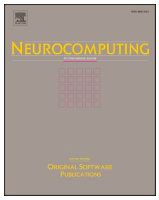
Neurocomputing 655 (2025) 131309

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Survey paper

Real-time human pose estimation and tracking on monocular videos: A systematic literature review

Yingying Chen [a,*], Zhenan Feng [a], Daniel Paes [a], Daniel Nilsson [b], Ruggiero Lovreglio [a]

[a] School of Built Environment, Massey University, Auckland 0632, New Zealand
[b] Department of Civil and Environmental Engineering, University of Canterbury, Christchurch 8041, New Zealand

ARTICLE INFO

Communicated by Zidong Wang

Keywords:
Real-time
Human pose estimation
Human pose tracking
Deep learning
2D and 3D pose
Monocular optical videos

ABSTRACT

Real-time human pose estimation and tracking on monocular videos is a fundamental task in computer vision with a wide range of applications. Recently, benefiting from deep learning-based methods, it has received impressive progress in performance. Although some works have reviewed and summarised the advancements in this field, few have specifically focused on real-time performance and monocular video-based solutions. The goal of this review is to bridge this gap by providing a comprehensive understanding of real-time monocular video-based human pose estimation and tracking, encompassing both 2D and 3D domains, as well as single-person and multi-person scenarios. To achieve this objective, this paper systematically reviews 68 papers published between 2014 and 2024 to answer six research questions. This review brings new insights into computational efficiency measures and hardware configurations of existing methods. Additionally, this review provides a deep discussion on trade-off strategies for accuracy and efficiency in real-time systems. Finally, this review highlights promising directions for future research and provides practical solutions for real-world applications.

1. Introduction

Human pose estimation and tracking are emerging fields in computer vision. Human pose estimation (HPE) involves determining the positions of different body joints from images or videos and outputting them into 2D or 3D coordinates. Human pose tracking (HPT) builds on the outputs of HPE by assigning a unique identification number to each person in the videos (refer to Section 2 for a comprehensive explanation). The two major approaches for HPE are traditional computer vision methods and deep learning methods. The traditional methods employ various hand-crafted feature extraction techniques, such as pictorial structure [3] and shape contexts [58] for body joint detections. However, they fail to address occlusion and noise issues as well as are time-consuming and challenging when handling more than one pose [69]. Since 2014, interest in HPE has increased due to the introduction of deep learning methods [78]. The impressive performance made the field shift their work from traditional models to deep learning models, ranging from simple neural networks to complex convolutional neural networks and, more recently, transformers-based architectures [43,57,90].

In human pose estimation and tracking, particularly in 3D HPE and HPT, diverse types of input data have been explored. Multi-view camera systems provide spatially diverse viewpoints, enabling precise 3D pose reconstruction through triangulation [8] or direct regression [82], but they require careful calibration and expensive setups. Depth sensors (e.g., Kinect) offer explicit distance information, alleviating depth ambiguity in 3D pose estimation [64]. However, they suffer from short-range constraints and reduced accuracy in bright or outdoor scenes. Wearable inertial measurement units (IMUs) can track body movement robustly without visual occlusion constraints [26]. Still, they are intrusive, require attachment to the body, and are not scalable for large populations or unstructured environments. In contrast, monocular video input, such as RGB footage from a single optical camera, offers a cost-effective, easily deployable, and non-intrusive solution. However, it introduces significant challenges associated with depth ambiguity [100], dynamic backgrounds, truncation or occlusion caused by dynamic interaction among people, motion blur from the fast motion of people or camera, changing lighting conditions, and pose variation [88]. Moreover, compared to image-based pose estimation, video-based pose estimation must consider temporal relationships of body poses across frames to alleviate pose inconsistency and motion jitter [51], which

* Corresponding author.
E-mail addresses: y.chen6@massey.ac.nz (Y. Chen), z.feng1@massey.ac.nz (Z. Feng), d.paes@massey.ac.nz (D. Paes), daniel.nilsson@canterbury.ac.nz (D. Nilsson), r.lovreglio@massey.ac.nz (R. Lovreglio).

https://doi.org/10.1016/j.neucom.2025.131309
Received 19 April 2025; Received in revised form 5 August 2025; Accepted 16 August 2025
Available online 23 August 2025

significantly introduces additional computational cost due to temporal redundancy and increases the complexity of spatio-temporal modelling. HPT faces additional challenges in maintaining consistent identities across frames, particularly under occlusion and fast movement [102]. These limitations make monocular video-based HPE and HPT a technically challenging yet practically important subfield.

These challenges are further amplified in real-time monocular HPE and HPT, where low-latency requirements and computational constraints further complicate the tasks [89]. Therefore, achieving a balance between accuracy and efficiency is essential for real-time video-based pose estimation and tracking [31]. Despite these difficulties, the demand for real-time and robust solutions has been growing. Real-time monocular HPE and HPT have been increasingly adapted in a wide range of practical applications, such as surveillance [14], animation [38], action recognition [34], immersive environments [52], and human-computer interaction research [68]. Recently, it has been applied in identifying violence and supporting healthcare [73] and it could be a key technology to reduce crowd accidents [28].

Several surveys and reviews have been conducted on HPE and HPT with various scopes and application focuses [5,15,51,80,100,102]. For instance, [80] exclusively took into account 2D HPE in their review, while [51,100] provided comprehensive reviews on HPE but did not involve HPT. The work by [15] focused on the contribution of HPE for training assistance. The aim of [5] was to explore the applications of HPE in sports and physical exercise. Although [102] attempted to cover both 2D and 3D HPE and HPT, it did not focus on the challenges of real-time inference from monocular video input, nor did it analyse the efficiency measures (e.g., inference speed) or hardware constraints, which are critical for deployment on mid-range devices. While existing review studies cover various aspects of 2D and 3D HPE, HPT, or specific application domains, few systematically focus on real-time human pose estimation and tracking using monocular video input, particularly from the perspective of computational efficiency and hardware deployment. This gap has also led to a limited understanding of how to balance accuracy and efficiency to achieve real-time performance on mid-range devices. To address these gaps, a tailored and thorough review of recent advancements in real-time human pose estimation and tracking from monocular videos is urgently needed.

This research aims to analyse various aspects that influence the performance of real-time human pose estimation and tracking on monocular videos, with a particular focus on efficiency metrics and inference hardware configurations. It further explores the underlying reasons for variation in computational efficiency, and highlights strategies introduced by recent works to balance accuracy and efficiency for real-time performance. To achieve this aim, we conducted a systematic literature review with 68 papers published between 2014 and 2024. These works were identified and analysed following the PRISMA guidelines [57] by answering six fundamental questions to evaluate the existing capabilities and limitations of real-time HPE and HPT for monocular videos. This research provides a comprehensive overview of recent advancements by evaluating the performance and complexity of different algorithms in relation to outputs and workflows, highlighting effective accuracy-efficiency trade-off strategies tailored to specific architectures and tasks. Findings could facilitate future development and deployment of robust real-time human pose estimators on resource-constrained hardware such as mid-range GPUs, CPUs, and even mobile devices.

The rest of this paper is organized as follows. Section 2 introduces background information on human pose estimation and tracking from monocular videos. Section 3 describes the methodology of our systematic review and introduces six research questions. Section 4 presents the results in response to these research questions, covering outputs, workflows, algorithms, and evaluation metrics for accuracy and efficiency, as well as the datasets and hardware configurations employed. This section also highlights common trade-off strategies between accuracy and efficiency adopted in the reviewed studies. Section 5 provides an insightful analysis of the strengths and limitations of different algorithms, and recommends practical strategies for achieving a balance between accuracy and efficiency. Furthermore, Section 5.2 discusses the limitations of this review and future research directions. Finally, Section 6 concludes the paper.

2. Background

Human pose estimation (HPE) and human pose tracking (HPT) are both fundamental computer vision tasks and have been intensively studied in recent years. HPE involves the detection of body joint positions while HPT aims to generate consistent human pose trajectories over time. The two tasks are intricately interconnected as pose tracking benefits from reliable body poses. Fig. 1 shows the difference and relationship between HPE and HPT using a multi-person example. This study emphasises the literature that tackles both tasks in a joint way within monocular video sequences. In this section, we provide the definitions and categories of these two tasks as well as the related challenges and applications.

2.1. Monocular video-based human pose estimation

Monocular video-based HPE is a complex computer vision task. It detects body joints, such as the head, knees, wrists and elbows, from monocular videos, and correctly connects them to form a skeleton structure or a body mesh. According to the spatial dimension of output, it can be categorized into two types: 2D HPE and 3D HPE. 2D HPE [25] estimates the x and y coordinates of body joints. Depending on the number of people in the videos, it can be further classified into single-person and multi-person HPE [7,10]. Similarly, 3D HPE [54] expands spatial dimensions to the x, y, and z coordinates for a more accurate representation of keypoints, where z represents depth information. Unlike 2D HPE, 3D HPE can output either relative coordinates (relative to the body root) [65] or absolute coordinates (relative to the camera) [56]. 3D HPE can be integrated with parametric body models such as the Skinned Multi-Person Linear Model (SMPL) to generate detailed 3D body meshes that include human body shapes and pose parameters [36].

Despite recent advancements, inferring human poses from monocular videos is still challenging, especially in multi-person scenarios [11]. First, the number of people is unknown in videos and people may be present at different positions and scales within a video. Second, interactions between people may cause occlusions, increasing the difficulty of pose detection. Third, runtime increases with an increased number of people, leading to the slowdown of algorithms dramatically. Beyond these challenges, video-based HPE needs to address motion jitter and temporal incoherence across adjacent frames [89]. Furthermore, 3D HPE on monocular videos introduces an additional challenge known as depth ambiguity due to the inherent difficulty of accurately estimating depth information (z-coordinates) from a single camera viewpoint [101].

2.2. Monocular video-based human pose tracking

Monocular video-based HPT is an extension of pose estimation in monocular videos, which first estimates human poses and then assigns a unique identification number (ID) to each pose across frames [87]. The main task of HPT is linking poses inter-frame poses. In other words, HPT compares pose similarity in adjacent frames by utilizing spatial-temporal relationships between a pair of poses. Depending on the number of people involved, HPT can be classified into single-person and multi-person tracking. However, little research is related to single-person HPT as it only involves tracking one person without the need to handle ID changes and merging. The main challenge of HPT is assigning correct IDs to different people and keeping consistency across frames. Therefore, HPT primarily focuses on the scene of multi-person

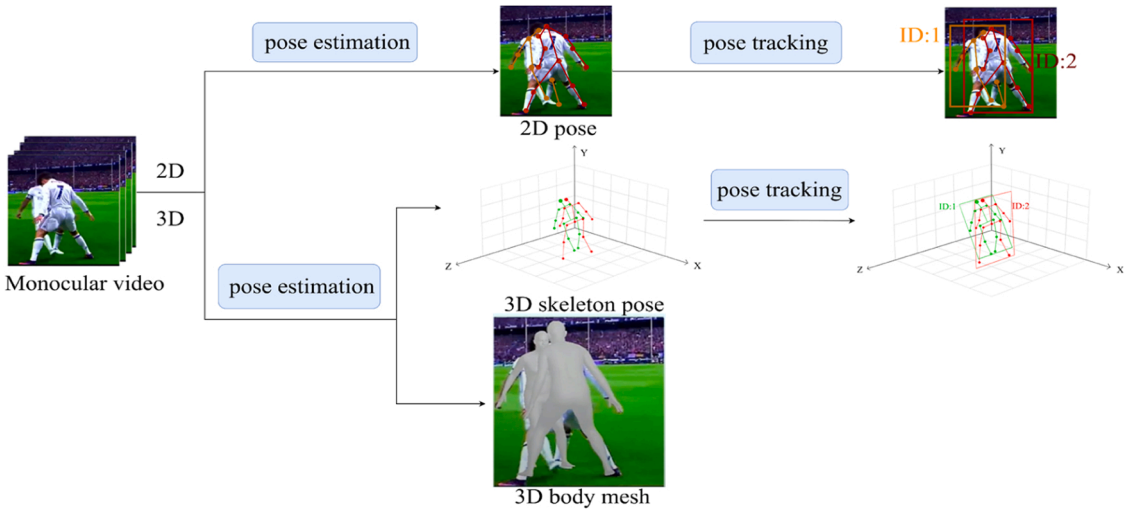


Fig. 1. The overview of multi-person video-based 2D and 3D pose estimation and tracking. The 3D HPE includes both skeleton pose estimation and body mesh recovery.

tracking [102]. The objective of multi-person HPT is to keep IDs consistent for same person across frames in the following possible scenarios: (1) some people disappear from the camera view or get occluded; (2) new people come in or previous people re-appear; (3) people walk across each other (i.e., two IDs may merge into one if not treated carefully); and (4) tracking fails due to fast camera shifting or zooming.

Multi-person HPT from monocular videos can be approached using two workflows: top-down and bottom-up. The top-down workflow [88] first detects human bounding boxes and locates body joints within these boxes, then tracks human poses over the entire video based on pose similarity. In contrast, the bottom-up workflow [67] begins with generating all body keypoints in each frame and constructing a spatio-temporal graph to represent their connections over time. Subsequently, the graph is partitioned into sub-graphs by resolving an integer linear problem, with each sub-graph corresponding to the pose trajectory of each person.

With the increasing availability of low-cost video acquisition equipment (e.g. smartphones) and powerful graphic processing devices, human pose estimation and tracking have attracted growing interest over the past decade. Despite the high performance achieved by recent neural networks-based approaches, challenges such as occlusion, depth ambiguity, missing tracking, and high latency still remain. Especially, in real-world applications, boosting efficiency without sacrificing accuracy is a critical problem. Given that, there is a need to understand how to effectively balance the accuracy and efficiency for designing and deploying robust and efficient models on mid-range devices.

3. Methodology

This study follows four of the five steps for conducting a systematic literature review outlined by [35]. These steps include the definition of research questions, the identification of relevant studies, the summary of the evidence, and the interpretation of the findings. It is worth noting that this review aims to identify and synthesise relevant and representative studies on real-time human pose estimation and tracking. Rather than assessing the quality of each study in depth, we focused on including works that meet predefined criteria and align with our research aim. Therefore, the quality assessment step outlined in [35] was omitted. This methodology section outlines the specific systematic steps followed in this review.

3.1. Research questions

This review aims to explore the factors influencing the real-time performance of monocular HPE and HPT models, with particular attention to computational efficiency and trade-offs between accuracy and efficiency. It further provides practical solutions for real-world applications on power-constrained devices, and highlights recent advancements and theoretical contributions that could guide future development and deployment. To achieve this aim, six research questions (RQ) were formulated as follows:

RQ 1. : What are the outputs of existing HPE algorithms?

RQ 2. : What workflows are used to estimate human poses?

RQ 3. : What algorithms are used for estimating and tracking human poses, and how do they operate?

RQ 4. : What are the measures to evaluate the accuracy of these algorithms?

RQ 5. : What hardware is used to implement these algorithms, and what are the measures to evaluate output efficiency?

RQ 6. : What strategies have been used to balance accuracy and efficiency for real-time applications?

3.2. Search strategy

We followed the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) workflow to identify eligible studies [63]. The studies included in this systematic literature review were sourced from Scopus, one of the largest scientific databases. To ensure completeness, a second searching approach, snowballing (both forward and backward) as described by [85], was employed to identify any potentially missing studies. We used the following keywords to collect relevant studies: (human OR pedestrian OR body OR person OR people) AND pose AND (detection OR estimation OR tracking) AND (real-time OR realtime OR online OR on-line) AND (video OR footage). Here, on-line indicates continuous processing of incoming data without batch delays, while real-time emphasizes producing results within strict time constraints. Their combination can ensure that robust and low-latency solutions for real-time pose estimation and tracking are searched comprehensively.

The first study employing a deep learning-based model for human pose estimation dates back to 2014 [78]. Therefore, the search spanned

the last ten years, from 2014 to 2024. Only English-language studies were considered. The following exclusion criteria (EC) were used to identify the relevant studies:

EC1: The study is a survey, literature review or conference introduction.

EC2: The study focuses solely on estimating specific body parts, such as the hand, foot or head.

EC3: The input videos for pose estimation are not optical and monocular.

EC4: The study targets specific human groups (e.g. patients, elderly or athletes) by training and testing their models on specialized datasets.

EC5: The study provides no evidence of the efficiency of the proposed method.

EC6: The study does not propose a new algorithm or model for pose estimation.

The titles and abstracts of each searched study were initially screened based on EC1, EC2, EC3 and EC4 to exclude clearly irrelevant papers. In the subsequent eligibility stage, the remaining studies were excluded by skimming the full text using EC5 and EC6. Additionally, we adopted the snowballing method to examine studies cited in the result evaluation sections. If a study was cited for performance comparison, it was deemed relevant and included for further evaluation. The studies searched by snowballing were evaluated and excluded based on EC5 and EC6 through a full-text review.

3.3. Search results

The initial search yielded 858 papers. Forty-three duplicate records were removed, leaving 815 papers for further filtering. The titles and abstracts of each paper were screened based on EC1, EC2, EC3 and EC4, resulting in the removal of 620 papers. In the subsequent eligibility stage, the other 151 papers were excluded by skimming the full text using EC5 and EC6. The remaining 43 papers were eligible for this review. Additionally, we adopted the snowballing method to select additional studies cited in the 43 initially included papers. These papers were evaluated based on EC5 and EC6 through a full-text review. As a result, 25 additional papers were added, expanding the total number of eligible papers to 68. Fig. 2 illustrates the complete selection process.

The eligible papers were then analysed to address the research questions framed in Section 3.1. Each paper was reviewed in detail to extract relevant data, such as outputs, workflows, algorithms, evaluation metrics, hardware configurations and datasets. The extracted data were subsequently analysed and organized to generate visualization results in the form of figures and tables, which support more effective comparisons between the reviewed studies.

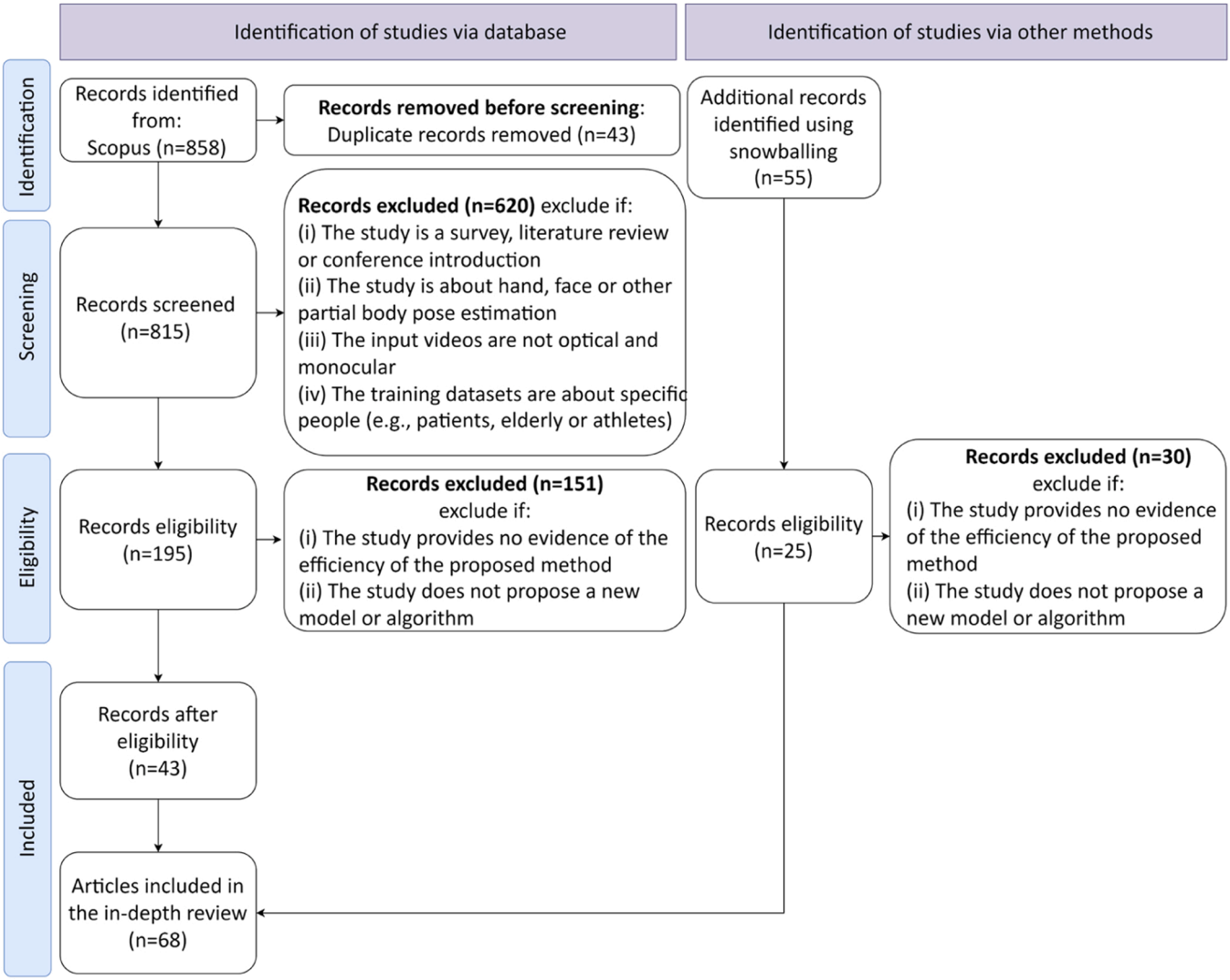


Fig. 2. Systematic literature selection process resulting in 68 papers for final in-depth review.

4. Critical analysis of selected studies

This section presents the findings extracted from the eligible papers in response to the six research questions outlined in Section 3.1. The results for RQ1 are detailed in Section 4.1, while the findings of RQ2 are presented in Section 4.2. Section 4.3 introduces the representative algorithms according to the above categories and workflows. Section 4.4 highlights the commonly used datasets and the measures for accuracy (RQ4). Section 4.5 provides efficiency measures and hardware configurations for testing these algorithms (RQ5). Lastly, we discuss the RQ6 in Section 4.6.

4.1. Outputs

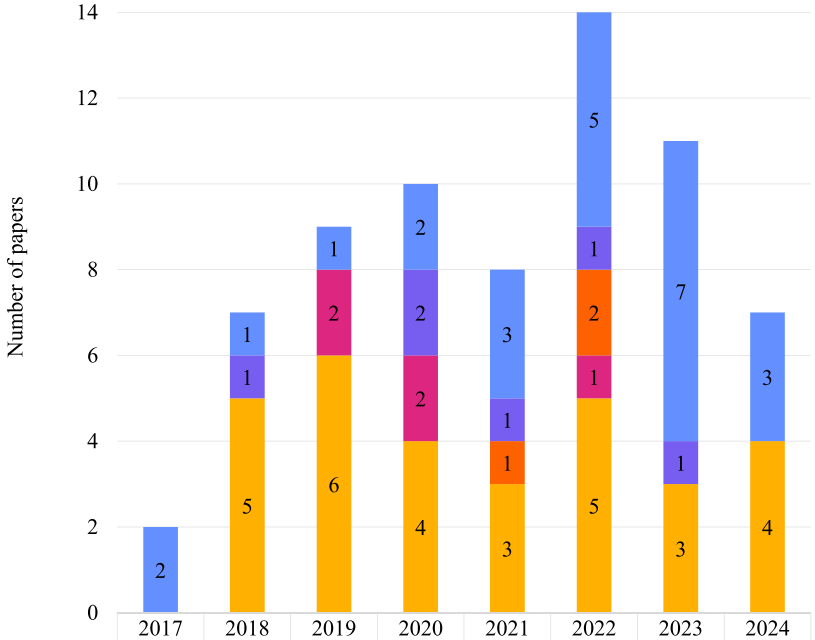
This section provides the outputs of the solutions proposed in the 68 selected works answering RQ1. The reviewed studies covered a range of outputs, which can be identified into five types: 2D-single pose, 2D-multi pose, 3D-single pose, 3D-multi pose and 2D and 3D-single pose. Given a monocular video, the outputs of HPE can be represented in either 2D or 3D forms in terms of space dimension. According to the number of people in the video, the outputs can be further divided into single-pose and multi-pose. Fig. 3 illustrates the distribution of different output types, including three studies that simultaneously generate both 2D and 3D poses, but only for single-person scenes.

In the field of 2D HPE, research on multi-person pose estimation (30 papers) significantly exceeds that on single-person pose estimation (5 papers). In contrast, 3D HPE shows an opposite trend, with single-person pose estimation (24 papers) being more prevalent than multi-person pose estimation (6 papers), highlighting a distinct shift in research between 2D and 3D HPE. Additionally, the growing interest in 3D HPE overtime may be attributed to the advancements in 2D HPE, which have encouraged researchers to explore the more complex and challenging 3D domain. As shown in Fig. 3, the x-axis starts in 2017, despite the reviewed literature spanning back to 2014, which indicates that before 2017, most works primarily focused on improving accuracy rather than real-time efficiency. Notably, seven papers [25,60,67,74,87,88,94] on pose tracking are contained in the 2D-multi pose category, as they build upon pose estimation as a foundational task. Moreover, the reviewed studies only involve 2D multi-pose tracking, with no inclusion of 3D pose tracking. Details of the algorithms are provided in Section 4.3.4.

4.2. Workflows

In this section, we discuss the different workflows for 2D and 3D HPE, as well as their associated advantages, limitations and challenges, thus answering RQ2. 2D HPE can generally be enforced in two workflows: top-down (20 papers) and bottom-up (15 papers). Fig. 4 provides an overview of these two workflows. The top-down workflow employs a detector to obtain the bounding box of each person and then adopts a single-person pose estimator on each bounding box for pose detection. It is more accurate and suffers less from scale variance, as each bounding box is rescaled to a fixed size, resulting in less degradation at smaller scales [94]. However, it also has several limitations. If the detector misses the person, pose estimation will fail. Furthermore, a separate pose estimator is required for each detected person, which leads to the runtime cost scales with the number of detected individuals [25]. The top-down workflow can be further divided into the two-stage approach (17 papers), where person detection and pose estimation are performed sequentially, and the unified approach (3 papers), which crops human bounding box and predicts keypoints from the corresponding feature maps simultaneously.

The bottom-up workflow detects all keypoints at first and then assigns them to individuals. It is more computationally efficient as it avoids repeatedly adopting a pose estimator for each detected person, thereby decoupling runtime cost from the number of people [61]. However, it may perform vulnerably in crowded scenes due to



	2017	2018	2019	2020	2021	2022	2023	2024
3D-single pose (24)	2	1	1	2	3	5	7	3
3D-multi pose (6)	0	1	0	2	1	1	1	0
2D and 3D-single pose (3)	0	0	0	0	1	2	0	0
2D-single pose (5)	0	0	2	2	0	1	0	0
2D-multi pose (30)	0	5	6	4	3	5	3	4

Fig. 3. The outputs of human pose estimation, categorized by spatial dimension and the number of people in the scene. The x-axis starts from 2017 to reflect the emerging focus on real-time efficiency, while the y-axis indicates the number of studies.

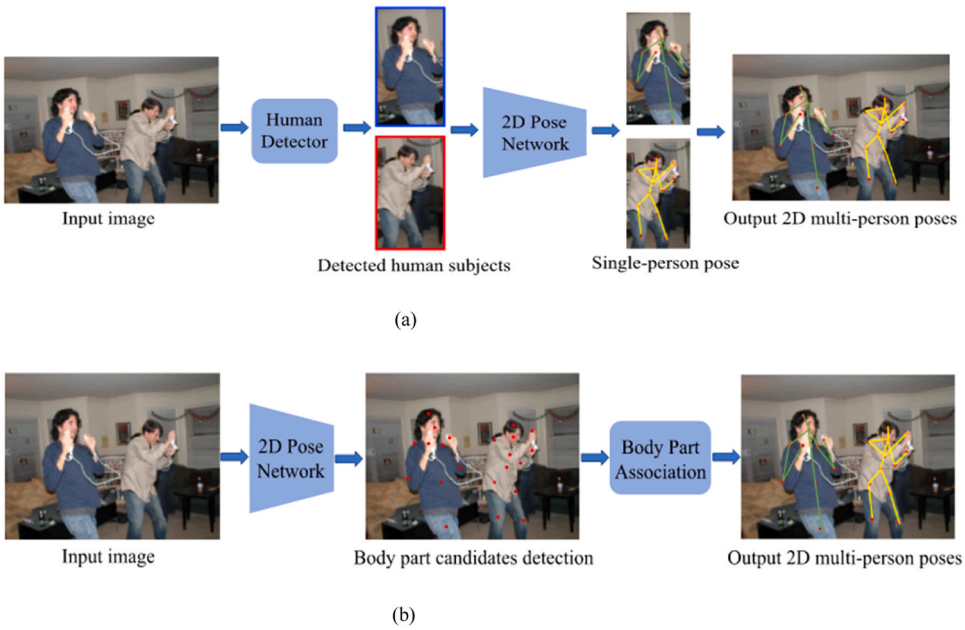


Fig. 4. Workflows of 2D multi-person HPE [100]. (a) Top-down workflow has two sub-tasks: human detection and pose estimation on each bounding box; (b) Bottom-up workflow also has two sub-tasks: detect all keypoints and assign them to individuals to form complete pose representations.

ambiguity in keypoint assignment, as it lacks instance-level global context and structural information from other body parts and people [88]. Additionally, it requires high-resolution inputs to accurately detect keypoints across different person sizes, making them sensitive to input resolution and scale variance [83]. Finally, The keypoint assignment process is an NP-hard integer linear problem, which can remarkably increase processing time [10]. According to Fig. 6, the top-down workflow is more commonly used for 2D HPE. This trend can be attributed to advancements in person detectors, which allow for decomposing the complex HPE task into two simpler and well-established subtasks: person detection and pose estimation.

3D HPE is inherently more challenging due to the high-dimensional variability and nonlinearity of human dynamics. Common challenges to face are depth ambiguities, occlusions, and the large variety of appearances and scenes [56]. 3D HPE can be executed by two workflows: end-to-end (12 papers) and 2D-to-3D (18 papers). As shown in Fig. 5, the end-to-end workflow directly infers 3D poses from RGB images or videos without the help of 2D pose representation. It requires large amounts of 3D pose-labelled datasets for supervision, which are difficult to obtain in the real world. In addition, extracting context information from images or videos results in substantial time and computation consumption. Recent research has integrated 3D body models (e.g. SMPL) to predict 3D body meshes [77]. Notably, three 2D and 3D HPE papers all apply end-to-end workflow to produce human poses [93].

2D-to-3D workflow reasons 3D human poses from estimated 2D poses. It divides a difficult problem into two decoupled subtasks: 2D pose detection and 3D pose estimation from 2D keypoints. Compared with raw monocular videos, 2D skeleton poses of each frame are much more memory-friendly, making it consume less computation and memory to model long-term frames [99]. Moreover, it also benefits from intermediate 2D pose supervision and advanced 2D pose estimators [65], which can interpret the different output trends between 2D and 3D domains. When 2D single-pose estimation achieved high-performance, researchers have shifted their attention to more complex tasks like 2D

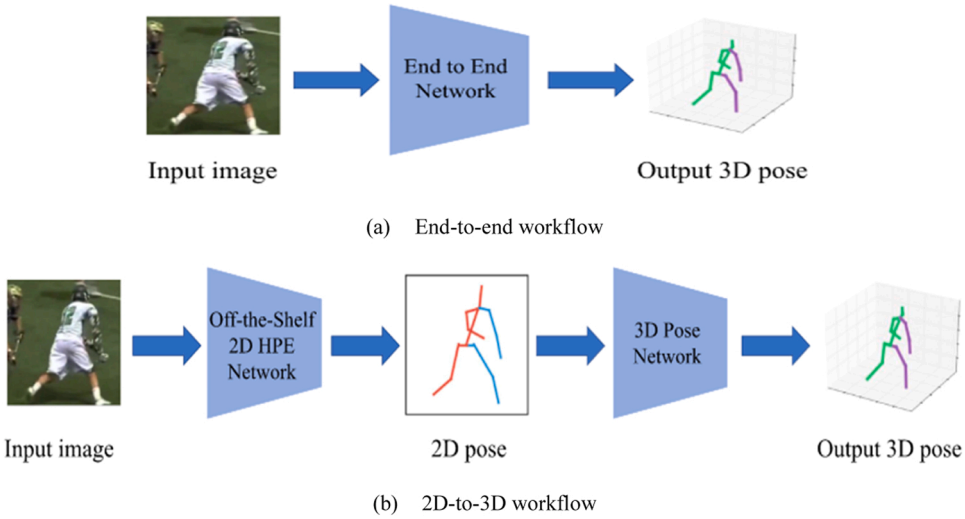


Fig. 5. Workflows of 3D single-person HPE [100]. (a) End-to-end workflow directly estimates 3D human pose from images; (b) 2D-to-3D workflow estimates 3D pose from intermediate 2D pose.

multi-person scenarios and 3D single-pose estimation. Despite their impressive performance, this workflow remains some limitation. First, if the 2D poses are not accurate will cause accumulation error in 3D pose estimating. Second, because of depth ambiguity and occlusion, a single 2D keypoint coordinate cannot uniquely determine its 3D position, leading to multiple potential 3D positions along the camera's line of sight. It can be seen from Fig. 6 that the 2D-to-3D workflow is more common than the end-to-end workflow, which is reasonable due to the advancements in 2D HPE, the wide availability of 2D pose datasets, and the inherent advantages of decoupling a complex task into multiple simple and accessible subtasks.

4.3. Algorithms

In this section, we organize the algorithms into four main categories to answer RQ3: 2D HPE, 3D HPE, 2D and 3D HPE, and HPT. Fig. 7 provides a summary of the algorithms used in HPE, linking them with the outputs they generate. In 2D HPE, Convolutional Neural Networks (CNNs) are the most commonly used method because the input data of 2D HPE are always images or videos, and CNNs are very effective for extracting features from such data [87]. In 3D HPE, the inputs can either be 2D keypoints or images. When using 2D keypoints as input, Transformers and Temporal Convolutional Networks (TCNs) are effective at modelling spatio-temporal relationships across frames to eliminate depth ambiguity and occlusion. Besides this, in order to improve efficiency without sacrificing accuracy, a combination of different types of networks is frequently employed in 3D HPE to utilize their respective advantages. In terms of HPT, an overview of the algorithms is presented in Table 1, which can be found in Section 4.3.4.

The following sections provide a detailed description of these algorithms. The 2D HPE algorithms are further discussed in Section 4.3.1 by top-down and bottom-up workflow. Similarly, the 3D HPE algorithms are presented in Section 4.3.2 through two subsections: end-to-end and 2D-to-3D workflow. Finally, Section 4.3.3 and Section 4.3.4 present the algorithms for 2D and 3D HPE, and HPT respectively.

4.3.1. 2D Human pose estimation

For 2D HPE, 35 papers employ either the top-down (20 papers) or bottom-up workflow (15 papers) to detect single or multiple poses. The specific algorithms are detailed below.

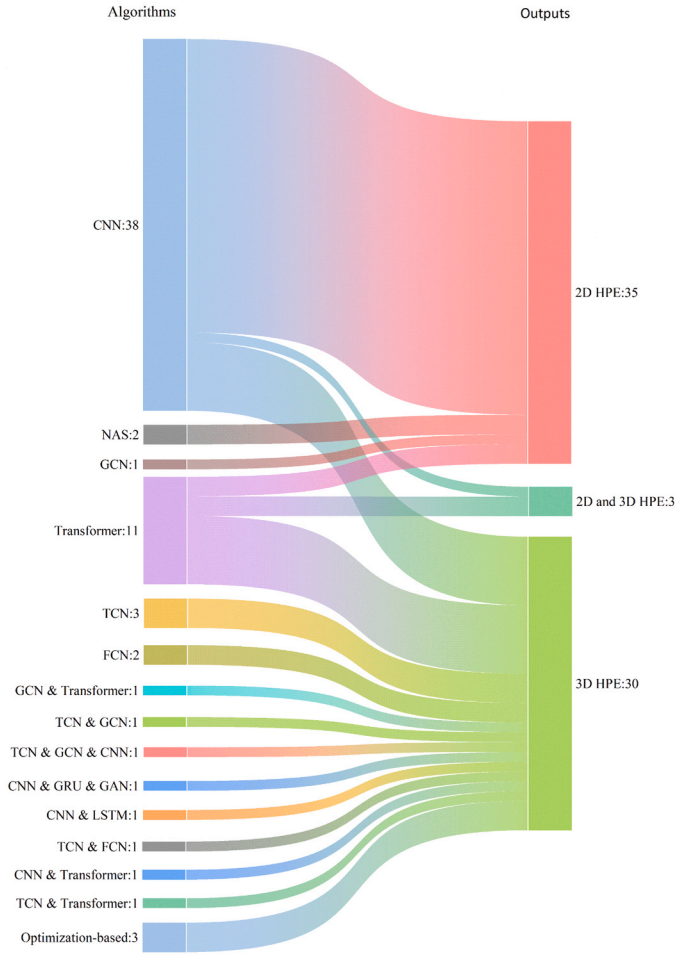
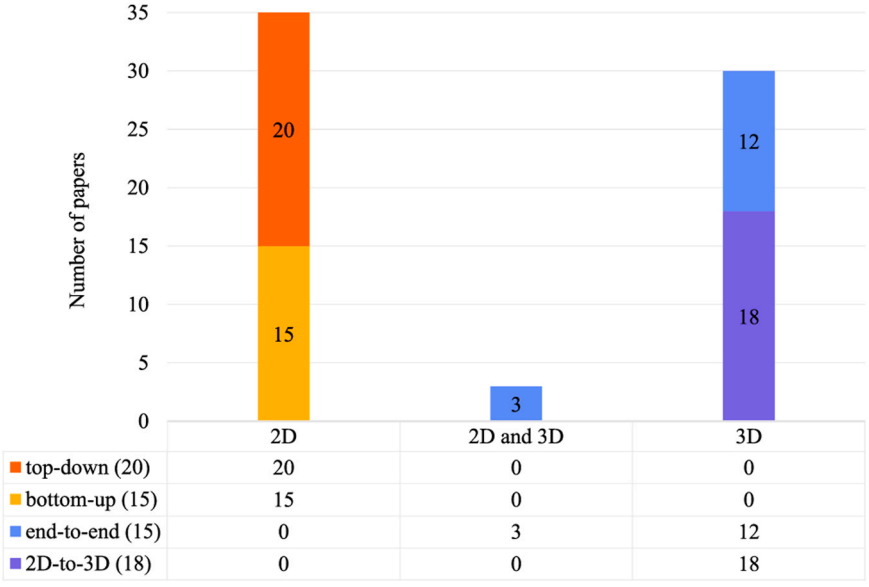


Fig. 7. The algorithms used in 2D and 3D human pose estimation. Algorithm categories are listed on the left, with corresponding usage frequencies for different output types indicated on the right. The width of each flow is proportional to the number of studies using that algorithm.



	2D	2D and 3D	3D
top-down (20)	20	0	0
bottom-up (15)	15	0	0
end-to-end (15)	0	3	12
2D-to-3D (18)	0	0	18

Fig. 6. The workflows used in 2D and 3D human pose estimation. Top-down and bottom-up workflows are typically used in 2D HPE, while end-to-end and 2D-to-3D workflows are commonly used in 3D HPE.

Table 1
Human pose estimation and tracking methods. "Off-the-shelf" indicates the method directly utilizes an existing pose estimator to detect human poses, only focuses on improving the tracking component.

Workflows	References	Pose estimation methods	Pose tracking methods
Top-down	Pose Flow [88]	CNN (off-the-shelf)	Pose-flow similarity
	Simple Baseline [87]	CNN	Flow-based pose similarity
	LightTrack [60]	CNN (off-the-shelf)	IoU of boxes & GCN
	FastPose [94]	CNN	Occlusion-aware re-ID features
	AlphaPose [25]	CNN	IoU of boxes, OKS of poses & pose-guide re-ID features
	KeyTrack [74]	CNN (off-the-shelf)	Transformer
Bottom-up	STAF [67]	CNN	TAF (CNN)

4.3.1.1. Top-down. CNNs have been the most significant and widely used networks in 2D HPE. [87] introduced Simple Baseline for pose estimation and tracking, which adopted three deconvolutional layers as a head to generate high-resolution feature maps and heatmaps, instead of using computationally expensive up-sampling to increase the feature map resolution. [94] proposed FastPose, a unified framework with three output heads for human detection, pose estimation, and person re-identification (Re-ID). This work addresses the common scale variation problem in unified workflow by exploiting Scale-Normalized Image and Feature Pyramids (SIFP). [7] introduced a lightweight CNN architecture, BlazePose, designed for real-time single-pose estimation on mobile devices. [74] used an off-the-shelf model as the pose estimation module and enhanced its performance through post-processing. They introduced box propagation to resolve the missing boxes problem and then calculated Temporal Object Keypoint Similarity (TOKS) among adjacent frames to preserve more accurate poses. This method outperforms approaches that only rely on bounding box confidence scores. [25] proposed AlphaPose, a unified system for whole-body pose estimation and tracking that localizes 136 points per person, including the face, body, hands and feet. AlphaPose employs a Dense Upsampling Convolution (DUC) module to generate keypoint heatmaps.

[59] avoided applying large networks to every frame by designing a lightweight CNN-based dynamic pose kernel distillatory, which transfers pose knowledge from previous frames to guide body joint localization in the current frame. [98] designed a lightweight, unsupervised network to select a small number of keyframes from a video sequence to be fed into a pose estimator. A pose interpolation module was then introduced to recover the poses of remaining non-keyframes based on keyframe poses. [55,98] got away from heatmap-based methods and regarded pose estimation as two object detection tasks. They utilized a dense detection network to simultaneously predict a set of keypoint objects, which supply precise joint positions, and pose objects, which capture the global relationships between joints. Finally, a matching algorithm fuses these two objects to yield human poses. [45] took advantage of motion vectors available in compressed videos to estimate human poses. They designed a fast pose warping module to propagate pose across consecutive frames using these motion vectors. In the RTMPose study [31], the authors reformulated 2D pose estimation as two classification tasks, predicting the x-axis and y-axis coordinates of keypoints, respectively. This approach removes the costly up-sampling layers, replacing them with two fully-connected layers for more efficient prediction.

In order to reduce computation costs without accuracy degradation, several papers have leveraged Neural Architecture Search (NAS) networks to find the most optimal network for HPE. For instance, [97] employed NAS to customize an efficient backbone and head for pose estimation, making the first attempt to search for an optimal backbone in this field. [89] also used NAS to explore optimal networks at both spatial and temporal levels, it was the first method to search for temporal connections cross frames and finally achieved CPU real-time performance. Additionally, Transformers ported from Natural Language Processing (NLP) have gained success in various computer vision tasks, including HPE. It consists of a self-attention module and a feed-forward network (FFN) and performs well in long-range dependency modelling. [90] introduced transformer-based networks as a backbone for 2D HPE.

4.3.1.2. Bottom-up. CNNs are the dominant networks in bottom-up workflow. [12] introduced Part Affinity Fields (PAFs), which represent keypoints as normalized 2D vector fields that encode both positions and orientations. The method employs a shared convolutional backbone, followed by two parallel branches to separately generate keypoint heatmaps and PAFs. Finally, a greedy bipartite graph matching algorithm is applied to associate the keypoints into human poses. [61] optimized the model proposed by [12] for deployment on edge devices. In the backbone, the authors replaced strided convolutions with dilated convolutions to preserve spatial resolution while enlarging the receptive field. In the refinement stage, only using a single branch, composed of depthwise separable convolution blocks, to efficiently produce keypoint heatmaps and PAFs. [67] proposed Spatial-Temporal Affinity Fields (STAF), which combine PAFs and Temporal Affinity Fields (TAFs). While PAFs encode connections across keypoints within a single frame, TAFs capture associations of corresponding keypoints across consecutive frames. STAF integrates these two components using a unique cross-linked limb topology to improve robustness against motion blur and occlusion, enabling simultaneous pose detection and tracking in video sequences.

[10] released OpenPose, the first open-source real-time system including body, foot, hand, and facial pose detection (in total 135 keypoints). [83] improved HRNet-based model proposed by [76], they designed a single-branch architecture for real-time HPE on edge devices by removing redundant high-resolution refinement branches. Additionally, a fusion deconvolution head and large-kernel convolutions were used to handle scale variation problem and enhance model capacity.

4.3.2. 3D human pose estimation

There are 30 papers related to 3D HPE. Among these, 18 papers adopt the 2D-to-3D workflow, while the remaining 12 papers follow the end-to-end workflow, some representative algorithms are presented below.

4.3.2.1. 2D-to-3D. 2D-to-3D workflow-based methods typically take 2D poses as both input and intermediate representations. To alleviate depth ambiguity and enhance motion coherence, Temporal Convolution Networks (TCNs), Transformers and Graph Convolution Networks (GCNs) are frequently employed and integrated into this workflow due to their strong capability to capture spatio-temporal dependencies within input sequences.

TCNs with dilated convolutions (1D convolutions over the time dimension), have been widely adopted to capture long-range temporal

dependencies and enlarge the receptive field. [65] used a fully convolutional network based on dilated temporal convolutions to model 2D pose sequences of arbitrary length for 3D pose estimation. Considering 3D pose labelled datasets are scarce, the authors re-projected the predicted 3D keypoints back into 2D space, enabling semi-supervised training on unlabelled videos. Similarly, [49,50] employed multi-scale dilated TCNs to model long-range dependencies among frames and proposed an attention-based TCN framework to adaptively identify significant frames and tensor through-puts across neural layers for optimal inference.

Transformers also exhibit excellent ability in modelling long-range temporal and spatial dependencies, particularly for discrete 2D pose joints. [41] replaced fully-connected layers in the FFN of Transformers with strided convolutions, effectively shrinking sequence length without accuracy loss. [95] observed different joints have distinct motions, a factor often ignored in previous studies. Therefore, the authors employed a temporal Transformer to independently learn the motion trajectories of each joint. [42] applied Transformers to learn spatio-temporal representations of multiple 3D pose hypotheses; based on the fact that estimating 3D pose from monocular videos is an ill-posed inverse problem with multiple feasible solutions. [99] exploited the frequency representation of 2D skeleton sequences to efficiently process long sequences for 3D pose estimation, fusing features in both time and frequency domains to improve robustness against noisy 2D keypoint detections. [20] adopted a masked token model in Transformers to generate upsampling tokens as placeholders for non-keyframes, enabling operations on temporally sparse 2D pose sequences while producing temporally dense 3D poses. [72] introduced a self-supervised spatio-temporal pre-training task that randomly masks some frames as well as some 2D joints in the remaining frames. The pre-trained encoder can effectively capture spatial and temporal dependencies from input sequences.

Recent studies have increasingly explored hybrid architectures that integrate TCNs, Transformers, and GCNs to efficiently capture both spatial and temporal dependencies in 2D-to-3D workflow. [47] synergistically interleaved attention-based GCNs and dilated TCNs to extract and fuse spatio-temporal information from 2D keypoint sequences to reconstruct 3D poses. Different from previous works focused on temporal contexts, this approach takes advantage of GCNs to explore local and global spatial constraints of body joints via attention mechanisms. [81] integrated Fully-Connected Networks (FCNs) and TCNs to predict bone length and direction respectively. Finally, the refined 3D positions of joints are derived using an additional FCN based on the estimated bone length and direction. [33] combined the strengths of GCNs and Transformers to explore local and global joint constraints respectively. Moreover, they provided an efficient module for embedding video sequences into single-frame models. Fig. 8 illustrates a qualitative comparison of 3D skeleton-based HPE results generated by three different methods that follow the 2D-to-3D workflow.

4.3.2.2. End-to-end. End-to-end workflow directly processes input video frames; therefore, CNNs remain the most preferable networks in this workflow, due to their strong balance between efficiency and accuracy in extracting features from images. [57] was the first method to generate a full global 3D skeletal pose of a human, including joint angles. It combines a shallow Fully-Convolutional Network (FCN) with a kinematic skeleton fitting method to yield a temporally stable 3D global pose. Building on this, [56] addressed the estimation of multi-person 3D absolute skeletal pose relative to the camera. The authors first utilized a CNN to detect visible body joints and subsequently used a lightweight

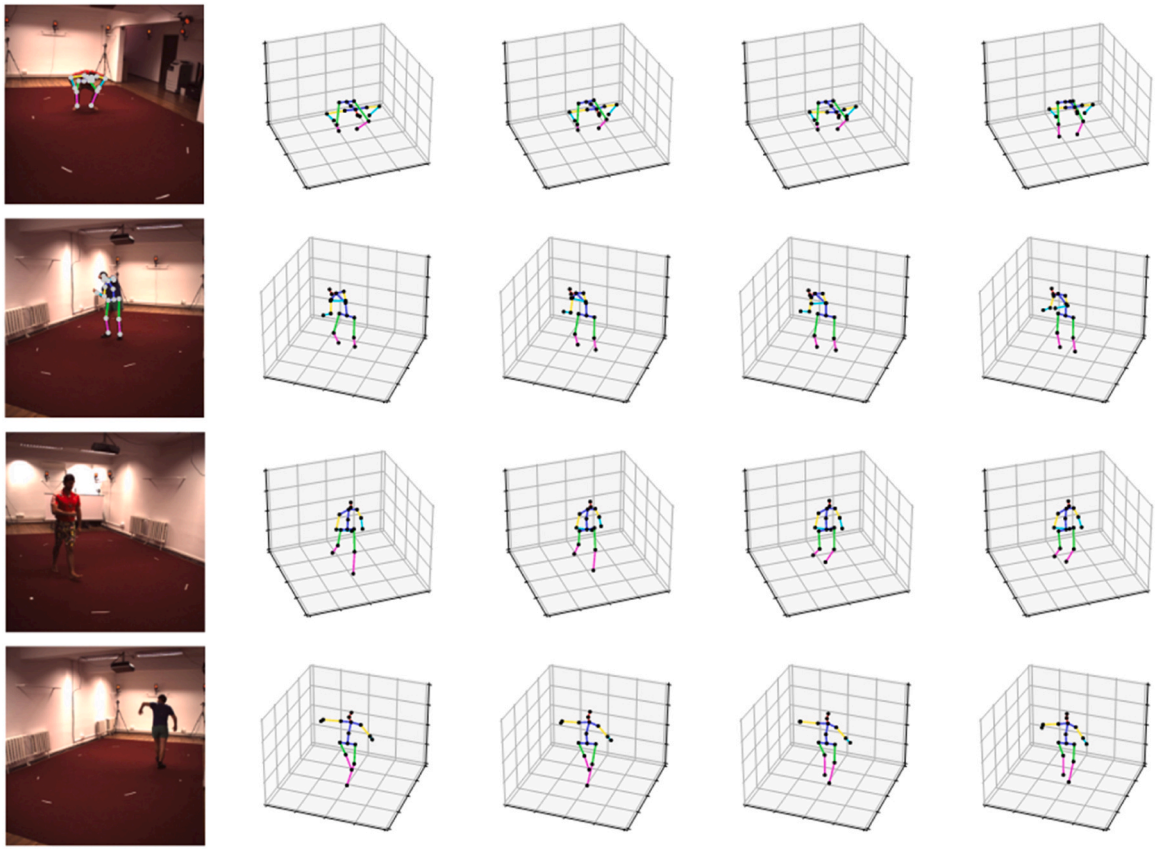


Fig. 8. Qualitative comparisons on Human3.6M dataset [20].

Fully-Connected Neural Network to infer the full 3D pose, including occluded joints, by leveraging full-body context.

In addition to skeletal pose estimation, mesh recovery with 3D pose within end-to-end workflow has emerged as an important research direction in recent years. [36] proposed VIBE, which employs a Generative Adversarial Network (GAN) to train a motion discriminator based on the large-scale motion capture dataset (AMASS). The generator consists of a CNN, a Gated Recurrent Unit (GRU) and a body parameter regressor to predict parameters of Skinned Multi-Person Linear (SMPL) model. The motion discriminator attempts to distinguish the difference between predicted poses and those sampled from the AMASS dataset by outputting a real or fake label on each frame. Finally, producing kinematically plausible 3D pose and shape sequences without requiring in-the-wild ground-truth 3D pose annotations. [77] adopted a one-stage multi--head architecture comprising a shared backbone and three heads for predicting the body centre heatmap, camera heatmap, and SMPL map simultaneously. The body centre heatmap provides the 2D positions of the body centre, which are then used to sample the corresponding 3D body mesh parameters. These sampled mesh parameter vectors are fed into the SMPL body model to derive the final 3D poses and shapes. Fig. 9 shows a qualitative comparison of 3D mesh-based HPE outputs produced by two different methods using the end-to-end workflow.

4.3.3. 2D and 3D human pose estimation

Among the reviewed studies, only 3 papers can simultaneously obtain 2D and 3D poses from monocular videos by employing either Transformer-based (2 papers) or CNN-based (1 paper) architectures. Specifically, [53] proposed a multi-task framework for jointly estimating 2D and 3D poses, as well as recognizing actions. For 3D pose estimation, they decoupled the task into 2D pose estimation and depth estimation. The final 3D pose is the concatenation of the 2D coordinates and the corresponding depth values. [93] only conducted costly pose estimator on sparsely sampled video frames (less than 10 % of the total frames). Instead of directly recovering the poses of remaining frames from sampled poses, the authors first denoised the detected poses with an efficient Transformer architecture before the recovering process.

As mentioned before, exploring spatio-temporal relationships of poses in adjacent frames is a promising and significant direction in video-based HPE, particularly for 3D HPE. According to the analysis of algorithms, the most striking observation is that Recurrent Neural Network (RNN)-based methods, including GRU, are rarely used in real-time systems due to expensive computation and unable to parallel processing. Additionally, RNNs suffer from vanishing or exploding gradient problems when handling long-range sequences. In contrast, TCNs and temporal Transformers can model long-range temporal dependencies with parallel computation. GCNs [47] and spatial Transformers [99] are commonly employed to capture spatial relationships of body joints in each frame. Consequently, many researchers have integrated different architectures into a unified framework to improve inference efficiency. In addition, bone length, direction and rotation information as priori knowledge usually have been introduced to ensure spatial and temporal consistency of poses [23,56,57,65,66,81], as relative bone lengths between body parts are scale-invariant and remain consistent during motion. A detailed comparison of different architectures is thoroughly discussed in Section 5.1.

4.3.4. Human pose tracking

HPT is another fundamental computer vision task, aiming to assign consistent identification numbers (IDs) to the same person in different video frames (See Section 2.2). In this work, we identify only 7 papers proposing a real-time solution, and they focus exclusively on 2D outputs.

Table 1 provides different methods used for pose estimation and pose tracking; several studies focus on improving tracking methods by employing existing state-of-the-art (off-the-shelf) pose estimators. Additionally, most tracking algorithms (6 papers) follow the top-down workflow, which involves first detecting bounding boxes and locating body joints and then tracking poses over the entire video based on pose similarity. As shown in Table 1, five primary methods are commonly employed to calculate pose similarity: (i) Intersection-over-Union (IoU) of bounding boxes, which becomes less accurate when people move fast leading to no overlap and confusion in crowd scenes. (ii) Object Keypoint Similarity (OKS) of poses, which is invalid when pose changes for the same person in different frames. (iii) GCN-based methods, which rely on keypoints spatial locations. (iv) CNN-based methods, which can feed bounding boxes or pure keypoints to match a pair of poses. (v) Transformer-based methods, which receive spatial information via the

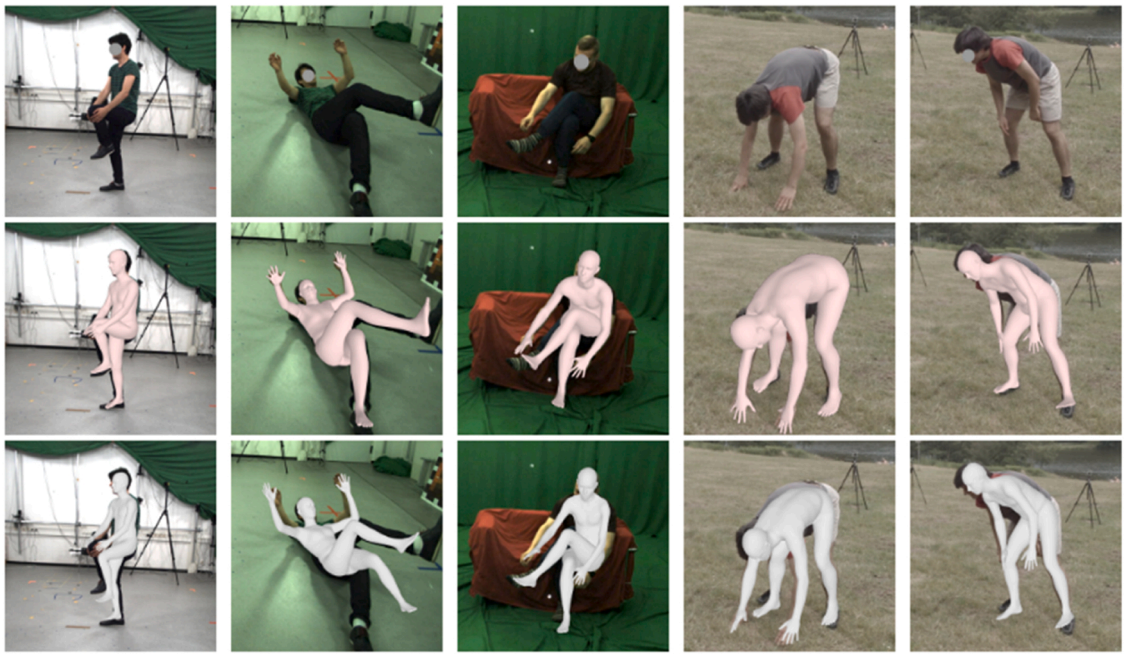


Fig. 9. Qualitative comparisons on MPI-INF-3DHP dataset [24], video frame (first row), the qualitative results of [24] (second row in pink), and the qualitative results of [37] (third row in grey).

position embeddings by projecting 2D keypoint coordinates into 1D linear embeddings.

For higher-resolution inputs, Transformers often need more parameters to achieve comparable performance to CNNs, and CNNs typically converge faster than Transformers [74]. Therefore, Transformers tend to outperform CNNs in low-resolution inputs, while GCNs can effectively model spatial relationships between keypoints but rely on high resolution and computation cost. For example, [87] proposed a flow-based pose similarity metric by comparing propagated joints with predicted joints in the current frame. [88] introduced the PoseFlow method, which first constructs pose flows for the same person across multiple frames, then adopts pose-flow Non-Maximum Suppression (NMS) to eliminate redundant pose flows and re-link temporally discontinuous poses.

Recent pose tracking methods increasingly utilize re-identification (re-ID) features to enhance tracking robustness. For example, [94] proposed FastPose, which integrates re-ID features to track people while improving the usage by inferring occlusion states from pose information. [25] proposed AlphaPose, a multi-person pose tracking model that employs pose-guided re-ID features to reduce background noise and computation load. It combines IoU of bounding boxes, OKS of poses, and re-ID features to assign final tracking IDs. [60] designed a Siamese Graph Convolution Network (SGCN) that associates pose IDs between adjacent frames by integrating GCN with bounding box IoU to ensure both temporal and spatial consistency. [74] presented a Transformer-based tracking method that relies solely on keypoints, without using any RGB or optical flow information. It classifies pairs of poses from different frames to determine whether one pose temporally follows another. Notably, [67] is the only study adopting a bottom-up workflow. It predicts Temporal Affinity Fields (TAFs), a set of 2D vector fields that encode associations between corresponding body joints across frames, thereby enabling the assignment of unique pose IDs to each person in a video.

4.4. Accuracy measures and datasets

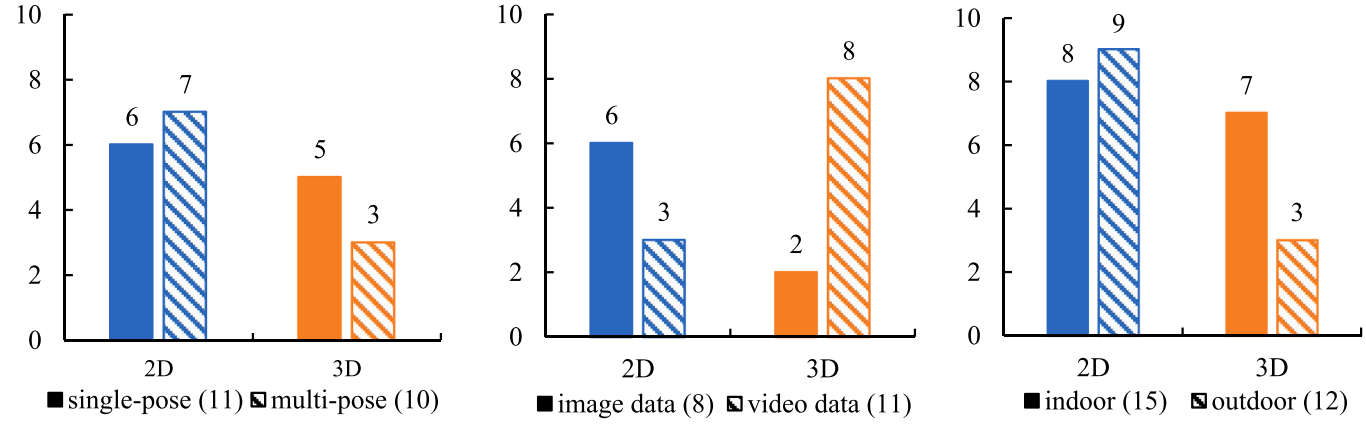
Different datasets may employ distinct evaluation metrics. Therefore, we present a combination of datasets and accuracy metrics in this section to answer RQ4.

Existing 3D datasets are primarily captured in controlled studio settings with limited pose and appearance diversity, or created by combining real and synthetic images. Collecting high-quality 3D pose annotations in outdoor environments is usually accompanied by high complexity and cost, resulting in a significant imbalance between indoor and outdoor datasets for 3D HPE. Therefore, one of the main challenges in 3D HPE is the lack of in-the-wild ground-truth 3D datasets. As shown in Figs. 10, 2D HPE datasets are relatively balanced in terms of indoor and outdoor scenarios. Additionally, single-person datasets are more available than multi-person datasets in 3D domain, which explains why most state-of-the-art methods focus on 3D single-pose estimation. The scarcity of multi-person 3D datasets limits the development of multi-person 3D pose estimations.

Another important finding is the video-based datasets in 2D HPE are scarce compared to image-based datasets. Indeed, 2D domain does not necessarily require exploiting temporal information based on video data, as employing a model on per-frame can obtain 2D poses, temporal context only improves the performance. Moreover, video-based HPE brings additional challenges such as motion blur and low-latency requirements. In contrast, 3D HPE has more video-based datasets, as it introduces a new challenge, depth uncertainty, which is difficult to predict from a single image. Therefore, leveraging the temporal co-dependency of poses among consecutive frames can solve depth ambiguity and reduce unnecessary calculations. Furthermore, temporal information can further handle occlusion and enhance robustness.

Table 2 provides the key characteristics of commonly used datasets and their corresponding accuracy evaluation metrics, including whether they support single-person or multi-person pose evaluation, image-based or video-based evaluation, and indoor or outdoor scenarios. In our view, these properties are essential for understanding each dataset's evaluation capability and limitations. Additional properties such as the year of release, number of annotated joints, dataset size, associated publications, and GitHub links can be found in [21,51,100]. Except for the datasets Halpe-FullBody [25], COCO-WholeBody [32] and RWS-CPE [14], the detailed descriptions and comparisons of the remaining datasets have already been thoroughly reviewed in previous literature [21, 51,100], and thus, they are not covered in this review.

Halpe-FullBody consists of a total of 136 keypoints for each person, including 20 for body, 6 for feet, 42 for hands and 68 for face (see Fig. 11). It contains 50 K instances for training and 5 K images for testing. Similarly, COCO-WholeBody defines 133 whole body keypoints based on the COCO dataset with 68 on the face, 42 on hands and 23 on the body and feet. The total training set contains 118 K images with 250 K instances, and the test set contains 5 K images. Compared to Halpe-FullBody, it does not contain annotations for the head, neck and hip points; the remaining keypoint definitions are consistent. Therefore, these two datasets are suitable for the whole-body pose estimation. RWS-CPE is a private dataset for real-world surveillance crowd pose estimation. It includes 100 frames selected from 11 different cameras with 4785 bounding boxes and 1894 poses manually annotated. Each



(a) single and multi (b) image and video (c) indoor and outdoor

Fig. 10. The dataset distribution of 2D and 3D pose estimation. The y-axis presents the number of datasets. (a) categorized by the number of people in the scene; (b) categorized by data type; (c) categorized by scene type.

Table 2
Summary of human pose datasets used in the reviewed studies and the corresponding accuracy evaluation metrics. "S/M" denotes datasets containing single-person or multi-person, "I/V" indicates datasets are image data or video data, "I/O" means datasets including indoor scenes or outdoor scenes. * means PoseTrack dataset provides both pose and identification annotations.

Types	Datasets	S/M	V/I	I/O	Metrics	# of Papers and References
2D	COCO	S&M	I	I&O	AP & AR	20 Papers: [10,13,16–18,25, 31,39,48,55,61,70, 71,76,83,87,89,90, 94,97]
	*PoseTrack 2017 & 2018	M	V	I&O	AP, AR, MOT, IDS, Prec & Rec	13 Papers: [14,25,45,60,67, 74,76,87–89,94]
	Halpe-FullBody	S&M	I	I&O	AP & AR	1 Papers: [25]
	COCO-WholeBody	S&M	I	I&O	AP & AR	3 Papers: [19,25,31]
	CrowdPose	M	I	I&O	AP & AR	7 Papers: [16–18,31,39,55, 83]
	Penn Action	S	V	I&O	PCK	4 Papers: [53,59,98,104]
	Sub-JHMDB	S	V	I&O	PCK	4 Papers: [59,93,98,104]
	RWS-CPE	M	I	O	AP & AR	1 Papers: [14]
	MPII	S&M	I	I&O	AP & PCK	6 Papers: [10,13,31,53,76, 97]
3D	Human3.6M	S	I&V	I	MPJPE, MPJVE, MRPE & VAR	25 Papers: [1,9,20,22–24,30, 33,36,41,42,47,49, 50,54,56,57,65,72, 75,81,86,95,99, 103]
	HumanEva I & II	S	V	I	MPJPE	7 Papers: [41,47,49,50,54, 65,95]
	MPI-INF−3DHP	S	I&V	I&O	MPJPE, 3D PCK, AUC & Accel	12 Papers: [6,20,24,33,36,42, 56,57,72,86,95, 99]
	AIST	S	V	I	MPJPE	1 Papers: [6]
	TotalCapture	S	V	I	MPJPE	1 Papers: [66]
	3DPW	M	V	O	MPJPE, PVE, MPJAE, 3D PCK, AUC & Accel	5 Papers: [1,6,36,56,77]
	CMU Panoptic	M	V	I	MPJPE	2 Papers: [56,77]
	MuPoTS−3D	M	V	I&O	MPJPE, 3D PCK, AUC, Accel & AP_{25}^{root}	4 Papers: [1,6,22,56]

frame contains at least a dozen and up to more than one hundred people, making it particularly suitable for crowd pose estimation in the wild.

As shown in Table 2, COCO is the most widely used dataset for both single-person and multi-person 2D pose estimation tasks. It offers comprehensive and fine-grained annotations of body joints across diverse environments, along with well-established evaluation metrics. In particular, for multi-person, COCO not only provides joint visibility flags but also defines a set of per-keypoint tolerance constants used in the OKS evaluation metric. These constants reflect the localization

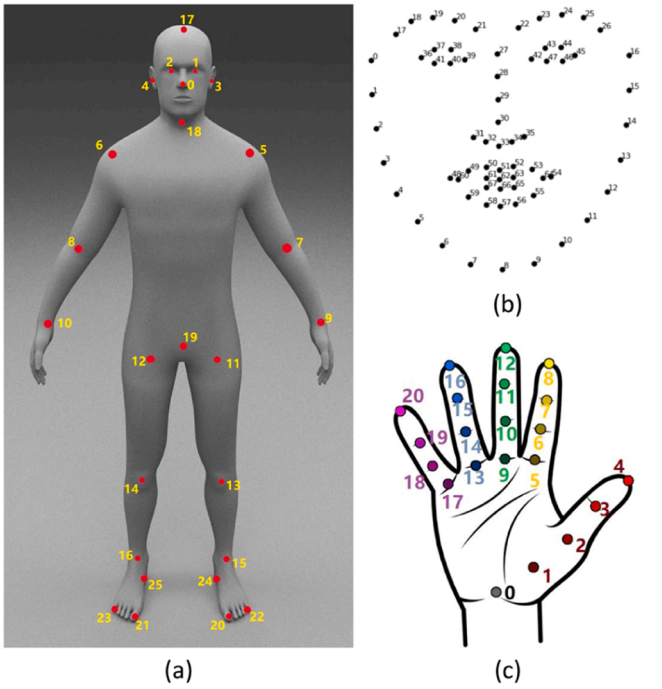


Fig. 11. Annotated keypoint format in Halpe-FullBody dataset for (a) body and foot, (b) face, (c) hand respectively [25].

uncertainty of different keypoints and determine the degree of error acceptable for each joint during evaluation. Another popular 2D dataset is PoseTrack (2017, 2018), which provides both pose annotations and identification number labels, supporting the tasks of pose estimation and pose tracking. Table S1 of the Supplementary Materials presents the performance of 2D multi-person HPE models evaluated on these two popular datasets. For 2D single-person HPE, the Penn Action dataset was chosen to show the performance of different single-pose models (see Table S2 in Supplementary Materials).

For 3D HPE, Human3.6M is the most commonly used dataset in single-person scenarios. It provides detailed 3D positions and rotation angles of body joints, as well as SMPL parameters. Additionally, camera extrinsic parameters (rotation and translation relative to world coordinates) and intrinsic parameters (focal length and principal point) are also available, making it possible to evaluate 3D absolute pose. However, it was captured entirely in indoor setups, and it may not generalize well to outdoor scenarios. To address this limitation, MPI-INF-3DHP is often used for evaluating models in outdoor single-person contexts. For multi-person 3D pose estimation, 3DPW is one of the most widely used in-the-wild datasets and also provides SMPL parameters. Its challenging real-world scenarios and high-quality annotations make it suitable for evaluating the generalization performance of 3D HPE models in unconstrained conditions. Table S3 of Supplementary Materials gives the performance of 3D single-person HPE models on the Human3.6M dataset and 3D multi-person HPE models on the 3DPW dataset, following different workflows to provide a clear quantitative comparison. Nevertheless, 3D pose tracking remains underexplored, largely due to the scarcity of available datasets for training and evaluation.

Table 3 presents the accuracy evaluation metrics and their variations under different thresholds, highlighting their distinct evaluation capability and characteristics. In 2D HPE, Average Precision (AP), Average Recall (AR), and Percentage of Correct Keypoints (PCK) are the three primary metrics for assessing accuracy. Notably, AP and AR, along with their threshold-based variants, were first introduced in the COCO dataset and have become standard evaluation metrics in 2D HPE. These metrics are now widely used by various datasets beyond COCO. The corresponding evaluation metrics for each dataset are summarized in

Table 3
Accuracy evaluation metrics for 2D and 3D human pose estimation and tracking. * indicates metrics used to evaluate the accuracy of pose tracking; ˆ means metrics used to evaluate the accuracy of body models (e.g., SMPL); a means metrics used to evaluate the absolute camera-centre pose estimation; † means metrics used to evaluate the speed consistency of motion.

Types	Metrics	Variations	Capability
2D	AR (average recall)	AR^{50}	AR at OKS= 0.5
		AR^{75}	AR at OKS= 0.75
		AR^{M}	AR for medium objects: 322 < area < 962
		AR^{L}	AR for large objects: area > 962
		mAR	AR at OKS = .50:.05:.95
	AP (average precision)	AP^{50}	loose metric: AP at OKS= 0.5
		AP^{75}	strict metric: AP at OKS= 0.75
		AP^{M}	AP for medium objects: 322 < area < 962
		AP^{L}	AP for large objects: area > 962
		mAP	mean AP based on different OKS or PCK
		AP^{E}	with easy crowd index (less occlusion)
		AP^{H}	with hard crowd index (more occlusion)
		AP^{T}	the AP score after tracking post-processing
	PCK (percentage of correct keypoints)	PCKh@0.5&0.1	assume the point to be detected correctly if the 2D Euclidean error is smaller than 50 % of the corresponding person's head size
		PCK@0.2&0.1&0.05	assume the point to be detected correctly if the 2D Euclidean error is smaller than 20 %/10 %/5 % of the corresponding person's torso size or person size
	*MOT	MOTA	Multiple-Object Tracking Accuracy
		MOTP	Multiple-Object Tracking Precision
	*IDS	ID Switches	the number of human ID switches
		% IDS	$\%IDSW^{i} = \sum_t IDSW_t^i / \sum_t GT_t^i$
	*Prec	Precision	$\frac{TP}{TP+FP}$
	*Rec	Recall	$\frac{TP}{TP+FN}$
3D	MPJPE (mm) (mean per-joint position error)	PA-MPJPE	Procrustes-aligned MPJPE: the error after alignment with the ground truth in translation, rotation, and scale
		N-MPJPE	normalized MPJPE: aligns predicted poses with the ground-truth only in scale
	†MPJVE	mean per-joint velocity error (mm/s)	corresponding to the MPJPE of the first derivative of the 3D pose sequences, measure the smoothness of the prediction sequence
	[a]MRPE	mean root position error	the average error of the absolute root joint (the hip) localization
	[a]AP_{25}^{root}	average precision	measure the 3D human root location prediction error which considers the prediction as correct when the Euclidean distance between the estimated and the ground truth coordinates is smaller than 25 cm
	VAR	Variance	the variance of MPJPE between action categories to evaluate the stability
	3D PCK (%)	3D PCK@150mm	Percentage of Correct Keypoint within 150 mm range
		PA−3DPCK@150mm	Procrustes-aligned variant of 3D PCK
		[a] $3DPCK_{abs}$@150mm	PCK without the root alignment used to evaluate the absolute poses
	AUC	Area Under Curve	area under curve for a range of 5–150 mm PCK thresholds
	†Accel	Acceleration error (mm/s2)	the average difference in acceleration between the ground-truth and predicted 3D joints.
	ˆPVE	Per Vertex Error	the mean Euclidean distance between predicted vertex positions and ground-truth vertex positions
	ˆMPJAE (mean per-joint angle error)	PA-MPJAE (rad)	Procrustes-aligned MPJAE: the error after alignment with the ground truth in translation, rotation, and scale

Table 2. For 2D single-person pose estimation, PCK is one of the most widely used metrics to measure the precision of keypoint detection. The formulation of PCK is as follows:

$$\mathrm{PCK} = \frac{\sum_{i=1}^{N} \mathbb{I}(d_i \le \alpha \cdot L)}{N} \tag{1}$$

where N is the number of total keypoints. $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition is satisfied and 0 otherwise. d_i represents the Euclidean distance between ground-truth and the i^{th} detected keypoint. L is a normalizing factor that can be set based on the person size, torso size or head size (PCKh). α is a hyperparameter, typically set to 0.2. This metric is considered relatively loose for evaluating model performance, particularly when the person size is relatively large. Moreover, PCK does not consider the visibility and tolerance of different body joints. As a result, it is mostly suitable for single-person pose estimation.

In contrast, AP and AR are more comprehensive metrics and widely adopted in multi-person pose estimation. To account for performance across different thresholds, the mean Average Precision (mAP) is calculated as:

$$\mathrm{mAP} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}\left(OKS_{\alpha_i}\right) \tag{2}$$

where N is the number of thresholds, usually be set 10, α_i represents different OKS thresholds (e.g., $\alpha = 0.50, 0.55, \ldots, 0.95$). The Object Keypoint Similarity (OKS) metric is defined as follows:

$$\mathrm{OKS} = \frac{\sum_i \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{3}$$

where d_i is the Euclidean distance between ground-truth and the i^{th} detected keypoint. s is the scale of person. k_i is a keypoint-specific constant that controls the falloff. v_i is the visibility flag (1 if visible, 0 if not). $\delta(v_i > 0)$ ensures that only visible keypoints contribute to the final score. OKS provides a more fine-grained and robust evaluation by

taking into account the scale of person, the visibility and tolerance of different body joints. It is therefore widely used in multi-person pose estimation.

Different from 2D HPE, the positions of 3D body joints can be categorized as either relative positions (with respect to the centre of person) or absolute positions (with respect to the camera). Therefore, the evaluation metrics for 3D HPE can be broadly divided into person-centric metrics and camera-centric metrics. Metrics such as Mean Root Position Error (MRPE), $3DPCK_{abs}@150mm$ and AP_{25}^{root} are specifically employed to evaluate the accuracy of absolute 3D body joint positions [21]. The formulation of MRPE is given by:

$$MRPE = \frac{1}{T}\sum_{t=1}^{T} \| \left(R^{(t)} - R^{*(t)}\right) \|_2 \tag{4}$$

where R and R^* denote the predicted and ground-truth root joint (the hip) respectively, T is the number of frames.

Except for the three metrics mentioned above, the remaining ones are person-centric and are used to evaluate relative 3D poses. As summarized in Table 2, the most widely used metric for evaluating relative 3D joint positions is the Mean Per Joint Position Error (MPJPE), which measures the mean Euclidean distance between ground-truth and the predicted 3D joint positions:

$$\text{MPJPE} = \frac{1}{N}\sum_{i=1}^{N} \| \widehat{\mathbf{J}}_i - \mathbf{J}_i^{\text{gt}} \|_2 \tag{5}$$

where N is the number of body joints, $\widehat{\boldsymbol{J}}_i = (\widehat{x}_i, \widehat{y}_i, \widehat{z}_i)$ is the predicted 3D coordinate of the i^{th} joint, $\boldsymbol{J}_i^{gt} = (x_i^{gt}, y_i^{gt}, z_i^{gt})$ is the ground-truth 3D coordinate of the i^{th} joint, the $\| \widehat{\boldsymbol{J}}_i - \boldsymbol{J}_i^{gt} \|_2$ is the Euclidean distance between ground-truth and the i^{th} detected keypoint. Depend on the alignment method applied during post-processing between the estimated pose and ground-truth poses, MPJPE has two common variants: PA-MPJPE and N-MPJPE (see Table 3 for detailed descriptions). However, evaluating the smoothness and temporal coherence of motion is also important for 3D video-based HPE. Metrics such as Acceleration Error (Accel) and Mean Per Joint Velocity Error (MPJVE) are exclusively introduced to evaluate the consistency of joint motion speeds over time.

Additionally, datasets like 3DPW and Human3.6M provide SMPL parameters, which allow the use of Per Vertex Error (PVE) and Mean Per Joint Angle Error (MPJAE) metrics to evaluate the accuracy of body shape. The formulation of PVE is as follows:

$$\text{PVE} = \frac{1}{M}\sum_{i=1}^{M} \| \widehat{\mathbf{v}}_i - \mathbf{v}_i^{\text{gt}} \|_2 \tag{6}$$

where M is the total number of mesh vertices, $\widehat{\boldsymbol{v}}_i$ is the predicted vertex position, and $\boldsymbol{v}_i^{gt}$ is the ground-truth vertex position. Similarly, MPJAE measures the mean per-joint rotation angle error between the predicted and ground-truth joint:

$$\text{MPJAE} = \frac{1}{N}\sum_{i=1}^{N} \text{Angle}\left(\widehat{\Theta}_i, \Theta_i^{\text{gt}}\right) \tag{7}$$

where $\widehat{\Theta}_i$ is the predicted joint rotation angle, Θ_i^{gt} is ground-truth joint rotation angle, N is the number of all body joints.

4.5. Hardware configurations and efficiency measures

Most reviewed studies emphasize computation efficiency rather than power efficiency or memory efficiency [2]. Commonly used metrics to evaluate the computation complexity of models include Floating-Point Operations (FLOPs) [97] and Multiply-Add Operations (MACs) [83]. Both metrics only reflect theoretical computation complexity. In addition, the number of model parameters is introduced to assess computation efficiency [33]. However, reducing parameters does not always result in computational speedups [56]. Because computation efficiency is not only related to the number of parameters, but also involves whether the model structure can utilize hardware resources. For instance, reducing model parameters leads to less requirement for memory of CPUs, while computationally intensive operations such as convolution or matrix multiplication still require a lot of computing resources of GPUs. Moreover, GPUs are generally not well-suited for sparse operations, especially unstructured sparsity, as modern GPUs are optimized for dense matrix and massive parallel computations efficiently. Therefore, GPUs benefit from less operations, whereas CPUs seem to be faster due to lower memory requirements [10].

To measure real-world performance, researchers implement their models on specific hardware and report inference speed using metrics such as FPS (frames per second) [70,89], Hz (Hertz) [44], and runtime (inference time of each frame) [23]. It is worth noting that evaluating computation efficiency based on hardware can reflect model deployment performance more accurately than theoretical complexity. However, power and memory efficiency have not received sufficient attention in many studies, which may become a critical direction for future research targeting mobile and edge devices. Table 4 provides a comprehensive overview of the practical implications, strengths, limitations, and relationships among different metrics for measuring computational efficiency. As a result, in 2D/3D HPE and HPT tasks, we recommend using FLOPs to reflect the theoretical computational complexity of models. While employing FPS to evaluate practical inference speed, the corresponding inference hardware configuration must be clearly specified. Tables S1, S2 and S3 in the Supplementary Materials provide the efficiency performance of different models, including the computational complexity (FLOPs) and inference speed (FPS) on different hardware.

Table 5 lists all the hardware types chosen by the reviewed papers to run their proposed models. We divide GPUs into three types: consumer GPUs, data centre GPUs and professional GPUs, based on their theoretical TFLOPs value and intended usage. The consumer GPUs category also contains high-end series, such as the Nvidia Titan series, which are designed for consumer-grade usage with TFLOPs exceeding 10, the Nvidia GeForce RTX 30 series including RTX 3060 (12.7 TFLOPs) and RTX 3090 (35.6 TFLOPs) are widely used in computer vison tasks due to its high performance. As shown in Fig. 12, deploying HPE models on mobile devices (e.g., smartphone or edge devices) and CPU-only devices is becoming increasingly popular, as these platforms are more closed to real-world applications. However, deploying HPE models on consumer GPUs is still prevalent (77 % percentage) because of their availability and excellent parallel computation capability compared to mobile and CPU-only devices.

However, some models are tested on different devices with the same settings but obtain different inference times and speeds [10,31,39,70,83]. This variation is mainly attributed to the difference in hardware computation and memory capacity, as well as the extent to which the model can effectively utilize the strengths of hardware. For instance, [83] proposed LitePose and tested its LitePose-S model (5.0 GMACs) on three platforms: Raspberry Pi 4B+ , Qualcomm Snapdragon 855 and NVIDIA Jetson Nano. The corresponding per-frame inference times are 420 ms, 76 ms, and 97 ms. The Raspberry Pi exhibits significantly longer inference time compared to the other two platforms, as Pi has no GPU and only relies on CPU for inference, resulting in limited parallelism for matrix operations, especially for CNNs. Another interesting finding is that when the model is further compressed into a more lightweight version (LitePose-XS with 1.2 GMACs), the per-frame inference time on Nano and Snapdragon drops to 22 ms and 27 ms, respectively. Although the Snapdragon 855 is theoretically more powerful than the Nano, the latter achieves faster inference speed for this smaller model. This is largely due to its efficient and low-overhead GPU execution via TensorRT. In contrast, Snapdragon relies on higher-overhead delegates, which involve runtime scheduling, memory

Table 4
The comparative analysis and relations among efficiency evaluation metrics used in HPE models.

Metrics	Definitions	Advantages	Limitations	Relations
MACs (Multiply-Add Operations)	One MAC consists of one multiplication followed by one addition	1. Reflects theoretical computation complexity and efficiency of models. 2. Only depends on the model and is independent of hardware platforms; thus, it is comparable cross different models. 3. Aligns well with convolutional layer design.	1. Ignores non-MAC operations (e.g., activation, normalization). 2. Cannot directly reflect practical runtime or inference speed of models. 3. Does not account for preprocessing, postprocessing, as well as data I/O.	1 MACs= 2 FLOPs
FLOPs (Floating-Point Operations)	One FLOP is defined as either a single floating-point multiplication or addition	1. Reflects theoretical computation complexity and efficiency of models. 2. Can be easily compared to the computation capability of hardware. 3. Only depends on the model and is independent of hardware; make it suitable for cross-model comparisons.	1. Ignores memory access and scheduling overhead. 2. Cannot directly reflect practical runtime and throughout of models. 3. Does not include the process of preprocessing, postprocessing, as well as data I/O.	1 FLOPs= 1/2 MACs
Params (Model Parameters)	Measures the model size and storage requirements	1. Indicates overfitting risk and the potential for compression or quantization. 2. Reflects memory and storage requirements.	1. A high Params does not indicate high computation cost. 2. Cannot directly reflect practical runtime or inference speed of models.	—
Inference Time (Per-frame Latency)	Measures the practical inference time or runtime to process a single frame.	1. Directly reflects real-time performance.	1. Varies across different hardware and optimization levels. 2. Only comparable under consistent measurement setup.	$\text{Inference Time(ms)} = \frac{1000}{\text{FPS}}$
FPS (Frames Per Second)	Measures the number of frames processed by the model per second	1. Directly reflects practical inference speed, especially in video-based tasks.	1. Strongly influenced by hardware and batch size. 2. Not comparable across different testing hardware.	$1\text{FPS} = \frac{1000}{\text{Inference Time(ms)}}$
Hz (Hertz, Processing Frequency)	Measures the number of times the model updates its outputs per second	1. Reflects real-time update frequency of models, especially in tracking tasks.	1. Strongly influenced by hardware and input frequency. 2. Not comparable across different hardware settings.	1 Hz ≈ 1 FPS

transfer and operator dispatching between CPU and GPU. These overheads become more pronounced in lightweight models like LitePose-XS, where the computation cost is already low and delegation setup time dominates the total inference time.

Similarly, [31] deployed their model RTMPose-m (1.93 GFLOPs) on three different platforms: NVIDIA GeForce GTX 1660 Ti GPU, Intel i7-11700 CPU, and Snapdragon 865 mobile device, achieving a per-frame inference time of 4.3 ms, 26.6 ms and 32.1 ms, respectively. While both the GTX 1660 Ti and Snapdragon 865 support GPU-based computation, they differ significantly in computation capacity. The GTX 1660 Ti can reach 5.4 TFLOPs in FP 32 computation, far surpassing the 0.5 TFLOPs of Snapdragon 865, which explains the huge gap in inference speed. Additionally, high-end desktop CPUs, such as Intel i7 generally provide higher performance due to faster processing speed, larger memory cache, and better support for parallel computing. This allows them to run small or medium-sized models efficiently, without the delays that often occur on mobile devices when switching between CPU and GPU during inference. When the model size increases to 4.16

Table 5
The hardware used for running HPE models. The listed hardware is categorized into three types: GPU, CPU, and mobile devices. GPUs are further grouped based on their computation capacity. * means the same HPE model was implemented on multiple platforms.

Hardware	Types		# of Papers and References
GPU (77 %)	Consumer GPUs (< 10TFLOPs)	Nvidia GeForce GTX 10 series	14 Papers: [10,22,23,27,47,53,56,59,67,74,77,79,81,87]
		Nvidia GeForce RTX 20 series	7 Papers: [25,33,36] [97]* [14] [9]* [70]
		Nvidia GeForce RTX 30 series	3 Papers: [16–18]
		Nvidia Titan series	9 Papers: [57] [71]* [44] [55]* [48–50,54,92]
		Nvidia GeForce GTX 900 seriesNvidia GeForce MX 200 seriesNvidia GTX 16 series	4 Papers: [19]*[31]* [40,75]
	Data Centre GPUs (10–20 TFLOPs)	Nvidia Tesla series(V100, P100, A100, P40, T4)	8 Papers: [60] [71]* [1,20,45,86,90,104]
	Professional GPUs (>20 TFLOPs)	Nvidia Quadro series	3 Papers: [6,24,65]
		Nvidia RTX A series	1 Papers: [13]
CPU (8 %)	i7-6850K, i7−8700, i7–8700K, i7–11700 Intel@ NUC 6i7KYB mini-PC		5 Papers: [89] [31]* [61]* [55]* [97]*
Mobile (15 %)	Mobile Devices	Mobile phone (Qualcomm Snapdragon 835/855/865GPU, etc.)	7 Papers: [7,30] [83]* [4] [31]* [96] [39]
	Edge Devices	Nvidia Jetson Nano; Coral USB Accelerator; Raspberry Pi 4B+	4 Papers: [19]* [83]* [103] [9]*

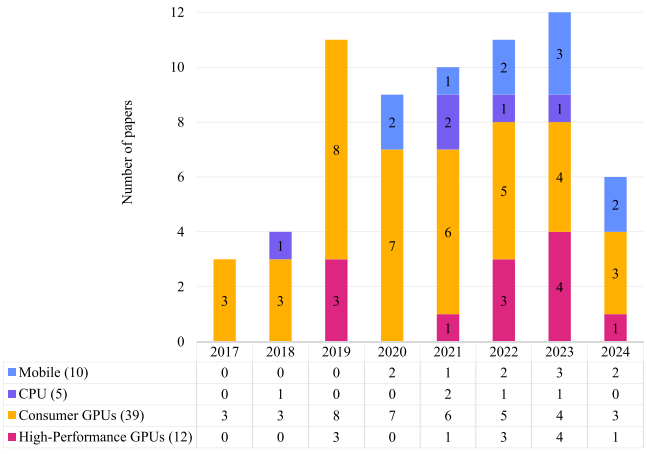


Fig. 12. The distribution of hardware used for implementing HPE models over the years. The high-performance GPUs include both data centre GPUs and professional GPUs.

GFLOPs (RTMPose-l), the inference time rises to 4.6 ms, 36.3 ms and 47.6 ms on the GPU, CPU, and mobile device, respectively. Notably, the inference time on GPU increases slightly compared to the CPU and mobile platform. This is because GPUs are highly parallel and optimized for large-scale tensor operations, allowing them to handle increased FLOPs more efficiently. In contrast, CPUs rely on limited thread-level parallelism, and mobile devices often suffer from delegate overheads and constrained memory bandwidth, both of which contribute to longer inference times as model size grows.

Figs. 13 and 14 show the dynamic changes over years in the computation complexity of HPE models and the computation capacity of implemented hardware. It is apparent that the computation complexity of models has been significantly reduced over time, as reflected by the decreasing GFLOPs (Giga FLOPs=10^9FLOPs). Conversely, hardware computation capacity has steadily improved, as shown by increasing TFLOPs (Tera FLOPs = 10^{12} FLOPs) for GPUs and GFLOPs for CPUs. Fig. 15 only presents the tendency in computation complexity of HPE models targeting mobile devices, as many studies only provide the GFLOPs of models without specifying the mobile device configurations. In order to deploy models on mobile devices, researchers commonly employ lightweight networks to improve efficiency, which may inevitably sacrifice accuracy. However, as the computing power of mobile devices continues to improve over time, researchers have gradually

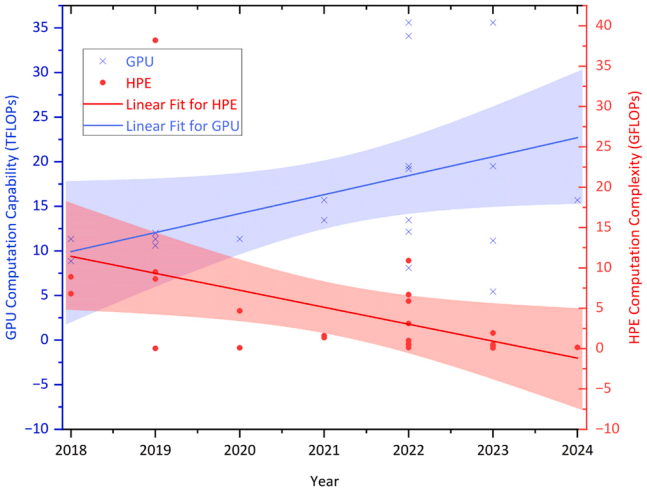


Fig. 13. The trends in the computation complexity of HPE models and the computation capacity of GPUs used to implement them over years.

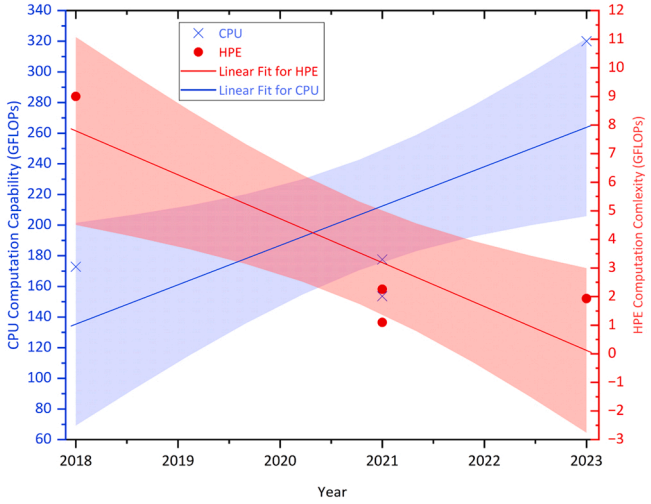


Fig. 14. The trends in the computation complexity of HPE models and the computation capacity of CPUs used to implement them over years.

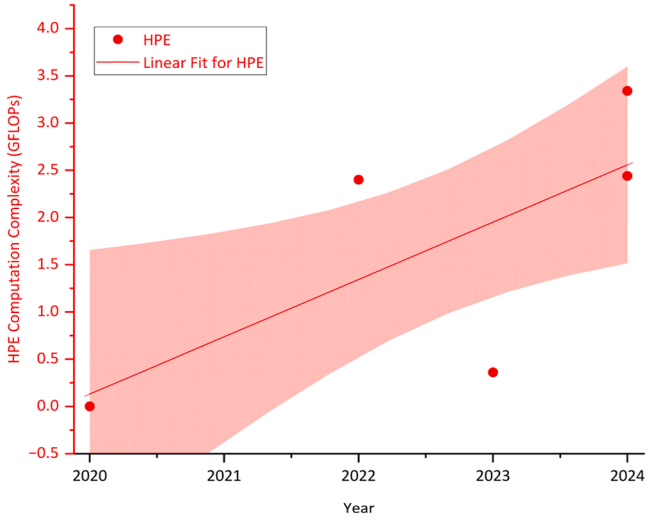


Fig. 15. The trends in the computation complexity of HPE models on mobile devices over years.

eased the strict GFLOPs constraints and shifted their focus toward model accuracy, rather than pursuing extremely low computation cost. Therefore, the increase in GFLOPs of mobile models reflects efforts to enhance accuracy. In addition, the GFLOPs are also influenced by factors such as the number of people in the scene, input resolution, the use of scale search, and the number of keypoints to be predicted. For instance, [39] reported that increasing the input resolution from 256 × 256–512 × 512 caused GFLOPs to rise from 2.81 to 11.23, highlighting the marked computational impact of input size. Moreover, the use of ground-truth bounding boxes in the top-down workflow and the application of multi-scale testing during inference can also significantly affect computation cost.

4.6. *Trade-offs between accuracy and efficiency*

This section discusses how the reviewed solutions balance accuracy and efficiency by answering RQ6. This balance demands reducing model latency while guaranteeing high accuracy. Real-time outputs can offer a better interaction experience, which is important for modern multimedia applications. Table 6 presents various strategies explored by reviewed studies to achieve trade-offs between accuracy and efficiency.

Table 6
The trade-off strategies used for balancing accuracy and efficiency in real-time video-based 2D/3D HPE.

Strategies	Detailed methods
1. Design lightweight networks or reduce redundant branch and steps	1.1 multi-task shares the same backbone 1.2 Knowledge Distillation 1.3 Generative Adversarial Networks (GANs) 1.4 Neural Architecture Search (NAS) 1.5 Gated Attention Unit (GAU) 1.6 selective concatenation-skip connections 1.7 depthwise separable convolution blocks 1.8 dilated convolutions 1.9 consecutive small convolution kernels 1.10 slim transposed convolution for up-sampling 1.11 avoid cropping the input images, searching the scale space and heatmaps refinement 1.12 convert to object detection or classification tasks
2. Utilize temporal redundancy and spatio-temporal relationships of poses across adjacent frames	2.1 select keyframes from a video and only apply HPE model on keyframes 2.2 temporal downsampling strategy to downsample the input 2D pose sequences 2.3 adaptively assign computer resources (i.e. Flops) for different frames 2.4 Temporal Convolutions Networks (TCNs) with dilated convolutions 2.5 Transformer-based architectures 2.6 leverage feature information from previous frames 2.7 propagate body joint locations or bounding boxes from previous frames to the current frame
3. Design multi-stage concurrent pipeline with FIFO queue	

The table illustrates three primary strategies along with their corresponding methods, which are discussed in the following subsections.

4.6.1. *Lightweight and compact models*

To enhance HPE real-time performance, researchers have adopted different strategies to reduce inference time, for instance, designing a lightweight network architecture [7,30,31,36,39,57,59,61,83,90,103]. Additionally, multi-tasks share the same backbone is also effective [53,94]. However, lightweight networks can result in accuracy degradation. To mitigate this drawback, some researchers adopt knowledge distillation (KD)-based training methods to transfer pose structure knowledge from a large teacher network to a small student network [30,90,104]. Another approach is employing GANs to train their proposed models, consisting of a generator and a discriminator. The generator generates temporally or spatially coherent poses, while the discriminator aims to distinguish between ground truth and predicted poses. Moreover, the discriminator can be removed during inference stage, resulting in no additional computation overhead [36,59]. Furthermore, [31] introduced a self-attention mechanism, the Gated Attention Unit (GAU), to more efficiently capture keypoint dependencies when combined with compact CNN-based representations. NAS has also been employed to find optimal network architectures [83,89,97].

Cropping input images to a fixed size is computationally expensive, [57] employed a fully-convolutional network that allows pose estimation directly on non-cropped images. Additionally, searching the scale space in each frame is another time-consuming process, the same authors used 2D keypoints from previous frame to track the bounding box of the current frame, avoiding searching process. Furthermore, [94] employed the scale-normalized image and feature pyramid on the feature extraction stage to avoid the multi-scale testing during inference. [56] introduced selective cross-module long-range and intra-module short-range concatenation-skip connections, which more efficiently enhance information flow through the network without compromising accuracy.

Enlarging the receptive fields can remarkably improve the accuracy of pose estimation and reduce motion jitter, as the large receptive fields can capture more contextual information and long-range spatial-temporal dependencies. Larger convolution kernels [31,83] are an effective way but with increasing computation cost and parameter complexity. To avoid extra computation and memory cost, dilated convolutions (e.g. 1D dilated convolutions for temporal modelling or 2D dilated convolutions for spatial feature extraction) are often employed to enlarge receptive fields without extra computation load [61,65]. Additionally, [48,61] replaced large convolution kernels with depthwise separable convolution blocks, which consist of depthwise convolution and pointwise convolution. [10,67] used consecutive small convolution kernels to replace each large kernel and concatenated the output of each small kernel to preserve both lower-level and higher-level features. Compared with large convolution kernels, these optimizations significantly reduce matrix operations.

Limited resolution is a key factor contributing to impact accuracy in the bottom-up workflow, as this workflow processes the entire image at once, leading to smaller resolution per person in crowded scenes. This limitation increases the difficulty of estimating body keypoints. Maintaining a high-resolution representation can alleviate this problem but consuming high computational resources. Therefore, [83] designed a single-branch architecture that integrates fusion deconvolution layers as the final prediction head. Additionally, the bottom-up workflow suffers from an NP-hard problem, especially the final keypoints grouping stage. [11] introduced a separate branch to generate PAFs, a set of 2D vectors encoding the position and orientation of each limb, and further decomposed a K-dimensional matching problem into bipartite matching subproblems. [67] extended PAFs to TAFs which present the connections between body joints across frames. These methods drastically reduce the computation complexity of the bottom-up workflow without sacrificing accuracy.

High-resolution heatmaps of keypoints can improve accuracy but typically demand high computation cost for extra refinement such as up-sampling. To address this, regression-based methods have been introduced, which straightly regress the keypoints coordinates without generating heatmaps. These methods are more efficient but less accurate compared with heatmap-based methods. Therefore, [7] used heatmaps and offsets to supervise the coordinates regression outputs in training stage, but in inference phrase they removed the corresponding output layers to reduce computation. [31] treated body joint localization as a coordinate classification task without extra refinement. [55] regarded pose estimation as two types of object detection: keypoint objects and pose objects, avoiding heatmap refinement and keypoint grouping. [97] employed slim transposed convolution for up-sampling and followed by a spatial information correction (SIC) module to achieve a trade-off between accuracy and efficiency.

4.6.2. *Temporal redundancy and pose relationships*

For monocular videos, there are three factors that influence HPE efficiency. First, adjacent frames share similar global context information, resulting in temporal redundancy. Second, poses in adjacent frames have strong spatial and temporal relationships. Third, frames are not equally informative and only a small number of frames are critical for capturing the global context. Therefore, applying the same model to each frame is a waste of computation. If only selecting the most informative frames, known as keyframes, from a video to apply pose estimator [98], and then utilizing the temporal correlation of poses to infer the poses of remaining frames can significantly reduce computation cost. Notably, recurrently leveraging information from previous frames is an effective strategy to reduce overall computation [36,57,59,67,89]. For instance, [89] adaptively selected the optimal level of features from previous frames and performed temporal features fusion to reduce

redundancy and enhance consistency. Similarly, [59] designed a Dynamic Kernel Distillation model to transfer pose knowledge from previous frames to current frame. Additionally, propagating body joint locations or bounding boxes from previous frames to the current frame not only alleviate missing or imperfect person detection problems but also reduce latency [7,10,31,45,60,74,87,96]. However, this method is based on individuals do not move fast when the frame rate is relatively high. Therefore, [45] used off-the-shelf motion vectors from compressed videos for fast pose propagation across adjacent frames.

The 2D-to-3D workflow can result in depth ambiguity, modelling temporal information can tackle this issue while improving robustness and motion coherence. However, efficiently exploiting temporal information from input sequences is still challenging in 3D domain. TCNs with dilated convolutions allow to capture long-term temporal dependencies without significantly increasing computation complexity [47,49,50,65,81]. This approach avoids storing numerous parameters and supports parallel processing input frames. In addition, temporal Transformers with self-attention mechanism are also commonly used to explore temporal relationships [20,33,41,42,72,93,95,99,101]. However, this approach limits the length of input sequences, as densely applying self-attention mechanism to long sequences brings a great computational overhead. Therefore, [99] exploited the frequency representation of 2D skeleton poses, they use low-frequency coefficients to represent the entire input sequences, which avoids applying self-attention to all frames as well as filters out high-frequency noise (e.g. jitters and outliers) brought by 2D pose detectors. [72] replaced self-attention operation with a simple MLP block to capture spatial relationships between joints within each frame. [41,72] replaced the parameter-heavy fully-connected layers with strided 1D temporal convolutions to eliminate temporal redundancy of 2D pose sequences. [20,72,93] employed a temporal downsampling strategy to downsample the input pose sequences while preserving the receptive fields. Especially, [20] downsampled input frames with a fixed interval and used the learnable upsampling token as the feature representations for other non-keyframes. [93] adopted a uniform sampling strategy to select 5∼10 % frames (under a certain ratio) of input videos to perform pose estimator. Notably, this method does not rely on frame's features to identify keyframes.

Based on the analysis above, there are three main directions to balance accuracy and efficiency (Table 6). The first strategy is designing lightweight and compact models. However, such models ineluctably suffer from accuracy reduction, especially in complex environments (e.g. occlusion). To avoid this, researchers have introduced some compensatory measurements, such as GAN-based training strategies, teacher-student knowledge distillation or post-processing optimization. The second direction is leveraging temporal redundancy across frames and pose consistency of motion. These methods adaptively operate different models on different frames for obtaining efficiency improvement. The third strategy focuses on designing parallel pipelines in engineering, enabling simultaneous processing of multiple tasks to improve overall efficiency [25]. Finally, different strategies are typically effective for specific network architectures and HPE tasks, a detailed discussion and quantitative comparisons of trade-offs are presented in Section 5.1.

5. Discussion

This paper presents some stable and robust solutions for real-time human pose estimation and tracking on monocular videos, it was done by analysing 68 related papers published between 2014 and 2024, the selection process and exclusion criteria are described in Section 3.2. The analysis of these selected papers was guided by six research questions outlined in Section 3.1. This review highlights recent advancements in monocular video-based HPE, covering both 2D and 3D domains as well as pose tracking. It also provides new insights into hardware configurations and efficiency measures for real-time models, and summarises key factors affecting real-time performance and computational complexity. It is important to note that this review includes literature up to 2024. As expected in a rapidly evolving field, some recent 2025 studies, such as those related to 2D HPE [29,62] and 3D HPE [46,84,91], were excluded to maintain methodological rigour and ensure consistency with the review protocol. This section further compares the performance and complexity of different methods for real-time video-based HPE/HPT and suggests potential directions for future improvement.

5.1. Performance and complexity analysis of different methods

As shown in Section 4.3.1, as well as Tables S1 and S2 of the Supplementary Material, CNNs are the most widely used architecture in real-time 2D video-based HPE and HPT. In contrast, GCNs and Transformers are less frequently adopted. Although their accuracy is comparable to that of CNNs, their high computational complexity makes practical deployment challenging, particularly on edge and mobile devices. In terms of pose estimation workflows, top-down methods generally outperform bottom-up methods in accuracy (see Table S1 of the Supplementary Material). Since top-down workflow first detects person bounding boxes and then estimates body joints within each box, thereby narrowing the search space for keypoint detection. However, on resource-constrained platforms, such as edge and mobile devices, bottom-up workflow has often been adopted due to their higher computational efficiency. Unlike top-down, bottom-up workflow does not require running a person detector for each person in every frame. As a result, they reduce considerable inference time, and their inference speed is irrelevant to the number of people in the scene.

Table S3 in the Supplementary Material shows that 2D-to-3D workflow consistently outperforms the end-to-end workflow in 3D HPE. The main reason is 2D-to-3D workflow relies on accurate 2D joint positions as intermediate representations, which provide strong spatial priors for subsequent 3D pose inference. In contrast, end-to-end workflow directly regresses 3D coordinates or generates heatmaps from videos without the supervision of intermediate 2D poses. Within the 2D-to-3D workflow, TCNs and Transformers are frequently used due to their strong capabilities in modelling spatio-temporal dependencies across consecutive frames. TCNs, particularly dilated TCNs, can model long-range temporal dependencies with parallel processing both over the number of sequences as well as the temporal dimension. However, their sparse sampling strategy limits temporal connectivity, as the dilation factor skips intermediate frames which may cause the loss of critical temporal information. Moreover, dilation increases the receptive field but requires many more layers to model long-time dependencies, making it inefficient and limited for processing long-range sequences. Similarly, temporal Transformers are excellent in exploring the relationships of pose sequences due to self-attention and parallel processing mechanism. Nevertheless, the time and memory complexity of the attention module grows quadratically with the sequence length. Furthermore, information redundancy in long-range 2D pose sequences leads to computation waste, particularly through the fully-connected layers of the FFN of Transformers. As a result, it is not feasible to infinitely enlarge the input sequence to expand the receptive field.

Apart from temporal relationships between body joints across frames, body joints within a single frame also exhibit strong spatial correlations and kinematic constraints, such as local connections, symmetries and global postural constraints. GCNs and spatial Transformers are commonly employed to capture spatial relationships of body joints in each frame. Therefore, they are often integrated with other architectures to explore the spatio-temporal dependencies to infer 3D poses from 2D keypoints. Compared to 2D HPE, such hybrid architecture designs have become increasingly common in 3D HPE. The advantages and limitations of these architectures, along with their effective accuracy-efficiency trade-off strategies, are comprehensively summarised in Table 7. In end-to-end workflow, CNNs remain the most prevalent backbone due to their balance between accuracy and computational

Table 7
Comparative analysis of architectures for real-time video-based HPE.

Architectures	Computation Complexity	Advantages	Limitations	Accuracy-efficiency Trade-off Strategies
CNN	Low *O(N)*	1. Excellent in modelling spatial relations. 2. Highly parallelizable and hardware-friendly on different devices.	1. Limited ability to model temporal dependencies across frames without additional modules.	1. Employ lightweight backbones (e.g., MobileNet, CSPNet). 2. Apply temporal propagation (e.g., keypoints or bounding boxes transfer) or reuse features from previous frames. 3. Regress keypoint coordinates without generating heatmaps
TCN	Low *O(N)*	1. Effective in modelling temporal relationships. 2. Supports parallel computation over time sequences.	1. Cannot directly model spatial relations; relies on input features extracted by CNN or GCN.	1. Use multi-scale dilated convolutions. 2. Combine with CNN for spatio-temporal modelling.
Transformer	High $O(N^2)$	1. Jointly models spatio-temporal dependencies. 2. Captures long-range dependencies effectively. 3. Typically achieves higher accuracy in video-based HPE.	1. High computation cost and memory usage. 2. Not hardware-friendly and challenging to deploy on edge/mobile devices.	1. Simplify attention and FFN modules. 2. Apply temporal downsampling strategies. 3. Leverage pre-training on large-scale motion capture datasets.
GCN	High $O(N^2)$	1. Captures complex spatial dependencies between joints. 2. Well-suited for structured data like human skeletons.	1. High computation cost; involves irregular computation limiting parallelism. 2. Low GPU utilization; challenging for deployment.	1. Integrate with TCN or Transformer for temporal modelling between joints.
Unified Framework	High (varies by design)	1. Combines strengths of different architectures (e.g., CNN, TCN, GCN, Transformer). 2. Offers flexibility in task-specific design.	1. Requires careful system design and optimization. 2. Potentially high overhead if not well-pruned.	1. Apply NAS, pruning, or distillation for optimization. 2. Employ keyframe selection or frame skipping strategies.

efficiency. Although Transformers have also been applied to end-to-end 3D HPE, they typically introduce higher computational cost, as indicated by their FLOPs values in Table S3 of the Supplementary Materials.

Consequently, it is essential to reduce model complexity without compromising accuracy to achieve trade-offs between accuracy and efficiency. CNN-based models benefit from strategies such as replacing heavy backbones with lightweight ones and leveraging temporal dependencies to maintain performance. For example, temporal propagation techniques transfer body joint locations or bounding boxes from previous frames to the current frame to avoid redundant computation [7,10,31,45,59,60,74,87,96]. Alternatively, compared to pixel-level or pose-level fusion strategies, feature-level temporal fusion methods reuse features extracted from previous frames to guide current predictions, enabling more efficient temporal modelling [36,57,59,67,89]. In [45], bounding boxes and keypoint positions are propagated from previous frames, eliminating the need for repeated person detection and multi-scale search in each frame. This strategy significantly increases inference speed from 6.7 to 20.8 FPS, while slightly reducing accuracy from 77.9 to 75.8 mAP. Similarly, [59] replaced the backbone of ResNet-50 with ResNet-18, combined with a lightweight network to transfer pose knowledge across frames, which halves the inference time from 11 ms to 6.5 ms per image with less than a 1 % drop in accuracy.

Another major computational bottleneck in CNN-based models is the generation and post-processing of heatmaps. Thus, directly regressing keypoint coordinates without generating heatmaps has become an effective alternative [7,31,55,97]. For example, [31] removed the upsampling layer typically used for heatmap generation, reducing computation cost from 5.5 to 4.03 GFLOPs. Further replacing the ResNet-50 backbone with a more compact CSPNeXt architecture reduces the GFLOPs to 1.93, without any loss in accuracy. Similarly, [7] adopted a heatmap-to-coordinate regression strategy that improves inference speed from 0.4 to 10 FPS, while maintaining acceptable accuracy (PCK@0.2 drops only from 87.8 to 84.1). These strategies provide substantial gains in computational efficiency while only slightly compromising accuracy, making them highly suitable for real-time applications.

A widely adopted strategy in TCN-based methods is the use of multi-scale dilated convolutions, which allow the model to capture long-term temporal dependencies without significantly increasing computational cost [47,49,50,65,81]. For instance, as reported in [65], a TCN model without dilation and a receptive field of 27 frames achieves an accuracy of 41.1 mm MPJPE at a computation complexity of 59.03 MFLOPs. When applying dilated convolutions with the same receptive field not only reduces the MFLOPs to 17.09, but also improves the accuracy to 40.6 mm MPJPE. Furthermore, even in the largest model with a receptive field of 243 frames using dilated convolutions, the MFLOPs increase modestly to 33.87, while the error is further reduced to 37.8 mm. This highlights that dilated convolutions can efficiently expand the receptive field, with computational complexity growing logarithmically rather than linearly. However, a larger receptive field does not always lead to better performance. Once the receptive field exceeds a certain threshold, the accuracy tends to saturate while the computation cost continues to increase, which suggests that 3D HPE does not require extremely long-term temporal modelling [65].

Transformers have been widely adopted in 3D HPE [20,33,41,42,72,93,95,99,101] due to their strong capability in modelling both spatial and temporal relationships among joints. However, these models bring a great computational overhead (as shown in Table S3), particularly when processing long pose sequences. Therefore, simplifying the attention and feed-forward network (FFN) modules has become an effective strategy to reduce computation load [33,41,72]. For example, [72] replaced the self-attention mechanism with a simple MLP block to capture spatial dependencies between joints, which reduces the computational cost from 1218 to 1094 MFLOPs and improves accuracy by lowering the MPJPE from 48.8 mm to 46.0 mm. In addition, combining temporal downsampling strategies has become a primary strategy to make a trade-off between accuracy and efficiency [20,72,93]. For instance, [72] showed that when the receptive field is fixed at 81 frames, increasing the downsampling rate from 1 to 3 significantly reduces MFLOPs from 493 to 163, with only a minor accuracy degradation (MPJPE increased from 45.6 mm to 46.8 mm). Similarly, [93] sampled only a subset (e.g., 5 %–10 %) of input frames to apply a heavy Transformer, followed by a lightweight interpolation module to recover the extra sequences. This approach adds merely 0.5 MFLOPs of overhead while decreasing MPJPE

from 54.7 mm to 52.8 mm.

Furthermore, [20] employed a fixed-interval downsampling strategy and only fed keyframe features into the Transformer. Non-keyframes were represented using learnable upsampling tokens. This strategy reduces the computation cost from 1358 to 543 MFLOPs, while only slightly increasing MPJPE from 44.3 mm to 45.5 mm. In addition, pre-training on large-scale motion capture datasets (e.g., AMASS) also improves accuracy without introducing additional computation overhead. For example, [20] showed that pre-training on AMASS reduced the MPJPE from 47.5 mm to 45.7 mm. In practice, many real-time models are not deployed or evaluated on high-performance GPUs. Instead, they are often tested on consumer GPUs, CPUs, or mobile devices (refers to Section 4.5). Therefore, single-scale inference without scale search is commonly adopted to ensure real-time performance in realistic deployment scenarios.

As a result, different HPE tasks adopt different strategies to balance accuracy and efficiency. For 2D HPE, employing lightweight networks and propagating body joint locations or bounding boxes from previous frames to the current frame have proven more effective. However, in 3D HPE, downsampling input 2D pose sequences and integrating different network architectures are commonly used to achieve trade-offs between accuracy and efficiency. Overall, leveraging temporal redundancy and pose relationships among frames is a promising direction for further exploration.

5.2. Limitations and future directions

The limitations of this systematic literature review are: (1) This review does not empirically test and evaluate the reviewed methods under standard hardware and input conditions to show their practical deployment limitations. Future work could build on this review by conducting unified experiments to assess the deployment feasibility and robustness of different methods. (2) This review excluded studies focusing on specific populations (e.g., patients, the elderly, or athletes) that train their models on specialized datasets. These studies often involve domain-specific datasets and constrained movement patterns, which limit the generalizability and comparability of their methods. However, such studies hold important application value and represent a meaningful direction for future exploration.

The findings of this review point out a few potential trends and directions for future studies. One of the main challenges in 3D HPE is the lack of in-the-wild ground-truth 3D datasets (see Fig. 10). To address this, some studies adopted semi-supervised approaches [65]. Several research leveraged the large-scale AMASS motion dataset to pre-train their models utilizing GAN-based learning approaches [6,20,36]. However, reducing reliance on ground-truth pose datasets or designing unsupervised models remain unaddressed. Additionally, how to quickly generate in-the-wild ground-truth 3D pose datasets using cost-friendly technologies requires further exploration.

3D HPE can output either absolute coordinates (camera-centre) or relative coordinates (person-centre). Among the 30 reviewed papers in the 3D domain, only a few works output 3D absolute coordinates [22,56], and a few metrics are introduced for evaluating the accuracy of absolute coordinates (as shown in Table 3). This is primarily because translating 2D poses from pixel coordinates to absolute 3D coordinates (in cm/m) requires camera intrinsic and extrinsic parameter matrices, which are unavailable in most datasets (see Section 4.4). Moreover, translating relative 3D coordinates to absolute ones also requires a reference object with known position and scale, which is inherently difficult to estimate from monocular video. Nevertheless, generating absolute 3D poses has promising potential for real-world applications [5]. Future research should focus on addressing these challenges from multiple perspectives, such as dataset development, evaluation metric design or prospective model optimization, to improve the robustness and accuracy of absolute 3D pose estimation.

The reviewed papers indicate that real-time 3D HPT on monocular videos remains underexplored, primarily due to limitations in the motion coherence and smoothness of current 3D HPE methods, the scarcity of 3D pose tracking datasets, and the inherent complexity of 3D multi-person HPE. Also, accuracy metrics for evaluating motion coherence are still scarce (see Table 3), inspiring the need to develop frame-level accuracy evaluation metrics for quantifying motion smoothness. Moreover, designing HPE and HPT models targeting mobile and edge devices is a promising future direction, but one must consider factors like power and memory efficiency.

6. Conclusion

In this study, we provide a systematic literature review of real-time human pose estimation and tracking on monocular videos, analysing data from 68 papers published from 2014 to 2024. This review identified different factors influencing the evolution of real-time solutions, encompassing outputs, workflows, algorithms, performance evaluation, as well as trade-off strategies between accuracy and efficiency. It provides an in-depth discussion of different algorithms by analysing their performance and complexity across various outputs and workflows. For performance evaluation, this study thoroughly compares different efficiency evaluation metrics, further revealing the reasons behind substantial gaps in inference speed across different hardware. Furthermore, three primary strategies employed by the reviewed studies to improve efficiency without compromising accuracy are identified. Based on the architecture types and target tasks, we provide tailored strategy recommendations and quantitative comparisons.

Finally, findings indicate that 2D multi-person and 3D single-person HPE are dominant in real-time fields based on monocular videos. While 3D multi-person real-time tracking is rarely explored due to its complexity, limited availability of 3D tracking datasets and the developments of 3D multi-person HPE.

Beyond summarising existing methods, this review aims to facilitate the practical deployment of human pose estimation and tracking systems by highlighting strategies that balance accuracy and efficiency, as well as hardware constraints. Its findings may serve as valuable guidance for developing real-time solutions adaptable to real-world applications, especially in safety-critical or resource-limited scenarios.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The NZ Ministry of Business, Innovation & Employment and Royal Society Te Apārangi fund this research through the Marsden Fast Start (MAU2204) and the Rutherford Discovery Fellowship (RDF-MAU2201). The authors acknowledge the use of ChatGPT for language improvements and take full responsibility for the content.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.neucom.2025.131309.

Data availability

No data was used for the research described in the article.

References

[1] Bilal Abdulrahman, Zhigang Zhu, Absolute-ROMP: absolute Multi-Person 3D mesh prediction from a single image. In Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2023, SCITEPRESS - Science and Technology Publications, Lisbon, Portugal, 2023, pp. 69–79, https://doi.org/10.5220/0011629500003417.

[2] Jinguji Akira, Tomoya Fujii, Shimpei Sato, Hiroki Nakahara, An FPGA realization of OpenPose based on a sparse weight convolutional neural network. In 2018 International Conference on Field-Programmable Technology (FPT), December 2018, IEEE, Naha, Okinawa, Japan, 2018, pp. 310–313, https://doi.org/10.1109/FPT.2018.00061.

[3] Mykhaylo Andriluka, Stefan Roth, Bernt Schiele, Pictorial structures revisited: people detection and articulated pose estimation. IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1014–1021, 10.1109/CVPR.2009.5206754.

[4] Yukihiko Aoyagi, Shigeki Yamada, Shigeo Ueda, Chifumi Iseki, Toshiyuki Kondo, Keisuke Mori, Yoshiyuki Kobayashi, Tadanori Fukami, Minoru Hoshimaru, Masatsune Ishikawa, Yasuyuki Ohta, Development of smartphone application for markerless Three-Dimensional motion capture based on deep learning model, Sensors 22 (14) (2022) 5282, https://doi.org/10.3390/s22145282.

[5] Aritz Badiola-Bengoa, Amaia Mendez-Zorrilla, A systematic review of the application of Camera-Based human pose estimation in the field of sport and physical exercise, Sensors 21 (18) (2021) 5996, https://doi.org/10.3390/s21185996.

[6] Fabien Baradel, Romain Bregier, Thibault Groueix, Philippe Weinzaepfel, Yannis Kalantidis, Gregory Rogez. PoseBERT: A Generic Transformer Module for Temporal 3D Human Modeling. Retrieved August 20, 20242022, ⟨http://arxiv.org/abs/2208.10211⟩.

[7] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, Matthias Grundmann, BlazePose: On-device Real-time, Body Pose Track. (2020), https://doi.org/10.48550/arXiv.2006.10204.

[8] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, Slobodan Ilic, 3D pictorial structures for multiple human pose estimation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Columbus, OH, USA, 2014, pp. 1669–1676, https://doi.org/10.1109/CVPR.2014.216.

[9] Michele Boldo, Damiano Carra, Davide Quaglia, Nicola Bombieri, A dynamic and collaborative deep inference framework for human motion analysis in telemedicine. IEEE International Conference on Edge Computing and Communications (EDGE), IEEE, Chicago, IL, USA, 2023, pp. 221–226, https://doi.org/10.1109/EDGE60047.2023.00043.

[10] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, Yaser Sheikh, OpenPose: realtime multi-person 2D pose estimation using part affinity fields, arXiv (2019), https://doi.org/10.48550/arXiv.1812.08008.

[11] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, Yaser Sheikh, OpenPose: realtime multi-person 2D pose estimation using part affinity fields, arXiv (2019), https://doi.org/10.48550/arXiv.1812.08008.

[12] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2016. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. arXiv.org. Retrieved August 6, 2024 from ⟨https://arxiv.org/abs/1611.08050v2⟩.

[13] Haijian Chen, Xinyun Jiang, Yonghui Dai, Shift pose: a lightweight Transformer-like neural network for human pose estimation, Sensors 22 (19) (2022) 7264, https://doi.org/10.3390/s22197264.

[14] Mickael Cormier, Aris Clepe, Andreas Specker, Jürgen Beyerer, Where are we with human pose estimation in Real-World surveillance, in: IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), January, 2022, 2022, pp. 591–601, https://doi.org/10.1109/WACVW54805.2022.00065.

[15] Gisela Miranda Difini, Marcio Garcia Martins, Jorge Luis Victória Barbosa, Human pose estimation for training assistance: a systematic literature review. In Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia '21), November 05, 2021. Association for Computing Machinery, 2021, pp. 189–196, https://doi.org/10.1145/3470482.3479633.

[16] Chengang Dong, Guodong Du, An enhanced real-time human pose estimation method based on modified YOLOv8 framework, Sci. Rep. 14 (1) (2024) 8012, https://doi.org/10.1038/s41598-024-58146-z.

[17] Chengang Dong, Yuhao Tang, Liyan Zhang, HDA-pose: a real-time 2D human pose estimation method based on modified YOLOv8, Signal Image Video Process 18 (8–9) (2024) 5823–5839, https://doi.org/10.1007/s11760-024-03274-2.

[18] Chengang Dong, Yuhao Tang, Liyan Zhang, MDA-YOLO person: a 2D human pose estimation model based on YOLO detection framework, Clust. Comput. 27 (9) (2024) 12323–12340, https://doi.org/10.1007/s10586-024-04608-y.

[19] Bruno Carlos Dos Santos Melício, Gábor Baranyi, Zsófia Gaál, Sohil Zidan, András Lőrincz, DeepRehab: real time pose estimation on the edge for knee injury rehabilitation, in: Igor Farkaš, Paolo Masulli, Sebastian Otte, Stefan Wermter (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2021, Springer International Publishing, Cham, 2021, pp. 380–391, https://doi.org/10.1007/978-3-030-86365-4_31.

[20] Moritz Einfalt, Katja Ludwig, Rainer Lienhart, 2023, Uplift and upsample: efficient 3D human pose estimation with uplifting transformers. arXiv.org. Retrieved September 5, 2024 from https://arxiv.org/abs/2210.06110v3.

[21] Amal El Kaid, Karim Baïna, A systematic review of recent deep learning approaches for 3D human pose estimation, J. Imaging 9 (12) (2023) 275, https://doi.org/10.3390/jimaging9120275.

[22] Amal El Kaid, Denis Brazey, Vincent Barra, Karim Baïna, Top-Down system for Multi-Person 3D absolute pose estimation from monocular videos, Sensors 22 (11) (2022) 4109, https://doi.org/10.3390/s22114109.

[23] Ahmed Elhayek, Onorina Kovalenko, Pramod Murthy, Jameel Malik, Didier Stricker, Fully automatic Multi-person human motion capture for VR applications. Virtual Reality and Augmented Reality, 2018, Springer International Publishing, Cham, 2018, pp. 28–47, https://doi.org/10.1007/978-3-030-01790-3_3.

[24] Taosha Fan, Kalyan Vasudev Alwala, Donglai Xiang, Weipeng Xu, Todd Murphey, Mustafa Mukadam, Revitalizing optimization for 3D human pose and shape estimation: a sparse constrained formulation, arXiv (2021), https://doi.org/10.48550/arXiv.2105.13965.

[25] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. 2022. AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time. Retrieved July 24, 2024 from ⟨http://arxiv.org/abs/2211.03375⟩.

[26] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. 2021. Human POSEitioning System (HPS): 3D Human Pose Estimation and Self-Localization in Large Scenes From Body-Mounted Sensors. 2021. 4318–4329. Retrieved August 4, 2025 from ⟨https://openaccess.thecvf.com/content/CVPR2021/html/Guzov_Human_POSEitioning_System_HPS_3D_Human_Pose_Estimation_and_Self-Localization_CVPR_2021_paper.html⟩.

[27] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2019. LiveCap: Real-time Human Performance Capture from Monocular Video. ⟨https://doi.org/10.48550/arXiv.1810.02648⟩.

[28] Milad Haghani, Ruggiero Lovreglio, Data-based tools can prevent crowd crushes, Science 378 (6624) (2022) 1060–1061, https://doi.org/10.1126/science.adf5949.

[29] Yuze He, Ke Chen, Juanjuan Hu, PoseTrackNet: integrating advanced techniques for accurate and robust human pose estimation in dynamic environments, Alex. Eng. J. 122 (2025) 152–164, https://doi.org/10.1016/j.aej.2025.02.094.

[30] Dong-Hyun Hwang, Suntae Kim, Nicolas Monet, Hideki Koike, Soonmin Bae, Lightweight 3D Hum. Pose Estim. Netw. Train. Using Teach. Stud. Learn. (2020), https://doi.org/10.48550/arXiv.2001.05097.

[31] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, Kai Chen, RTMPose: real-time multi-person pose estimation based on MMPose, Pose Estim. Based MMPose (2023), https://doi.org/10.48550/arXiv.2303.07399.

[32] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, Ping Luo, Whole-Body human pose estimation in the wild. Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 196–214, https://doi.org/10.1007/978-3-030-58545-7_12.

[33] Hongbo Kang, Yong Wang, Mengyuan Liu, Doudou Wu, Peng Liu, and Wenming Yang. 2023. Double-chain Constraints for 3D Human Pose Estimation in Images and Videos. Retrieved July 29, 2024 from ⟨http://arxiv.org/abs/2308.05298⟩.

[34] Min-Seok Kang, Dongoh Kang, HanSaem Kim, Efficient Skeleton-Based action recognition via Joint-Mapping strategies. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), IEEE, Waikoloa, HI, USA, 2023, pp. 3392–3401, https://doi.org/10.1109/WACV56688.2023.00340.

[35] Khalid S. Khan, Regina Kunz, Jos Kleijnen, Gerd Antes, Five steps to conducting a systematic review, J. R. Soc. Med. 96 (3) (2003) 118–121, https://doi.org/10.1177/014107680309600304.

[36] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. 2020. VIBE: Video Inference for Human Body Pose and Shape Estimation. Retrieved July 24, 2024 from ⟨http://arxiv.org/abs/1912.05656⟩.

[37] Nikos Kolotouros, Georgios Pavlakos, J.Black Michael, Kostas Daniilidis, Learning to reconstruct 3D human pose and shape via model-fitting in the loop, arXiv (2019), https://doi.org/10.48550/arXiv.1909.12828.

[38] Laxman Kumarapu, Prerana Mukherjee, AnimePose: Multi-person 3D pose estimation and animation, Pattern Recognit. Lett. 147 (2021) 16–24, https://doi.org/10.1016/j.patrec.2021.03.028.

[39] Gongjin Lan, Yu Wu, Qi Hao, DIR-BHRNet: a lightweight network for Real-Time Vision-Based multiperson pose estimation on smartphones, IEEE Trans. Ind. Inf. (2024) 1–9, https://doi.org/10.1109/TII.2024.3421511.

[40] Lorenzo Landolfi, Paolo Tripicchio, Alessandro Filippeschi, Carlo Alberto Avizzano, Fast and fluid human pose tracking. In 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR), IEEE, Irkutsk, Russia, 2019, pp. 24–29, https://doi.org/10.1109/RCAR47638.2019.9044037.

[41] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, Wenming Yang, Exploiting temporal contexts with strided transformer for 3D human pose estimation, arXiv (2022), https://doi.org/10.48550/arXiv.2103.14304.

[42] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, Luc Van Gool, MHFormer: Multi-hypothesis transformer for 3D human pose estimation, arXiv (2022), https://doi.org/10.48550/arXiv.2111.12707.

[43] Marko Linna, Juho Kannala, Esa Rahtu, Real-time human pose estimation from video with convolutional neural networks, arXiv (2016), https://doi.org/10.48550/arXiv.1609.07420.

[44] Marko Linna, Juho Kannala, Esa Rahtu, Real-time human pose estimation with convolutional neural networks, in: In Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2018, SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal, 2018, pp. 335–342, https://doi.org/10.5220/0006624403350342.

[45] Huan Liu, Wentao Liu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, Jin Tang, 2023. Fast human pose estimation in compressed videos, IEEE Trans. Multimed. 25 (2023) 1390–1400, https://doi.org/10.1109/TMM.2022.3141888.

[46] Jiajie Liu, Mengyuan Liu, Hong Liu, Wenhao Li, TCPFormer: Learning Temporal Correlation with Implicit Pose Proxy for 3D Human Pose Estimation, arXiv (2025), https://doi.org/10.48550/arXiv.2501.01770.
[47] Junfa Liu, Juan Rojas, Zhijun Liang, Yihui Li, Yisheng Guan, A Graph Attention Spatio-temporal Convolutional Network for 3D Human Pose Estimation in Video, arXiv (2020), https://doi.org/10.48550/arXiv.2003.14179.
[48] Leiming Liu, Linghao Lin, Jiangyun Li, Research on human pose estimation and object detection in the field of unmanned retail, in: Huansheng Ning, Feifei Shi (Eds.), Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health, Springer Singapore, Singapore, 2020, pp. 131–141, https://doi.org/10.1007/978-981-33-4336-8_11.
[49] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, Vijayan Asari, Attention mechanism exploits temporal contexts: Real-Time 3D human pose reconstruction. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Seattle, WA, USA, 2020, pp. 5063–5072, https://doi.org/10.1109/CVPR42600.2020.00511.
[50] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, Vijayan K. Asari, Enhanced 3D Human Pose Estimation from Videos by using Attention-Based Neural Network with Dilated Convolutions, arXiv (2021), https://doi.org/10.48550/arXiv.2103.03170.
[51] Wu Liu, Qian Bao, Yu Sun, Tao Mei, Recent advances of monocular 2D and 3D human pose estimation: a deep learning perspective, ACM Comput. Surv. 55 (4) (2022), https://doi.org/10.1145/3524497.
[52] Xiaoxu Liu, Xiaoyi Feng, Shijie Pan, Jinye Peng, Xuan Zhao, Skeleton tracking based on kinect camera and the application in virtual reality system. In Proceedings of the 4th International Conference on Virtual Reality (ICVR 2018), Association for Computing Machinery, New York, NY, USA, 2018, pp. 21–25, https://doi.org/10.1145/3198910.3198915.
[53] Diogo C. Luvizon, Hedi Tabia, David Picard, Multi-task deep learning for Real-Time 3D human pose estimation and action recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2020) 1, https://doi.org/10.1109/TPAMI.2020.2976014.
[54] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. 2017. A simple yet effective baseline for 3d human pose estimation. arXiv.org. Retrieved September 2, 2024 from ⟨https://arxiv.org/abs/1705.03098v2⟩.
[55] William McNally, Kanav Vats, Alexander Wong, John McPhee, Rethinking Keypoint Representations: Modeling Keypoints and Poses as Objects for Multi-Person Human Pose Estimation, arXiv (2022), https://doi.org/10.48550/arXiv.2111.08557.
[56] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. 2020. XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera. 2020. Retrieved August 21, 2024 from ⟨https://vcai.mpi-inf.mpg.de/projects/XNect/⟩.
[57] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, Christian Theobalt, VNect: Real-time 3D human pose estimation with a single RGB camera, ACM Trans. Graph 36 (4) (2017) 1–14, https://doi.org/10.1145/3072959.3073596.
[58] G. Mori, J. Malik, Recovering 3D human body configurations using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 28 (7) (2006) 1052–1062, https://doi.org/10.1109/TPAMI.2006.149.
[59] Xuecheng Nie, Yuncheng Li, Linjie Luo, Ning Zhang, Jiashi Feng, Dynamic kernel distillation for efficient pose estimation in videos. IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Seoul, Korea (South), 2019, pp. 6941–6949, https://doi.org/10.1109/ICCV.2019.00704.
[60] Guanghan Ning, Heng Huang, LightTrack: A Generic Framework for Online Top-Down Human Pose Tracking, arXiv (2019), https://doi.org/10.48550/arXiv.1905.02822.
[61] Daniil Osokin. 2018. Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose. ⟨https://doi.org/10.48550/arXiv.1811.12004⟩.
[62] Cesare Davide Pace, Alessandro Marco De Nunzio, Claudio De Stefano, Francesco Fontanella, and Mario Molinara. 2025. Poseidon: A ViT-based Architecture for Multi-Frame Pose Estimation with Adaptive Frame Weighting and Multi-Scale Feature Fusion. ⟨https://doi.org/10.48550/arXiv.2501.08446⟩.
[63] Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjorn Hrobjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, David Moher, The PRISMA 2020 statement: an updated guideline for reporting systematic reviews, BMJ 372 (2021), https://doi.org/10.1136/bmj.n71.
[64] David Pascual-Hernández, Nuria Oyaga De Frutos, Inmaculada Mora-Jiménez, José María Cañas-Plaza, Efficient 3D human pose estimation from RGBD sensors, Displays 74 (2022) 102225, https://doi.org/10.1016/j.displa.2022.102225.
[65] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 3D human pose estimation in video with temporal convolutions and semi-supervised training. Retrieved July 29, 2024 from ⟨http://arxiv.org/abs/1811.11742⟩.
[66] Ilias Poulios, Theodora Pistola, Spyridon Symeonidis, Sotiris Diplaris, Konstantinos Ioannidis, Stefanos Vrochidis, Ioannis Kompatsiaris, Enhanced real-time motion transfer to 3D avatars using RGB-based human 3D pose estimation. Proceedings of the 2024 ACM International Conference on Interactive Media Experiences Workshops, ACM, Stockholm Sweden, 2024, pp. 88–99, https://doi.org/10.1145/3672406.3672427.
[67] Yaadhav Raaj, Haroon Idrees, Gines Hidalgo, Yaser Sheikh, Efficient Online Multi-Person 2D Pose Tracking with Recurrent Spatio-Temporal Affinity Fields, arXiv (2019), https://doi.org/10.48550/arXiv.1811.11975.
[68] Samuele Salti, Oliver Schreer, Luigi Di Stefano, Real-time 3d arm pose estimation from monocular video for enhanced HCI. In Proceedings of the 1st ACM workshop on Vision networks for behavior analysis (VNBA '08), Association for Computing Machinery, New York, NY, USA, 2008, pp. 1–8, https://doi.org/10.1145/1461893.1461895.
[69] Esraa Samkari, Muhammad Arif, Manal Alghamdi, Mohammed A. Al Ghamdi, Human pose estimation using deep learning: a systematic literature review, Mach. Learn. Knowl. Extr. 5 (4) (2023) 1612–1659, https://doi.org/10.3390/make5040081.
[70] Djalma Lúcio Luiz Schirmer, Luiz Velho Alberto Raposo, Hélio Lopes, A lightweight 2D pose machine with attention enhancement. 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2020, pp. 324–331, https://doi.org/10.1109/SIBGRAPI51738.2020.00051.
[71] Luiz José Schirmer Silva, Djalma Lúcio Soares da Silva, Alberto Barbosa Raposo, Luiz Velho, Hélio Côrtes Vieira Lopes, Tensorpose: Real-time pose estimation for interactive applications, Comput. Graph 85 (2019) 1–14, https://doi.org/10.1016/j.cag.2019.08.013.
[72] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. 2022. P-STMO: Pre-Trained Spatial Temporal Many-to-One Model for 3D Human Pose Estimation. Retrieved September 11, 2024 from ⟨http://arxiv.org/abs/2203.07628⟩.
[73] Serhii Shapoval, Begoña García Zapirain, Amaia Mendez Zorrilla, Iranzu Mugueta-Aguinaga, Biofeedback applied to interactive serious games to monitor frailty in an elderly population, Appl. Sci. 11 (8) (2021) 3502, https://doi.org/10.3390/app11083502.
[74] Michael Snower, Asim Kadav, Farley Lai, Hans Peter Graf, 15 Keypoints Is. All You Need (2020), https://doi.org/10.48550/arXiv.1912.02323.
[75] Haixun Sun, Yanyan Zhang, Yijie Zheng, Jianxin Luo, Zhisong Pan, G2O-Pose: Real-Time monocular 3D human pose estimation based on general graph optimization, Sensors 22 (21) (2022) 8335, https://doi.org/10.3390/s22218335.
[76] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. ⟨https://doi.org/10.48550/arXiv.1902.09212⟩.
[77] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J. Black, Tao Mei, Monocular, One-stage, Regression of Multiple 3D People, arXiv (2021), https://doi.org/10.48550/arXiv.2008.12272.
[78] Alexander Toshev and Christian Szegedy. 2014. DeepPose: Human Pose Estimation via Deep Neural Networks. 2014. 1653–1660. Retrieved December 18, 2024 from ⟨https://openaccess.thecvf.com/content_cvpr_2014/html/Toshev_DeepPose_Human_Pose_2014_CVPR_paper.html⟩.
[79] Madhawa Vidanpathirana, Imesha Sudasingha, Jayan Vidanapathirana, Pasindu Kanchana, Indika Perera, Tracking and frame-rate enhancement for real-time 2D human pose estimation, Vis. Comput. 36 (7) (2020) 1501–1519, https://doi.org/10.1007/s00371-019-01757-9.
[80] Chen Wang, Feng Zhang, Shuzhi Sam Ge, A comprehensive survey on 2D multi-person pose estimation methods, Eng. Appl. Artif. Intell. 102 (2021) 104260, https://doi.org/10.1016/j.engappai.2021.104260.
[81] Guangming Wang, Honghao Zeng, Ziliang Wang, Zhe Liu, and Hesheng Wang. 2021. Motion Projection Consistency Based 3D Human Pose Estimation with Virtual Bones from Monocular Videos. arXiv.org. Retrieved September 5, 2024 from ⟨https://arxiv.org/abs/2106.14706v2⟩.
[82] tao wang, Jianfeng Zhang, Yujun Cai, Shuicheng Yan, Jiashi Feng, Direct Multi-view Multi-person 3D pose estimation, in: Advances in Neural Information Processing Systems, 2021, Curran Associates, Inc, 2021, pp. 13153–13164, in: ⟨https://proceedings.neurips.cc/paper/2021/hash/6da9003b743b65f4c0ccd295cc484e57-Abstract.html⟩.
[83] Yihan Wang, Muyang Li, Han Cai, Wei-Ming Chen, Song Han, Lite Pose: Efficient Architecture Design for 2D Human Pose Estimation, arXiv (2022), https://doi.org/10.48550/arXiv.2205.01271.
[84] Mingjie Wei, Xuemei Xie, Yutong Zhong, Guangming Shi, Learning Pyramid-structured Long-range dependencies for 3D human pose estimation, IEEE Trans. Multimed. (2025) 1–14, https://doi.org/10.1109/TMM.2025.3535349.
[85] Claes Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering. Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14), Association for Computing Machinery, New York, NY, USA, 2014, pp. 1–10, https://doi.org/10.1145/2601248.2601268.
[86] Xinyao Xi, Chen Zhang, Wen Jia, Ruxue Jiang, Enhancing human pose estimation in sports training: integrating spatiotemporal transformer for improved accuracy and real-time performance, Alex. Eng. J. 109 (2024) 144–156, https://doi.org/10.1016/j.aej.2024.08.072.
[87] Bin Xiao, Haiping Wu, Yichen Wei, Simple Baselines for Human Pose Estimation and Tracking, arXiv (2018), https://doi.org/10.48550/arXiv.1804.06208.
[88] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, Cewu Lu, Pose Flow: Efficient Online Pose Tracking, arXiv (2018), https://doi.org/10.48550/arXiv.1802.00977.
[89] Lumin Xu, Yingda Guan, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, Xiaogang Wang, ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search, arXiv (2021), https://doi.org/10.48550/arXiv.2105.10154.
[90] Yufei Xu, Jing Zhang, Qiming Zhang, Dacheng Tao, ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation, arXiv (2022), https://doi.org/10.48550/arXiv.2204.12484.

[91] Mingrui Ye, Lianping Yang, Hegui Zhu, Zenghao Zheng, Xin Wang, Yantao Lo, Dual-stream Transformer-GCN Model with Contextualized Representations Learning for Monocular 3D Human Pose Estimation, arXiv (2025), https://doi.org/10.48550/arXiv.2504.01764.
[92] Anastasios Yiannakides, Andreas Aristidou, Yiorgos Chrysanthou, Real-time 3D human pose and motion reconstruction from monocular RGB videos, Comput. Animat. Virtual Worlds 30 (3–4) (2019) e1887, https://doi.org/10.1002/cav.1887.
[93] Ailing Zeng, Xuan Ju, Lei Yang, Ruiyuan Gao, Xizhou Zhu, Bo Dai, Qiang Xu, DeciWatch: A Simple Baseline for 10x Efficient 2D and 3D Pose Estimation, arXiv (2022), https://doi.org/10.48550/arXiv.2203.08713.
[94] Jiabin Zhang, Zheng Zhu, Wei Zou, Peng Li, Yanwei Li, Hu Su, Guan Huang, FastPose: Towards Real-time Pose Estimation and Tracking via Scale-normalized Multi-task Networks, arXiv (2019), https://doi.org/10.48550/arXiv.1908.05593.
[95] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, Junsong Yuan, MixSTE: Seq2seq mixed Spatio-Temporal encoder for 3D human pose estimation in video. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA, 2022, pp. 13222–13232, https://doi.org/10.1109/CVPR52688.2022.01288.
[96] Jinrui Zhang, Deyu Zhang, Huan Yang, Yunxin Liu, Ju Ren, Xiaohui Xu, Fucheng Jia, Yaoxue Zhang, MVPose: realtime Multi-Person pose estimation using motion vector on mobile devices, IEEE Trans. Mob. Comput. 22 (6) (2023) 3508–3524, https://doi.org/10.1109/TMC.2021.3139940.
[97] Wenqiang Zhang, Jiemin Fang, Xinggang Wang, Wenyu Liu, EfficientPose: Efficient Human Pose Estimation with Neural Architecture Search, arXiv (2020). ⟨http://arxiv.org/abs/2012.07086⟩.
[98] Yuexi Zhang, Yin Wang, Octavia Camps, Mario Sznaier, Key Frame Proposal Network for Efficient Pose Estimation in Videos, arXiv (2020), https://doi.org/10.48550/arXiv.2007.15217.
[99] Qitao Zhao, Ce Zheng, Mengyuan Liu, Pichao Wang, Chen Chen. 2023. PoseFormerV2: Exploring Frequency Domain for Efficient and Robust 3D Human Pose Estimation. Retrieved July 29, 2024 from , and ⟨http://arxiv.org/abs/2303.17472⟩.
[100] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, Mubarak Shah, Deep Learning-based human pose estimation: a survey, ACM Comput. Surv. 56 (1) (2023), https://doi.org/10.1145/3603618. Chen.
[101] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, Zhengming Ding, 3D Human Pose Estimation with Spatial and Temporal Transformers, arXiv (2021), https://doi.org/10.48550/arXiv.2103.10455. Chen.
[102] Lijuan Zhou, Xiang Meng, Zhihuan Liu, Mengqi Wu, Zhimin Gao, Pichao Wang, Human Pose-based Estimation, Tracking and Action Recognition with Deep Learning: A Survey, arXiv (2023), https://doi.org/10.48550/arXiv.2310.13039.
[103] Qihua Zhou, Song Guo, Jun Pan, Jiacheng Liang, Jingcai Guo, Zhenda Xu, Jingren Zhou, PASS: patch automatic skip scheme for efficient On-Device video perception, IEEE Trans. Pattern Anal. Mach. Intell. 46 (5) (2024) 3938–3954, https://doi.org/10.1109/TPAMI.2024.3350380.
[104] Xiaomao Zhou, Xiaolong Yu, Cheng Xu, Fast and accurate pose estimation in videos based on knowledge distillation and pose propagation. In 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022, pp. 1–8, https://doi.org/10.1109/IJCNN55064.2022.9892706.

Yingying Chen is a PhD student at the School of Built Environment at Massey University (NZ). Her research investigates pedestrian evacuation dynamics using computer vision and synthetic data.

Dr Zhenan Feng is a Senior Lecturer in Digital Built Environment in the School of Built Environment at Massey University (NZ). Dr Feng has teaching and research experience on digital technologies for built environment, such as virtual reality, augmented reality, serious games, building information modelling, and digital construction.

Dr Daniel Paes is a Senior Lecturer in Digital Built Environment in the School of Built Environment at Massey University (NZ). Dr Paes's works explore how virtual and augmented reality, serious games, BIM, unmanned-aerial systems, and other digital-construction tools can improve design decision-making, workforce training, and on-site safety.

Dr Daniel Nilsson is a Professor at the Department of Civil and Natural Resources Engineering at the University of Canterbury. His area of expertise is Evacuation, and my main interest is the interaction between people and evacuation systems, e.g., way-finding systems and notification systems.

Dr Ruggiero Lovreglio is a Professor at the School of Built Environment at Massey University (NZ). He is a Rutherford Discovery Fellow for the Royal Society Te Apārangi. Prof Lovreglio has teaching and research experience on digital technologies for built environment, such as virtual reality, augmented reality, serious games, building information modelling, and digital construction