ORIGINAL ARTICLE

Journal Section

Instance-Motion Aware Network for Human Pose Estimation

Xue Wang[1,2] | **Runyang Feng**[3*] | **Haoming Chen**[3*] | **Yingying Jiao**[1*] | **Roger Zimmermann**[4] | **Zhenguang Liu**[3*]

[1]College of Computer Science and Technology, Jilin University, Changchun, China

[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China

[3]College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China

[4]School of Computing, National University of Singapore, Singapore, Singapore

Correspondence
Runyang Feng, Haoming Chen, Yingying Jiao, Zhenguang Liu
Email:
runyang2019.feng@gmail.com (R. F.),
chenhaomingbob@gmail.com (H. C.),
jiaoyingy17@gmail.com (Y. J.),
liuzhenguang2008@gmail.com (Z. L.)

Funding information
This paper is supported by the Natural Science Foundation of Zhejiang Province, China (Grant No. LQ19F020001), the National Natural Science Foundation of China (No. 61902348), the Key R&D Program of Zhejiang Province (No. 2021C01104), and the Department of Education of Zhejiang Province (No. Y202045070)

Estimating human poses from an image or a video is at the foundation of many visual intelligent systems. Various convolutional neural network based methods have been proposed, achieving state-of-the-art performance on different image datasets. However, most existing approaches are image-based, which deliever unreliable estimations in videos since they fail to model temporal consistency across video frames. Recently, another line of work leverages temporal cues for multi-frame person pose estimation, yet still in an **instance-unaware** fashion, disregarding the specific traits of different instances (persons) or different joints. In this paper, we propose a novel approach to build keypoint motion representations of a person, termed Instance-Motion Aware Network (**IMAPose**), which is equipped with the advantage of **instance-aware** and **motion-aware**. In the IMAPose, we devise three components: (i) an **Instance-Sensitive Extractor** that adaptively computes the spatial feature according to human physical characteristics; (ii) a **Keypoint Motion Encoder** that separately generates convolution kernels with fine-grained keypoint motion encoding; (iii) a **Motion Driven Decoder** that parses multi-frame spatial fea-

*Corresponding Authors.

tures of the same person to provide precise human pose estimations. Extensive experiments on PoseTrack 2017 and PoseTrack 2018 datasets demonstrate that our approach greatly improves the performance of multi-frame human pose estimation . It is worth mentioning that our approach surpasses the state-of-the-art method (PoseWarpper) by + 1.7 mAP and achieves 82.9 mAP in PoseTrack2017 dataset.

KEYWORDS
Human pose estimation, multi-person, multi-frame, dynamic convolution, keypoint motion, fine-grained

1 | INTRODUCTION

The goal of human pose estimation is to locate keypoints (*e.g.*, knee, shoulder) of persons from unconstrained images. Since poses contain abundant human anatomical information, precise pose estimation is coveted in a broad spectrum of human-centric visual applications, including pedestrian tracking, automatic driving, human behavior understanding, and human-computer interaction. Therefore, pose estimation has received increasing attention in the past decade.

Recently, we have witnessed the remarkable success of image-based approaches [1, 2, 3, 4, 5, 6, 7], which are built on successful convolutional neural networks (CNNs). Unfortunately, most state-of-the-art methods are designed for static images, their performances are significantly diminished when they are applied to videos. Undoubtedly, both video and image are important mediums for information dissemination. The video, by nature, brings more challenges such as inadvertently camera shift, rapid object movement, and uncommon human action, which result in frame quality deterioration frequently. The inability of the image-based approaches to compensate for the damaged visual information in frames, ultimately, leads to their unsatisfactory results on video.

To address this issue, researchers [8, 9, 10, 11, 12] investigated how to utilize temporal clues for pose estimation in videos. Specifically, [13, 14] exploit dense optical flow to predict motion fields between every two frames, and levearge motion vectors for aligning features temporally across multiple frames. Such methods are effective when optical flow can be computed accurately. LSTM Pose Machine [15] extends the Pose Machine [6], which handles images in a multi-stage manner, to videos, achieveing good results on simple scenes. DetTrack [16] couples the temporal dimension with HRNet [4] to develop a 3D-HRNet, allowing each pixel to learn the spatial domain and also to preceive the temporal domain for predicting the pose sequence. DCPose [17] analyzes the spatiotemporal context in the video, proposing to construct keypoints search regions and resample upon the region using pose residuals.

Despite the promising results, an important aspect that has been ignored so far is that existing methods do not take into consider the personalized traits of different persons and different joints. By investigating the released implementations of existing methods, we empirically observe that their CNN architectures with the same and fixed convolution kernels for all persons can not handle the personalized appearance of each person well. Meanwhile, current methods tend to treat different joints equally, fail to account for their differences.

In this paper, we argue that estimating the keypoint position accurately in a video needs to learn the representation that is both **instance-aware** and **motion-aware**. Towards this aim, we present a concise and effecitve architecture, termed instance-motion aware network (IMAPose), which generates different convolution kernels for different

persons and different joints. Our framework consists of three components, an *Instance-Sensitive Extractor*, a *Keypoint Motion Encoder*, and a *Motion Driven Decoder*. In particular, **1)** the Instance-Sensitive Extractor extracts the customized spatial features conditioned on human physical properties, which is implemented by deformable convolutions with flexible sampling of both position and weighting. **2)** The Keypoint Motion Encoder takes into account the differences between instance keypoints, and encodes abundant motion contexts of different keypoints into different kernels. **3)** The Motion Driven Decoder collects the spatial feature from current frame and neighboring frames, and aggregates collections into a spatio-temporal representation. The spatio-temporal representation is then utilized to predict the human pose. As illustrated in Fig.1, traditional methods leverage the same kernel for different person instances and use fixed kernels for different joints, in contrast, our method utilizes distinct kernels for different person instances and uses dynamic kernels for different joints.

Experimental results on the widely used large-scale benchmark datasets PoseTrack17 and PoseTrack18 show that our framework IMAPose prominently improves the keypoint localization in videos.

It is worth mentioning that IMAPose achieves a significant +5.6 mAP gain over cutting-edge single-frame method HRNet [4]. We also provide extensive comparison with other existing methods and analyze the impact of each component in the proposed approach.

Our contributions can be summarized as follows:

- We present a novel **Instance-Motion Aware Network (IMAPose)** that automatically generates convolution kernels based on the specific traits of different persons and different joints to facilitate the multi-person pose estimation task in videos.
- We design three components: i) an instance-sensitive extractor that flexibly extracts spatial features according to physical characteristics of different persons, ii) a keypoint motion encoder that condenses motion contexts into multiple convolution kernels for different joints, iii) a motion driven decoder that predicts the human poses with the spatio-temporal features and keypoint motion kernels.
- Our method significantly outperforms prior methods and achieves new state-of-the-art pose estimation results on large-scale benchmark datasets: PoseTrack2017 and PoseTrack2018.

The rest of this paper is organized as follows. Related work is summarized in Sect. 2. A detailed introduction to the proposed IMAPose method is given in Sect. 3. Qualitative and quantitative experimental evaluations, as well as ablation analysis, are demonstrated in Sect. 4. Finally, conclusion and future work are presented in Sect. 5.

2 | RELATED WORK

Briefly, the closely related works can be roughtly cast into three categories, namely *single-frame human pose estimation*, *multi-frame human pose estimation*, and *dynamic convolution*.

2.1 | Single-Frame Human Pose Estimation

Existing single-frame human pose estimation methods can be broadly categorized into two streams, *bottom-up* [18, 19, 20, 21, 22, 23] methods and *top-down* [7, 13, 24, 25, 26] methods.

Bottom-up methods detect all keypoints in an identity-free schema and then groups them to obtain instance-level keypoints. AE [27] proposes to assemble keypoints through learning identity embeddings simultaneously with

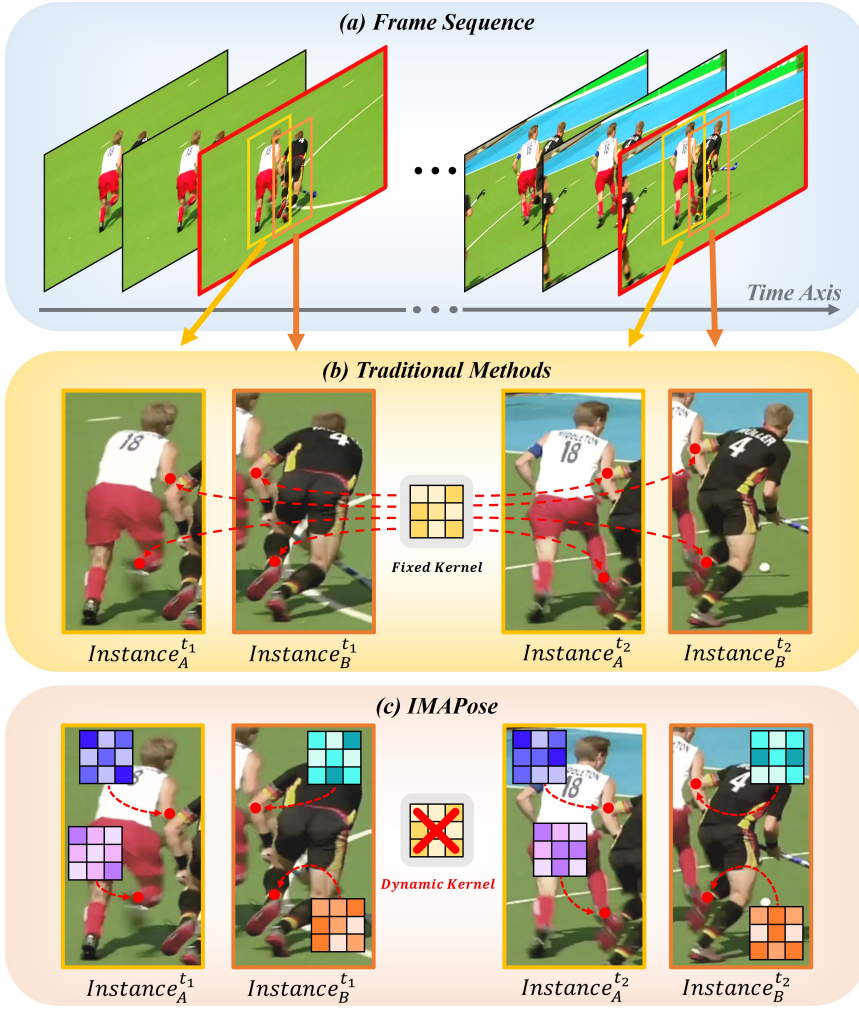

FIGURE 1 An illustration of traditional methods and our IMAPose. **(a)**: Original frame sequence in the dataset. We aim to detect poses in all frames. First, an off-the-shelf human detector is employed to provide the bounding box for each person, as indicated by the yellow and orange solid rectangles in the image. Then the single person pose estimation is performed for each detected person. In what follows, we leverage (b) and (c) to illustrate the traditional methods and our method, respectively. **(b)**: In the traditional methods, the network parameters are freezed once trained. Therefore, the unique convolution kernel is employed for both different people and various keypoints. **(c)** Our proposed IMAPose, which has the advantages of Instance-Aware and Motion-Aware. Regarding instance-aware, our IMAPose adaptively generates diverse convolution kernels for different persons ($Instance_A^{t_1}$ and $Instance_B^{t_1}$) to extract specialized features. For motion-aware, IMAPose dynamically generates a respective convolution kernel for each joint to model fine-grained motion contexts.

detecting keypoints. OpenPose [3] employs part affinity fields to capture pairwise relationships between different body parts. PifPaf [28] uses part intensity fields and part association fields for localizing keypoint and capturing pairwise relationships, respectively. HigherHRNet [29] maintains high-resolution image features throughout the inference process, and achieves favorable keypoint detection performance.

Conversely, top-down methods first detect every person in an image via an off-the-shelf human detector [30, 31], then predict each person's body joints wihin their bounding box range. DeepPose [32] employs an iterative architecture, which first extracts image features with the cascaded convolutional neural network and then regresses the coordinate of joints with multilayer perceptron. Alphapose [2] utilizes the symmetric spatial transformer network to promote inaccurate bounding box into high-quality single person region for human pose estimation. SimpleBaseline [5] explores a simple and effective feature upsampling step, which uses deconvolution layers instead of traditional bilinear interpolation. HRNet [4] keeps high-resolution image features throughout the network architecture, significantly improving the performance of joint detection. Lite-HRNet [33] introduces a lightweight computational unit, which reduces the redundancy and complexity of HRNet and alleviates the performance degradation due to the reduced model size.

Compared to bottom-up approaches, top-down approaches achieve outstanding performance by setting a local region for each person and allowing the single-person pose detector to focus on the local region. In IMAPose, we adopt the top-down paradigm to obtain the bounding boxes of a person in consecutive frames. In Sect. 3, we introduce the paradigm in detail.

2.2 | Multi-Frame Human Pose Estimation

Temporal cues can promote performance in multi-frame human pose estimation. We broadly classify the existing approaches into *Optical-Flow* based [13, 14, 1, 34, 35], *RNNs* (Recurrent Neural Networks) based [15, 36, 37], *Pose-Tracking* based [16, 38, 39, 40], and *keyframe* based [41, 17, 42]. Optical-Flow based methods [13, 14, 1] propose to compute the optical flow between every two frames to obtain human pose predictions. An RNNs based approach, the LSTM pose machine [15], uses the ConvLSTM component to process video frame sequences, achieving good results in simple scenes. Pose tracking-based methods use historical or future poses for pose estimation, such as DetTrack [16], which combines the temporal dimension with HRNet [4] so that features learn both the temporal and spatial domains to obtain a sequence of human poses and optimize the pose with the tracking results. DCPose [17], a state-of-the-art keyframe based method, obtains excellent human pose estimation results by first generating a keypoint search region at the keyframe and then resampling in the region according to the pose residuals.

Comparison on these four categories of methods. Optical flow-based approaches introduce considerable overhead and have performance degradation problems when generating inaccurate optical flows. So far, RNNs based methods cannot be well applied to multi-person pose estimation, since the memory unit can only robustly memorize single-person information. The tracking-based strategy has the advantage of using the same person's historical or future pose sequence, but ensuring the correctness of the pose sequence is a huge challenge especially in fast-moving or crowded scenes. Our system belongs to the keyframe strategy, which performs human pose estimation by focusing on the keyframe and its neighboring frames, and shows better performance.

Contrast to previous approaches that model different people or different key points uniformly, our IMAPose obtains excellent performance by generating distinct convolution kernels for different person appearances and different keypoint motion contexts.

2.3 | Dynamic Convolution

Due to the adaptability, dynamic (or conditional) convolution has attracted extensive attention [43, 44, 45, 46, 47, 48, 49, 18, 50] in the computer vision community. Traditional convolution freezes the kernel *neighborhoods* and *weights* once they are trained. However, for the dynamic convolution, the kernel neighborhoods and/or kernel weights are dynamically modified or predicted based on the input features.

For the kernel neighborhoods, methods in [47, 51, 52] revise the convolution dilation rate to change the receptive field, and approaches in [53, 18] compute the neighborhood sampling grid. For the kernel weights, existing methods focus on modulating or generating the weights of the kernel. DCNV2 [54] builds on the successful DCNV1 [53], introducing a modulation mechanism to modulate the input feature amplitude. Semi-dynamic convolution [45, 55, 56] sets up several expert kernels and predicts the fusion coefficients to obtain a combined kernel. Full-dynamic convolution [57, 48, 58] generates the kernels for a plane or per pixel to improve generalization performance.

In this work, we employ DCNV2 convolution to extract robust spatial features for different persons and utilize full-dynamic convolutions to encode the motion context for different keypoints.

3 | METHODOLOGY

The pipeline of our proposed IMAPose is illustrated in Fig. 2. Given $2D+1$ consecutive frames $I_{[t-D:t+D]}$, D indicates the distance from middle frame I_t to the farthest neighboring frame, we denote the middle frame I_t as the *keyframe*. The aim of IMAPose is to estimate the pose $\hat{P}^i$ in keyframe I_t. A pose P^i depicts the joint locations of person i and is composed of a series of two-dimensional coordinates, each corresponds to the location of a joint.

Particularly, the IMAPose approach consists of three components, which will be presented below. Before that, we first introduce our **data pre-processing** strategy. To obtain the bounding boxes for the same person in consecutive frames, we first obatin the bounding box B_i of person instance i by applying a human detector to the keyframe I_t. Then, considering the motion of the person, we enlarge the box B_i by 25% and employ the enlarged box to crop the same person in frames $I_{[t-D:t+D]}$. Person i in frames $I_{[t-D:t+D]}$ will thus be represented as a cropped video segment, which we denote as tube $\mathcal{T}_t^i$ with radius D. The number of tubes is equal to the number of person instances in the keyframe I_t. Each tube $\mathcal{T}_t^i$ is independently fed into the three components.

In a high level, tube $\mathcal{T}_t^i$ is first fed into an Instance-Sensitive Extractor that serves to output the spatial features $F^S_{I_{t+d}}$ for the person in each frame I_{t+d}. Subsequently, in the Keypoint Motion Encoder, the differences between spatial features of all neighboring frames and keyframe are aggregated as Φ_t, which is then fed into J keypoint encoding branches to obtain J keypoint motion convolution kernels $\mathcal{K}_{1:J}$. Note that J denotes the number of joints. Finally, in Motion Driven Decoder, all spatial features $F^S_{I_{t+d}}$ within the tube $\mathcal{T}_t^i$ are integrated into the spatio-temporal features Ψ_t. The J keypoint motion convolution kernels $\mathcal{K}_{1:J}$ are respectively convolved with the features Ψ_t and followed by a 1×1 convolution to output the heatmap $\mathcal{H}_j$ of the corresponding keypoint j. In what follow, we will present the three components one by one.

3.1 | Instance-Sensitive Extractor

The motivation for our Instance-Sensitive Extractor comes from the following observations and heuristics. 1) Although existing image based pose estimation methods suffer from performance degeneration on videos, we observe that they still provide useful appearance information for an instance. 2) Another heuristic is personalization, *i.e.*, comparing two persons in the same frame or a person in different frames, we observe differences in the *scale/shape* of the human

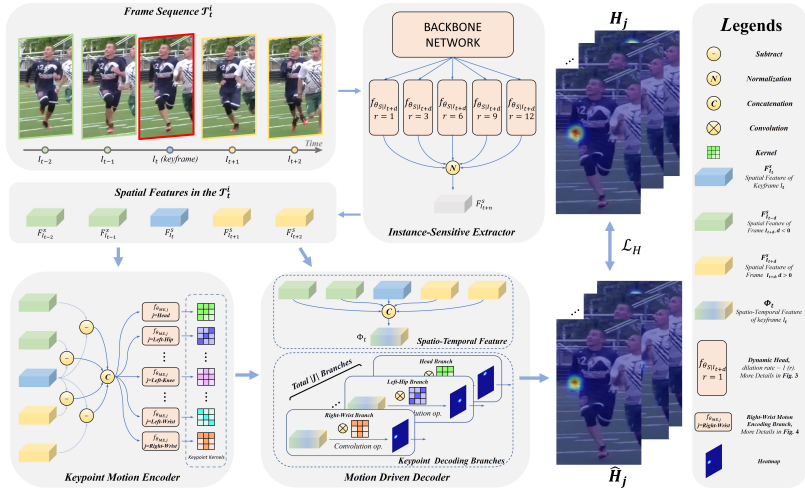

FIGURE 2 Overall pipeline of the proposed IMAPose method. Given a cropped image sequence of a person, the goal is to recognize and locate human keypoints in the keyframe I_t. First, the spatial features of the image sequence are extracted by the Instance-Sensitive Extractor. Differences between features of keyframe and neighboring frames are subsequently leveraged by the keypoint motion encoder to generate convolution kernels for each keypoint. Finally, all keypoint kernels are applied by the motion driven decoder to parse the spatial-temporal features, which outputs the final keypoint heatmaps.

bodies. Therefore, we design an instance-sensitive extractor to extract the unique spatial feature for each person.

In order to extract the high-quality and unique spatial feature for a person, we employ a backbone network to generate the coarse spatial feature $F^A_{I_{t+d}}$, and then use a dynamic head to produce the refined spatial feature $F^S_{I_{t+d}}$. This intuition can be formalized as:

$$\begin{aligned} F^A_{I_{t+d}} &= f_{\theta_A}(I_{t+d}), \\ F^S_{I_{t+d}} &= f_{\theta_{S|I_{t+d}}}(F^A_{I_{t+d}}), \end{aligned} \tag{1}$$

where f_{θ_A} denotes the backbone network with fixed parameters θ_A. The parameters $\theta_{S|I_{t+d}}$ of dynamic head $f_{\theta_{S|I_{t+d}}}$ are subject to I_{t+d}. This allows the dynamic head to distinguish the visual characteristics of different persons.

In conventional convolutions, the kernels after the training phase are frozen and shared by all inputs. A problem arises is that the convolutions optimized on one distribution might be sub-optimal on others. In consider of this, we design a dynamic head to alleviate this issue. The core belief behind proposing the **dynamic head** is that modifying both neighborhoods and weights of the convolution kernels should depend on the input features. Each flexible convolution kernel [54] in our dynamic head involves K sampling locations. We denote W and G as the sampling weights and sampling grid, respectively. For example, a 3×3 kernel is defined with $K = 9$, $W \in \{w_1, w_2, ..., w_9\}$, and $G \in \{(-1,-1), (-1,0), ..., (1,1)\}$, each pixel location q on the spatial feature $F^S_{I_{t+d}}$ is computed as:

$$F^S_{I_{t+d}}(q) = \sum_{k=1}^{K} w_k \cdot F^A_{I_{t+d}}(q + g_k + \triangle o_k) \cdot \triangle w_k \tag{2}$$

where g_k enumerates the locations in the set G. $\triangle o_k$ and $\triangle w_k$ are kernel offset and kernel modulation scalar, which are obtained by the following procedure:

$$\begin{aligned} \triangle o_k &= f_{\theta_O}(F^A_{I_{t+d}}), \\ \triangle w_k &= f_{\theta_W}(F^A_{I_{t+d}}), \end{aligned} \tag{3}$$

where f_{θ_O} and f_{θ_W} denote the networks for computing kernel offset and kernel modulation scalar, respectively.

The receptive field of Instance-Sensitive Extractor is insufficient, since it contains only one dynamic head with a dilation rate of 1. To ease this problem, multiple dynamic heads with different dilation rates $r \in R$ are employed to generate the spatial features that combine local details and global contexts , where $R = \{1, 3, 6, 9, 12\}$. An example of a dynamic head with $r = 2$ is illustrated in Fig. 3. Different dilation rates correspond to varying the size of the effective receptive field whereby enlarging the dilation rate increases the scope of the receptive field. A smaller dilation rate focuses on subtle appearance. Conversely, a larger dilation rate captures global contexts of the human body. We reformulate equation (2) and equation (3) as follows:

$$\begin{aligned} F^S_{I_{t+d}} &= \frac{1}{|R|}\sum_{r\in R} F^S_{I_{t+d,r}}, \\ F^S_{I_{t+d,r}}(q) &= \sum_{k=1}^{K} w_{k,r} \cdot F^A_{I_{t+d}}(q + g_{k,r} + \triangle o_{k,r}) \cdot \triangle w_{k,r}, \\ \triangle o_{k,r} &= f_{\theta_{O,r}}(F^A_{I_{t+d}}), \\ \triangle m_{k,r} &= f_{\theta_{W,r}}(F^A_{I_{t+d}}), \end{aligned} \tag{4}$$

where the subscript r denotes the dilation rate. Therefore, our Instance-Senstive Extractor is instance-aware, which generates the robust spatial features for the person accroding to its physical characteristics.

3.2 | Keypoint Motion Encoder

The Keypoint Motion Encoder leverages the spatial features from instance-sensitive extractor to model motion contexts for all joints. In particular, we first integrate the spatial feature differences between keyframe and neighboring frames. The feature differences are then fed into multiple independent keypoint motion encoding branches to obatin motion kernels for keypoints. We would like to point out that the keypoint motion encoding branches generate convolution kernels **from scratch**, instead of generating convolution kernels by **deforming a fixed kernel** (as done by the dynamic heads in the instance-sensitive extractor). The dynamic heads, modifying kernel sampling position based on the input, is robust to the person appearance. However, we observe that the kernel modulation scalar $\triangle w_k$ in the dynamic head plays an essential role yet has limited modulation capability. It is hard to take into account the human kinematic chains with high freedom degrees [59] when generating convolution kernels by deforming a fixed kernel.

Specifically, the differences between spatial features of all neighboring frames and the keyframe are aggregated as Φ_t. The differences Φ_t are then fed into J keypoint encoding branches to obtain J keypoint motion convolution

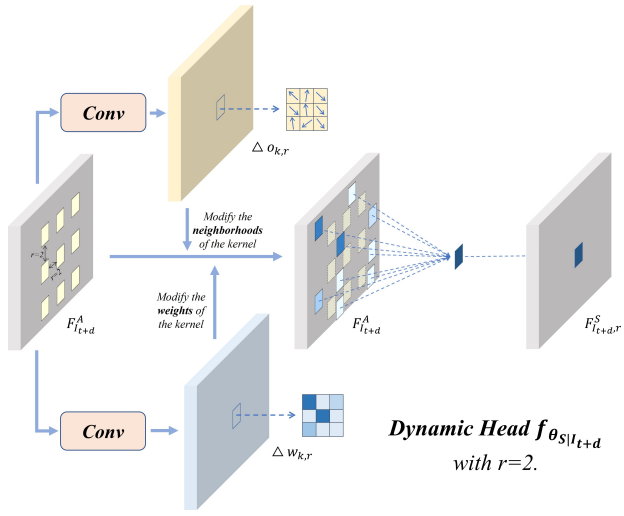

FIGURE 3 Illustration of dynamic head $f_{\theta_{s|I_{t+d}}}$ with $r = 2$. The yellow tensor $\triangle o_{k,r}$ and the blue tensor $\triangle w_{k,r}$ are obtained via two sets of independent convolutions. In particular, the yellow tensor $\triangle o_{k,r}$ is employed to modify the neighborhoods of the convolution kernels, whereas the blue tensor $\triangle w_{k,r}$ is utilized to modify the weights of the convolution kernels.

kernels $\mathcal{K}_{1:J}$. We formalize this process as:

$$\left(\bigoplus_{t+m}^{m\in[-D:-1]} F^S_{I_t} - F^S_{I_{t+m}}\right) \oplus \left(\bigoplus_{t+n}^{n\in[+1:+D]} F^S_{I_{t+n}} - F^S_{I_t}\right) \xrightarrow[\text{convolution}]{3\times3} \Phi_t,$$
$$\mathcal{K}_j = f_{\theta_{ME,j}}(\Phi_t), j \in J, \tag{5}$$

where $t + m$ and $t + n$ are frame indices. Symbol $\oplus$ indicates the concatenation operation, $f_{\theta_{ME,j}}$ denotes the j-th keypoint motion encoding branch, which is responsible for generating the keypoint motion kernel $\mathcal{K}_j$.

In a multi-branch manner, the keypoint motion encoder obtains the fine-grained motion contexts for all keypoints in the keyframe. The Keypoint Motion Encoder further equips our IMAPose with the ability of motion-aware. The procedure for encoding the motion contexts of the right wrist is illustrated in Fig. 4.

3.3 | Motion Driven Decoder

The Motion Driven Decoder performs convolution operation between keypoint motion kernels and spatio-temporal features to provide the final human poses. Specifically, all spatial features in the $\mathcal{T}_t^i$ are concatenated and fed into a 3× 3 convolution layer to yield the spatio-temporal feature Ψ_t. The different keypoint motion kernels $\mathcal{K}_j$ are subsequently used to convolve with the feature Ψ_t separately, followed by a 1×1 convolution outputs corresponding keypoint

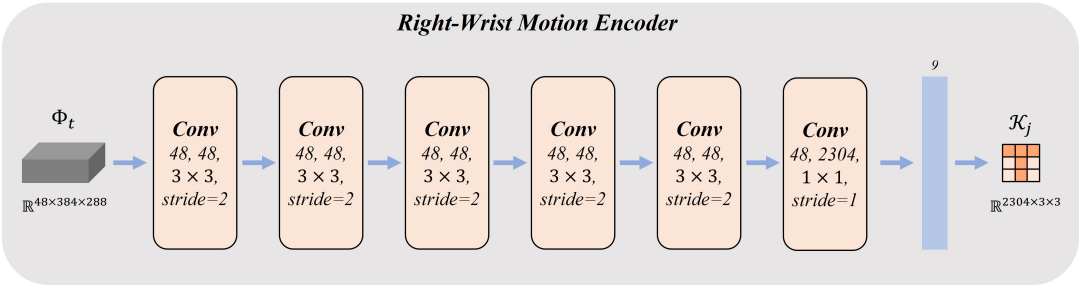

FIGURE 4 We take the Motion Encoder of *right-wrist* as an example. The feature tensor Φ_t is fed into a sequential convolution structure with six convolution layers and one fully connected layer to obtain the convolution kernel $\mathcal{K}_j$ for *right-wrist*. The descriptions of the convolution layers from top to bottom denote input channel and output channel, kernel size, and stride, respectively.

heatmaps:

$$\begin{aligned} \bigoplus_{t+d}^{d\in[-D:+D]} F^{\mathcal{S}}_{I_{t+d}} &\xrightarrow[\text{convolution}]{3\times3} \Psi_t, \\ \Psi_t \otimes \mathcal{K}_j &\xrightarrow[\text{convolution}]{1\times1} \mathcal{H}_j, \end{aligned} \tag{6}$$

where $\otimes$ denotes the convolution operation. Ψ_t and $\mathcal{K}_j$ are input tensor and convolutional kernel, respectively.

To obtain the keypoint coordinate, let $p_j = (x, y)$ denotes the position of keypoint j in the image, p_j can be computed as:

$$p_j = argmax(H_j). \tag{7}$$

Ultimately, the above procedure is performed for each tube $\mathcal{T}_t^i$. By effectively generating specific representations for different persons (instance-aware) and various joints (motion-aware) in our IMAPose network, the final heatmaps is predicted more precisely.

3.4 | Loss Function

We employ the standard pose estimation loss function as our cost function. The training aims to minimize the total L2 distance between prediction and ground truth heatmaps for all joints. The loss function is defined as:

$$\mathcal{L}_H = \frac{1}{|J|} \sum_{j}^{j\in J} v_j \times \ell_2(H_j, \hat{H}_j) \tag{8}$$

where $\hat{H}_j$, H_j, and v_j denote the prediction heatmap, ground truth heatmap and visibility of joint j.

4 | EXPERIMENTS

In this section, we evaluate the proposed method on two widely used benchmark datasets: PoseTrack2017 and PoseTrack2018 datasets[1]. We seek to answer the following research questions.

- **RQ1:** How is the proposed method comparing to state-of-the-art human pose estimation approaches on **quantitative** results?
- **RQ2:** How is the proposed method comparing to state-of-the-art human pose estimation approaches on **visual** results?
- **RQ3:** How much do different components of IMAPose contribute to its performance?

Next, we first present the experimental settings, followed by answering the above research questions one by one.

4.1 | Experimental Settings

Dataset PoseTrack is a large-scale public dataset for human pose estimation and articulated tracking in unconstrained videos. **PoseTrack2017** dataset contains 514 video clips and 16,219 pose annotations, with 250 clips reserved for training, 50 clips for validation, and 214 clips for testing. The **PoseTrack2018** dataset greatly increased the number of video clips and contains 593 for training, 170 for validation, and 375 for testing. The training videos are annotated densely within the center 30 frames of each video clip. For the validation videos, annotations are provided every four frames across the whole video clip besides the dense annotation of the center 30 frames. Both PoseTrack2017 and PoseTrack2018 identify 15 joints, with an additional label for joint visibility. We train and evaluate our model only for visible joints.

Evaluation Metric As done in the literature [60], we employ average precision (**AP**) to evaluate the performance of our model in terms of human pose estimation. We compute this metric independently on each joint and then obtain the final mean **AP** (mAP) by averaging over all joints.

Parameter Settings We independently train the IMAPose network on PoseTrack2017 and PoseTrack2018 using same configurations. During training, we incorporate the data augmentation including random rotation $[-45°, 45°]$, ramdom scale [0.65,1.35], truncation and horizontal flip. Input image size is fixed to 384×288. The default radius D of tube $\mathcal{T}_t^i$ is set to 2. We employ the HRNet-W48 [4] pretrained on COCO dataset as our backbone network. All subsequent weight parameters are randomly initialized from a Gaussian distribution with $\mu = 0$ and $\sigma = 0.001$, while bias is consistently initialized to 0. We use the Adam optimizer. The base learning rate is set to $1e-3$, which drops to $1e-4$, $1e-5$ at the 8^{th} and 15^{th} epochs, respectively. We train our model on 4 Nvidia GeForce 2080Ti GPUs, and all training process is terminated within 20 epochs.

4.2 | Quantitative Comparison with Existing Methods (RQ1)

Results on PoseTrack2017 Dataset We first evaluate our approach on PoseTrack2017 validation set and full test set using the widely adopted average precision (AP) metric. Table 1 summarizes quantitative comparisons with state-of-the-art methods on the PoseTrack2017 validation set, including the APs of keypoint, such as Head, Shoulder, Knee, and Elbow, as well as the mAP for all joints. Our IMAPose surpasses existing methods and obtains a significant 82.9 mAP on the validation set.

[1] https://posetrack.net

Results on the PoseTrack2017 test set are provided in Table 2. These results are obtained by uploading the prediction results to the PoseTrack evaluation server (https://posetrack.net/leaderboard.php) due to the unavailability of test annotations. Our IMAPose network achieves state-of-the-art results on the test set, and obtains a 79.3 mAP. We also receive encouraging results for relatively difficult keypoints, with final AP of 76.1 and 71.0 for the wrist and ankle, respectively. Compared to the leading single-frame pose estimation model, HRNet[4], our method also improves the accuracy by **4.4** mAP. Some visual results on PoseTrack2017 dataset are illustrated in Fig. 5, which demonstrate the robustness of our method in challenging situations.

Results on PoseTrack2018 Dataset We also benchmark our model on PoseTrack2018 dataset, and the AP results of validation and test set are tabulated in Table 3 and Table 4, respectively. As shown in the tables, our model once again achieves state-of-the-art results on both validation and test sets, consistently outperforming all existing methods. We obtain the final mAP of 79.0 on the test set with promising improvements on challenging joints, wrist (77.0) and ankle (71.6). A few visual results on PoseTrack2018 datasets are depicted in Fig. 7.

TABLE 1 Quantitative results of Instance-Motion Aware Network (IMAPose) and state-of-the-art methods on **PoseTrack2017 validation set**.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	**mAP**
PoseTracker[61]	67.5	70.2	62.0	51.7	60.7	58.7	49.8	60.6
PoseFlow[62]	66.7	73.3	68.3	61.1	67.5	67.0	61.3	66.5
JointFlow[63]	-	-	-	-	-	-	-	69.3
FastPose[64]	80.0	80.3	69.5	59.1	71.4	67.5	59.4	70.3
SimpleBaseline[5]	81.7	83.4	80.0	72.4	75.3	74.8	67.1	76.7
STEmbedding[8]	83.8	81.6	77.1	70.0	77.4	74.5	70.8	77.0
MDPN[9]	85.2	88.5	83.9	77.5	79.0	77.0	71.4	80.7
HRNet[4]	82.1	83.6	80.4	73.3	75.5	75.3	68.5	77.3
Dynamic-GNN[38]	88.4	88.4	82.0	74.5	79.1	78.3	73.1	81.1
PoseWarper[41]	81.4	88.3	83.9	78.0	82.4	80.5	73.6	81.2
IMAPose	**88.2**	**88.7**	**84.3**	**78.4**	**82.9**	**81.7**	**73.6**	**82.9**

4.3 | Visual Comparison with Existing Methods (RQ2)

To validate the robustness of our model in complex scenarios, we illustrate in Fig. 6 the side-by-side comparisons of our IMAPose network against state-of-the-art methods. Each column describes different challenging scenarios including *nearby person*, *pose occlusions* and *rapid motion*, whilst each row depicts the pose detections from different approaches. Obviously our method produces more accurate predictions on such challenging situations compared to state-of-the-art methods, PoseWarper and HRNet-W48. HRNet-W48 is designed and trained on still images and does not model motion contexts for different keypoints, which leads to unsatisfactory results in videos. PoseWarper

TABLE 2 Quantitative results of Instance-Motion Aware Network (IMAPose) and state-of-the-art methods on **PoseTrack2017 test set**.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	**mAP**
PoseTracker[61]	–	–	–	51.5	–	–	50.17	59.6
PoseFlow[62]	64.9	67.5	65.0	59.0	62.5	62.8	57.9	63.0
JointFlow[63]	-	-	-	53.1	-	-	50.4	63.4
KeyTrack[65]	-	-	-	71.9	-	-	65.0	74.0
DetTrack[16]	-	-	-	69.8	-	-	65.9	74.1
SimpleBaseline[5]	80.1	80.2	76.9	71.5	72.5	72.4	65.7	74.6
HRNet[4]	80.1	80.2	76.9	72.0	73.4	75.2	67.0	74.9
PoseWarper[41]	79.5	84.3	80.1	75.8	77.6	76.8	70.8	77.9
IMAPose	**84.9**	**84.6**	**80.5**	**76.1**	**77.9**	**77.5**	**71.0**	**79.3**

TABLE 3 Quantitative results of Instance-Motion Aware Network (IMAPose) and state-of-the-art methods on **PoseTrack2018 validation set**.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	**mAP**
TML++[66]	–	–	–	60.2	–	–	56.9	67.8
AlphaPose[2]	63.9	78.7	77.4	71.0	73.7	73.0	69.7	71.9
MDPN[9]	75.4	81.2	79.0	74.1	72.4	73.0	69.9	75.0
PGPT[10]	-	-	-	72.3	-	-	72.2	76.8
STAF[11]	-	-	-	64.7	-	-	62	70.4
Dynamic-GNN[38]	80.6	84.5	80.6	74.4	75.0	76.7	71.8	77.9
PoseWarper[38]	79.9	86.3	82.4	77.5	79.8	78.8	**73.2**	79.7
IMAPose	**84.2**	**86.8**	**82.7**	**77.9**	**80.4**	**79.4**	**73.2**	**80.9**

models motion contexts using only one auxiliary frame, which might be insufficient. Moreover, the personalized representations for different person instances and keypoints are also absent in the two state-of-the-art methods. Our IMAPose network simultaneously generates specific convolution kernels for different persons and joints, and achieves new state-of-the-art both quantitatively and qualitatively.

FIGURE 5 Visual results of some examples in the **PoseTrack2017** dataset, which contain challenging scences including motion blur, occlusions, and multiple persons.

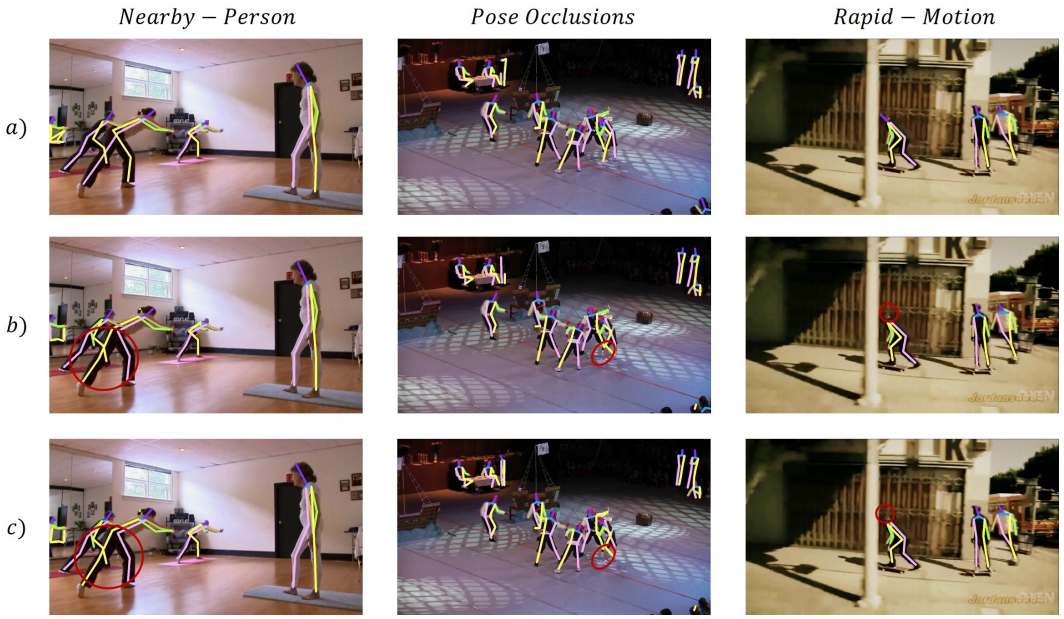

FIGURE 6 Visual results of the pose predictions of our IMAPose network(a), PoseWarper(b), and HRNet(c) on the challenging situations from the PoseTrack2017 and PoseTrack2018 datasets. Each column from left to right denotes the Nearby-Person, Pose Occlusions, and Rapid-Motion scene, respectively. Inaccurate keypoint predictions are highlighted with the red solid circles.

TABLE 4 Quantitative results of Instance-Motion Aware Network (IMAPose) and state-of-the-art methods on **PoseTrack2018 test set**.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	**mAP**
TML++[66]	–	–	–	60.2	–	–	56.8	76.4
OpenSVAI[12]	-	-	-	66.8	-	-	62.4	69.4
FlowTrack[2]	-	-	-	73.0	-	-	69.0	74.0
MDPN[9]	-	-	-	74.5	-	-	71.8	76.4
DetTrack[16]	-	-	-	69.8	-	-	67.1	73.5
OpenSVAI[12]	-	-	-	66.8	-	-	62.4	69.4
Miracle[39]	-	-	-	68.1	-	-	66.1	70.9
PoseWarper[38]	78.9	**84.4**	**80.9**	76.8	75.6	77.5	**71.8**	78.0
IMAPose	**83.3**	84.2	**80.9**	**77.0**	**76.0**	**77.6**	71.6	**79.0**

4.4 | Ablation Study (RQ3)

We further conduct extensive ablation experiments on PoseTrack2017 dataset to examine the influence of individual components in the proposed framework. Through ablating the various modular networks including Instance-Sensitive Extractor, Keypoint Motion Encoder, and Motion Driven Decoder, we evaluate their respective contributions toward the overall performance. We also investigate the effect of tube radius D. The results are presented in Table 5.

To more clearly evaluate the validity of our designs, we propose an IMAPose-Baseline network which removes all key designs from our IMAPose. Specifically, we delete the dynamic heads of Instance-Sensitive Extractor, and then replace the independent keypoint kernels for all joints in the Motion Encoder with a shared pose kernel. The strengthened spatio-temporal feature in Motion Driven Decoder is also replaced by the keyframe spatial feature. Under this configuration, the performance dramatically decreases from 82.9 to 80.5. This significant performance

FIGURE 7 Visual results of some examples in the **PoseTrack2018** dataset, which contain challenging scences including motion blur, occlusions, and multiple persons.

TABLE 5 Ablation study of different components in our IMAPose performed on PoseTrack2017 validation set. "r/m X" refers to removing X module in our network. The complete IMAPose consistently achieves the best results which are highlighted.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
IMAPose, Complete	**88.2**	**88.7**	**84.3**	**78.4**	**82.9**	**81.7**	**73.6**	**82.9**
IMAPose, Baseline.	85.0	88.3	83.7	76.5	80.3	77.3	71.4	80.5
Tube $\mathcal{T}_t^i$								
D=1	87.8	88.5	83.8	77.6	82.7	81.0	73.2	82.5
D=3	87.9	88.5	83.4	77.4	82.1	80.9	72.8	82.3
Instance-Sensitive Extractor								
r/m dynamic heads	87.4	88.0	83.4	77.3	82.3	79.8	72.3	81.8
1 dynamic head, r = 1	87.7	88.4	83.7	77.3	82.4	80.4	73.2	82.3
2 dynamic heads, r = 1,3	87.8	88.2	83.6	77.6	82.6	80.8	73.1	82.4
3 dynamic heads, r = 1,3,5	88.3	88.7	83.9	78.2	82.7	81.3	73.2	82.7
4 dynamic heads, r = 1, 3, 6, 9	88.2	88.4	83.8	78.0	82.7	81.3	73.3	82.6
Keypoint Motion Encoder								
keypoint-level ← human-level	87.6	88.3	83.6	77.5	82.5	80.6	72.5	82.2
Motion Driven Decoder								
r/m feature temporal aggregation	88.0	88.3	83.7	77.5	82.8	80.8	72.9	82.4

degoration (2.4 mAP) suggests the validity of our network design.

Instance-Sensitive Extractor We study the effects of adopting different sets of dilations for the dynamic heads in the instance-sensitive extractor. Five different settings are experimented: *without dynamic heads*, $r \in \{1\}$, $r \in \{1, 3\}$, $r \in \{1, 3, 6\}$, and $r \in \{1, 3, 6, 9\}$ whereas the complete IMAPose framework setting has $r \in \{1, 3, 6, 9, 12\}$. From the results of Table 5, we can observe that the mAP gradually improves with the increase in the number of the dilation rates, namely from 81.8→82.3→82.4→82.7→82.6→82.9. This confirms our intuitions, i) dynamic heads allow generating personalized representations for different human subjects, which greatly increase the modeling ability of our network; ii) parallel configuration of dilation convolutions further equips the network with multiple different receptive fields, combining the global consistency of the full human body and detailed descriptions of different body parts leads to more robust spatial features.

Keypoint Motion Encoder In this ablation setting, we focus on the performance of fine-grained motion modeling. We remove the specific keypoint kernels and utilize a shared pose kernel to model the human-level motion, which produces the final accuracy of 82.2 mAP. The significant performance degradation upon removal of the keypoint kernels can be attributed to that motion contexts of different joints are distinct, and the failure of modeling fine-grained keypoint motions leads to the sub-optimal results.

Motion Driven Decoder For the studies of Motion Driven Decoder, we examine the validity of feature temporal aggregation. Specifically, we directly replace the aggregated representations Ψ_t with the spatial feature of the keyframe: $\Psi_t = F_{I_t}^S$. Experimental results in Table 5 show a drop of 0.5 on mAP. This reduction on performance re-

flects the importance of temporal aggregation, *i.e.*, aggregation of features at different frames effectively enlarges the searching scopes for all keypoints, resulting in a more accurate keypoint localization.

Tube Length The radius D of the tube is a hyper-parameter whose default value is set to 2. We experiment with stretching and compressing the input tube, where D is set to 1, 2, 3 respectively, which correspond to the tube length of 3, 5 and 7. The mAP results decrease from 82.9 for $D = 2$ to 82.5, 82.3 at $D = 1, 3$, respectively. This is in accordance with our expectations, shorter frame sequence in the tube contains limited motion information which provides finite guidance for modeling motion contexts of different keypoints. Conversely, overlength tube leads to more uncertainty on the identity of a single person (pending pose estimation), and the temporal cues in this case are often interfering. This also explains our choice of the tube length.

5 | CONCLUSION

In this paper, we present an instance-motion aware network for multi-frame person pose estimation, which dynamically generates convolution kernels depending on the particular traits of different persons and different keypoints. Three components are designed in our framework. An Instance-Sensitive Extractor flexibly extracts robust spatial features for different persons while a Keypoint Motion Encoder compresses the rich motion contexts into a set of convolution kernels for different keypoints. These are then processed via our Motion Driven Decoder for effectively estimating of human poses. With respect to the sophisticated scenarios such as pose occlusions, nearby person, and rapid motion, the proposed method can produce robust pose estimations due to its instance-aware and motion-aware properties. Extensive experiments confirm that our method significantly surpasses existing methods on the large-scale benchmark datasets PoseTrack2017 and PoseTrack2018.

references

[1] Weinzaepfel P, Revaud J, Harchaoui Z, Schmid C. DeepFlow: Large displacement optical flow with deep matching. In: Proceedings of the IEEE international conference on computer vision; 2013. p. 1385–1392.

[2] Fang HS, Xie S, Tai YW, Lu C. Rmpe: Regional multi-person pose estimation. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2334–2343.

[3] Cao Z, Simon T, Wei SE, Sheikh Y. Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 7291–7299.

[4] Sun K, Xiao B, Liu D, Wang J. Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2019. p. 5693–5703.

[5] Xiao B, Wu H, Wei Y. Simple baselines for human pose estimation and tracking. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 466–481.

[6] Wei SE, Ramakrishna V, Kanade T, Sheikh Y. Convolutional pose machines. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition; 2016. p. 4724–4732.

[7] Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation. In: European conference on computer vision Springer; 2016. p. 483–499.

[8] Jin S, Liu W, Ouyang W, Qian C. Multi-person articulated tracking with spatial and temporal embeddings. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. p. 5664–5673.

[9] Guo H, Tang T, Luo G, Chen R, Lu Y, Wen L. Multi-domain pose network for multi-person pose estimation and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 0–0.

[10] Bao Q, Liu W, Cheng Y, Zhou B, Mei T. Pose-Guided Tracking-by-Detection: Robust Multi-Person Pose Tracking. IEEE Transactions on Multimedia 2020;.

[11] Raaj Y, Idrees H, Hidalgo G, Sheikh Y. Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. p. 4620–4628.

[12] Ning G, Liu P, Fan X, Zhang C. A top-down approach to articulated human pose estimation and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 0–0.

[13] Pfister T, Charles J, Zisserman A. Flowing convnets for human pose estimation in videos. In: Proceedings of the IEEE International Conference on Computer Vision; 2015. p. 1913–1921.

[14] Song J, Wang L, Van Gool L, Hilliges O. Thin-slicing network: A deep structured model for pose estimation in videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 4220–4229.

[15] Luo Y, Ren J, Wang Z, Sun W, Pan J, Liu J, et al. Lstm pose machines. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 5207–5215.

[16] Wang M, Tighe J, Modolo D. Combining detection and tracking for human pose estimation in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 11088–11096.

[17] Liu Z, Chen H, Feng R, Wu S, Ji S, Yang B, et al. Deep Dual Consecutive Network for Human Pose Estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 525–534.

[18] Geng Z, Sun K, Xiao B, Zhang Z, Wang J. Bottom-Up Human Pose Estimation Via Disentangled Keypoint Regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 14676–14686.

[19] Luo Z, Wang Z, Huang Y, Wang L, Tan T, Zhou E. Rethinking the Heatmap Regression for Bottom-up Human Pose Estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 13264–13273.

[20] Wei F, Sun X, Li H, Wang J, Lin S. Point-set anchors for object detection, instance segmentation and pose estimation. In: European Conference on Computer Vision Springer; 2020. p. 527–544.

[21] Jin S, Liu W, Xie E, Wang W, Qian C, Ouyang W, et al. Differentiable hierarchical graph grouping for multi-person pose estimation. In: European Conference on Computer Vision Springer; 2020. p. 718–734.

[22] Li J, Su W, Wang Z. Simple pose: Rethinking and improving a bottom-up approach for multi-person pose estimation. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34; 2020. p. 11354–11361.

[23] Nie X, Feng J, Xing J, Yan S. Pose partition networks for multi-person pose estimation. In: Proceedings of the european conference on computer vision (eccv); 2018. p. 684–699.

[24] Nie X, Li Y, Luo L, Zhang N, Feng J. Dynamic kernel distillation for efficient pose estimation in videos. In: Proceedings of the IEEE International Conference on Computer Vision; 2019. p. 6942–6950.

[25] Zhang F, Zhu X, Ye M. Fast human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 3517–3526.

[26] Moon G, Chang JY, Lee KM. Posefix: Model-agnostic general human pose refinement network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 7773–7781.

[27] Newell A, Huang Z, Deng J. Associative Embedding: End-to-End Learning for Joint Detection and Grouping. Advances in Neural Information Processing Systems 2017;30.

[28] Kreiss S, Bertoni L, Alahi A. Pifpaf: Composite fields for human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. p. 11977–11986.

[29] Cheng B, Xiao B, Wang J, Shi H, Huang TS, Zhang L. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 5386–5395.

[30] Qiao S, Chen LC, Yuille A. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 10213–10224.

[31] Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 6154–6162.

[32] Güler RA, Neverova N, Kokkinos I. Densepose: Dense human pose estimation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 7297–7306.

[33] Yu C, Xiao B, Gao C, Yuan L, Zhang L, Sang N, et al. Lite-hrnet: A lightweight high-resolution network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 10440–10450.

[34] Zhang D, Shah M. Human pose estimation in videos. In: Proceedings of the IEEE International Conference on Computer Vision; 2015. p. 2012–2020.

[35] Zhang D, Guo G, Huang D, Han J. Poseflow: A deep motion representation for understanding human behaviors in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 6762–6770.

[36] Artacho B, Savakis A. Unipose: Unified human pose estimation in single images and videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 7035–7044.

[37] Gkioxari G, Toshev A, Jaitly N. Chained predictions using convolutional neural networks. In: European Conference on Computer Vision Springer; 2016. p. 728–743.

[38] Yang Y, Ren Z, Li H, Zhou C, Wang X, Hua G. Learning Dynamics via Graph Neural Networks for Human Pose Estimation and Tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 8074–8084.

[39] Yu D, Su K, Sun J, Wang C. Multi-person pose estimation for pose tracking with enhanced cascaded pyramid network. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 0–0.

[40] Zhou C, Ren Z, Hua G. Temporal Keypoint Matching and Refinement Network for Pose Estimation and Tracking. In: European Conference on Computer Vision Springer; 2020. p. 680–695.

[41] Bertasius G, Feichtenhofer C, Tran D, Shi J, Torresani L. Learning temporal pose estimation from sparsely-labeled videos. In: Advances in Neural Information Processing Systems; 2019. p. 3027–3038.

[42] Zhang Y, Wang Y, Camps O, Sznaier M. Key Frame Proposal Network for Efficient Pose Estimation in Videos. In: European Conference on Computer Vision Springer; 2020. p. 609–625.

[43] Jia X, De Brabandere B, Tuytelaars T, Gool LV. Dynamic filter networks. Advances in neural information processing systems 2016;29:667–675.

[44] Zamora Esquivel J, Cruz Vargas A, Lopez Meyer P, Tickoo O. Adaptive convolutional kernels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops; 2019. p. 0–0.

[45] Chen Y, Dai X, Liu M, Chen D, Yuan L, Liu Z. Dynamic convolution: Attention over convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 11030–11039.

[46] Zhou J, Jampani V, Pi Z, Liu Q, Yang MH. Decoupled Dynamic Filter Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 6647–6656.

[47] Zhang R, Tang S, Zhang Y, Li J, Yan S. Scale-adaptive convolutions for scene parsing. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 2031–2039.

[48] Tian Z, Shen C, Chen H. Conditional convolutions for instance segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16 Springer; 2020. p. 282–298.

[49] Pang Y, Zhang L, Zhao X, Lu H. Hierarchical dynamic filtering network for rgb-d salient object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16 Springer; 2020. p. 235–252.

[50] Zhang J, Xie Y, Xia Y, Shen C. DoDNet: Learning to segment multi-organ and tumors from multiple partially labeled datasets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 1195–1204.

[51] Tabernik D, Kristan M, Leonardis A. Spatially-adaptive filter units for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 9388–9396.

[52] Tabernik D, Kristan M, Leonardis A. Spatially-adaptive filter units for compact and efficient deep neural networks. International Journal of Computer Vision 2020;128(8):2049–2067.

[53] Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, et al. Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 764–773.

[54] Zhu X, Hu H, Lin S, Dai J. Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 9308–9316.

[55] Ma N, Zhang X, Huang J, Sun J. Weightnet: Revisiting the design space of weight networks. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16 Springer; 2020. p. 776–792.

[56] Yang B, Bender G, Le QV, Ngiam J. CondConv: Conditionally Parameterized Convolutions for Efficient Inference. Advances in Neural Information Processing Systems 2019;32:1307–1318.

[57] Wang J, Chen K, Xu R, Liu Z, Loy CC, Lin D. Carafe: Content-aware reassembly of features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. p. 3007–3016.

[58] Wang X, Zhang R, Kong T, Li L, Shen C. Solov2: Dynamic, faster and stronger. arXiv e-prints 2020;p. arXiv–2003.

[59] Liu Z, Wu S, Jin S, Liu Q, Lu S, Zimmermann R, et al. Towards natural and accurate future motion prediction of humans and animals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 10004–10012.

[60] Andriluka M, Iqbal U, Insafutdinov E, Pishchulin L, Milan A, Gall J, et al. Posetrack: A benchmark for human pose estimation and tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 5167–5176.

[61] Girdhar R, Gkioxari G, Torresani L, Paluri M, Tran D. Detect-and-track: Efficient pose estimation in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 350–359.

[62] Xiu Y, Li J, Wang H, Fang Y, Lu C. Pose flow: Efficient online pose tracking. arXiv preprint arXiv:180200977 2018;.

[63] Doering A, Iqbal U, Gall J. Joint flow: Temporal flow fields for multi person tracking. arXiv preprint arXiv:180504596 2018;.

[64] Zhang J, Zhu Z, Zou W, Li P, Li Y, Su H, et al. Fastpose: Towards real-time pose estimation and tracking via scale-normalized multi-task networks. arXiv preprint arXiv:190805593 2019;.

[65] Snower M, Kadav A, Lai F, Graf HP. 15 keypoints is all you need. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 6738–6748.

[66] Hwang J, Lee J, Park S, Kwak N. Pose estimator and tracker using temporal flow maps for limbs. In: 2019 International Joint Conference on Neural Networks (IJCNN) IEEE; 2019. p. 1–8.