Learning Dynamics via Graph Neural Networks for Human Pose Estimation and Tracking

Yiding Yang[1]*, Zhou Ren[2], Haoxiang Li[2], Chunluan Zhou[2], Xinchao Wang[1,3]†, Gang Hua[2]
[1]Stevens Institute of Technology, [2]Wormpex AI Research, [3]National University of Singapore
{yyang99, hli18, xinchao.wang}@stevens.edu, renzhou200622@gmail.com,
czhou002@e.ntu.edu.sg, ganghua@gmail.com

Abstract

Multi-person pose estimation and tracking serve as crucial steps for video understanding. Most state-of-the-art approaches rely on first estimating poses in each frame and only then implementing data association and refinement. Despite the promising results achieved, such a strategy is inevitably prone to missed detections especially in heavily-cluttered scenes, since this tracking-by-detection paradigm is, by nature, largely dependent on visual evidences that are absent in the case of occlusion. In this paper, we propose a novel online approach to learning the pose dynamics, which are independent of pose detections in current fame, and hence may serve as a robust estimation even in challenging scenarios including occlusion. Specifically, we derive this prediction of dynamics through a graph neural network (GNN) that explicitly accounts for both spatial-temporal and visual information. It takes as input the historical pose tracklets and directly predicts the corresponding poses in the following frame for each tracklet. The predicted poses will then be aggregated with the detected poses, if any, at the same frame so as to produce the final pose, potentially recovering the occluded joints missed by the estimator. Experiments on PoseTrack 2017 and PoseTrack 2018 datasets demonstrate that the proposed method achieves results superior to the state of the art on both human pose estimation and tracking tasks.

1. Introduction

Multi-person pose estimation and tracking find their applications in a wide spectrum of scenarios including behavior analysis and action recognition, and have therefore received increasing attention in recent years [45, 32, 19]. Despite often coupled together, they focus on slightly different aspects: the former aims to locate human joints in each frame of an input video, while the latter one aims to associate joints that belong to the same human across frames. It has been long considered as a challenging task due to various factors, including but not limited to camera motions, complex backgrounds, and mutual occlusions.

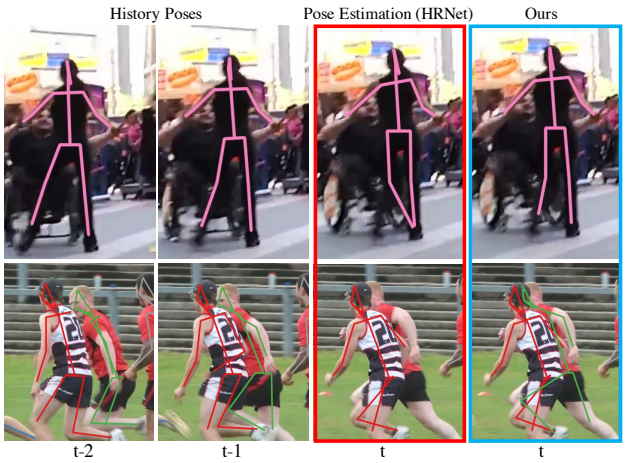

Figure 1. By modeling the pose dynamics from history poses through a graph neural network, our method learns a pose prediction that is robust to challenging scenes, such as motion blur (top) and occlusion (bottom). In both cases, the visual-based HRNet [37] fails to locate the joints, yet our approach delivers dependable pose estimations.

Thanks to the recent advances of deep learning techniques, pose estimation and tracking have witnessed unprecedented results in the past years. Existing methods can be broadly categorized into two streams, bottom-up methods [32, 19, 55, 18] and top-down methods [45, 53, 40]. Bottom-up methods first generate joint candidates and then group the joints into a person detection. The grouped joints are then associated across frames to generate the final pose tracking results. Top-down methods, on the other hand, first detect human candidates in a single frame and then estimate the human poses for each candidate. The estimated human poses are associated across frames to achieve pose tracking. Methods from both streams have produced promising

*The work is partially done when the author is an internship at Wormpex AI Research.
†Corresponding author.

results on various scenarios [53, 40].

In spite of the encouraging results, state-of-the-art pose estimation and tracking approaches remain prone to missed detections especially in highly-cluttered and fast-motion scenes. This is not totally unexpected, since by nature they rely on first detecting either joints or human bodies in a scene using a visual-based detector, and only then carrying out data association to link the detections into tracks. In challenging scenarios such as crowded or blurred scenes, the joint- or human-detector would inevitably fail due to the absent image evidences. Although some succeeding refinement steps would mildly remedy the flawed estimations, they are are still largely dependent on visual cues and hence incompetent to fully tackle missed detections.

We propose in this paper a novel approach by explicitly looking into the *dynamics* of human poses within image sequences. In contrast to state-of-the-art approaches that rely on first detecting human or joints in each frame, which is again prone to failures in the absence of detection evidences, our approach first *predicts* poses in a frame from a track of history without looking at any detection cue. This strategy allows us to free our dependency on the detection evidences and consequently produce a legitimate state of human pose at the very first place. Specifically, in our approach this prediction step is accomplished through a graph neural network (GNN) that takes as input a track of history poses in previous frames. Next, the predicted pose is aggregated with the detected poses, if any, in the same frame to produce the final pose, in which way both dynamical and visual information are exploited. At a conceptual level, our approach follows a similar spirit of Bayesian filters, expect that in our approach all parameters and features are learned end to end. A qualitative example is shown in Figure 1, where our dynamic-based approach yields dependable pose estimation results in the cases of motion blur and occlusion.

Apart from the strength of recovering missed poses from predictions, the proposed approach also enjoys other merits. First, prior approaches match poses between two *consecutive* frames, which is brittle to identify switches due to factors such as intersection of poses and fast motion. Our approach, by contrast, aggregates poses within the *same frame*, thanks to our prediction-based nature, allowing us to significantly reduce the mismatched rate. Second, as compared to state-of-the-art methods, our approach tackles pose tracking from an additional perspective, *i.e*. the motion dynamics, which complements the visual cues that are in many cases absent, resulting in gratifying final poses.

We evaluate the effectiveness of the proposed method on two widely used benchmark datasets, PoseTrack 2017 and PoseTrack 2018. Empirical evaluations showcase that our method outperforms state-of-the-art approaches by a considerably large margin on both pose estimation and tracking tasks. We also provide extensive analyses on the impact of each component in the proposed method, and demonstrate the superiority of learning pose dynamics using our method.

2. Related Work

We briefly review the following three related topics, including single-frame human pose estimation, human pose tracking, and graph neural networks.

2.1. Single-Frame Human Pose Estimation

Human pose estimation methods from single images can be generally categorized into top-down methods and bottom-up methods. Bottom-up methods [6, 27, 29, 17, 8] do not rely on human detectors. These methods first detect all the body joints and then group them to form human poses. The major challenges are robustly detecting joints in complex situations (e.g. various scales, poses and cluttered background) and correctly grouping joints from different persons particularly in crowds with heavy occlusions.

Top-down methods first detect the human bounding boxes from an image and then estimate the human pose within each bounding box. Most top-down methods adopt off-the-shelf human detectors [33, 7, 54] and focus on designing efficient human pose estimators [37, 28]. Pose estimation is confined for a single person within a small area at a fixed scale. With a reliable human detector, the top-down methods can achieve accurate human pose estimation.

2.2. Human Pose Tracking

Extending the pose estimation to video lead to the human pose tracking problem, where the human poses are estimated for each frame and associated across frames. As a result, pose tracking is often tackled together with human-location tracking [42, 43, 25, 24, 21].

Bottom-up methods [32, 18, 39] in pose tracking associated the joints spatially and temporally without detecting human bounding boxes. For example, Raaj *et al*. [32] extended the Part Affinity Field (PAF) [6] designed for single image pose estimation to include temporal modeling for pose tracking. Jin *et al*. [18] proposed ST-Embed to learn the Spatial-Temporal Embedding of joints based on the idea of Associative Embedding [27]. Both methods only model relationships of joints between two frames.

Top-down methods focus on improving single-frame pose estimation by exploiting temporal context and associating the estimated poses into human pose tracklets. In the simple baseline method [45], the estimated human poses are associated by the similarity computed based on the optical flow between consecutive frames. Detect-and-Track (DAT) [15] utilizes a 3D Mask R-CNN model to detect persons with key-points from a video clip and then associates them by comparing the locations of person detections. CombDet [40] extends a 3D network as the backbone for

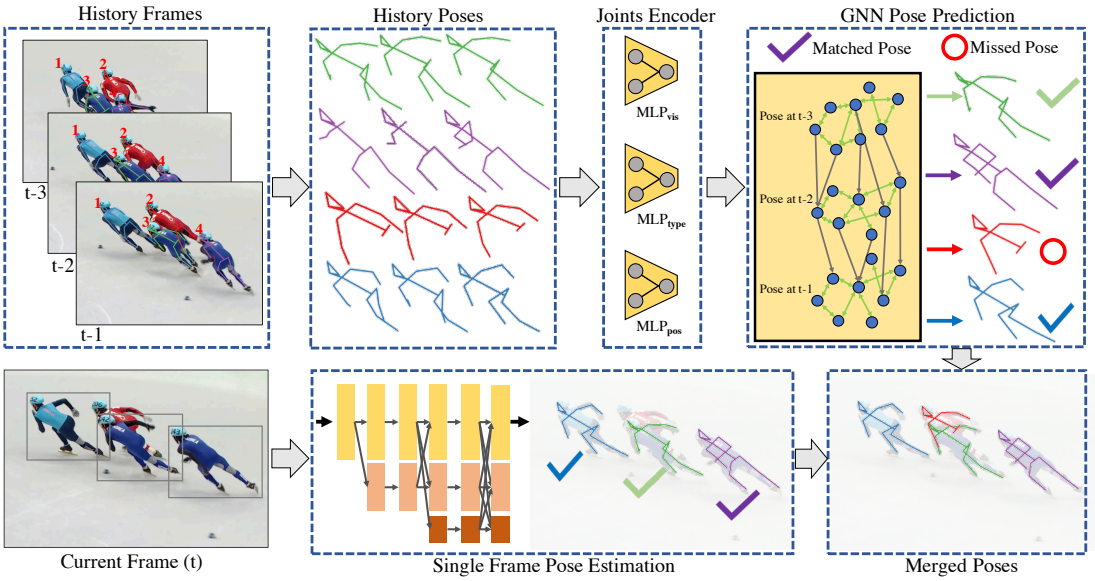

Figure 2. Overall pipeline of the proposed method. Given the history of poses and the current frame, the GNN model predicts poses for each tracklet in the history memory. The predicted poses are then matched and merged with the estimated poses to obtain the final poses in the current frame.

pose estimation to generate a tube of poses by directly propagating a bounding box to the neighboring frames. KeyTrack [36] associates the estimated human poses pose similarities. TKMRNet [53] matches human poses by learning appearance embeddings of joints and refines joints by exploiting temporal context from tracked poses.

Although some of the prior methods utilize multiple consecutive frames to help improve pose estimation and tracking, none of them explicitly model the spatial-temporal and visual dynamics of human joints. Our method models the pose tracking process with a Graph Neural Networks to learn the dynamics across frames from data.

2.3. Graph Neural Networks

Graph Neural Networks (GNNs) was first developed for graph analysis such as node classification [20] and link prediction [52]. It shows great potential in dealing with non-grid data [14, 49, 47] and has been applied to process point clouds and images [12, 44, 26, 48, 30]. For example, DGMPN [51] utilize GNN to capture the long range dependence among pixels in images to enhance the feature representation.

GNN has been used to model human poses for pose-based action recognition [23, 9, 35] and single-frame pose estimation [41, 4]. For example, DGCN [31] adopts several learnt graphs to model the relations of different joints and propagates among them to obtain the enhanced joint feature for better human pose estimation.

There are prior works that use GNNs for generic object tracking [13, 2]. Gao *et al*. [13] proposed to divide an object into several parts and learn a spatial-temporal template of the object for tracking. Bao *et al*. [2] utilized GNN in their pose tracking method to exploit human structural relations to help associate human poses across frames. This method relies on a strong human detector as well as a strong pose estimator to generate human poses for association.

In this paper, we propose a GNN-based predictor to estimate a potential configuration for each human pose tracklet frame by frame via leveraging the tracked pose history. The learnable predictor naturally models the pose tracking process and captures the dynamics of pose tracklets across video frames. Our proposed framework is capable of predicting the poses of missed human detections, which makes it robust to heavy occlusions and motion blur.

3. Method

Figure 2 shows the overall pipeline of the proposed method. For each incoming frame, two sets of poses are computed separately by the single-frame pose estimation module and the GNN-based pose prediction module. These two sets of poses are matched and merged together to generate the final human poses for the current frame. We introduce each components of the proposed method in the following sections.

3.1. Single-Frame Pose Estimation

We follow the standard pipeline of recent top-down pose trackers [45, 40, 53] to perform pose estimation for each frame. Each human detection in a frame is first cropped and rescaled to a fixed size (e.g. 384×288 when HRNet is used as the backbone of human pose estimation). The human pose estimator takes the scaled image as input and outputs a set of feature maps as well as a set of heatmaps $\mathbf{H}$. The

size of the generated heatmaps is typically smaller than the input image (e.g. 96×72 with HRNet as the backbone). The number of heatmaps is set to be the number of joints, which is 15 on PoseTrack 2017 and PoseTrack 2018 datasets. Let $\mathbf{H}_{ijk}$ be the value at the (i, j) location of the k-th heatmap. The position of the k-th joint can be computed as

$$l_k^* = \arg\max_{(i,j)} \mathbf{H}_{ijk}, \tag{1}$$

where l_k^* is the position within the heatmap and can be transformed to the position in the frame according to the center and scale information of the cropped image.

The training loss of the single-frame pose estimation model is computed against the heatmaps. A cropped human example is first scaled to a fixed size and the corresponding ground-truth joints are properly transformed to the coordinates in heatmaps. Let l_k be the ground-truth location of the k-th joint in the heatmap. The ground truth heatmap is generated following a 2D Gaussian distribution: $\mathbf{H}_{ijk}^{gt} = \exp(-\frac{||(i,j)-l_k||_2^2}{\sigma^2})$. σ is set to be 3 in all our experiments. We train the human pose estimation model by minimizing the following loss:

$$\mathcal{L}_e = \sum_i^H \sum_j^W \sum_k^K ||\mathbf{H}_{ijk}^{pred} - \mathbf{H}_{ijk}^{gt}||_2^2, \tag{2}$$

where H and W represent the height and width of heatmaps, and K is the number of joints.

3.2. Dynamics Modeling via GNN

As shown in Figure 2, given the tracked poses of the same identity from prior frames, we design a GNN-based model to explicitly capture the spatial-temporal human motion dynamics from history poses and make prediction for the subject's pose in current frame.

The GNN as a human pose dynamics model has joints of tracklets as the nodes. Edges between all pair-wise joints within-frame and between consecutive frames help capture the relative location constraints between joints as well as human motion dynamics. When applied to history tracklets as shown in the Joint Aggregation part in Figure 3, the GNN updates features on the nodes with respective to the learned dynamics. For pose prediction, each location in the current frame is considered as a node and is connected to the joints of the last pose in the tracklet. The GNN performs feature aggregation for the locations in the current frame and classifies each location by its aggregated features to determine the joint type of the location.

Let t be the total number of frames involved in the GNN. A FIFO queue is used to maintain the history poses with the same identity. We denote a human pose as $\mathbf{P}_r$, where $r \in \{1, \ldots, t\}$. $\mathbf{P}_{1,\ldots,t-1}$ are from history tracklets and $\mathbf{P}_t$ represents the predicted pose in the current frame.

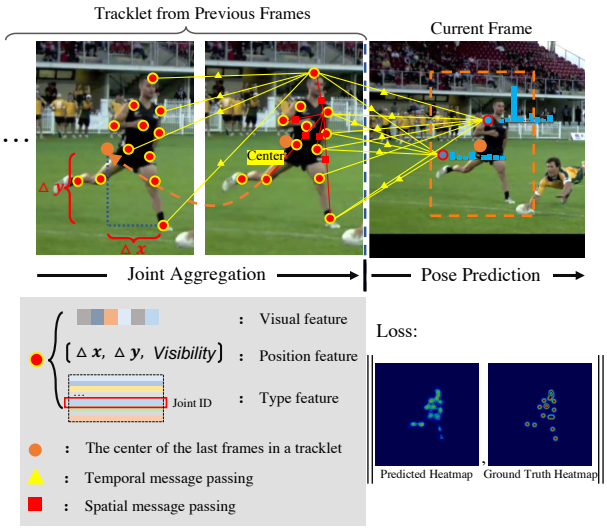

Figure 3. Illustration of our GNN model. Nodes in the tracklet are the joints of poses, while edges are the connections between joints within the same pose or across consecutive poses. During the pose prediction, we model each position in the current frame as a node and generate the heatmaps by classifying all the nodes. L_2 norm is used as the loss function to train the GNN model.

3.2.1 Nodes in the proposed GNN model

Joints of history tracklets and potential joints of the human pose in current frame are used as nodes in our GNN model. For each frame, we incorporate three kinds of cues on each joint to construct the input node feature, the visual feature from the backbone CNN of our single-frame pose estimator as v_k, the encoding of its joint type with a learnable lookup table [10] as c_k, and its 2D position and confidence score from pose estimator as p_k. For the potential joint in the current frame we set its confidence to 1. All the 2D position of joints are normalized according to the center of the last tracked pose $\mathbf{P}_{t-1}$. Normalizing joint positions with respect to the same center help capture the full body movement. Here $k \in 1, \ldots, K$ denotes the k-th joint type of a given human pose.

We use Multilayer Perceptron (MLP) to transform all the joint features to have the same dimension and merge them with average pooling, i.e. The final feature of the k-th joint is computed as follows:

$$\mathbf{J}_k = \textbf{Pooling}\big(\mathbf{MLP}_{vis}(v_k), \mathbf{MLP}_{pos}(p_k), \mathbf{MLP}_{type}(c_k)\big). \tag{3}$$

The three $\mathbf{MLP}_*$ encoders above ($\mathbf{MLP}_{vis}$, $\mathbf{MLP}_{pos}$, $\mathbf{MLP}_{type}$) for different cues do not share parameters. When constructing $\mathbf{J}_k$ for potential joints in the current frame the c_k part is ignored.

3.2.2 Edges in the proposed GNN model

The graph is constructed with two different types of edges: the connections between joints within the same frame and the connections across consecutive frames. Edges within the same frame enable the GNN to capture relative movements and spatial structure of human joints while the cross-frame edges model the temporal human pose dynamics. We use two sets of GNN parameters when aggregating features from these two types of edges.

3.2.3 Joint aggregation

In each layer of the GNN model, node features are updated via massage passing, i.e.,

$$\mathbf{J}_k^{l+1} = \mathbf{J}_k^l + \mathbf{MLP}\Big(\big[\,\mathbf{J}_k^l \,||\mathbf{M}(\,\mathbf{J}_{k',k'\in\mathcal{N}_{\mathbf{J}_k^l}}^l|\mathbf{J}_k^l)\big]\Big), \quad (4)$$

where $\mathbf{J}_k^l$ represents the feature of the k-th joint at the l-th layer. $\mathcal{N}_{\mathbf{J}_k^l}$ represents the set of neighbours of the k-th joint, $\mathbf{M}$ represents the message aggregating function that takes all the neighbours as inputs and computes the aggregated feature, and $[\cdot||\cdot]$ represents the concatenation of vectors.

We use self-attention [38] mechanism in function $\mathbf{M}$ to compute the aggregated feature. To aggregate the features from all the neighbours, the query representation of $\mathbf{J_k}$ is computed as $\mathbf{J_{kq}}$ and then each joint $\mathbf{J_{k'}}$ is first transformed to two different representations include value $\mathbf{J_{k'v}}$ and key $\mathbf{J_{k'k}}$. The final aggregated feature can be computed as the weighted average of all the values of the neighbours:

$$\begin{gathered}\mathbf{M}(\,\mathbf{J}_{k',k'\in\mathcal{N}_{\mathbf{J}_k}}|\mathbf{J}_k) = \sum_{k'\in\mathcal{N}_{\mathbf{J}_k}} \alpha_{kk'}\mathbf{J}_{k'v}, \\ \text{where } \alpha_{kk'} = \mathbf{Softmax}_{k'}(\mathbf{J}_{kq}^\top \mathbf{J}_{k'k}).\end{gathered} \quad (5)$$

$\mathbf{J}^\top$ represents the transpose of the feature vector $\mathbf{J}$ and the similarity is computed as the dot product between the query and keys. $\alpha_{kk'}$ is computed as the softmax normalization over the similarities.

The information comes from the different types of edges plays different roles: edges within the same frame model the spatial dynamics while edges across frames incorporate the temporal dynamics. We keep separated parameters for the two dynamics. Specifically, the $\mathbf{MLP}(\cdot)$ in Equation 4 is switched between two implementations from layer to layer. In the l-th layer, the implementation is set to be $\mathbf{MLP}_{spatial}(\cdot)$ working on neighbors defined by the edges within the same frame and in the next $(l+1)$-th layer, it is switched to $\mathbf{MLP}_{temporal}(\cdot)$ working on neighbors defined by edges across frames, and so on so forth. The aggregated features of joints from $\mathbf{P}_{t-1}$ are used for the pose prediction step.

3.2.4 Pose prediction

This step aims to locate the poses in current frame by the GNN model, with neither human detection nor single-frame human pose estimator. To reduce computation, we select potential joints only from a confined scope. We propagate the bounding box of the last tracked pose $\mathbf{P}_{t-1}$ to current frame and scale it up by a factor of 1.5 vertically and 2 horizontally at the same center to support fast-motion scenes, shown as the dotted orange box in Figure 3.

A graph is constructed with potential joints in the current frame and joints from $\mathbf{P}_{t-1}$. The learned GNN model is then applied to this graph to update joint features via message passing as explained above.

On top of the final features from GNN as $\mathbf{J}$, the prediction is conducted via another MLP over each potential joint in current frame, i.e.,

$$\mathbf{Prob} = \mathbf{MLP}_{pred}(\mathbf{J}), \quad (6)$$

where $\mathbf{Prob}$ denotes the probability distribution over all joint types of the input node. The predicted probability distributions of all potential joints in current frame generate the predicted heatmaps for all joints.

3.2.5 Training

As in Equation 2, we generate ground-truth heatmaps from labeled human pose and compute L_2 loss against the predicted joint heatmaps. Since the full GNN predictor is differentiable, we optimize the parameters and learn the dynamics from end to end.

3.3. Online Tracking Pipeline

In the current frame, given the poses from the GNN-based predictor and the poses from the single-frame pose estimator, we match and fuse them to obtain the final tracked human poses. In this process, the poses from the predictor and that from the estimator are complimentary to each other as the poses missed by the single-frame estimator due to occlusion and motion blur can be recovered by the predictor.

Specifically, we apply Hungarian matching to compute an one-to-one mapping between the predicted poses and the estimated poses. The similarity used in the Hungarian algorithm is the object keypoint similarity [45] computed based on the positions of the joints.

After matching, we propagate the tracking IDs from the predicted poses to the estimated poses if they are matched. A new ID is assigned to the estimated pose without a matched predicted pose, which is likely to be a newly observed one. For all the matched poses, the joint heatmaps of the two poses are first aligned according to their centers and then merged together by averaging the heatmaps. Refined poses are then decoded from the fused heatmaps.

We store the tracked results in a FIFO manner while keeping a fixed size of each tracklet. The history tracklets are then used as inputs to the GNN model for the following frame. The proposed framework is hence implemented to be an online tracker, as shown in Figure 2.

4. Experiments

4.1. Datasets

We evaluate the proposed method on two widely used datasets for human pose estimation and tracking, PoseTrack 2017 and PoseTrack 2018 [1]. These datasets contain several video sequences of articulated people that perform various actions. Specially, PoseTrack 2017 contains 250 video sequences for training and 50 video sequences for validation, PoseTrack 2018 increases the number of video sequences and contains 593 for training and 170 for validations. Both datasets are annotated with 15 joints, each of them are associated with an ID for the corresponding person. The training videos are annotated densely within the middle 30 frames of each video sequences. The validation videos are annotated every forth frame across the whole video sequences beside the densely annotation of the middle. We use the training set for training and validation set for testing, which is a common setup in previous works [53, 15].

The performance of the proposed method is evaluated from two aspects: human pose estimation and human pose tracking. We use mean Average Precision (mAP) [22, 34] to evaluate the performance of human pose estimation, and Multi Object Tracking Accuracy (MOTA) to evaluate human pose tracking. MOTA is evaluated based on three kinds of errors: missing rate, false positive rate, and switch rate. Both metrics are computed independently for each joint and then averaged across all joints. Since the evaluation of human pose tracking requires filtering the joints according to some certain thresholds, we can either evaluated the performance of human pose estimation independently or based on the filtered joints. The former one provides us an illustration of of the trade-off between human pose tracking and human pose estimation while the latter one provides us the pure performance of human pose estimation. We report both results for pose estimation.

4.2. Implementation Details

For the single-frame human pose estimation, we used HRNet [37] as the backbone. Following the training strategies of [3, 53], the HRNet is first trained on COCO dataset and then fine-tuned on PoseTrack 2017 and PoseTrack 2018 independently. For the fine-tuning process, we train the model for 20 epochs with Adam optimizer. The initial learning rate is set to be 0.0001 and reduced by a factor of 10 at the 10th and 15th epochs. We add several data augmentation strategies as used in [3], including random rotation, random flip, randomly using half of body, and random scale. Flip test is used in our work as in [40]. We adopt Faster R-CNN [33] with feature pyramid network and deformable convolutional network as the human detector [53]. The human detector is pre-trained on COCO dataset and then fine-tuned on PoseTrack 2017 and PoseTrack 2018 separately.

For the human detector, Non-Maximum Suppresion (NMS) is applied to remove duplicate detected bounding boxes which is a common operation in detection. Specifically, we use Soft-NMS [5] and set the threshold to 0.7. As articulated human pose tracking in a video often involves complex interaction and heavy person-to-person occlusions, traditional NMS in object detection that merely rely on the Intersection Over Union (IOU) of the bounding boxes is prone to fail [53]. Since we have the pose information, Pose-based Non-Maximum Suppresion (pNMS) [11] is adopted to help further remove the duplicate human poses. In pNMS, the IOU is not computed based on the bounding boxes but the weighted sum of all the joints' distances with respect to the scale of the pose. The threshold of pNMS is set to be 0.5.

For the training of the GNN pose prediction model, the fine-tuned backbone model is used to compute the visual feature of the joints. Specifically, we obtain the feature maps that are in the same resolution as the heatmaps, from all the three stages of the HRNet. The feature maps then are concatenated together and form the final feature maps with depth of 144. The visual feature of each joint can be obtained according to the joint position in the heatmap. Several data augmentation strategies are used during the GNN training process, including random rotation of the tube, random flip, random scale of the tube, and randomly selecting the gap between consecutive frames in the tube. We train the GNN model for 10 epochs with Adam optimizer. The initial learning rate is set to be 0.0001 and reduced by a factor of 10 at the 5th and 8th epochs. The length of pose history is set to be three.

4.3. Results on PoseTrack 2017

We compare our proposed method with the state-of-the-art methods in human pose estimation and human pose tracking, which are shown in Table 1, Table 2, and Table 3. In Table 2 and Table 3, the upper methods are bottom-up fashion and lower methods are top-down fashion.

Human pose estimation. In Table 1 and Table 2, we

Method	Head	Shou	Elb	Wri	Hip	Knee	Ankl	Total
PoseWarper [3]	81.4	88.3	83.9	78.0	82.4	80.5	73.6	81.2
CombDet [40]	89.4	89.7	85.5	**79.5**	82.4	80.8	76.4	83.8
Ours	**90.9**	**90.7**	**86.0**	79.2	**83.8**	**82.7**	**78.0**	**84.9**

Table 1. Comparison of state-of-the-art methods on pure human pose estimation (without filtering) on the PoseTrack 2017 validation set, where the performance is evaluated as mAP and all joints are counted.

Method	Head	Shou	Elb	Wri	Hip	Knee	Ankl	Total
BUTD [19]	79.1	77.3	69.9	58.3	66.2	63.5	54.9	67.8
RPAF [55]	83.8	84.9	76.2	64.0	72.2	64.5	56.6	72.6
ArtTrack [1]	78.7	76.2	70.4	62.3	68.1	66.7	58.4	68.7
PoseFlow [46]	66.7	73.3	68.3	61.1	67.5	67.0	61.3	66.5
STAF [32]	-	-	-	65.0	-	-	62.7	72.6
ST-Embed [18]	83.8	81.6	77.1	70.0	77.4	74.5	70.8	77.0
DAT [15]	67.5	70.2	62.0	51.7	60.7	58.7	49.8	60.6
FlowTrack [45]	81.7	83.4	80.0	72.4	75.3	74.8	67.1	76.9
TKMRNet [53]	85.3	88.2	79.5	71.6	76.9	76.9	**73.1**	79.5
Ours	**88.4**	**88.4**	**82.0**	**74.5**	**79.1**	**78.3**	**73.1**	**81.1**

Table 2. Comparison with state-of-the-art methods on human pose estimation (with filtering) on the PoseTrack 2017 Validation set, where thresholds are used to filtering low confidence joints for pose tracking. Evaluated in mAP and all joints are counted.

Method	Head	Shou	Elb	Wri	Hip	Knee	Ankl	Total
BUTD [19]	71.5	70.3	56.3	45.1	55.5	50.8	37.5	56.4
ArtTrack [1]	66.2	64.2	53.2	43.7	53.0	51.6	41.7	53.4
PoseFlow [46]	59.8	67.0	59.8	51.6	60.0	58.4	50.5	58.3
STAF [32]	-	-	-	-	-	-	-	62.7
ST-Embed [18]	78.7	79.2	71.2	61.1	**74.5**	69.7	**64.5**	71.8
DAT [15]	61.7	65.5	57.3	45.7	54.3	53.1	45.7	55.2
FlowTrack [45]	73.9	75.9	63.7	56.1	65.5	65.1	53.5	65.4
PGPT [2]	75.4	77.2	69.4	71.5	65.8	67.2	59.0	68.4
TKMRNet [53]	81.0	82.9	69.8	63.6	72.0	71.1	60.8	72.2
CombDet [40]	80.5	80.9	71.6	**63.8**	70.1	68.2	62.0	71.6
Ours	**82.0**	**83.1**	**73.4**	63.5	72.3	**71.3**	63.5	**73.4**

Table 3. Comparison of state-of-the-art methods on human pose tracking on the PoseTrack 2017 validation set. The performance is evaluated as MOTA and all joints are counted.

evaluate pure human pose estimation in videos where the estimated poses are directly evaluated without filtering, as well as the filtered human pose estimation performance in the context of pose tracking. As shown in Table 1, the proposed method achieves the best performance, outperforming the previous best method [40] by 1.1 mAP. Note that CombDet [40] utilizes a heavier 3D convolutional backbone and uses 9 frames as input. Since human pose tracking needs to firstly filter some estimated joints, the mAP result in Table 2 is lower than that in Table 1. As shown in Table 3, our method outperforms the best top-down method [53] by 1.6 mAP and the best bottom-up method [18] by 4.1 mAP.

Human pose tracking. As shown in Table 3, our method achieves state-of-the-art pose tracking performance and outperform the best top-down method [53] by 1.2 MOTA, and the best bottom-up method [18] by 1.6 MOTA.

Qualitative samples. To provide an intuitive understanding of our method, in Figure 4 we visualize some samples of the history pose tracklets, the pose estimation result in the current frame, and the final outputs of our full method. Different skeleton colors represents different person identity and the red circles in the 4th column highlight the missed or incorrect estimated joints that are corrected by the proposed GNN model.

4.4. Results on PoseTrack 2018

We show in Table 4, Table 5 and Table 6 the comparison of our proposed method and existing methods on the PoseTrack 2018 validation set. Again, our method achieves

Method	Head	Shou	Elb	Wri	Hip	Knee	Ankl	Total
PT_CPN++ [50]	82.4	88.8	86.2	79.4	72.0	80.6	76.2	80.9
KeyTrack [36]	84.1	87.2	**85.3**	79.2	77.1	80.6	76.5	81.6
CombDet [40]	84.9	87.4	84.8	79.2	77.6	79.7	75.3	81.5
Ours	**85.1**	**87.7**	**85.3**	**80.0**	**81.1**	**81.6**	**77.2**	**82.7**

Table 4. Comparison of state-of-the-art methods on pure human pose estimation (without filtering) on the validation set of PoseTrack 2018. Evaluated in mAP and all joints are counted.

Method	Head	Shou	Elb	Wri	Hip	Knee	Ankl	Total
STAF [32]	-	-	-	64.7	-	-	62.0	70.4
TML++ [16]	-	-	-	-	-	-	-	74.6
TKMRNet [53]	-	-	-	-	-	-	-	76.7
Ours	80.6	84.5	80.6	74.4	75.0	76.7	71.9	**77.9**

Table 5. Comparison of state-of-the-art methods on human pose estimation (with filtering) on the PoseTrack 2018 validation set, where thresholds are used to filtering low confidence joints for pose tracking.

Method	Head	Shou	Elb	Wri	Hip	Knee	Ankl	Total
STAF [32]	-	-	-	-	-	-	-	60.9
TML++ [16]	**76.0**	76.9	66.1	56.4	65.1	61.6	52.4	65.7
PT_CPN++ [50]	68.8	73.5	65.6	61.2	54.9	64.6	56.7	64.0
TKMRNet [53]	-	-	-	-	-	-	-	68.9
KeyTrack [36]	-	-	-	-	-	-	-	66.6
CombDet [40]	74.2	76.4	71.2	64.1	64.5	65.8	**61.9**	68.7
Ours	74.3	**77.6**	**71.4**	**64.3**	**65.6**	**66.7**	61.7	**69.2**

Table 6. Comparison of state-of-the-art methods on human pose tracking on the PoseTrack 2018 validation set. Evaluated in MOTA and all joints are counted.

the best performances in pure human pose estimation, pose estimation with filtering, and human pose tracking. Specifically, as in Table 4, the proposed method improves pure human pose estimation without filtering by 1.1 mAP over the state-of-the-art method [36]. As shown in Table 5, our method outperforms the best existing human pose estimation [53] with filtering for pose tracking by 1.2 mAP. And for human pose tracking, the proposed method also achieves the state-of-the-art performance improving the MOTA by 0.3 over [53], as shown in Table 6.

The superior performance on both PoseTrack 2017 and 2018 datasets in all three tasks (pure pose estimation in video, pose estimation with filtering, and pose tracking) validates the effectiveness of modeling dynamics by GNN.

4.5. Model Analysis

We provide here analyses on the proposed method, including ablation studies, visualization of the attentions among joints learnt from the GNN model, and sensitively analysis of the memory length and GNN model size.

Ablation study. We examine the effectiveness of the proposed method by conducting ablation experiments on several key components. As shown in Table.7, Matching w/ IOU and Matching w/ OKS means we associate the estimated poses between consecutive frames using the IOU and OKS as the similarity measure. Matching w/ GNN means we only use the predicted poses for matching measure, and the final poses are not refined by the predicted poses. Full model is our proposed model. It can be seen that using the

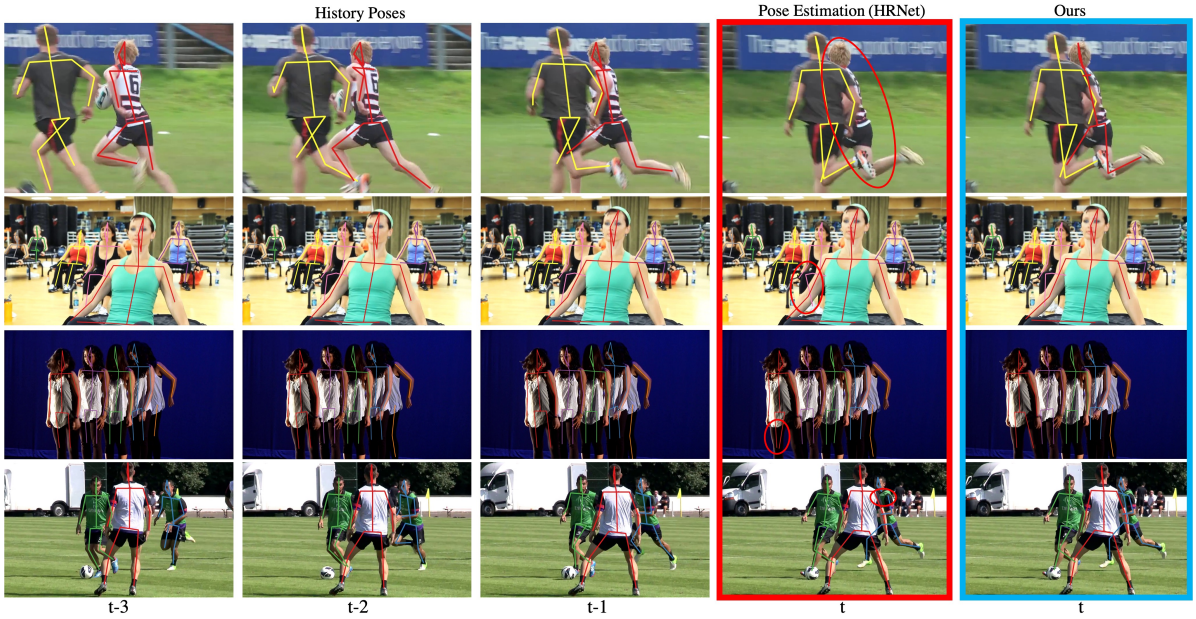

Figure 4. Qualitative examples of the proposed method on the PoseTrack 2017 validation set. The first three columns show the poses in the memory, the fourth column shows the estimated poses from HRNet, and the last column shows the final poses of our proposed method. Red dot circles highlight the incorrect or missed poses that are corrected.

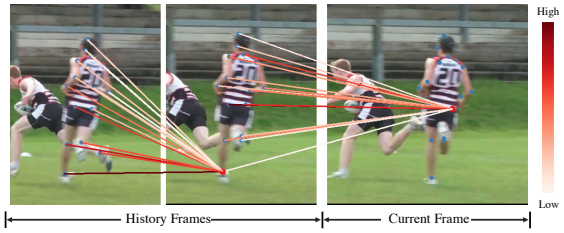

Figure 5. Visualization of the attention among different joints within the GNN model. Red nodes are the centers for aggregation and the colors of lines indicate the attention values. We zoom the current frame for a better visualization.

predicted poses for matching metric can improve the MOTA performance and reduce the switch rate over IOU and OKS metrics, the full model with pose refinement by pose merging can improve both the mAP and MOTA further more.

Visualization of GNN model. In order to provide a thorough understanding of the GNN model, we visualize in Figure 5 the computed attention weights $\alpha_{kk'}$ as computed in Equation 5. It can be observed that the hip in current frame is influenced by the hip, shoulder, and knee in the consecutive pose mostly. The ankle in the middle frame is influence mostly by the lower part of the previous pose.

Length of memory tube and model size. In Table 8 we show the results with different lengths of memory and different model size, where smaller model means the dimension of the output of $\mathbf{MLP}_*$ (as in Equation 3) is halved. It can be seen that the performance is improved when changing the memory length from two to four frames and being saturated when using more memory. Enlarging the model size improves both mAP and MOTA.

Method	mAP	MOTA	Miss (%)	Switch (%)	FP (%)
Matching w/ IOU	79.9	71.8	17.4	1.8	9.0
Matching w/ OKS	79.9	72.1	17.4	1.6	8.9
Matching w/ GNN	79.9	73.1	17.1	1.4	8.4
Full model	81.1	73.4	16.9	1.3	8.4

Table 7. Ablation studies on the PoseTrack 2017 validation set, where Miss, Switch, FP stand for the missing rate, switch rate and false positive rate (the lower the better) in MOTA.

Method	mAP	MOTA	Miss (%)	Switch (%)	FP (%)
Two frames	80.6	72.9	17.2	1.4	8.5
Four frames	81.3	73.4	16.9	1.3	8.4
Smaller model	80.8	73.2	17.1	1.3	8.4
Full model	81.1	73.4	16.9	1.3	8.4

Table 8. Influence of model capacity and length of memory.

5. Conclusion

We present in this paper a novel approach for human pose estimation and tracking. In our method, a GNN model is designed to explicitly model the dynamics of the pose tracklets and predict the corresponding poses in an incoming frame, independent of the estimations. When combining with the human pose estimation model, the proposed method takes advantages of both the visual information and the dynamics, thereby enabling the recovery of missed poses and refinement of estimated poses. Extensive experiments on PoseTrack 2017 and PoseTrack 2018 datasets validate the superiority of the proposed method in both human pose estimation and human pose tracking tasks. In our future work, we would like to explore a more flexible manner to aggregate the predicted results and the new observation, making the whole pipeline even more adaptive.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5167–5176, 2018. 4326, 4327

[2] Qian Bao, Wu Liu, Yuhao Cheng, Boyan Zhou, and Tao Mei. Pose-guided tracking-by-detection: Robust multi-person pose tracking. *IEEE Transactions on Multimedia*, 2020. 4323, 4327

[3] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning temporal pose estimation from sparsely-labeled videos. In *Advances in Neural Information Processing Systems*, pages 3027–3038, 2019. 4326

[4] Yanrui Bin, Zhao-Min Chen, Xiu-Shen Wei, Xinya Chen, Changxin Gao, and Nong Sang. Structure-aware human pose estimation with graph convolutional networks. *Pattern Recognition*, page 107410, 2020. 4323

[5] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 4326

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 4322

[7] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4974–4983, 2019. 4322

[8] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5386–5395, 2020. 4322

[9] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020. 4323

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4324

[11] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017. 4326

[12] Matthias Fey, Jan E. Lenssen, Frank Weichert, and Heinrich Müller. SplineCNN: Fast geometric deep learning with continuous B-spline kernels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4323

[13] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. Graph convolutional tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4649–4659, 2019. 4323

[14] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017. 4323

[15] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 350–359, 2018. 4322, 4326, 4327

[16] Jihye Hwang, Jieun Lee, Sungheon Park, and Nojun Kwak. Pose estimator and tracker using temporal flow maps for limbs. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 4327

[17] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016. 4322

[18] Sheng Jin, Wentao Liu, Wanli Ouyang, and Chen Qian. Multi-person articulated tracking with spatial and temporal embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5664–5673, 2019. 4321, 4322, 4327

[19] Sheng Jin, Xujie Ma, Zhipeng Han, Yue Wu, Wei Yang, Wentao Liu, Chen Qian, and Wanli Ouyang. Towards multi-person pose tracking: Bottom-up and top-down methods. In *ICCV PoseTrack Workshop*, volume 2, page 7, 2017. 4321, 4327

[20] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 4323

[21] Long Lan, Xinchao Wang, Gang Hua, Thomas S. Huang, and Dacheng Tao. Semi-online multi-people tracking by re-identification. *International Journal on Computer Vision*, 128:1937–1955, 2020. 4322

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4326

[23] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020. 4323

[24] Andrii Maksai, Xinchao Wang, Francois Fleuret, and Pascal Fua. Non-markovian globally consistent multi-object tracking. In *International Conference on Computer Vision*, 2017. 4322

[25] Andrii Maksai, Xinchao Wang, and Pascal Fua. What players do with the ball: A physically constrained interaction modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 4322

[26] Gonzalo Mena, David Belanger, Gonzalo Munoz, and Jasper Snoek. Sinkhorn networks: Using optimal transport techniques to learn permutations. In *NIPS Workshop in Optimal Transport and Machine Learning*, 2017. 4323

[27] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in neural information processing systems*, pages 2277–2287, 2017. 4322

[28] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015. 4322

[29] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016. 4322

[30] Jiayan Qiu, Yiding Yang, Xinchao Wang, and Dacheng Tao. Hallucinating visual instances in total absentia. In *European Conference on Computer Vision*, 2020. 4323

[31] Zhongwei Qiu, Kai Qiu, Jianlong Fu, and Dongmei Fu. Dgcn: Dynamic graph convolutional network for efficient multi-person pose estimation. In *AAAI*, pages 11924–11931, 2020. 4323

[32] Yaadhav Raaj, Haroon Idrees, Gines Hidalgo, and Yaser Sheikh. 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*, pages 4620–4628, 2019. 4321, 4322, 4327

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 4322, 4326

[34] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 369–378, 2017. 4326

[35] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1227–1236, 2019. 4323

[36] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 keypoints is all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2020. 4323, 4327

[37] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 4321, 4322, 4326

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4325

[39] Jue Wang, Shaoli Huang, Xinchao Wang, and Dacheng Tao. Not all parts are created equal: 3d pose estimation by modelling bi-directional dependencies of body parts. In *International Conference on Computer Vision*, 2019. 4322

[40] Manchen Wang, Joseph Tighe, and Davide Modolo. Combining detection and tracking for human pose estimation in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11088–11096, 2020. 4321, 4322, 4323, 4326, 4327

[41] Rui Wang, Chenyang Huang, and Xiangyang Wang. Global relation reasoning graph convolutional networks for human pose estimation. *IEEE Access*, 8:38472–38480, 2020. 4323

[42] Xinchao Wang, Engin Turetken, Francois Fleuret, and Pascal Fua. Tracking interacting objects optimally using integer programming. In *European Conference on Computer Vision*, 2014. 4322

[43] Xinchao Wang, Engin Turetken, Francois Fleuret, and Pascal Fua. Tracking interacting objects using intertwined flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:2312–2326, 2016. 4322

[44] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019. 4323

[45] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 4321, 4322, 4323, 4325, 4327

[46] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose flow: Efficient online pose tracking. *arXiv preprint arXiv:1802.00977*, 2018. 4327

[47] Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. Factorizable graph convolutional networks. *Advances in Neural Information Processing Systems*, 33, 2020. 4323

[48] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7074–7083, 2020. 4323

[49] Yiding Yang, Xinchao Wang, Mingli Song, Junsong Yuan, and Dacheng Tao. Spagan: Shortest path graph attention network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4099–4105. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 4323

[50] Dongdong Yu, Kai Su, Jia Sun, and Changhu Wang. Multi-person pose estimation for pose tracking with enhanced cascaded pyramid network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 4327

[51] Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. Dynamic graph message passing networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3726–3735, 2020. 4323

[52] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, pages 5165–5175, 2018. 4323

[53] Chunluan Zhou, Zhou Ren, and Gang Hua. Temporal keypoint matching and refinement network for pose estimation and tracking. 4321, 4322, 4323, 4326, 4327

[54] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 4322

[55] Xiangyu Zhu, Yingying Jiang, and Zhenbo Luo. Multi-person pose estimation for posetrack with enhanced part affinity fields. In *ICCV PoseTrack Workshop*, volume 7, 2017. 4321, 4327