15] Hongji Guo, Alexander Aved, Collen Roller, Erika Ardiles-Cruz, and Qiang Ji. Skeleton-based human action recognition with a physics-augmented encoder-decoder network. In *Geospatial Informatics XIII*, volume 12525, pages 193–202. SPIE, 2023. 3

[16] Hongji Guo, Hanjing Wang, and Qiang Ji. Uncertainty-guided probabilistic transformer for complex action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20052–20061, 2022. 8

[17] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013. 2

[18] Md Mahedi Hasan and Hossen Asiful Mustafa. Learning view-invariant features using stacked autoencoder for skeleton-based gait recognition. *IET Computer Vision*, 2021. 2

[19] Eric Heiden, David Millard, Erwin Coumans, and Gaurav S Sukhatme. Augmenting differentiable simulators with neural networks to close the sim2real gap. *arXiv preprint arXiv:2007.06045*, 2020. 3

[20] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. 1

[21] Andreas Hochlehnert, Alexander Terenin, Steindór Sæmundsson, and Marc Deisenroth. Learning contact dynamics using physically structured neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 2152–2160. PMLR, 2021. 2

[22] Saihui Hou, Chunshui Cao, Xu Liu, and Yongzhen Huang. Gait lateral network: Learning discriminative and compact representations for gait recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX*, pages 382–398. Springer, 2020. 9

[23] Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019. 3

[24] Xiaohu Huang, Duowang Zhu, Hao Wang, Xinggang Wang, Bo Yang, Botao He, Wenyu Liu, and Bin Feng. Context-sensitive temporal feature learning for gait recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12909–12918, 2021. 9

[25] Thien Huynh-The, Cam-Hao Hua, Nguyen Anh Tu, and Dong-Seong Kim. Learning 3d spatiotemporal gait feature by convolutional network for person identification. *Neurocomputing*, 397:192–202, 2020. 2

[26] Kooksung Jun, Deok-Won Lee, Kyoobin Lee, Sanghyub Lee, and Mun Sang Kim. Feature extraction using an rnn autoencoder for skeleton-based abnormal gait recognition. *IEEE Access*, 8:19196–19207, 2020. 2

[27] K Kashinath, M Mustafa, A Albert, JL Wu, C Jiang, S Esmaeilzadeh, K Azizzadenesheli, R Wang, A Chattopadhyay, A Singh, et al. Physics-informed machine learning: case

studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194):20200093, 2021. 2

[28] Nirattaya Khamsemanan, Cholwich Nattee, and Nitchan Jianwattanapaisarn. Human identification from freestyle walks using posture-based gait feature. *IEEE Transactions on Information Forensics and Security*, 13(1):119–128, 2017. 2, 6

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[30] Tiziana Lencioni, Ilaria Carpinella, Marco Rabuffetti, Alberto Marzegan, and Maurizio Ferrarin. Human kinematic, kinetic and emg data during different walking and stair ascending and descending tasks. *Scientific data*, 6(1):1–10, 2019. 8

[31] Jie Li, Lin Qi, Aite Zhao, Xingnan Chen, and Junyu Dong. Dynamic long short-term memory network for skeleton-based gait recognition. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6. IEEE, 2017. 6

[32] Xiang Li, Yasushi Makihara, Chi Xu, Yasushi Yagi, and Mingwu Ren. Gait recognition via semi-supervised disentangled representation learning to identity and covariate features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13309–13319, 2020. 1

[33] Rijun Liao, Zhu Li, Shuvra S Bhattacharyya, and George York. Posemapgait: A model-based gait recognition method with pose estimation maps and graph convolutional networks. *Neurocomputing*, 501:514–528, 2022. 2

[34] Rijun Liao, Shiqi Yu, Weizhi An, and Yongzhen Huang. A model-based gait recognition method with body pose and human prior knowledge. *Pattern Recognition*, 98:107069, 2020. 6, 7, 8, 9

[35] Beibei Lin, Shunli Zhang, and Xin Yu. Gait recognition via effective global-local feature representation and local temporal aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14648–14656, 2021. 9

[36] Yu Liu, Xinghao Jiang, Tanfeng Sun, and Ke Xu. 3D gait recognition based on a CNN-LSTM network with the fusion of SkeGEI and DA features. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8. IEEE, 2019. 2, 6

[37] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490*, 2019. 2

[38] Samuel E Otto and Clarence W Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019. 3

[39] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[40] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 2

[41] Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming C Lin. Scalable differentiable physics for learning and control. *arXiv preprint arXiv:2007.02168*, 2020. 3

[42] Apoorva Rajagopal, Christopher L Dembia, Matthew S DeMers, Denny D Delp, Jennifer L Hicks, and Scott L Delp. Full-body musculoskeletal model for muscle-driven simulation of human gait. *IEEE transactions on biomedical engineering*, 63(10):2068–2079, 2016. 4

[43] Yanzhi Ren, Yingying Chen, Mooi Choo Chuah, and Jie Yang. User verification leveraging gait recognition for smartphone enabled mobile healthcare systems. *IEEE Transactions on Mobile Computing*, 14(9):1961–1974, 2014. 1

[44] Kohei Shiraga, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. Geinet: View-invariant gait recognition using a convolutional neural network. In *2016 international conference on biometrics (ICB)*, pages 1–8. IEEE, 2016. 9

[45] Jiande Sun, Yufei Wang, Jing Li, Wenbo Wan, De Cheng, and Huaxiang Zhang. View-invariant gait recognition based on kinect skeleton feature. *Multimedia Tools and Applications*, 77(19):24909–24935, 2018. 2

[46] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 2

[47] Naoya Takeishi and Alexandros Kalousis. Variational autoencoder with differentiable physics engine for human gait analysis and synthesis. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 3

[48] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[49] Torben Teepe, Johannes Gilg, Fabian Herzog, Stefan Hörmann, and Gerhard Rigoll. Towards a deeper understanding of skeleton-based gait recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1569–1577, 2022. 1, 2, 6, 7, 8, 9

[50] Torben Teepe, Ali Khan, Johannes Gilg, Fabian Herzog, Stefan Hörmann, and Gerhard Rigoll. Gaitgraph: Graph convolutional network for skeleton-based gait recognition. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2314–2318. IEEE, 2021. 2, 6, 7, 8, 9

[51] Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789*, 2019. 3

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 8

[53] Likai Wang, Jinyan Chen, Zhenghang Chen, Yuxin Liu, and Haolin Yang. Multi-stream part-fused graph convolutional networks for skeleton-based gait recognition. *Connection Science*, 34(1):652–669, 2022. 2

[54] Likai Wang, Jinyan Chen, and Yuxin Liu. Frame-level refinement networks for skeleton-based gait recognition. *Computer Vision and Image Understanding*, 222:103500, 2022. 2

[55] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1457–1466, 2020. 2

[56] Keenon Werling, Dalton Omens, Jeongseok Lee, Ioannis Exarchos, and C Karen Liu. Fast and feature-complete differentiable physics for articulated rigid bodies with contact. *arXiv preprint arXiv:2103.16021*, 2021. 4, 6

[57] 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, 2016. 9

[58] Zhaoyang Xia, Genming Ding, Hui Wang, and Feng Xu. Person identification with millimeter-wave radar in realistic smart home scenarios. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2021. 1

[59] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 4

[60] Ke Yang, Yong Dou, Shaohe Lv, Fei Zhang, and Qi Lv. Relative distance features for gait recognition with Kinect. *Journal of Visual Communication and Image Representation*, 39:209–217, 2016. 2, 6

[61] Tsung-Yen Yang, Justinian P Rosca, Karthik R Narasimhan, and Peter Ramadge. Learning physics constrained dynamics using autoencoders. In *Advances in Neural Information Processing Systems*. 1, 3

[62] Miloš Zefran and Francesco Bullo. Lagrangian dynamics. *Robotics and Automation Handbook*, pages 5–1, 2005. 2

[63] Ruiyang Zhang, Yang Liu, and Hao Sun. Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling. *Engineering Structures*, 215:110704, 2020. 2

[64] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012. 2

[65] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20228–20237, 2022. 2, 5, 9

[66] Zheng Zhu, Xianda Guo, Tian Yang, Junjie Huang, Jiankang Deng, Guan Huang, Dalong Du, Jiwen Lu, and Jie Zhou. Gait recognition in the wild: A benchmark. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14789–14799, 2021. 2, 5