Video-Based Human Pose Regression via Decoupled Space-Time Aggregation

Jijie He, Wenwu Yang*
Zhejiang Gongshang University, China

Abstract

*By leveraging temporal dependency in video sequences, multi-frame human pose estimation algorithms have demonstrated remarkable results in complicated situations, such as occlusion, motion blur, and video defocus. These algorithms are predominantly based on heatmaps, resulting in high computation and storage requirements per frame, which limits their flexibility and real-time application in video scenarios, particularly on edge devices. In this paper, we develop an efficient and effective video-based human pose regression method, which bypasses intermediate representations such as heatmaps and instead directly maps the input to the output joint coordinates. Despite the inherent spatial correlation among adjacent joints of the human pose, the temporal trajectory of each individual joint exhibits relative independence. In light of this, we propose a novel Decoupled Space-Time Aggregation network (DSTA) to separately capture the spatial contexts between adjacent joints and the temporal cues of each individual joint, thereby avoiding the conflation of spatiotemporal dimensions. Concretely, DSTA learns a dedicated feature token for each joint to facilitate the modeling of their spatiotemporal dependencies. With the proposed joint-wise local-awareness attention mechanism, our method is capable of efficiently and flexibly utilizing the spatial dependency of adjacent joints and the temporal dependency of each joint itself. Extensive experiments demonstrate the superiority of our method. Compared to previous regression-based single-frame human pose estimation methods, DSTA significantly enhances performance, achieving an **8.9** mAP improvement on PoseTrack2017. Furthermore, our approach either surpasses or is on par with the state-of-the-art heatmap-based multi-frame human pose estimation methods. Project page: https://github.com/zgspose/DSTA.*

1. Introduction

Human pose estimation, which aims at identifying anatomical keypoints (*e.g.*, elbow, knee, etc.) of human bodies from images or videos, has been extensively studied in the computer vision community [8, 31, 37, 41]. It plays a crucial role in a variety of human-centric tasks, including motion capture, activity analysis, surveillance, and human-robot interaction [43]. Recently, significant progress has been made in the field of human pose estimation, particularly with the advent of deep convolutional neural networks (CNNs) [11, 25, 26] and Transformer networks [30, 36]. While the majority of recent methods focus on estimating human poses in *static images*, it has been demonstrated in [2, 7, 20] that the significance of *dynamic cues* (*i.e.*, temporal dependency and geometric consistency) across video frames cannot be overlooked. To address inherent challenges in human motion images, such as motion blur, video defocus, and pose occlusions, it is essential to sufficiently exploit the temporal cues in video sequences.

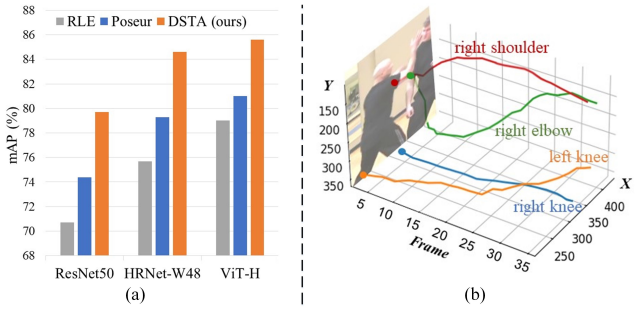

Figure 1. (a) Compared to our proposed video-based regression method, previous image-based regression methods of RLE [18] and Poseur [23] have a substantial performance decline when processing video input, *e.g.*, the dataset of PoseTrack2017 [14]. (b) Despite the intrinsic spatial correlations among human body joints, each joint exhibits independent motion trajectories temporally.

Existing methods of human pose estimation can be divided into two categories: heatmap-based [8, 20, 26, 28, 31, 34, 37, 41], and regression-based [18, 19, 23, 29, 40]. Heatmap-based methods generate a likelihood heatmap for each joint, whereas regression-based methods directly map the input to the output joint coordinates. Owing to their superior performance, heatmap-based methods dominate in the field of human pose estimation, particularly among video-based approaches [2, 7, 15, 20]. The high computation and storage requirements of heatmap-based methods, however, make them expensive in 3D contexts (tem-

*Correspondence Author (wwyang@zjgsu.edu.cn)

poral), which restricts their versatility and real-time deployment in video applications, especially on edge devices. On the other hand, regression-based methods are more flexible and efficient. According to [18], while a standard heatmap head (3 deconv layers) costs $1.4\times$ FLOPs of the ResNet-50 backbone, the regression head only costs 1/20000 FLOPs of the same backbone. Moreover, recent regression-based approaches [18, 23] have demonstrated outstanding performance that is on par with heatmap-based methods. Unfortunately, these regression-based approaches are all built for static images and neglect the temporal dependency between video frames, leading to a marked decline in performance when handling video input, as shown in Fig. 1(a).

In this work, we explore the video-based human pose regression to facilitate multi-person pose estimation in video sequences. Regression-based approaches primarily focus on regressing the coordinates of pose joints, often overlooking the rich structural information inherent in the pose [27]. As demonstrated in [23], the self-attention module employed in the Transformer architecture [30] can be used across the pose joints to naturally capture their spatial dependency. A simple and direct extension is to use the self-attention module across all the joints from consecutive video frames to capture both the structural information of the pose and its temporal dependency in video sequences. However, as illustrated in Fig. 1(b), while there's an inherent spatial correlation between adjacent joints of the human pose, the temporal trajectory of each joint tends to be rather independent. This implies that the spatial structure of the pose and its temporal dynamics across video frames cannot be conflated and must be captured separately.

To this end, we propose a novel and effective video-based human pose regression method, named Decoupled Space-Time Aggregation (DSTA), that models the spatial structure between adjacent joints and the temporal dynamic of each individual joint separately, thereby avoiding the conflation of spatiotemporal information. Rather than using the output feature maps of a CNN backbone to regress the joints' coordinates as in existing regression models [18, 23], DSTA converts the backbone's output into a sequence of tokens, with each token uniquely representing a joint. Intuitively, each token embodies the feature embedding of its corresponding joint; therefore, it is natural to use them to model the spatiotemporal dependencies of pose joints. Specifically, DSTA first establishes the feature token for each joint via Joint-centric Feature Decoder (JFD) module, which are hence used to capture the spatiotemporal relations of pose joints in the Space-Time Decoupling (STD) module. To efficiently and flexibly model the spatial dependency between adjacent joints and the temporal dependency of each joint itself, we introduce a joint-wise local-awareness attention mechanism to ensure each joint only attends to those joints that are structurally or temporally relevant. The aggregated spatial and temporal information is utilized to determine the coordinates of the joints. During training, the JFD and STD modules are optimized simultaneously, with the entire model undergoing end-to-end training.

To the best of our knowledge, this is an original effort on regression-based framework for multi-person pose estimation in video sequences. We evaluate our method through the widely-utilized video-based benchmarks for human pose estimation: PoseTrack datasets [1, 6, 14]. With a simple yet effective architecture, DSTA achieves a notable improvement of **8.9** mAP over previous regression-based methods tailored for static images and obtains superior performance to the heatmap-based methods for video sequences. Moreover, it offers greater efficiency of computation and storage than heatmap-based multi-frame human pose estimation methods, making it more suitable for real-time video applications and easier to deploy, particularly on edge devices. For instance, utilizing the HRNet-W48 backbone, our regression-based DSTA achieves **83.4** mAP on the PoseTrack2017 [14] dataset with a head computation of merely **0.02** GFLOPs, while heatmap-based DCPose [20] attains 82.8 mAP on the same dataset with a significantly higher head computation of 11.0 GFLOPs.

Our main contributions can be summarized as follows:

- We propose **DSTA**, a novel and effective video-based human pose regression framework. The proposed method efficiently and flexibly models the spatiotemporal dependencies of pose joints in the video sequences.
- Our method is the first regression-based method for multi-frame human pose estimation. Compared to heatmap-based methods, our method is efficient and flexible, opening up new possibilities for real-time video applications.
- We demonstrate the effectiveness of our approach with extensive experiments. Our method not only delivers a marked improvement over prior regression-based methods designed for static images, but also achieves performance superior to the heatmap-based methods.

2. Related Work

Heatmap-based Human Pose Estimation. Since the introduction of likelihood heatmaps to represent human joint positions [28], heatmap-based methods have become predominant in the field of 2D human pose estimation [13, 26, 37, 38, 41], owing to their superior performance. To perform multi-person human pose estiamtion, the top-down approaches initially identify person bounding boxes and subsequently conduct single-person pose estimation within the cropped regions [12, 26, 37]. Conversely, bottom-up methods commence by detecting identity-free keypoints for all individuals and then cluster these keypoints into distinct persons [3, 4, 17, 22]. Recently, the heatmaps or CNN features from adjacent frames have been utilized to extract the temporal dependencies of human poses, thereby enhanc-

ing the performance of multi-person human pose estimation in video sequences [2, 20, 21]. Despite its effectiveness, the heatmap representation inherently suffers from several drawbacks, such as quantization errors and the high computational and storage demands associated with maintaining high-resolution heatmaps.

Regression-based Human Pose Estimation. Regression-based methods forgo the use of intermediate heatmaps, opting instead to map the input directly to the output joint coordinates [29]. This approach is flexible and efficient for a wide range of human pose estimation tasks and real-time applications, especially on edge devices. Despite their efficiency, regression-based methods have traditionally lagged behind heatmap-based methods in accuracy within the realm of human pose estimation, leading to less focus on their development [19, 24, 29, 33]. Recently, advancements such as RLE [18] and Poseur [23] have significantly propelled regression-based approaches, elevating their performance to a level comparable with heatmap-based methods. However, these regression-based methods are designed exclusively for static images. When these image-based methods are directly applied to video sequences, they tend to yield suboptimal predictions due to their inability to capture temporal dependencies between frames. As a result, such models struggle with challenges inherent to video inputs, such as motion blur, defocusing, and pose occlusions, which are common in dynamic scenes.

In this work, we present for the first time a regression-based approach for multi-person human pose estimation in video sequences, outperforming or is on par with state-of-the-art heatmap-based methods for video sequences.

3. Method

3.1. Overview

Given a video frame $\mathcal{I}(t)$ at time t containing multiple persons, we are interested in estimating locations of pose joints for each person. To enhance pose estimation for the frame $\mathcal{I}(t)$, we leverage the temporal dynamics from a consecutive frame sequence $\mathcal{S} = \langle \mathcal{I}(t-T), \ldots, \mathcal{I}(t), \ldots, \mathcal{I}(t+T) \rangle$, where T is a predefined temporal span. Our method follows the top-down paradigm. Initially, we use an human detector to identify individual persons in the frame $\mathcal{I}(t)$. Subsequently, each detected bounding box is expanded by 25% to extract the same individual across the frame sequence $\mathcal{S}$, resulting in a cropped video clip $\mathcal{S}_i = \langle \mathcal{I}_i(t-T), \ldots, \mathcal{I}_i(t), \ldots, \mathcal{I}_i(t+T) \rangle$ for every individual i. The goal of estimating human pose for individual i within the specified video frame $\mathcal{I}(t)$ can then be denoted as

$$\{\mathbf{x}_i^j(t)\}_{j=1}^n = \text{HPE}(\mathcal{S}_i),$$

where $\text{HPE}(\cdot)$ denotes the human pose estimation module, $\mathbf{x}_i^j(t)$ is the j-th pose joint for the individual i in video frame $\mathcal{I}(t)$, and n represents the number of joints for each person, *e.g.*, $n = 15$ for PoseTrack datasets [1, 6, 14]. For simplicity in the following description of our algorithm, unless otherwise specified, we will refer to a specific individual i.

We adopt the regression-based method to implement the human pose estimation module $\text{HPE}(\cdot)$. Compared to the heatmap-based method, the regression-based method offers several advantages: i) It eliminates the need for high-resolution heatmaps, resulting in reduced computation and storage demands. This makes it more apt for real-time video applications and facilitates deployment, especially on edge devices. ii) It provides continuous outputs, avoiding the quantization issues inherent in heatmap methods. In the regression-based pose estimation module $\text{HPE}(\cdot)$, the global feature maps extracted by a CNN backbone are fed into a regression module, which then directly produces the coordinates of the joints, *i.e.*,

$$\{\mathbf{x}_i^j(t)\}_{j=1}^n = \text{REG}(\hat{\mathcal{S}}_i), \tag{1}$$

where $\text{REG}(\cdot)$ denotes the regression module and $\hat{\mathcal{S}}_i = \langle \mathcal{F}_i(t-T), \ldots, \mathcal{F}_i(t), \ldots, \mathcal{F}_i(t+T) \rangle$. Here, $\mathcal{F}_i(t')$ with $t' \in [t-T, t+T]$ is the global feature maps extracted by the CNN backbone from the cropped image $\mathcal{I}_i(t')$. Note, when regressing the pose joints at the current frame time t, we aim to utilize the temporal feature information across the video clip $\mathcal{S}_i$, rather than solely relying on the feature information from the current frame $\mathcal{I}_i(t)$.

Prior work, such as Poseur [23], has demonstrated that the spatial dependencies among pose joints can be naturally captured by applying the self-attention mechanism [30] over them. It follows that we can also employ the self-attention mechanism on the global pose features in the temporal sequence $\hat{\mathcal{S}}_i$ to discern the temporal dependency of individual's pose over the time interval $[t-T, t+T]$. As depicted in Fig. 1(b), each pose joint exhibits a relatively independent temporal trajectory. Hence, it's more appropriate to model the temporal dependency at the joint level rather than for the entire pose. To this end, extra efforts are required to convert each global feature $\mathcal{F}_i(t')$ into a set of joint-aware feature embeddings. This procedure is finished in the Joint-centric Feature Decoder (JFD) , which can be denoted as

$$\{\mathcal{F}_i^j(t')\}_{j=1}^n = \text{JFD}(\mathcal{F}_i(t')), \quad t' \in [t-T, t+T], \tag{2}$$

where $\mathcal{F}_i^j(t')$, termed a feature token, represents the feature embedding of the j-th joint of the pose at the video frame of time t'. Let's represent the feature tokens for each joint of the pose over the time span $[t-T, t+T]$ as $\tilde{\mathcal{S}}_i$. That is, $\tilde{\mathcal{S}}_i = \langle \{\mathcal{F}_i^j(t-T)\}_{j=1}^n, \ldots, \{\mathcal{F}_i^j(t)\}_{j=1}^n, \ldots, \{\mathcal{F}_i^j(t+T)\}_{j=1}^n \rangle$. We subsequently utilize the feature tokens in $\tilde{\mathcal{S}}_i$ to model

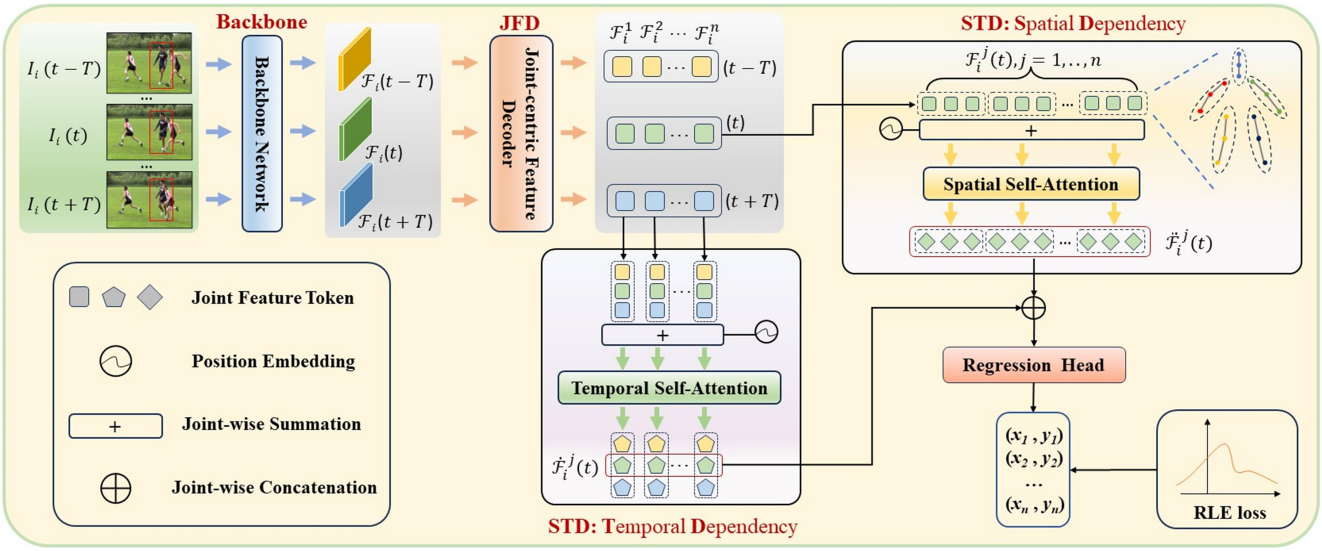

Figure 2. The pipeline of the proposed Decoupled Space-Time Aggregation (DSTA). The goal is to detect the human pose of the key frame $\mathcal{I}_i(t)$. Given a video sequence $\langle \mathcal{I}_i(t-T), \ldots, \mathcal{I}_i(t), \ldots, \mathcal{I}_i(t+T) \rangle$, DSTA uses a backbone network to extract their feature maps. From these maps, Joint-centric Feature Decoder (JFD) extracts feature tokens to individually represent each joint. Space-Time Decoupling (STD) then models the temporal dynamic dependencies and spatial structural dependencies of joints separately, producing aggregated space-time features for the current key frame. Each of these aggregated features is utilized to regress the coordinates of the corresponding joint.

the spatiotemporal dependencies of pose joints via Space-Time Decoupling (STD),

$$\{\mathbf{f}_i^j(t)\}_{j=1}^n = \text{STD}(\tilde{\mathcal{S}}_i), \tag{3}$$

where $\mathbf{f}_i^j(t)$ is the aggregated space-time feature for the j-th joint of the pose at the current video frame t, which is then fed to a joint-wise fully connected feed-forward network to produce the coordinates of the joint.

Our proposed Decoupled Space-Time Aggregation Network (DSTA) is composed of three primary modules: the backbone, JFD$(\cdot)$, and STD$(\cdot)$. We learn DSTA by training the backbone, JFD, STD modules in an end-to-end manner. The architecture and workflow of DSTA are depicted in Fig. 2. Subsequent sections delve into the specifics of the JFD, STD, and the computation of the loss function.

3.2. Joint-centric Feature Decoder

As shown in Fig. 2, the purpose of JFD is to extract the feature embedding for each joint from the given global feature maps $\mathcal{F}_i(t')$ with $t' \in [t-T, t+T]$. As suggested by [23], one potential approach to construct the joint embedding is as follows: initially, a traditional regression method such as [18] is utilized to regress the joint coordinates from $\mathcal{F}_i(t')$. For each joint, its x-y coordinates are converted into position embedding using sine-cosine position encoding [30]. Concurrently, a learnable class embedding is designated for every joint type. The final feature embedding for each joint is derived by summing its position embedding with the respective class embedding. However, this approach loses crucial contextual information of the joints within the pose that is learned in the global feature maps $\mathcal{F}_i(t')$. Though we can augment each joint with relevant contextual feature from the global feature maps, such as the approach in [23] which uses the joint's embedding as a query and applies a multi-scale deformable attention module to sample features for each joint from the feature maps, this method incurs significant computational costs.

We employ a straightforward yet efficient approach to construct the joint embeddings from the provided global feature maps $\mathcal{F}_i(t')$. Given the global feature maps produced by the backbone, previous heatmap-based methods convolve these maps via convolution layers to generate a heatmap feature for each joint [2, 20]. We follow this strategy, deriving the feature embedding for each joint from $\mathcal{F}_i(t')$ through a convolution layer or a fully connected layer (FC). In our setup, the ResNet backbones (like ResNet50 or ResNet152) are followed by a global average pooling layer and a FC layer. The FC layer comprises $2048 \times K$ neurons. Here, 2048 represents the dimensionality of $\mathcal{F}_i(t')$ after undergoing global average pooling and flattening. Meanwhile, K is calculated as $n \times 32$, where n indicates the number of pose joints, and 32 signifies the dimension of the joint embedding. The output of the FC layer is the feature embedding for each joint, where the output is evenly divided into n parts, denoted as $\{\mathcal{F}_i^j(t')\}_{j=1}^n$, with each part representing a feature embedding for a joint. Further implementation details regarding more backbones (*e.g.*, HRNet backbone) can be found in the supplementary material.

3.3. Space-Time Decoupling

STD is designated to model the spatial and temporal dependencies between joints based on their embeddings over the time span $[t-T, t+T]$, *i.e.*, all feature tokens in $\tilde{\mathcal{S}}_i = \langle \{\mathcal{F}_i^j(t-T)\}_{j=1}^n, \ldots, \{\mathcal{F}_i^j(t)\}_{j=1}^n, \ldots, \{\mathcal{F}_i^j(t+T)\}_{j=1}^n \rangle$. In numerous applications [5, 23], the self-attention mechanism's proficiency in capturing long-distance dependencies within sequences has been thoroughly demonstrated [30]. Thus, a direct approach to capturing the spatio-temporal dependencies between joints is to apply the self-attention module to the sequence of feature tokens in $\tilde{\mathcal{S}}_i$,

$$\{\hat{\mathcal{F}}_i^j(t)\}_{j=1}^n = \text{S-ATT}(\tilde{\mathcal{S}}_i), \tag{4}$$

where S-ATT$(\cdot)$ denotes the self-attention module [30], and $\hat{\mathcal{F}}_i^j(t)$, which encodes the spatial and temporal information learned by the S-ATT module, represents the updated feature token for the j-th joint at the current video frame t. In our implementation, the S-ATT module adheres to the conventional Transformer architecture [30]. In our setup, 4 identical layers are stacked sequentially. Each layer comprises two sub-layers: the first one employs a multi-head self-attention mechanism, and the second one utilizes a simple, token-wise, fully connected feed-forward network. The input feature tokens pass through these modules in sequence, each producing an updated version that serves as the input for the subsequent layer. Additionally, each of the initial input feature tokens is equipped with a learnable position embedding, and their sum forms the final input.

3.3.1 Decoupled Space-Time Aggregation

However, as illustrated in Fig. 1(b), despite the inherent spatial correlation among adjacent joints of the human pose, the temporal trajectory of each individual joint tends to be rather independent. So, as shown in Fig. 2, our proposed DSTA models the temporal dynamic dependencies and spatial structure dependencies separately, instead of modeling spatial and temporal dependencies together as in Eq. 4. This approach allows for a more nuanced capture of the unique dependency characteristics that joints exhibit separately in both the temporal and spatial dimensions. Then, by fusing the captured spatial and temporal information, an aggregated spatio-temporal feature for each joint of current frame t, *i.e.*, $\mathbf{f}_i^j(t)$, is derived:

$$\{\mathbf{f}_i^j(t)\}_{j=1}^n = \text{SD}(\tilde{\mathcal{S}}_i) \bigoplus \text{TD}(\tilde{\mathcal{S}}_i), \tag{5}$$

where $\bigoplus$ denotes the concatenation operation, which is individually applied to each pair of corresponding updated feature tokens associated with each joint. By utilizing the local-awareness attention introduced below (Sec. 3.3.2), the SD$(\cdot)$ module learns the spatial dependencies between adjacent joints and correspondingly generates an updated feature token for each joint in the current frame. Concurrently, the TD$(\cdot)$ module discerns the temporal dependencies of each joint, resulting in another updated feature token for each joint in the current frame. Subsequently, the aggregated features of joints $\{\mathbf{f}_i^j(t)\}_{j=1}^n$ are fed into a joint-wise fully connected feed-forward network, producing the coordinates of the joints $\{\mathbf{x}_i^j(t)\}_{j=1}^n$:

$$\{\mathbf{f}_i^j(t)\}_{j=1}^n \xrightarrow[\text{feed-forward network}]{\text{joint-wise fully connected}} \{\mathbf{x}_i^j(t)\}_{j=1}^n. \tag{6}$$

3.3.2 Local-awareness Attention

From a temporal perspective, each joint is intimately connected only with its corresponding joints in preceding and succeeding frames, having no relevance with other joints. From a spatial perspective, the structure dependencies of joints are primarily manifested between adjacent joints within a single frame. Therefore, we introduce a joint-wise local-awareness attention mechanism, ensuring that each joint only attends to those that are structurally or temporally relevant. This local-awareness attention mechanism is elaborated upon, demonstrating its application in implementing the aforementioned SD$(\cdot)$ and TD$(\cdot)$ modules.

In the TD module, we capture the temporal dynamic dependency for each joint j at the current frame t. To this end, our proposed local-awareness attention selectively applies the self-attention module S-ATT in Eq. 4 across the corresponding joints over the time span $[t-T, t+T]$,

$$\dot{\mathcal{F}}_i^j(t) = \text{S-ATT}(\tilde{\mathcal{S}}_i^j), \quad j = 1, 2, \ldots, n, \tag{7}$$

where $\tilde{\mathcal{S}}_i^j = \langle \mathcal{F}_i^j(t-T), \ldots, \mathcal{F}_i^j(t), \ldots, \mathcal{F}_i^j(t+T) \rangle$, and $\dot{\mathcal{F}}_i^j(t)$ denotes the updated feature token for the j-th joint at the current video frame t, encoding the temporal dependency information of this joint embedded within the sequence $\tilde{\mathcal{S}}_i^j$. Since the sequence $\tilde{\mathcal{S}}_i^j$ only includes the feature tokens of joint j over the time span $[t-T, t+T]$, the temporal dependency encoded in $\dot{\mathcal{F}}_i^j(t)$ is solely related to the joint itself, without any relevance to other joints.

In the SD module, we capture the spatial structure dependency among joints within the current frame t. A straightforward way is to directly apply the self-attention module S-ATT from Eq. 4 to all joints in the current frame. To allow each joint to focus more closely on the adjacent joints that are intimately associated with it in structure, we divide the joints into K groups according to the semantic structure of the human pose, as shown in the top right of Fig. 2. Our proposed local-awareness attention conducts the self-attention module S-ATT separately for each group,

$$\{\ddot{\mathcal{F}}_i^j(t)\}_{j \in G(k)} = \text{S-ATT}(\langle \mathcal{F}(t)_i^j \rangle_{j \in G(k)}), \quad k = 1, \ldots, K, \tag{8}$$

where $G(k)$ represents the set of joint indices in group k, and $\ddot{\mathcal{F}}_i^j(t)$ denotes the updated feature token for the j-th

joint at the current video frame t, encapsulating the spatial structure dependencies of this joint within the pose.

Through the modules TD and SD , we have captured the spatial and temporal contexts for each joint in the current frame, obtaining the corresponding updated feature tokens, $\dot{\mathcal{F}}_i^j(t)$ and $\ddot{\mathcal{F}}_i^j(t)$. Consequently, the spatio-temporal aggregated feature $\mathbf{f}_i^j(t)$ for each joint j at the current frame t, as per Eq. 5, can be explicitly computed as follows:

$$\mathbf{f}_i^j(t) = \dot{\mathcal{F}}_i^j(t) \oplus \ddot{\mathcal{F}}_i^j(t), \quad j = 1, 2, \ldots, n, \tag{9}$$

where $\oplus$ denotes the concatenation operation.

Discussion: Compared with the *global* attention method as defined in Eq. 4, our proposed *local-awareness* attention ensures that each joint only attends to those that are structurally or temporally relevant. This approach not only avoids the undesired conflation of spatiotemporal dimensions but also reduces computational overhead. For example, the computational cost of the S-ATT module is mainly determined by the multi-head self-attention mechanism [30], where the computational complexity is proportionate to the square of the quantity of feature tokens. Consequently, the computational complexity of the global attention method delineated in Eq. 4 is approximately $O((n\times(2T+1))^2) = O(4n^2T^2+4n^2T+n^2)$. In contrast, the total computational complexity of our local-awareness attention methods, corresponding to Eqs. 7 and 8, is approximately $O(n\times(2T+1)^2+K\times(\frac{n}{K})^2) = O(4nT^2+4nT+n+\frac{n^2}{K})$. In the experiment, the value of T is quite small, for instance $T = 1$, thus our local-awareness attention method reduces the time complexity from $O(9n^2)$ to $O(\frac{n^2}{K}+9n)$ with $K = 5$ in our implementation, thereby achieving a speedup close to 45 times.

3.4. Loss Computation

During training, the entire model undergoes end-to-end optimization, aiming to minimize the discrepancy between the coordinates of the predicted joints and the ground truth joints in the current frame t. To boost the regression performance, we employ the residual log-likelihood estimation loss (RLE) as proposed in [18], in lieu of the conventional regression loss (l_1 or l_2). We extend the RLE loss, originally designed for image-based pose regression, to the context of video-based pose regression. Given an input cropped video clip, $\mathcal{S}_i$, for individual i, we calculate a distribution, $P_{\theta,\phi}(\{\mathbf{x}_i^j(t)\}_{j=1}^n|\mathcal{S}_i)$, which reflects the likelihood that the ground truth at the current frame t appears at the predicted locations $\{\mathbf{x}_i^j(t)\}_{j=1}^n$. Here, θ represents the parameters of our model, and ϕ represents the parameters of a flow model. The flow model is not required to operate during inference, thereby introducing no additional overhead at test time. The learning process involves the simultaneous optimizations of the model parameters θ and ϕ, aiming to maximize the probability of observing the ground truth μ_g. This is achieved by defining the RLE loss as follows:

$$\mathcal{L}_{rle} = -P_{\theta,\phi}(\{\mathbf{x}_i^j(t)\}_{j=1}^n|\mathcal{S}_i)\Big|_{\{\mathbf{x}_i^j(t)\}_{j=1}^n=\mu_g}. \tag{10}$$

For a more detailed discussion and further information about the RLE loss, we refer readers to [18].

Method	ResNet-50	HRNet-W48	ViT-H
image-based			
RLE [18]	70.7	75.7	79.0
Poseur [23]	74.4	79.3	81.0
video-based			
DSTA (Ours)	**79.7**	**84.6**	**85.6**

Table 1. **Comparison with image-based regression** (mAP) on PoseTrack2017 val. set.

4. Experiments

4.1. Experimental Settings

We have conducted evaluations on three widely-utilized video-based benchmarks for human pose estimation: PoseTrack2017 [14], PoseTrack2018 [1], and PoseTrack21 [6]. These datasets contain video sequences of complex scenarios involving rapid movements of highly occluded individuals in crowded environments. To assess the performance of our models, we utilize the Average Precision (AP) metric [2, 18, 20, 26]. The AP is calculated for each joint, and the mean AP across all joints is denoted as mAP. Our method is implemented using PyTorch. Unless otherwise specified, the input image size is 384×288 when using the HRNet-w48 backbone, while for other backbones, the input image size is 256×192. We pretrained the backbones on the COCO dataset. For further implementation details, please refer to the supplementary materials provided.

4.2. Main Results

4.2.1 Comparison with Image-based Regression

To study the effectiveness of the proposed regression method on video input, we compare it with existing state-of-the-art image-based regression methods, namely RLE [18] and Poseur [23]. For thorough and fair comparisons, we utilized three distinct backbone networks—ResNet-50, HRNet-W48, and ViT-H—and ensured that each approach applied the identical pre-trained model to each backbone network. The experimental results on the PoseTrack2017 validation set are presented in Table 1. As shown, our proposed video-based method achieves significant performance improvements across all backbone networks when compared to image-based methods. For instance, our method outperforms RLE [18] by a notable margin of **8.9** mAP (or **9.0** mAP) when utilizing the HRNet-W48 (or ResNet-50) backbone. This demonstrates the importance of incorporating temporal cues from neighboring frames.

Figure 3. **Qualitative comparison** of a) our DSTA, b) DCPose [20], c) Poseur [23], and d) RLE [18] on the PoseTrack datasets, featuring challenges such as occlusions, nearby-person interactions, and motion blur. Inaccurate predictions are marked with red solid circles.

By leveraging temporal dependencies across consecutive frames, our video-based regression is better equipped to handle challenging situations such as occlusion or motion blur commonly encountered in video scenarios, as demonstrated in Fig. 3. These experiments demonstrate the superior performance of our proposed video-based regression framework, which significantly outstrips the capabilities of prior image-based methods when handling the video input.

4.2.2 Comparison with State-of-the-art Methods

Current state-of-the-art algorithms for video-based human pose estimation are predominantly based on heatmaps. We first conduct a performance comparison of our method with these heatmap-based approaches on the PoseTrack datasets. Subsequently, we compare the computational complexity between the heatmap-based methods and our proposed regression-based approach. Furthermore, we examine the varying impacts of input resolution on both the heatmap-based methods and our regression-based approach.

Results on the PoseTrack Datasets. Table 2 presents the quantitative results of different approaches on PoseTrack2017 validation set. Our method achieves comparable performance to the state-of-the-art methods. For example, employing the HRNet-W48 backbone, our method attains an mAP of **84.6**, which surpasses the adopted backbone network HRNet-W48 [26] by **7.3** points. When compared to video-based approaches utilizing the same backbone, our method outperforms DCPose [20] by **1.8** points while maintaining a performance level on par with the state-of-the-art FAMI-Pose [21]. Our approach is flexible and can be easily integrated into various backbone networks. When using the ViT-H backbone, our method further pushes the performance boundary and achieves an mAP of **85.6**. The performance enhancement for the relatively challenging joints is truly encouraging: an mAP of **82.6** (↑ **2.6**) for wrists and an mAP of **77.8** (↑ **0.8**) for ankles. It is worth noting that methods utilizing temporal information, such as PoseWarper [2], DCPose [20], DetTrack [32], FAMI-Pose [21], and our DSTA, consistently outperform those relying solely on a single key frame, such as HRNet [26]. This reaffirms the importance of incorporating temporal cues from adjacent frames. Qualitative results are shown in Fig. 3.

Method	Bkbone	Head	Should.	Elbow	Wrist	Hip	Knee	Ankle	**Mean**
heatmap-based									
PoseTrack [9]	ResNet-101	67.5	70.2	62.0	51.7	60.7	58.7	49.8	60.6
PoseFlow [35]	ResNet-152	66.7	73.3	68.3	61.1	67.5	67.0	61.3	66.5
FastPose [42]	ResNet-101	80.0	80.3	69.5	59.1	71.4	67.5	59.4	70.3
SimBase. [34]	ResNet-152	81.7	83.4	80.0	72.4	75.3	74.8	67.1	76.7
STEmbed. [16]	ResNet-152	83.8	81.6	77.1	70.0	77.4	74.5	70.8	77.0
HRNet [26]	HRNet-W48	82.1	83.6	80.4	73.3	75.5	75.3	68.5	77.3
MDPN [10]	ResNet-152	85.2	88.5	83.9	77.5	79.0	77.0	71.4	80.7
Dyn.-GNN [39]	HRNet-W48	88.4	88.4	82.0	74.5	79.1	78.3	73.1	81.1
PoseWarp. [2]	HRNet-W48	81.4	88.3	83.9	78.0	82.4	80.5	73.6	81.2
DCPose [20]	HRNet-W48	88.0	88.7	84.1	78.4	83.0	81.4	74.2	82.8
DetTrack [32]	HRNet-W48	89.4	89.7	85.5	79.5	82.4	80.8	76.4	83.8
FAMIPose [21]	HRNet-W48	89.6	90.1	**86.3**	**80.0**	84.6	**83.4**	**77.0**	**84.8**
regression-based									
DSTA (Ours)	ResNet-152	88.3	88.1	83.3	76.0	82.5	81.1	70.0	81.8
DSTA (Ours)	HRNet-W48	**89.8**	**90.8**	**86.2**	79.3	**85.2**	82.2	75.9	**84.6**
DSTA (Ours)	ViT-H	89.3	90.6	87.3	82.6	84.5	85.1	77.8	85.6

Table 2. **Comparison with the SOTA** on PoseTrack2017 val. set. Similar to FAMI-Pose [21], our proposed DSTA sets the temporal span T to 2, consisting of two preceding and two subsequent frames, totalling four auxiliary frames.

Method	#Params	GFLOPs of Backbone	GFLOPs of Net. Head	mAP
heatmap-based				
PoseWarper [2]	71.1M	35.5	156.7	81.0
DCPose [20]	65.2M	35.5	11.0	82.8
regression-based				
DSTA (Ours)	63.9M	35.5	**0.02**	**83.4**

Table 3. **Computation complexity** with HRNet-W48 backbone. #Params includes the parameters of entire network. All methods utilize the same two auxiliary frames as in [20].

We further evaluate our model on the PoseTrack2018 and PoseTrack21 datasets. Due to space limitation, we present these results in the supplementary materials. Based on these results, it is evident that our approach either outperforms or is on par with the state-of-the-art heatmap-based methods. Using the HRNet-w48 backbone, we achieve **82.1** mAP and **82.0** mAP on these two datasets, respectively, while using the ViT-H backbone, we further improve performance by **1.3** and **1.5** points, respectively.

Computation Complexity. We conduct experiments to assess computation complexity using the PoseTrack2017 validation set, and the results are presented in Table 3. To ensure a fair comparison, all methods utilize the identical HRNet-W48 backbone and adopt the identical two auxiliary frames. Our proposed method outperforms heatmap-based methods while utilizing significantly lower computation complexity and fewer model parameters. The FLOPs of our

JFD		w/o	w/	✓	✓	✓
STD	SD			✓		✓
	TD				✓	✓
mAP		73.8	74.8	71.4	78.1	**78.6**

Table 4. **Ablation of different modules** in DSTA.

JFD Method	MFLOPs	mAP
[23] (a)	0.3	74.6
[23] (b)	19.3	77.9
Ours	5.0	**78.6**

Table 5. **Different methods for constructing joint embeddings.**

#Auxiliary Frame	ResNet-50	HRNet-W48	ViT-H
1 {-1}	78.0	82.6	82.6
2 {-1, +1}	78.6	83.4	84.3
4 {-2, -1, +1, +2}	**79.7**	**84.6**	**85.6**

Table 6. **Different number of auxiliary frames**. '-' indicates previous frames while '+' indicates subsequent frames.

Method	Input Size			
	384×288	256×192	128×128	64×64
heatmap-based				
DCPose [20]	82.8	81.2	71.7	35.1
regression-based				
DSTA (Ours)	**83.4**	**82.3**	**77.9**	**55.4**

Table 7. **Performance with different input resolutions**. Note that, as in [20], only two auxiliary frames are used in DSTA.

regression-based head are an almost negligible **1/7835** or **1/550** of those heatmap-based heads. We encourage our readers to refer to the supplementary materials for additional comparisons using smaller backbones, *i.e.*, ResNet and MobileNet. The computational superiority of our proposed regression framework is of great value in the industry, particularly for real-time video applications.

Gains on Low-resolution Input. In practical applications, especially on some edge devices with limited computation resources, it is common to use low-resolution images for reduced computational cost. To explore the robustness of our model under different input resolutions, we compare our method with heatmap-based DCPose [20] on the PoseTrack2017 validation set. As shown in Table 7, our method consistently outperforms DCPose across all input sizes. The results also show that the performance of heatmap-based methods decreases significantly with low-resolution input. For example, at an input resolution of 64×64, our proposed method outperforms DCPose by **20.3** mAP.

4.3. Ablation Study

We conduct ablation experiments to analyze the influence of each component using the PoseTrack2017 validation set. The temporal span T is set to 1, consisting of one preceding and one subsequent frames, totalling 2 auxiliary frames, and the ResNet-50 backbone is employed.

Impact of different modules. Table 4 lists the performance impact of each module of our approach. When we adopt global pose features instead of modeling the temporal dependency at the joint level, *i.e.*, without the JFD and STD modules, the algorithm achieves an mAP of 73.8, decreasing **4.8** points. When capturing spatiotemporal relations based on joints using Eq. 4, *i.e.*, Space-Time coupling, the accuracy reaches **74.8** mAP. It aligns with our assumption that modeling temporal dependency at the joint level, as opposed to the entire pose, is more appropriate. Furthermore, when using the SD and TD modules to separately model the temporal dynamic dependencies and spatial structure dependencies, the algorithm achieves the highest accuracy of **78.6** mAP. It proves that the temporal dependencies of each joint should be individually captured, as every joint exhibits an independent temporal trajectory. Meanwhile, we can see that the TD module capturing temporal dependencies has a much greater impact (**78.1**) on overall performance than the SD module capturing spatial dependencies (**71.4**). We believe this is mainly because the feature token extracted by the JFD module for each joint already contains its spatial context information within the pose. This means the spatial structure information complemented by the SD module is limited.

Choice of JFD. As discussed in Sec. 3.2, [23] proposes an alternative approach to constructing feature embeddings for each joint. We compare our JFD module with this method in Table 5, where [23] (b) augments the feature embedding of each joint with relevant contextual features while [23] (a) does not. As can be seen, our approach improves accuracy by **4.0** points with a relatively small computational overhead, achieving the highest accuracy.

Auxiliary Frames. In addition, we investigate the impact of using different numbers of auxiliary frames. The results presented in Table 6 consistently demonstrate that increasing the number of auxiliary frames leads to improved performance across various backbone networks. It aligns with our intuition that more auxiliary frames can provide more complementary information, thereby facilitating the enhancement of pose estimation for the key frame.

5. Conclusion

In this paper, we propose a novel and effective regression framework for video-based human pose estimation. Through the proposed Decoupled Space-Time Aggregation network (DSTA), we efficiently leverage temporal dependencies in video sequences for multi-frame human pose estimation, while reducing computational and storage requirements. Extensive experiments demonstrate the superiority of our approach over image-based regression methods as well as heatmap-based methods, opening up new possibilities for real-time video applications.

Acknowledgment

This work is supported by "Pioneer" and "Leading Goose" R&D Program of Zhejiang Province (2024C01167).

References

[1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *CVPR*, pages 5167–5176, 2018. 2, 3, 6

[2] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning temporal pose estimation from sparsely labeled videos. In *NIPS*, 2019. 1, 3, 4, 6, 7

[3] Guillem Brasó, Nikita Kister, and Laura Leal-Taixé. The center of attention: Center-keypoint grouping via attention for multi-person pose estimation. In *ICCV*, pages 11853–11863, 2021. 2

[4] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, pages 5386–5395, 2020. 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186, 2019. 5

[6] Andreas Doering, Di Chen, Shanshan Zhang, Bernt Schiele, and Juergen Gall. Posetrack21: A dataset for person search, multi-object tracking and multi-person pose tracking. In *CVPR*, pages 20931–20940, 2022. 2, 3, 6

[7] Runyang Feng, Yixing Gao, Xueqing Ma, Tze Ho Elden Tse, and Hyung Jin Chang. Mutual information-based temporal difference learning for human pose estimation in video. In *CVPR*, pages 17131–17141, 2023. 1

[8] Zigang Geng, Chunyu Wang, Yixuan Wei, Ze Liu, Houqiang Li, and Han Hu. Human pose as compositional tokens. In *CVPR*, pages 660–671, 2023. 1

[9] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *CVPR*, pages 350–359, 2018. 7

[10] Hengkai Guo, Tang Tang, Guozhong Luo, Riwei Chen, Yongchen Lu, and Linfu Wen. Multi-domain pose network for multi-person pose estimation and tracking. In *ECCV Workshops*, 2018. 7

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2

[13] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *CVPR*, pages 5700–5709, 2020. 2

[14] U. Iqbal, A. Milan, and J. Gall. Posetrack: Joint multi-person pose estimation and tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4654–4663, 2017. 1, 2, 3, 6

[15] Kyung-Min Jin, Gun-Hee Lee, and Seong-Whan Lee. Otpose: Occlusion-aware transformer for pose estimation in sparsely-labeled videos. In *2022 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3255–3260, 2022. 1

[16] Sheng Jin, Wentao Liu, Wanli Ouyang, and Chen Qian. Multi-person articulated tracking with spatial and temporal embeddings. In *CVPR*, pages 5664–5673, 2019. 7

[17] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *CVPR*, pages 11977–11986, 2019. 2

[18] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *ICCV*, pages 11005–11014, 2021. 1, 2, 3, 4, 6, 7

[19] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *CVPR*, pages 1944–1953, 2021. 1, 3

[20] Zhenguang Liu, Haoming Chen, Runyang Feng, Shuang Wu, Shouling Ji, Bailin Yang, and Xun Wang. Deep dual consecutive network for human pose estimation. In *CVPR*, pages 525–534, 2021. 1, 2, 3, 4, 6, 7, 8

[21] Z. Liu, R. Feng, H. Chen, S. Wu, Y. Gao, Y. Gao, and X. Wang. Temporal feature alignment and mutual information maximization for video-based human pose estimation. In *CVPR*, pages 10996–11006, 2022. 3, 7

[22] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Liang Wang, Tieniu Tan, and Erjin Zhou. Rethinking the heatmap regression for bottom-up human pose estimation. In *CVPR*, pages 13264–13273, 2021. 2

[23] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. Poseur: Direct human pose regression with transformers. In *ECCV*, pages 72–88, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[24] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *ICCV*, pages 6951–6960, 2019. 3

[25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 1

[26] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5686–5696, 2019. 1, 2, 6, 7

[27] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *ICCV*, pages 2621–2630, 2017. 2

[28] Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, page 1799–1807, 2014. 1, 2

[29] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014. 1, 3

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, page 6000–6010, 2017. 1, 2, 3, 4, 5, 6

[31] Dongkai Wang and Shiliang Zhang. Contextual instance decoupling for robust multi-person pose estimation. In *CVPR*, pages 11060–11068, 2022. 1

[32] Manchen Wang, Joseph Tighe, and Davide Modolo. Combining detection and tracking for human pose estimation in videos. In *CVPR*, pages 11088–11096, 2020. 7

[33] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *ECCV*, pages 527–544, 2020. 3

[34] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 1, 7

[35] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose flow: Efficient online pose tracking. *arXiv preprint arXiv:1802.00977*, 2018. 7

[36] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *NIPS*, 34:28522–28535, 2021. 1

[37] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *NIPS*, 35:38571–38584, 2022. 1, 2

[38] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *ICCV*, pages 11802–11812, 2021. 2

[39] Yiding Yang, Zhou Ren, Haoxiang Li, Chunluan Zhou, Xinchao Wang, and Gang Hua. Learning dynamics via graph neural networks for human pose estimation and tracking. In *CVPR*, pages 8074–8084, 2021. 7

[40] Suhang Ye, Yingyi Zhang, Jie Hu, Liujuan Cao, Shengchuan Zhang, Lei Shen, Jun Wang, Shouhong Ding, and Rongrong Ji. Distilpose: Tokenized pose regression with heatmap distillation. In *CVPR*, pages 2163–2172, 2023. 1

[41] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. *NIPS*, 34: 7281–7293, 2021. 1, 2

[42] Jiabin Zhang, Zheng Zhu, Wei Zou, Peng Li, Yanwei Li, Hu Su, and Guan Huang. Fastpose: Towards real-time pose estimation and tracking via scale-normalized multi-task networks. *arXiv preprint arXiv:1908.05593*, 2019. 7

[43] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *ICCV*, pages 11656–11665, 2021. 1