Sequential Joint Dependency Aware Human Pose Estimation with State Space Model

Hanxi Yin[1], Shaodi You[1], Jungong Han[2], Zhixiang Chen[2]*

[1]University of Amsterdam
[2]University of Sheffield
{h.yin, s.you}@uva.nl, {jungong.han, zhixiang.chen}@sheffield.ac.uk

Abstract

In this paper, we present a sequential joint dependency aware model for monocular 2D-to-3D human pose estimation. While existing estimators leverage the (bi)directional joint dependency with graph convolutions and attention, we further propose to exploit the sequential dependency between joints with state space model (SSM) with a pose SSM module. Our sequential dependency takes into consideration the information of kinematic chain, joint hierarchy and the body part. We design a sequential dependency aware representation to transform the pose data into sequential data for our pose SSM module. We tailor the SSM layer in the pose SSM module for pose estimation by learning joint-dependent parameters and introducing pose aware hidden state initialization. Extensive experiments are conducted on two datasets to validate the effectiveness of our proposed SSM module, and the results demonstrate that our pose estimator can deliver impressive performance.

Code — https://github.com/yinhanxi/PoseSSM

Introduction

3D human pose estimation (HPE) is a longstanding computer vision problem and has broad application in virtual reality, human tracking (Mehta et al. 2017b), human-robot interaction (Errity 2016), computer animation, and human behavior analysis. Monocular 3D HPE aims to generate 3D human joints locations from a single-frame image. In comparison to multi-frame approach, single-frame approach does not requires high computational costs and expensive multi-view capture system, making it widely used (Zhai et al. 2023). Monocular 3D HPE is an ill-posed problem due to the depth ambiguity caused by the many-to-one 3D-to-2D projection. Existing methods can be broadly categorized into end-to-end learning architectures that infer 3D pose directly from images and two stage solutions that estimate 2D joint from images followed by 3D pose estimation from 2D joint detection. Similar to (Gong et al. 2023; Zhao, Wang, and Tian 2022; Zhai et al. 2023), in this paper, we study the latter and focus on the lifting from 2D joints to 3D joints as the first stage can leverage the high performance of existing 2D keypoint detectors (Dabral et al. 2018; Pham et al. 2020).

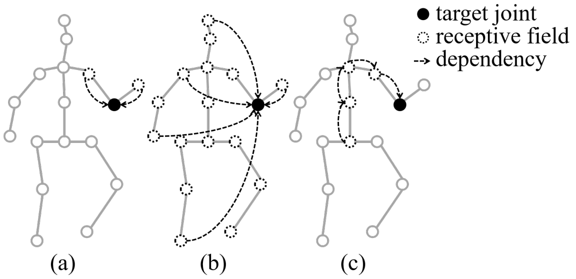


Figure 1: An illustration of (a) the dependency in graph convolutions, (b) the dependency between any two joints in attention and (c) the sequential dependency in our pose SSM module.

The latest progress leverages graph convolutional networks (GCNs) (Kipf and Welling 2017) and Transformer (Vaswani et al. 2017) to capture the relationship between human body joints. As human skeleton can be well represented by a graph, GCNs are utilized to extract local features by aggregating neighbor nodes in skeletal structure. While GCNs have been proven effective in dealing with skeleton graph (Zhao et al. 2019; Ci et al. 2019; Liu et al. 2020; Zou and Tang 2021; Xu and Takano 2021; Zeng et al. 2021), their performance are limited for only considering first-order neighbors and failing to understand global information. Stacking multiple GCN layers may increase model's receptive fields but will encounter the inherent over-smooth problem (Li, Han, and Wu 2018). On the other hand, transformer-based methods (Zheng et al. 2021; Zhang et al. 2022; Zhu et al. 2023) treat human joints as a fully-visible sequence. The self-attention model is employed to capture dependencies among all joint nodes by calculating the joints similarities, and thus obtain global receptive fields. However, this self-attention mechanism lacks some of the inductive biases inherent to GCNs such as structural information in human joints, which may results in limited capacity to capture the underlying patterns in skeleton graph and unsatisfied generalization. Some recent works (Zhao, Wang, and Tian 2022; Zhai et al. 2023; Kang et al. 2023) have applied both GCNs and transformer to extract local and global features from human joints, achieving promising results. Generally, GCNs and transformers, with different receptive

[*] Corresponding author

fields and inductive biases, are effective components for extracting different pose features by treating human skeleton as a graph and a fully-visible sequence, respectively.

Different from these two representations of human pose data, we argue that the sequential joint dependency is beneficial to the 3D pose estimation. This is based on the observation that human skeleton defined in Human3.6M dataset (Ionescu et al. 2014) can also be viewed as a kinematic structure (Xu et al. 2020; Chen et al. 2022; Wang et al. 2021; Cai et al. 2024). As shown in Fig. 2(a), the kinematic chain starts at the root joint, extending through the legs, torso, and arms, ultimately terminating at feet, head and wrists. In this hierarchical structure, there exist sequential relationships between the parent nodes (lower-hierarchy joints) and their child nodes (higher-hierarchy joints). Specifically, the location of child joint is influenced by that of its parent joint along with the length and direction of the corresponding bone. To model such sequential dependency in human kinematic chain, some researchers (Xu et al. 2020; Chen et al. 2022; Wang et al. 2021) explicitly regress the bone length and direction, followed by composing the whole skeleton through forward kinematics to obtain 3D joint locations. However, these methods suffer from error accumulation across hierarchies, often leading to deteriorated results. Cai et al. (Cai et al. 2024) designed a complex transformer-based module to inject hierarchical information into joint features and aggregate hierarchical adjacent joints, which resembles the combination of transformer and GCNs.

In contrast with them, we introduce state space model (SSM) to capture such dependency by designing a sequential dependency aware pose representation and a joint-dependent SSM. This is motivated by the ability of SSMs like Mamba (Gu and Dao 2023) to process causal chain as suggested in MambaOut (Yu and Wang 2024) and MLLA (Han et al. 2024). Our main contributions are:

- We introduce a new 3D human pose estimator to capture the sequential dependency between human joints using a pose state space model (SSM).
- We design a sequential dependency aware representation to transform the pose data into sequential data, which considers the kinematic chain, joint hierarchy and body part information.
- We tailor a pose SSM module with joint-dependent parameters and pose aware hidden state initialization for sequential joint modeling.

We conduct extensive experiments on two widely used datasets. The ablation study shows the effectiveness of our designed components. The comparison with existing methods demonstrates the superiority of our proposed method.

Related Work

3D Human Pose Estimation

There exist two main paradigms in 3D human pose estimation: direct regression from RGB images to 3D joints (Zhao et al. 2019; Pavlakos et al. 2017; Sun et al. 2018), and a two-stage approach that first estimates 2D pose from images and then infers 3D pose from the detected 2D joints. In this paper, we concentrate on the latter paradigm, which, with the rapid development of 2D pose estimation (Huang et al. 2023; Wang et al. 2022; Sun et al. 2019; Chen et al. 2018), reveals immense potential for better performance.

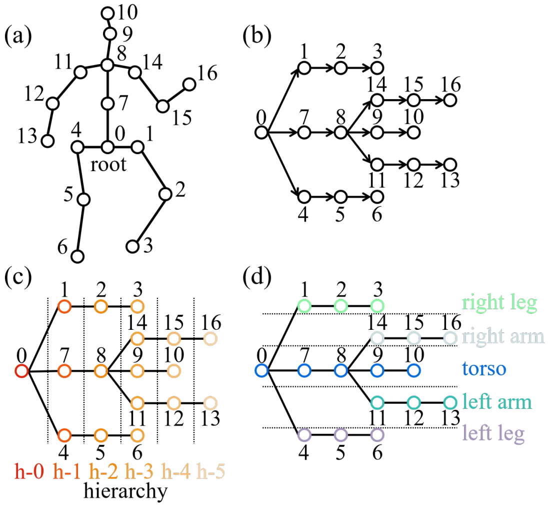
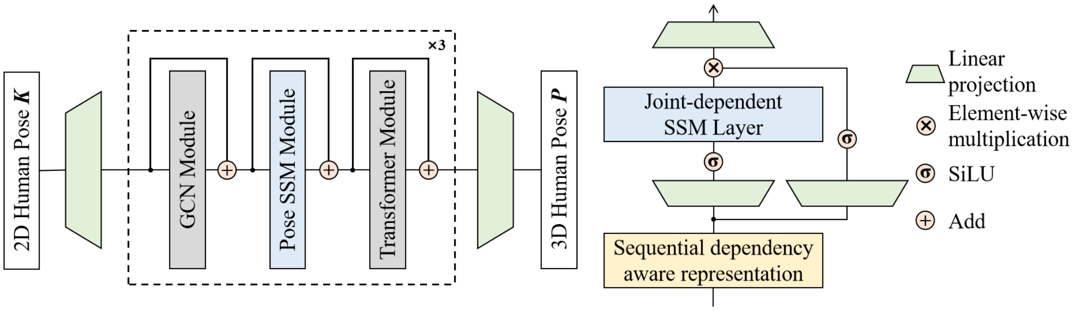


Figure 2: An illustration of (a) the human skeleton, (b) the kinematic chain, (c) the corresponding joints hierarchy, and (d) five body parts.

Martinez et al. (Martinez et al. 2017) firstly introduced a simple baseline for 2D-to-3D lifting by using fully connected layer with residual connection, batch normalization, dropout and activation. In the following works, some researchers (Pavllo et al. 2019; Li et al. 2020; Zeng et al. 2020) further advanced the 3D pose estimator based on linear layers. As the human skeleton can be naturally represented by a graph, some works (Zhao et al. 2019; Liu et al. 2020; Zou and Tang 2021; Xu and Takano 2021) utilized graph neural network (GCN) (Kipf and Welling 2017) to capture the relation between joints, mainly working on improving the graph construction and joint feature aggregation. There are also some attempts (Zheng et al. 2021; Zhang et al. 2022; Zhu et al. 2023) to extract both spatial and temporal features with transformer (Vaswani et al. 2017). Moreover, subsequent studies(Zhao, Wang, and Tian 2022; Zhai et al. 2023; Kang et al. 2023) leveraged the combination of transformer and GCN to extract diverse types of joint features, achieving impressive results. In distinction to them, we are the first to use state space model (SSM) (Dorf and Bishop 2008) to exploit the sequential dependency between joints. More recently, Gong et al. (Gong et al. 2023) proposed a diffusion-based approach that formulates 3D pose estimation as a reverse diffusion process, which is perpendicular to our work.

State Space Models

Recently, state space models have gained great attention for their impressive results in language models. As a pioneer work, Gu et al. (Gu, Goel, and Ré 2022) introduced a structured state-space sequence model (S4) to handle long-range dependencies in time sequences with linear complexity. Subsequently, various works (Smith, Warrington, and



(a) Overall Architecture

(b) Pose SSM Module

Figure 3: (a) The overall framework of our proposed pose estimator. (b) The pose SSM module consists of the sequential dependency aware representation (Fig. 4) and the joint-dependent SSM layer (Alg. 1) with simplified selective mechanism.

Linderman 2023; Fu et al. 2023) further improved S4 by introducing different architectural enhancements. More recently, Mamba (Zhu et al. 2024) excels other methods owing to a time- and input-dependent selective mechanism and efficient hardware design.

In the vision domain, many works (Zhu et al. 2024; Liu et al. 2024) apply SSM as a novel alternative to CNN or transformer by enabling 1D scanning in 2D image space. Moreover, some attempts (Islam and Bertasius 2022; Islam et al. 2023; Wang et al. 2023) use SSM to handle the long-range temporal dependencies in videos. In contrast to them, we utilize SSM in human pose estimation to exploit the sequential dependency between human joints.

Sequential Dependency Aware Pose Estimator

3D Human Pose Estimation. In this work, we consider the estimation of 3D human poses from 2D human keypoints, i.e. pose lifting. Let $\boldsymbol{K} = \{\boldsymbol{k}_x, \boldsymbol{k}_y\} \in \mathbb{R}^{L\times 2}$ denote the two coordinates of the L joints of a human in a projected view, where L defines the number of keypoints. We aim to learn a mapping Ψ from $\boldsymbol{K}$ to $\boldsymbol{P} = \{\boldsymbol{p}_x, \boldsymbol{p}_y, \boldsymbol{p}_z\} \in \mathbb{R}^{L\times 3}$, which are the three coordinates of the L joints of a human in a global coordinate system. The 2D keypoints could be obtained from 2D images by using an off-the-shelf human keypoint detector like (Chen et al. 2018) or projecting the ground truth measured 3D keypoints to an image view. For the 3D coordinates, we follow previous work (Martinez et al. 2017) to choose the camera coordinate system as the global coordinate system.

To learn the mapping $\Psi : \boldsymbol{K} \mapsto \boldsymbol{P}$ from pairs of training samples, we propose a sequential dependency aware pose estimator to exploit the relation between joints. As illustrated in Fig. 2, there are three kinds of relation considered in this work. Fig. 2(a) shows the topology of the pose template used in this work. In Fig. 2(b), the kinematic chain captures the assembly of rigid parts in the predefined topology. It models the physical dependency between joints for body movements with the hip as the root. In Fig. 2(c), the joint hierarchy measures the distance of each joint to the root hip joint. The distance is calculated by the number of hops. In Fig. 2(d), the body part information categorizes each joint into one of the five functional groups, i.e. right leg, right arm, torso, left arm, and left leg. There are sequential dependency in both the kinematic chain and the joint hierarchy.

To model the sequential dependency between joints, we introduce the state space model (SSM) (Dorf and Bishop 2008) to learn the mapping Ψ and design an representation of the pose data for the SSM module. The overall framework of the proposed model is shown in Fig. 3(a). We adopt a linear projection layer to transform the 2D keypoints $\boldsymbol{K}$ to an embedded feature $\boldsymbol{F} \in \mathbb{R}^{L\times D}$, where D denotes the number of feature channels. The embedded feature $\boldsymbol{F}$ is further processed by the backbone of the pose estimator. Following the practice in (Zou and Tang 2021; Kang et al. 2023), there are three repetitive blocks to learn multi-level features to enhance the model capacity. Each block consists of three modules. Inspired by the success of recent works (Zhao, Wang, and Tian 2022; Kang et al. 2023), we utilize a GCN module and a Transformer module along with our proposed pose SSM module to update the joint features by exploiting different properties of the pose data. The GCN module takes the spatial skeleton graph into consideration to update the joint features. The Transformer module models the dependency between any two joints to update the features with the attention mechanism. The pose SSM module exploits the sequential dependency between joints to learn joint features with a customized SSM layer, which is a recurrent model. To feed the joint features into the SSM layer, we convert the joint features in graph to sequence with a sequential dependency aware data representation for human pose. The output head consists of a linear projection to regress the 3D pose $\boldsymbol{P}$ from the updated joint features.

Sequential Dependency Aware Representation

To minimize the negative impact of sequencing joint features for SSM, we consider to preserve three key information, kinematic chain, joint hierarchy, and body parts in the sequencing. Our assumption of this sequencing is that the kinematic chain is the key to represent the skeleton graph. We exploit two options, BPS and JHS to preserve the joint hierarchy and the body part information.

BPS. The body part-sorted sequence serializes the joints according to the kinematic branch information with the root joint as the first joint, as shown in Fig. 4 (a). The body parts are arranged in the following order: right leg, left leg, torso,

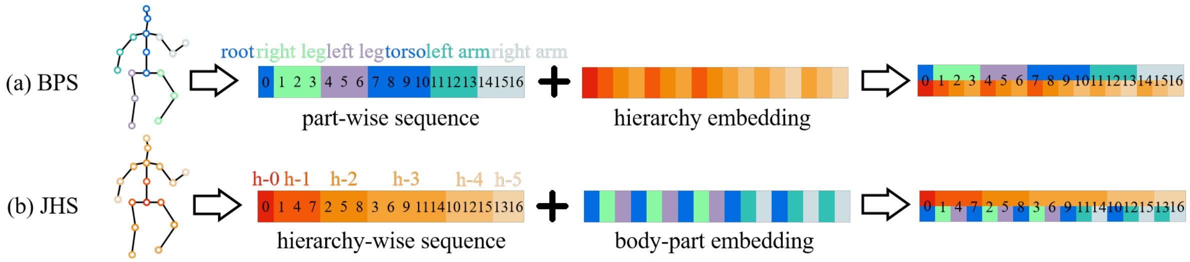


Figure 4: Sequential dependency aware representation. (a) Body part sorted serialization (BPS). (b) Joint hierarchy sorted serialization (JHS).

left arm, and right arm. This sequencing makes it easier to exploit the dependency between joints within the same body part as they are close to each other. For instance, joints ‘1’, ‘2’, and ‘3’ are close to each other. The feature of joint ‘3’ could be updated based on the status of the joints ‘1’ and ‘2’. However, this sequencing overlooks the joint hierarchy information across body parts. As a remedy, a learnable embedding could be introduced to explicitly represent the joint hierarchy, which is further combined with the joint features $\boldsymbol{F}$ to form the representation of each joint. Nevertheless, such representation cannot resolve the issue due to the recurrent nature of the SSM. For instance, joint ‘4’ is placed after joint ‘3’. There is no mechanism for SSM to exploit the features of joint ‘4’ to update the status of joint ‘3’. This situation persists even after the order of the body parts is changed. We argue that the preservation of the joint hierarchy information is critical when using SSM for human pose estimation task. This motivates us to propose JHS.

JHS. As shown in Fig. 4 (b), we serialize the kinematic chain according to the joint hierarchy in joint hierarchy-sorted serialization (JHS). This preserves the relative order of joints in the kinematic chain and the joint hierarchy. We put joints with high joint hierarchy towards the end of the sequence as they are likely dependent on the joints with low joint hierarchy in the same branch. Taking the branch ‘0-1-2-3’ as an example, we place them at the first, second, fifth, and eight position in the sequence, respectively. However, this sequencing overlooks the body part information which indicates whether two joints are part of the same branch. For example, there is no sequential dependency between joints ‘1’ and ‘4’ despite their proximity in the sequence. To address this drawback, we propose to learn a body-part embedding for each body part. The joints within each body part share the same body-part embedding. Specifically, we divide human skeleton into five parts based on the distinct kinematic chain branches: left/right arms, left/right legs and the torso. The body part embedding $\boldsymbol{E}_{bp} \in \mathbb{R}^{L \times D}$ is further integrated with the reordered joint features $\boldsymbol{F} \in \mathbb{R}^{L \times D}$ through an element-wise addition. This combination results in the sequential dependency aware representation for all joints $\boldsymbol{F}_s \in \mathbb{R}^{L \times D}$. The sequential dependency between joints could then be exploited by the SSM as the causal relationship between tokens.

```
Algorithm 1: Joint-Dependent SSM Layer
Input: $\boldsymbol{F}_x$: (B, L, $\mathrm{D_s}$)
Output: $\boldsymbol{F}_y$: (B, L, $\mathrm{D_s}$)
$\boldsymbol{A}$: ($\mathrm{D_s}$, N) $\leftarrow$ Parameter
                    $\triangleright$ Represents structured N $\times$ N matrix
$\boldsymbol{B}$: (L, N) $\leftarrow$ Parameter
$\boldsymbol{C}$: (L, N) $\leftarrow$ Parameter
$\boldsymbol{\Delta}$: (L, $\mathrm{D_s}$) $\leftarrow$ Softplus(Parameter)
$\bar{\boldsymbol{A}}$, $\bar{\boldsymbol{B}}$: (L, $\mathrm{D_s}$, N) $\leftarrow$ discretize($\boldsymbol{\Delta}$, $\boldsymbol{A}$, $\boldsymbol{B}$)
$\hat{\boldsymbol{h}}$: (B, $\mathrm{D_s}$, N) $\leftarrow$ initialize($\bar{\boldsymbol{B}}$, $\boldsymbol{F}_x$) with Eq. 2
$\boldsymbol{F}_y \leftarrow$ SSM($\bar{\boldsymbol{A}}$, $\bar{\boldsymbol{B}}$, $\boldsymbol{C}$)($\boldsymbol{F}_x$, $\hat{\boldsymbol{h}}$) with Eqs. 3 and 4
                    $\triangleright$ Joint-dependent
return $\boldsymbol{F}_y$
```

The advantage of JHS over BPS is two-folded. Firstly, the relative orders within the joint hierarchy and the kinematic branch are preserved for each joint in JHS, which makes it possible for SSM to exploit the dependency between joints across body parts. Secondly, placing joints with a lower joint hierarchy at the beginning of the sequence helps capture the global feature as joints from different body parts are observed. This benefits the update of hidden state in the SSM.

Joint-Dependent SSM

State Space Model (SSM). The state space model (Dorf and Bishop 2008) is a two-stage continuous system where the input $x_t \in \mathbb{R}$ at time t is mapped to an output $y_t \in \mathbb{R}$ via a hidden state $\boldsymbol{h}_t \in \mathbb{R}^{N \times 1}$. This system is defined with three parameters $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, $\boldsymbol{B} \in \mathbb{R}^{N \times 1}$ and $\boldsymbol{C} \in \mathbb{R}^{1 \times N}$. More specifically, the system could be described as:

$$\begin{aligned} \boldsymbol{h}_t &= \boldsymbol{A}\boldsymbol{h}_{t-1} + \boldsymbol{B}x_t, \\ y_t &= \boldsymbol{C}\boldsymbol{h}_t. \end{aligned} \tag{1}$$

There is a timescale parameter $\Delta \in \mathbb{R}$ for the discretization of SSM for the implementation with deep neural networks. By leveraging SSM to process sequential pose data, we aim to capture the dependencies between joints in a way similar to the dependencies between tokens in other sequence data modalities.

We build our joint-dependent SSM on top of the simplified selective mechanism in Mamba (Gu and Dao 2023), which allows the model to selectively propagate or forget information along the sequence length dimension. As shown in Fig. 3(b), with the sequential dependency aware representation $\boldsymbol{F}_s \in \mathbb{R}^{L\times D}$ at hand, we apply two separated linear layer to project them to SSM input features $\boldsymbol{F}_x \in \mathbb{R}^{L\times D_s}$ and gating features $\boldsymbol{F}_z \in \mathbb{R}^{L\times D_s}$, where $D_s = D \times K$ and K is the expansion factor. The features $\boldsymbol{F}_x$ is further processed by a joint-dependent SSM layer before gated by $\boldsymbol{F}_z$. By controlling the flow of information, the gated mechanism can dynamically select features and thereby enhance model's expressivity. The gated features $\boldsymbol{F}_y$ are then projected back to the original feature dimension D with a linear layer and reordered back for other modules of the estimator.

The overall description of our joint-dependent SSM layer is presented in Algorithm. 1. We make two key contributions to the adaptation of SSM as joint-dependent SSM for our human pose estimation task. This is based on the observation of the difference between pose data and other sequence data. Different to other general sequence data with varying length, the pose data is with fixed length, which is the total number of joints. Moreover, each position in the sequence has a fixed physical meaning, which represents a specific joint.

Firstly, we let SSM parameters ($\boldsymbol{B}$, $\boldsymbol{C}$, and Δ) be functions of the index of the each token in the sequence, where a token is the feature of a joint. For our pose data with sequence length L, we have parameters $\boldsymbol{A} \in \mathbb{R}^{D_s\times N}$, $\boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^{L\times N}$ and $\boldsymbol{\Delta} \in \mathbb{R}^{L\times D_s}$. We follow the discretization in S4 (Gu, Goel, and Ré 2022) to obtain the discrete version of parameters $\boldsymbol{A}$ and $\boldsymbol{B}$ as $\bar{\boldsymbol{A}} \in \mathbb{R}^{L\times D_s\times N}$, $\bar{\boldsymbol{B}} \in \mathbb{R}^{L\times D_s\times N}$. The SSM then becomes joint-dependent as the parameters for each joint are distinct. Note that we share the parameters across poses to make them joint dependent rather than pose dependent. The joint dependent parameters allow the model to learn how each joint contributes to the update of the hidden state in SSM and how the hidden state impacts the outputs of each joint. In practice, we simply assign learnable parameters for each joint.

Secondly, we leverage the SSM input features $\boldsymbol{F}_x \in \mathbb{R}^{L\times D_s}$ to initialize the hidden state as $\hat{\boldsymbol{h}}$ in SSM. Specifically, we calculate $\hat{\boldsymbol{h}}$ as

$$\hat{\boldsymbol{h}} = \sum_{l=0}^{L-1} \bar{\boldsymbol{B}}(l,:,:)\boldsymbol{F}_x(l,:)/L. \tag{2}$$

This works in a way similar to how the input features contribute to the hidden state as in Eq. 1. $\bar{\boldsymbol{B}}$ can be viewed as a gate measuring the utility factor of the features $\boldsymbol{F}_x$ to the initial hidden state. We take the average of the contributions from all joints as the initialization of the hidden state. In comparison to the zero initialization employed in S4 (Gu, Goel, and Ré 2022) and Mamba (Zhu et al. 2024), our initialization could provide a prior of the pose for the sequence processing. To obtain the output feature $\boldsymbol{F}_y$, our joint-dependent SSM recurrently updates the hidden state and maps it to the output by using the following equations:

$$\boldsymbol{h}_l = \begin{cases} \bar{\boldsymbol{A}}(0,:,:)\hat{\boldsymbol{h}} + \bar{\boldsymbol{B}}(0,:,:)\boldsymbol{F}_x(0,:), & l=0, \\ \bar{\boldsymbol{A}}(l,:,:)\boldsymbol{h}_{l-1} + \bar{\boldsymbol{B}}(l,:,:)\boldsymbol{F}_x(l,:), & 1 \le l \le L-1, \end{cases} \tag{3}$$

$$\boldsymbol{F}_y(l,:) = \boldsymbol{C}(l,:)\boldsymbol{h}_l, \;\; 0 \le l \le L-1. \tag{4}$$

Learning Loss. To learn the pose estimator in Fig. 3(a), we use paired 2D pose $\boldsymbol{K}$ and the ground truth 3D pose $\hat{\boldsymbol{P}}$ to minimize the weighted L_2 loss

$$\mathcal{L} = \frac{1}{L}\sum_{l=0}^{L-1}(w_l\|\boldsymbol{P}_l - \hat{\boldsymbol{P}}_l\|_2), \tag{5}$$

where $\boldsymbol{P}$ is the output of the pose estimator.

Experiments

Datasets and Evaluation Protocols

Our method is evaluated on Human3.6M (Ionescu et al. 2014) and MPI-INF-3DHP (Mehta et al. 2017a).

Human3.6M. Following previous works (Gong, Zhang, and Feng 2021; Wandt and Rosenhahn 2019; Martinez et al. 2017; Zeng et al. 2020; Chen et al. 2018), we utilize subjects 1, 5, 6, 7 and 8 for training, and subjects 9 and 11 for evaluation. We evaluate model performance on Human3.6M with two metrics: Mean Per Joint Position Error (MPJPE) in millimeters and MPJPE with Procrustes alignment between ground truth and predicted poses (P-MPJPE), which are referred as Protocol #1 (P1) and Protocol #2 (P2) respectively.

MPI-INF-3DHP. MPI-INF-3DHP is a large 3D pose dataset commonly used for cross-dataset evaluation (Gong, Zhang, and Feng 2021; Wandt and Rosenhahn 2019; Zeng et al. 2020; Li et al. 2020). Compared to the data samples in Human3.6M, there are unseen poses and actions in this dataset. We use the evaluation metrics, Percentage of Correct Keypoints (PCK) and Area Under the Curve (AUC) to compare model performance. We report PCK within 150mm and AUC calculated over a range of PCK thresholds from 0 to 150mm with a step of 5mm, in line with previous work (Gong, Zhang, and Feng 2021).

Implementation Details

In line with previous works (Zou and Tang 2021; Kang et al. 2023; Zhai et al. 2023), we utilize both the ground truth 2D poses (GT) and the 2D poses detected by the cascaded pyramid network (Chen et al. 2018) (CPN) as inputs. Our method is implemented on a single NVIDIA GeForce RTX 4090 GPU. We train the model 30 epochs with a batch size of 512. The learning rate is initialized at 0.0005, decayed by 0.95 per epoch and halved every 5 epochs. Horizontal flip is applied as data augmentation in training. No additional refinement module is used. We run experiments 3 times to report the best results. We follow previous work (Zhang et al. 2022) to set the weighting factor w in Eq. 5.

In our model, we use the Local Constraint Module (LCM) and Global Constraint Module (GCM) in DC-GCT (Kang et al. 2023) as the GCN module and the Transformer module. We sequentially arrange the modules as it performs better than parallel arrangement. We apply weight of 2 for the residual connections in each module. Experimentally, we set D, K and N as 160, 2 and 4 with the optimal MPJPE on Human3.6M by searching each parameter independently. We search D from 96 to 240, with a step size of 16, K from $\{2,4,8\}$ and N from $\{2,4,8,16\}$. We initialize SSM parameters $\boldsymbol{A}$ and Δ according to Mamba (Gu and Dao 2023), and randomly initialize $\boldsymbol{B}$ and $\boldsymbol{C}$.

Methods	P1(CPN)	P2(CPN)	P1(GT)
(Martinez et al. 2017)	62.9	47.7	45.5
(Zhao et al. 2019)	57.6	-	43.8
(Ci et al. 2019)	52.7	42.2	-
(Pavllo et al. 2019)	51.8	-	-
(Cai et al. 2019)	50.6	40.2	38.1
(Liu et al. 2020)	52.4	41.2	37.8
(Zeng et al. 2020)	49.9	39.4	36.4
(Zou and Tang 2021)	49.4	39.1	37.4
(Xu and Takano 2021)	51.9	-	-
(Zhao, Wang, and Tian 2022)	51.8	-	35.2
(Cai et al. 2023)	48.9	39.0	34.0
(Li et al. 2023)	50.5	-	-
(Gong et al. 2023)(†)	49.7	-	<u>31.6</u>
(Zhai et al. 2023)	<u>48.5</u>	-	32.7
(Chen et al. 2024)	49.8	<u>38.9</u>	32.4
Ours	**48.1**	**37.9**	**31.3**

Table 1: Quantitative comparison on Human3.6M dataset with 2D poses detected by CPN and ground truth (GT) 2D poses as inputs under Protocol #1 and Protocol #2. The best results are highlighted in bold and the second-best results are underlined. (†) indicates probabilistic methods.

Comparison with State-of-the-art

We first compare our method with state-of-the-art methods on the Human3.6M dataset to validate the effectiveness of our approach both quantitatively and qualitatively, and then perform cross-dataset experiments on the MPI-INF-3DHP dataset to verify its generalization capability.

Comparison on Human3.6M. We first conduct an experiment using 2D poses detected by CPN (Chen et al. 2018) as inputs, following previous methods (Zhai et al. 2023; Gong et al. 2023). The quantitative results under Protocol #1 and Protocol #2 are shown in the second and third columns of Table 1 respectively. It can be observed that our method achieves a performance of 48.1mm under Protocol #1 and 37.9mm under Protocol #2, outperforming all previous approaches.

Given the interference from the uncertainty in detected 2D poses, we further evaluate our method's performance by leveraging ground truth 2D poses as inputs. As shown in the fourth column of Table 1, our method achieves the best result of 31.3mm under Protocol #1, excelling even the probabilistic method (Gong et al. 2023). In comparison with other deterministic methods, our model obtains a 1.1mm improvement in Protocol #1 over the best result by (Chen et al. 2024).

The qualitative performance of our method on the Human3.6M test set is presented in Fig. 5. We compare our approach with GraFormer (Zhao, Wang, and Tian 2022) and HTNet (Cai et al. 2023) as their source codes are available. We utilize detected 2D poses as inputs. We can observe that our method predicts 3D human poses with better accuracy than the two compared methods in different actions. We attribute the enhancement to the exploitation of sequential dependency by the pose SSM module, which models the sequential relationship between joints of varying hierarchies in the kinematic chain.

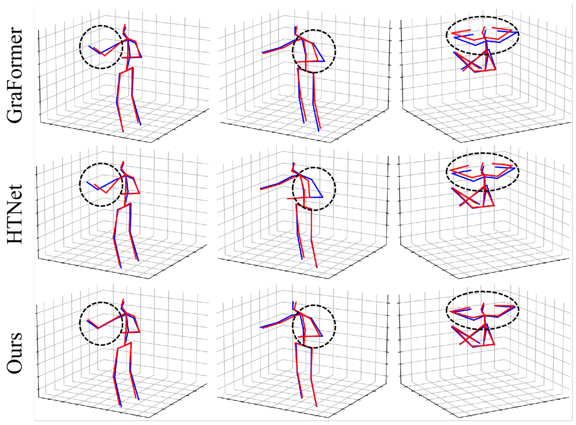


Figure 5: Qualitative results of GraFormer (Zhao, Wang, and Tian 2022), HTNet (Cai et al. 2023) and our proposed method on the Human3.6M test set. The blue lines represent the 3D ground truth poses, and the red lines show the predicted 3D poses.

Comparison on MPI-INF-3DHP. In Table 2, we assess the generalization capability of our method. We train our model on the Human3.6M train set and validate its performance on the MPI-INF-3DHP test set. From the table, we can find that our method achieves the best PCK results in 2 out of 3 scenarios (no Green Screen and Outdoor) and rank top in the other scenario (Green Screen) in the MPI-INF-3DHP test set, which demonstrates outstanding stability in diverse scenarios. Due to the imbalanced samples in each scenario, our method places second in terms of both PCK (All) and AUC metrics, demonstrating effectiveness in generalizing to unseen situations.

Computational efficiency. Our model has 2.4 million parameters with 0.0356 GFLOPs. Our method achieves the best performance with a model size comparable to other methods (Chen et al. 2024; Zhai et al. 2023; Cai et al. 2023; Xu and Takano 2021; Liu et al. 2020). Additionally, our method achieves a frame rate of 16747 FPS on an NVIDIA GeForce RTX 4090 GPU.

Ablation Studies

To evaluate the contribution of each component in our proposed method, we conduct intensive ablation studies on the Human3.6M dataset using the detected 2D poses serve as inputs, following previous works (Kang et al. 2023; Gong et al. 2023; Cai et al. 2023).

Pose SSM module. In Table 3, we conduct ablation experiments to analyze the impact of the pose SSM module. Taking the fourth row as an anchor point, in the first row, we remove the pose SSM module from each block. In the second and third rows, we replace the pose SSM module with a GCN module and a Transformer module respectively. Comparing the last row to the top three, the pose SSM module significantly enhances the model's performance with comparable amounts of parameters and GFLOPs. Specifically,

Methods	PCK ↑				AUC ↑
	GS	no GS	OD	All	
(Martinez et al. 2017)	49.8	42.5	31.2	42.5	17.0
(Ci et al. 2019)	74.8	70.8	77.3	74.0	36.7
(Li and Lee 2019)	70.1	68.2	66.6	66.9	-
(Zeng et al. 2020)	-	-	80.3	77.6	43.8
(Liu et al. 2020)	77.6	80.5	80.1	79.3	47.6
(Zou and Tang 2021)	86.4	86.0	85.7	86.1	53.7
(Zhao, Wang, and Tian 2022)	80.1	77.9	74.1	79.0	43.8
(Cai et al. 2023)	86.9	86.2	85.9	86.7	54.1
(Zhai et al. 2023)	**89.1**	85.9	85.9	**87.2**	**57.0**
(Chen et al. 2024)	-	-	-	85.5	53.6
Ours	86.7	**86.8**	**86.6**	86.7	54.5

Table 2: Quantitative comparisons on the MPI-INF-3DHP test set. GS and OD denotes Green Screen and Outdoor respectively.

	#Param	GFLOPs ↓	P1	P2
G-T	1.9 M	0.0330	48.7	38.5
G-G-T	3.6 M	0.0606	48.9	38.6
G-T-T	2.3 M	0.0382	48.8	38.5
G-S-T (Ours)	2.4 M	0.0356	**48.1**	**37.9**

Table 3: Ablation experiments on the pose SSM module (S). G and T denote GCN and Transformer module respectively. All modules are sequentially arranged.

when compared to the first, second, and third rows, improvements of 0.6mm, 0.8mm, 0.7mm in Protocol #1 and 0.6mm, 0.7mm, 0.6mm in Protocol #2 are observed, respectively. We attribute this to the exploitation of sequential dependency among joints and learning of features with selective state space model.

BPS vs. JHS. In Table 4, we report the results with different skeleton serialization strategies, BPS and JHS. Compared with the results without pose SSM module in Table 3, we can find out that both serialization strategies are beneficial to the pose estimation. Although both serializations include the information of kinematic chain, joint hierarchy, and body part, further optimizing their representation in the serialization lead to better performance. This is evidenced by the improvement of JHS over BPS on both Protocol #1 and Protocol #2 metrics, where the improvements are 0.4mm and 0.2mm, respectively.

Joint-dependent SSM layer. In Table 5, we validate the two task-related designs in our joint-dependent SSM layer, namely the joint-dependent parameters and the pose-aware hidden state initialization $\hat{\boldsymbol{h}}$. We conduct experiments on three different settings for the SSM parameters. The first row shows the results of the setting that the SSM parameters are shared across joints and poses as implemented in S4 (Gu, Goel, and Ré 2022). The second row shows the results of the setting that the SSM parameters are variants of both joints and poses as implemented in Mamba (Gu and Dao 2023). The third row shows the results of our setting that the SSM parameters are variants of joints but shared across poses. We can observe from the results that setting SSM parameters related to the joints and poses improve the performance in both evaluated metrics. Considering the specific characteristics of pose data, making the SSM parameters invariant to the poses further enhances the performance. In these three experiments, we use the zero initialization for the hidden state initialization as adopted in previous SSM methods. Comparing the last two rows in the table, we can find that our hidden state initialization from the input features in Equation 2 is effective, leading to 0.1mm improvement under both evaluated metrics.

serialization	Protocol #1	Protocol #2
BPS	48.5	38.1
JHS	**48.1**	**37.9**

Table 4: Comparison of two skeleton serialization strategies, BPS and JHS.

Dependency		$\hat{\boldsymbol{h}}$	Protocol #1	Protocol #2
joint	pose			
			48.6	38.5
✓	✓		48.3	38.3
✓			48.2	38.0
✓		✓	**48.1**	**37.9**

Table 5: Ablation experiments on our tailored joint-dependent SSM layer on the parameter setting and hidden state initialization.

Conclusion

In this paper, we propose a sequential joint dependency aware pose estimator for monocular 2D-to-3D human pose estimation. We propose a pose SSM module to exploit the sequential dependency between joints, while using GCN and Transformer based modules to learn features in the local and global manners. In our pose SSM module, we transform the graph-like pose data into sequential data with a sequential dependency aware representation. Such representation considers the preservation of kinematic chain, joint hierarchy, and semantic body part information, which are critical to the pose estimation task. Furthermore, we customize the SSM layer in the pose SSM module with joint-dependent parameters and pose-aware hidden state initialization. The proposed method achieves state-of-the-art results on two widely used datasets. The ablation study shows the effectiveness of our tailored SSM and the components in our designed module.

As the performance on the two evaluated datasets are saturated, we plan to extend our experiment on a more challenging dataset to evaluate our method. Furthermore, our method has limitations in crowd and occluded challenging scenarios, which is a common challenge for other HPE methods. For example, we found that actions like sitting down and photoing are still challenging. We plan to investigate mechanisms to address these in the future.

Acknowledgements

Han was partially supported by Shenzhen International Research Cooperation Project (Grant No. GJHZ20220913142611021). This work made use of the facilities of the N8 Centre of Excellence in Computationally Intensive Research (N8 CIR) provided and funded by the N8 research partnership and EPSRC (Grant No. EP/T022167/1). The Centre is co-ordinated by the Universities of Durham, Manchester and York. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Cai, J.; Liu, H.; Ding, R.; Li, W.; Wu, J.; and Ban, M. 2023. HTNet: Human Topology aware network for 3d Human pose estimation. In *International Conference on Acoustics, Speech and Signal Processing*.

Cai, Q.; Hu, X.; Hou, S.; Yao, L.; and Huang, Y. 2024. Disentangled Diffusion-Based 3D Human Pose Estimation with Hierarchical Spatial and Temporal Denoiser. In *AAAI Conference on Artificial Intelligence*, 882–890.

Cai, Y.; Ge, L.; Liu, J.; Cai, J.; Cham, T.; Yuan, J.; and Magnenat-Thalmann, N. 2019. Exploiting Spatial-Temporal Relationships for 3D Pose Estimation via Graph Convolutional Networks. In *International Conference on Computer Vision*, 2272–2281.

Chen, T.; Fang, C.; Shen, X.; Zhu, Y.; Chen, Z.; and Luo, J. 2022. Anatomy-Aware 3D Human Pose Estimation With Bone-Based Pose Decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1): 198–209.

Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; and Sun, J. 2018. Cascaded Pyramid Network for Multi-Person Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 7103–7112.

Chen, Z.; Dai, J.; Bai, J.; and Pan, J. 2024. DGFormer: Dynamic graph transformer for 3D human pose estimation. *Pattern Recognit.*, 152: 110446.

Ci, H.; Wang, C.; Ma, X.; and Wang, Y. 2019. Optimizing Network Structure for 3D Human Pose Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2262–2271.

Dabral, R.; Mundhada, A.; Kusupati, U.; Afaque, S.; Sharma, A.; and Jain, A. 2018. Learning 3D Human Pose from Structure and Motion. In *European Conference on Computer Vision*, 679–696.

Dorf, R.; and Bishop, R. 2008. *Modern Control Systems*. Pearson Prentice Hall.

Errity, A. 2016. Human–computer interaction. In *An Introduction to Cyberpsychology*, 240–256. Routledge.

Fu, D. Y.; Dao, T.; Saab, K. K.; Thomas, A. W.; Rudra, A.; and Ré, C. 2023. Hungry Hungry Hippos: Towards Language Modeling with State Space Models. In *International Conference on Learning Representations*.

Gong, J.; Foo, L. G.; Fan, Z.; Ke, Q.; Rahmani, H.; and Liu, J. 2023. DiffPose: Toward More Reliable 3D Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 13041–13051.

Gong, K.; Zhang, J.; and Feng, J. 2021. PoseAug: A Differentiable Pose Augmentation Framework for 3D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 8575–8584.

Gu, A.; and Dao, T. 2023. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *CoRR*.

Gu, A.; Goel, K.; and Ré, C. 2022. Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations*.

Han, D.; Wang, Z.; Xia, Z.; Han, Y.; Pu, Y.; Ge, C.; Song, J.; Song, S.; Zheng, B.; and Huang, G. 2024. Demystify Mamba in Vision: A Linear Attention Perspective. *arXiv preprint*.

Huang, L.; Li, Y.; Tian, H.; Yang, Y.; Li, X.; Deng, W.; and Ye, J. 2023. Semi-Supervised 2D Human Pose Estimation Driven by Position Inconsistency Pseudo Label Correction Module. In *Conference on Computer Vision and Pattern Recognition*.

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2014. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7): 1325–1339.

Islam, M. M.; and Bertasius, G. 2022. Long Movie Clip Classification with State-Space Video Models. In *European Conference on Computer Vision*.

Islam, M. M.; Hasan, M.; Athrey, K. S.; Braskich, T.; and Bertasius, G. 2023. Efficient Movie Scene Detection using State-Space Transformers. In *Conference on Computer Vision and Pattern Recognition*.

Kang, H.; Wang, Y.; Liu, M.; Wu, D.; Liu, P.; and Yang, W. 2023. Double-chain Constraints for 3D Human Pose Estimation in Images and Videos. *CoRR*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Li, C.; and Lee, G. H. 2019. Generating Multiple Hypotheses for 3D Human Pose Estimation With Mixture Density Network. In *Conference on Computer Vision and Pattern Recognition*, 9887–9895.

Li, H.; Shi, B.; Dai, W.; Zheng, H.; Wang, B.; Sun, Y.; Guo, M.; Li, C.; Zou, J.; and Xiong, H. 2023. Pose-Oriented Transformer with Uncertainty-Guided Refinement for 2D-to-3D Human Pose Estimation. In *AAAI Conference on Artificial Intelligence*, 1296–1304.

Li, Q.; Han, Z.; and Wu, X. 2018. Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning. In *AAAI Conference on Artificial Intelligence*, 3538–3545.

Li, S.; Ke, L.; Pratama, K.; Tai, Y.; Tang, C.; and Cheng, K. 2020. Cascaded Deep Monocular 3D Human Pose Estimation With Evolutionary Training Data. In *Conference on Computer Vision and Pattern Recognition*, 6172–6182.

Liu, K.; Ding, R.; Zou, Z.; Wang, L.; and Tang, W. 2020. A Comprehensive Study of Weight Sharing in Graph Networks for 3D Human Pose Estimation. In *European Conference on Computer Vision*, 318–334.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024. VMamba: Visual State Space Model. *CoRR*.

Martinez, J.; Hossain, R.; Romero, J.; and Little, J. J. 2017. A Simple Yet Effective Baseline for 3d Human Pose Estimation. In *International Conference on Computer Vision*, 2659–2668.

Mehta, D.; Rhodin, H.; Casas, D.; Fua, P.; Sotnychenko, O.; Xu, W.; and Theobalt, C. 2017a. Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision. In *International Conference on 3D Vision*, 506–516.

Mehta, D.; Sridhar, S.; Sotnychenko, O.; Rhodin, H.; Shafiei, M.; Seidel, H.; Xu, W.; Casas, D.; and Theobalt, C. 2017b. VNect: real-time 3D human pose estimation with a single RGB camera. *ACM Transactions on Graphics*, 36(4): 44:1–44:14.

Pavlakos, G.; Zhou, X.; Derpanis, K. G.; and Daniilidis, K. 2017. Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. In *Conference on Computer Vision and Pattern Recognition*.

Pavllo, D.; Feichtenhofer, C.; Grangier, D.; and Auli, M. 2019. 3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training. In *Conference on Computer Vision and Pattern Recognition*, 7753–7762.

Pham, H.; Salmane, H.; Khoudour, L.; Crouzil, A.; Velastin, S. A.; and Zegers, P. 2020. A Unified Deep Framework for Joint 3D Pose Estimation and Action Recognition from a Single RGB Camera. *Sensors*, 20(7): 1825.

Smith, J. T. H.; Warrington, A.; and Linderman, S. W. 2023. Simplified State Space Layers for Sequence Modeling. In *International Conference on Learning Representations*.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*.

Sun, X.; Xiao, B.; Wei, F.; Liang, S.; and Wei, Y. 2018. Integral Human Pose Regression. In *European Conference on Computer Vision*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wandt, B.; and Rosenhahn, B. 2019. RepNet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 7782–7791.

Wang, G.; Zeng, H.; Wang, Z.; Liu, Z.; and Wang, H. 2021. Motion Projection Consistency Based 3D Human Pose Estimation with Virtual Bones from Monocular Videos. *CoRR*.

Wang, J.; Zhu, W.; Wang, P.; Yu, X.; Liu, L.; Omar, M.; and Hamid, R. 2023. Selective Structured State-Spaces for Long-Form Video Understanding. In *Conference on Computer Vision and Pattern Recognition*.

Wang, Y.; Li, M.; Cai, H.; Chen, W.; and Han, S. 2022. Lite Pose: Efficient Architecture Design for 2D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*.

Xu, J.; Yu, Z.; Ni, B.; Yang, J.; Yang, X.; and Zhang, W. 2020. Deep Kinematics Analysis for Monocular 3D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 896–905.

Xu, T.; and Takano, W. 2021. Graph Stacked Hourglass Networks for 3D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 16105–16114.

Yu, W.; and Wang, X. 2024. MambaOut: Do We Really Need Mamba for Vision? *CoRR*.

Zeng, A.; Sun, X.; Huang, F.; Liu, M.; Xu, Q.; and Lin, S. 2020. SRNet: Improving Generalization in 3D Human Pose Estimation with a Split-and-Recombine Approach. In *European Conference on Computer Vision*, 507–523.

Zeng, A.; Sun, X.; Yang, L.; Zhao, N.; Liu, M.; and Xu, Q. 2021. Learning Skeletal Graph Neural Networks for Hard 3D Pose Estimation. In *International Conference on Computer Vision*, 11416–11425.

Zhai, K.; Nie, Q.; Ouyang, B.; Li, X.; and Yang, S. 2023. HopFIR: Hop-wise GraphFormer with Intragroup Joint Refinement for 3D Human Pose Estimation. In *International Conference on Computer Vision*.

Zhang, J.; Tu, Z.; Yang, J.; Chen, Y.; and Yuan, J. 2022. MixSTE: Seq2seq Mixed Spatio-Temporal Encoder for 3D Human Pose Estimation in Video. In *Conference on Computer Vision and Pattern Recognition*, 13222–13232.

Zhao, L.; Peng, X.; Tian, Y.; Kapadia, M.; and Metaxas, D. N. 2019. Semantic Graph Convolutional Networks for 3D Human Pose Regression. In *Conference on Computer Vision and Pattern Recognition*, 3425–3435.

Zhao, W.; Wang, W.; and Tian, Y. 2022. GraFormer: Graph-oriented Transformer for 3D Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 20406–20415.

Zheng, C.; Zhu, S.; Mendieta, M.; Yang, T.; Chen, C.; and Ding, Z. 2021. 3D Human Pose Estimation with Spatial and Temporal Transformers. In *International Conference on Computer Vision*, 11636–11645.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. *CoRR*.

Zhu, W.; Ma, X.; Liu, Z.; Liu, L.; Wu, W.; and Wang, Y. 2023. MotionBERT: A Unified Perspective on Learning Human Motion Representations. In *International Conference on Computer Vision*, 15039–15053.

Zou, Z.; and Tang, W. 2021. Modulated Graph Convolutional Network for 3D Human Pose Estimation. In *International Conference on Computer Vision*, 11457–11467.