POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Self-Supervised 3D Human Pose Estimation in Sports Aerial Videos

Tesi di Laurea Magistrale in
Computer Science and Engineering -
Ingegneria Informatica

Author: **Angelo Attivissimo**

Student ID: 233484
Advisor: Prof. Giacomo Boracchi
Co-advisors: Diego Martin
Academic Year: 2024-25

Abstract

3D Human Pose Estimation (3D HPE) aims to recover the three-dimensional configuration of the human body from visual data and has achieved remarkable progress in recent years thanks to deep learning and large-scale datasets. In sports, accurate 3D pose reconstruction can support performance analysis, injury prevention, and tactical evaluation. However, current state-of-the-art methods are mainly designed for controlled environments or ground-level cameras, and are not tailored to aerial videos.

Drone platforms offer a flexible and scalable solution for outdoor sports scenarios, where the monitored area can be very large and the installation of multiple calibrated fixed cameras is impractical or economically unfeasible. Unmanned Aerial Vehicles (UAVs) can dynamically adjust viewpoint, height, and coverage, enabling data acquisition in complex environments. Nevertheless, aerial footage introduces additional challenges, including viewpoint variability, motion blur, scale changes, and frequent occlusions.

In this thesis, I propose Brancher, a deep learning framework built upon a state-of-the-art backbone for single-person, single-view 3D HPE from sports aerial videos. To the best of our knowledge, this is among the first works to explicitly address 3D human pose reconstruction from monocular drone footage using a fully deep learning approach. The model follows a self-supervised training strategy and extends standard 3D regression by jointly modeling spatio-temporal joint dynamics, limb rotations, and prediction uncertainty. By explicitly accounting for rotational alignment and heteroscedastic uncertainty, Brancher improves robustness to noisy pseudo-labels and limited visibility, enabling stable and temporally coherent 3D motion reconstruction from challenging aerial sports footage. These results highlight the potential of uncertainty-aware self-supervision for advancing 3D HPE in real-world UAV applications.

Keywords: Deep Learning, Computer Vision, 3D Human Pose Estimation, UAV Footage, Self-Supervised Learning

Abstract in lingua italiana

La stima della posa umana 3D (3D HPE) mira a ricostruire la configurazione tridimensionale del corpo umano a partire da dati visivi e ha compiuto progressi significativi negli ultimi anni grazie al deep learning e alla disponibilità di dataset su larga scala. In ambito sportivo, una ricostruzione della posa 3D può supportare l'analisi delle prestazioni, la prevenzione degli infortuni e la valutazione tattica. Tuttavia, gli attuali metodi allo stato dell'arte sono progettati principalmente per ambienti controllati o telecamere poste ad altezza uomo, e non risultano ottimizzati per le riprese aeree.

Le piattaforme basate su droni offrono una soluzione flessibile e scalabile per scenari sportivi all'aperto, dove l'area monitorata può essere molto vasta e l'installazione di sistemi multicamera fissi e calibrati risulta spesso logisticamente o economicamente proibitiva. Gli aeromobili a pilotaggio remoto (UAV) possono regolare dinamicamente il punto di vista, l'altezza e la copertura, consentendo l'acquisizione di dati in ambienti complessi. Ciononostante, i filmati aerei introducono ulteriori sfide, tra cui la variabilità del punto di vista, il motion blur, i cambiamenti di scala e le frequenti occlusioni.

In questa tesi, si propone Brancher, un framework di deep learning basato su una backbone allo stato dell'arte per la 3D HPE monoculare di un singolo soggetto da video sportivi aerei. Questo lavoro è tra i primi ad affrontare esplicitamente la ricostruzione della posa umana 3D da riprese monoculari di droni attraverso un approccio basato sul deep learning. Il modello adotta una strategia di addestramento auto-supervisionata ed estende la regressione 3D standard modellando congiuntamente la dinamica spazio-temporale delle articolazioni, le rotazioni degli arti e l'incertezza della predizione. Tenendo esplicitamente conto dell'allineamento rotazionale e dell'incertezza eteroschedastica, Brancher migliora la robustezza rispetto a pseudo-label rumorose e alla visibilità limitata, consentendo una ricostruzione del movimento 3D stabile e temporalmente coerente da riprese aeree complesse. Questi risultati evidenziano il potenziale dell'auto-supervisione consapevole dell'incertezza per l'avanzamento della 3D HPE in applicazioni reali con sistemi UAV.

Parole chiave: Deep Learning, Visione Artificiale, Stima della posa umana 3D, Riprese da UAV, Apprendimento auto-supervisionato

Contents

1 | Introduction

1.1. Human Pose Estimation

Human Pose Estimation (HPE) is a fundamental task in **computer vision** that aims to localize human body joints, such as the head, shoulders, elbows, wrists, hips, knees, and ankles, within images or video sequences. These joints are subsequently connected according to a predefined kinematic structure to form a skeletal representation of the human body. By providing a compact and semantically meaningful description of human posture and motion, human pose estimation enables machines to interpret and reason about human actions, movements, and behaviors in visual data.

Depending on the target representation, human pose estimation can be formulated either as a **two-dimensional (2D)** or a **three-dimensional (3D)** problem. In 2D human pose estimation, joint locations are predicted in the image plane, typically using mathematical methods that operate directly on RGB images or videos. While 2D pose estimation has achieved high accuracy under controlled conditions, it inherently lacks depth information and is therefore insufficient for capturing the full spatial structure of human motion. In contrast, 3D human pose estimation aims to reconstruct joint positions in three-dimensional space, providing a more complete and physically meaningful representation of the human body. However, recovering 3D pose from visual data, especially from monocular images, introduces significant challenges due to depth ambiguity and the loss of geometric information during the image projection process.

Historically, accurate 3D human pose acquisition has been achieved through **marker-based** motion capture systems, which rely on reflective markers attached to the subject's body and multiple calibrated cameras to triangulate joint positions. Although such systems provide highly precise measurements, they require specialized hardware, controlled environments, and intrusive setups, limiting their applicability to laboratory conditions. More generally, motion capture can also be achieved through **sensor-based** approaches, which rely on wearable devices and instrumented environments to track body motion with high accuracy. While these systems are effective in controlled settings, they remain

dependent on dedicated infrastructure and do not fall within the scope of vision-based human pose estimation.
To overcome the limitations of marker-based and sensor-based systems, **markerless** pose estimation approaches have been developed, leveraging computer vision and learning-based methods to estimate human pose directly from images or videos without the need for physical markers or wearable devices. Despite their increased flexibility, markerless methods, particularly in 3D, remain sensitive to occlusions, viewpoint changes, and the availability of annotated training data.

Human pose estimation plays a crucial role in a wide range of application domains, including healthcare and rehabilitation, surveillance and safety monitoring, human-computer interaction, animation, and sports performance analysis. Among these, sports analysis represents one of the most challenging scenarios, as it involves fast and complex motions, self-occlusions, and highly dynamic viewpoints. These characteristics make sports an ideal testbed for advancing robust and generalizable human pose estimation techniques, particularly in unconstrained real-world environments.

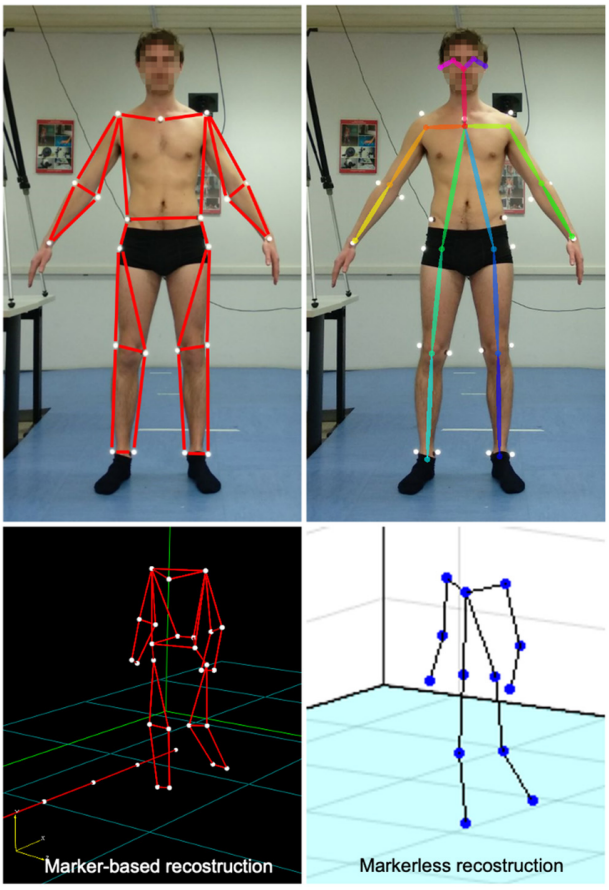

Figure 1.1: An example of a marker-based system vs a marker-less system from [77]

1.2. State of the Art and Limitations of Current Approaches

In recent years, human pose estimation has witnessed substantial progress, largely driven by the rapid evolution of deep learning techniques. The growing relevance of this research area is reflected in the extensive body of literature, with millions of publications addressing human pose estimation and closely related topics. This remarkable expansion highlights both the scientific interest and the practical importance of accurately modeling human posture and motion from visual data.

Modern deep learning-based approaches have significantly improved the performance of human pose estimation systems, particularly in controlled environments. State-of-the-art architectures, including convolutional and transformer-based models, are capable of producing highly accurate and stable pose predictions when operating under favorable conditions. Such conditions typically include limited occlusions, well-defined and relatively static camera viewpoints, consistent lighting, and the availability of large-scale annotated datasets. Under these assumptions, both 2D and 3D pose estimation methods have achieved impressive results, often approaching human-level accuracy in benchmark datasets.

Despite these advances, the strong performance of current methods is often tightly coupled to the constraints under which they are developed and evaluated. Many state-of-the-art models rely on assumptions that are difficult to satisfy in real-world scenarios, such as fixed or frontal viewpoints, minimal camera motion, and the presence of training data closely matching the target domain. As a result, the generalization capabilities of these models remain limited when deployed in unconstrained environments.

Several challenges remain open and continue to hinder the robustness of human pose estimation systems. Occlusions caused by self-intersections or external objects can lead to missing or inaccurate joint predictions. Viewpoint variability introduces significant appearance changes, making it difficult for models trained on canonical perspectives to generalize effectively. In monocular settings, depth ambiguity represents a fundamental limitation, as multiple three-dimensional poses may correspond to the same two-dimensional projection. Furthermore, the scarcity of accurately annotated 3D pose data, especially in outdoor and in-the-wild conditions, poses a major obstacle to supervised learning approaches.

As the field of human pose estimation has reached a high level of maturity, further progress increasingly depends on addressing specific and challenging application scenarios where

existing methods exhibit clear limitations. Rather than pursuing incremental improvements on well-established benchmarks, contemporary research efforts focus on identifying underexplored problem settings that better reflect real-world conditions. Within this context, constrained yet practically relevant scenarios, such as monocular 3D pose estimation in dynamic outdoor environments, represent a meaningful direction for advancing the applicability and robustness of human pose estimation systems.

1.3. Human Pose Estimation in Outdoor Sports

Sports performance analysis represents one of the most demanding and impactful application domains for human pose estimation. Accurate modeling of an athlete's body motion enables detailed biomechanical analysis, quantitative assessment of technical execution, performance optimization, and injury prevention. For coaches, athletes, and sports scientists, access to reliable pose information provides valuable insights into movement efficiency, coordination patterns, and load distribution during complex athletic actions.

Traditionally, high-precision motion analysis in sports has been achieved through marker-based motion capture systems. While these systems offer accurate 3D measurements, they require reflective markers to be attached to the athlete's body, specialized equipment, and controlled acquisition environments. In many sports scenarios, the use of physical markers is impractical or undesirable, as it may interfere with natural movement, restrict the athlete's performance, or be incompatible with competitive settings. For this reason, markerless pose estimation approaches are particularly attractive for sports applications, as they allow motion analysis to be conducted without altering the athlete's behavior.

In indoor sports environments, such as volleyball, basketball, or tennis played in controlled arenas, effective markerless solutions have been developed by leveraging multi-camera systems. By using multiple synchronized and calibrated cameras, accurate 3D pose reconstruction can be achieved through geometric triangulation. The fixed camera setup, limited capture volume, and controlled lighting conditions make these environments well-suited for multi-view pose estimation techniques. As a result, reliable 3D motion analysis has become feasible in several indoor sports contexts.

However, extending these approaches to **outdoor** sports introduces substantial challenges. Outdoor sports often take place over large spatial areas, making the installation and calibration of dense networks of static cameras logistically complex and economically costly. Environmental factors such as varying illumination, weather conditions, and background clutter further complicate visual data acquisition. Moreover, athletes may move freely across wide regions, frequently leaving the field of view of fixed cameras. As a conse-

quence, traditional multi-camera motion capture systems struggle to scale beyond controlled indoor environments and fail to capture natural athletic movements in real-world outdoor settings.

These limitations highlight the need for flexible and non-intrusive pose estimation solutions capable of operating in unconstrained outdoor sports scenarios, where controlled camera setups are not feasible.

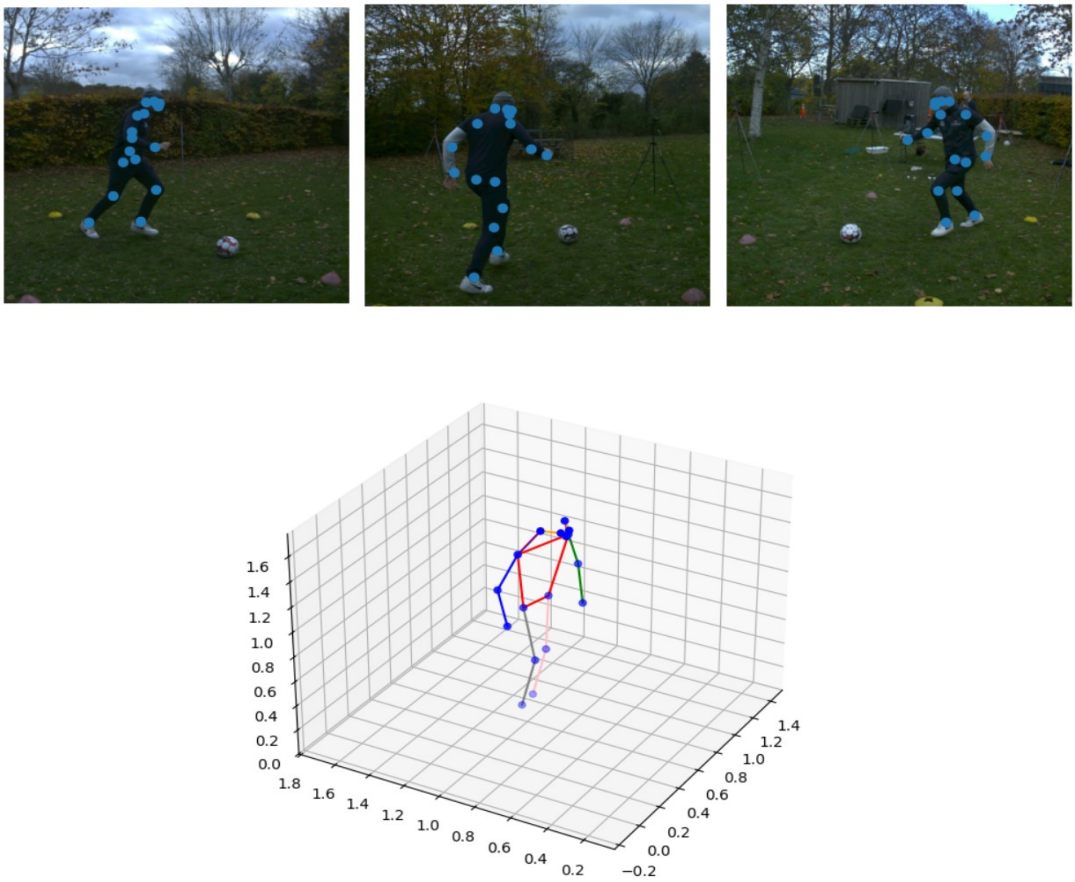

Figure 1.2: [18] [19] presents an example of outdoor multi-view HPE for sports scenarios. However, such systems rely on controlled camera setups and predefined capture areas, which are often impractical in real-world sports settings. In many disciplines, athletes move over large and dynamic environments, making it difficult to confine the action within a fixed multi-camera configuration.

1.4. Drones as a Data Acquisition Platform

The limitations of fixed camera systems in outdoor sports scenarios motivate the adoption of mobile sensing platforms capable of flexibly following athletes across large and uncon-

strained environments. In this context, unmanned aerial vehicles (UAVs), commonly referred to as drones, represent a natural evolution of visual data acquisition systems for sports analysis.

Unlike static camera installations, drones provide a mobile and adaptive viewpoint, allowing continuous tracking of athletes across wide spatial areas without requiring predefined capture volumes. This flexibility is particularly valuable in outdoor sports, where actions are rarely confined to a restricted region and where the installation of fixed multi-camera setups may be impractical or economically unfeasible.

Over the past decade, drone technology has significantly matured. Modern consumer-grade UAVs integrate stabilized gimbal systems, visual-inertial odometry, autonomous navigation, and subject-tracking capabilities. These advancements enable the acquisition of high-resolution and temporally consistent video data, even during dynamic flight maneuvers. As a result, drones have become reliable platforms for large-scale motion capture in real-world outdoor environments.

However, this flexibility comes at a cost. Drone-acquired videos are characterized by continuously changing viewpoints, varying camera-to-subject distances, and non-negligible camera motion. These factors introduce substantial scale variations, motion blur, and viewpoint-dependent appearance changes. Moreover, drone systems typically operate with a single moving camera, which removes the geometric redundancy available in traditional multi-view capture setups.

These characteristics make drone footage fundamentally different from controlled laboratory recordings and pose new challenges for computer vision algorithms, particularly for 3D human pose estimation.

1.5. Drone-based Monocular 3D Human Pose Estimation

Building upon the characteristics described above, performing 3D HPE from drone videos emerges as a challenging and underexplored problem.

In traditional 3D pose estimation pipelines, depth ambiguity is commonly resolved using multi-view camera systems. However, replicating such configurations in drone-based scenarios would require multiple coordinated UAVs, significantly increasing hardware costs, system complexity, and operational constraints. In high-speed sports contexts, maintaining sufficiently diverse and complementary viewpoints across multiple drones is particu-

Figure 1.3: An example of a drone shot in outer space, ready for use with Human Pose Estimation

larly difficult, as aerial platforms may converge to similar trajectories while tracking the same athlete, thus providing limited additional geometric information.

For these reasons, a single moving camera represents the most realistic and scalable configuration for drone-based sports analysis. Nevertheless, monocular 3D pose estimation is inherently ill-posed: multiple three-dimensional body configurations can project to the same two-dimensional keypoints, making purely geometric reconstruction insufficient.

To overcome this ambiguity, modern approaches rely on deep neural networks to learn strong priors on human body kinematics and motion dynamics. These priors allow models to infer plausible 3D poses consistent with the observed 2D evidence, even in the absence of multi-view constraints.

A further and critical limitation in this domain is the scarcity of annotated 3D datasets acquired from aerial viewpoints. Unlike indoor benchmarks captured in controlled environments with motion capture systems, large-scale 3D ground-truth annotations for outdoor drone footage are extremely limited. The acquisition of accurate 3D labels in such settings is technically complex and costly, hindering the direct application of fully supervised learning strategies.

Consequently, drone-based monocular 3D human pose estimation requires models that are not only robust to viewpoint and scale variability, but also capable of learning under limited or absent 3D supervision.

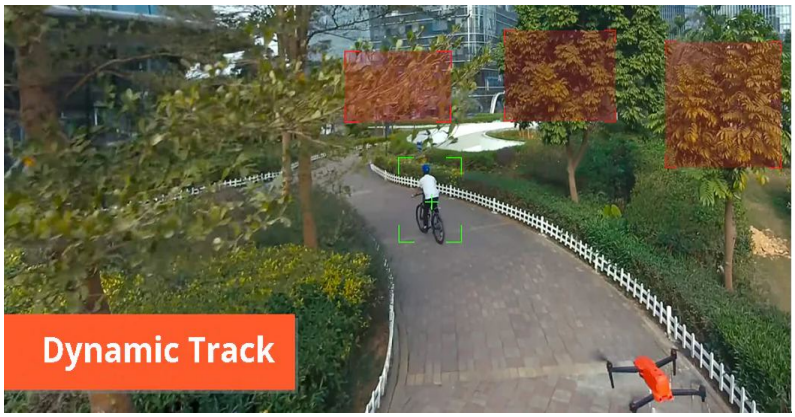

Figure 1.4: An example of drone dynamic tracking

These limitations highlight the need for a new generation of methods specifically designed for drone-based monocular 3D human pose estimation. Such methods must simultaneously address viewpoint variability, scale changes, and motion dynamics, while reducing reliance on large-scale 3D annotated datasets. In this thesis, I tackle this challenge by proposing a self-supervised framework tailored to aerial sports footage.

1.6. Thesis Contributions

In this thesis, I address the problem of monocular 3D Human Pose Estimation from drone videos in outdoor sports scenarios. A major limitation in this domain is the scarcity of large-scale annotated 3D datasets acquired from aerial viewpoints. Collecting accurate 3D ground-truth poses outdoors is costly and technically challenging, making fully supervised approaches difficult to scale. For this reason, I focus on designing a self-supervised solution that can operate without requiring extensive 3D annotations.

To this end, I introduce **Brancher**, a self-supervised neural network for single-person, single-view 3D Human Pose Estimation from UAV videos. The model is specifically designed to handle high-speed sports movements and the geometric challenges introduced by aerial viewpoints.

The main contributions of this thesis can be summarized as follows:

- **A self-supervised framework for 3D HPE.**
  I propose a novel architecture that learns motion-aware pose representations from high-speed sports videos and leverages self-supervision to reduce dependence on large-scale 3D annotated datasets.

- **Self-supervised adaptation to drone footage.**
  I develop a fine-tuning strategy tailored to UAV-acquired videos, addressing domain shift, viewpoint variability, and scale changes typical of aerial recordings, without requiring 3D ground-truth annotations.

- **A custom drone-based sports dataset.**
  I collect and curate a novel dataset of outdoor sports videos captured from drones, specifically designed to support research in monocular 3D human pose estimation from aerial perspectives.

Through these contributions, this thesis aims to bridge the gap between state-of-the-art 3D pose estimation methods and practical drone-based sports analysis in unconstrained outdoor environments.

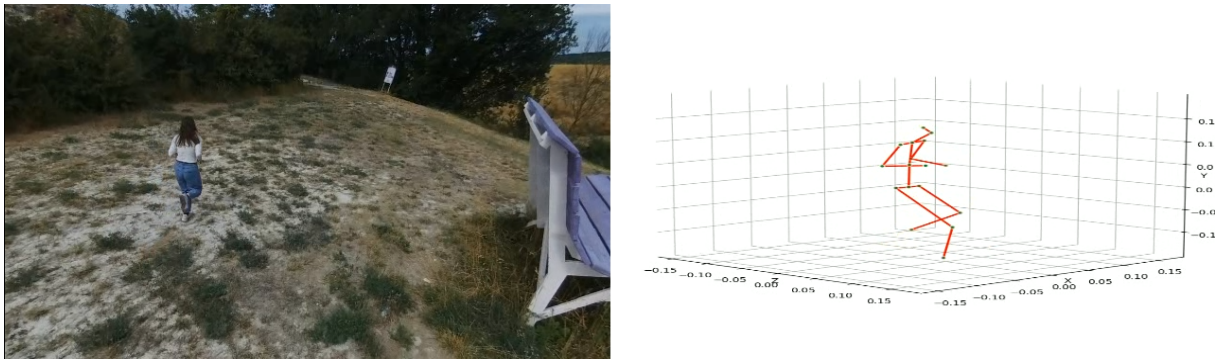

Figure 1.5: A demo qualitative visualization of the proposed framework Brancher

Figure 1.6: A data example from the custom drone dataset

2 | Domain background

2.1. Keypoint and keypoint detection

A **keypoint** is a semantically meaningful point in an image corresponding to a relevant part of an object or subject. In general, keypoints are characterized by the following properties:

1. **Distinctiveness:** keypoints should be easily distinguishable from their surroundings, allowing them to be uniquely identified.
2. **Invariance:** keypoints should be robust to common image transformations such as rotation, scaling, and changes in illumination.
3. **Repeatability:** keypoints should be consistently detectable across different images or frames of the same object or scene.

In the context of Human Pose Estimation, keypoints typically correspond to predefined human body joints or landmarks, such as the head, shoulders, elbows, wrists, hips, knees, and ankles.
Formally, a keypoint is represented by a spatial location, expressed as 2D coordinates (x, y) in the image plane or as 3D coordinates (x, y, z) in the physical space. In many approaches, a confidence score is also associated with each keypoint to indicate the reliability of the estimated location. The collection of keypoints belonging to a person defines a structured representation of the human body, commonly referred to as a pose or skeleton, which captures the geometric configuration of the subject and serves as a compact and interpretable representation for downstream tasks.

Keypoint detection refers to the task of automatically identifying the spatial locations of keypoints within an image or video. In the context of Human Pose Estimation, the objective is to localize each predefined human body keypoint according to a fixed anatomical convention, given a visual input.
Rather than directly regressing keypoint coordinates, modern deep learning approaches typically formulate keypoint detection as a probabilistic localization problem. In this

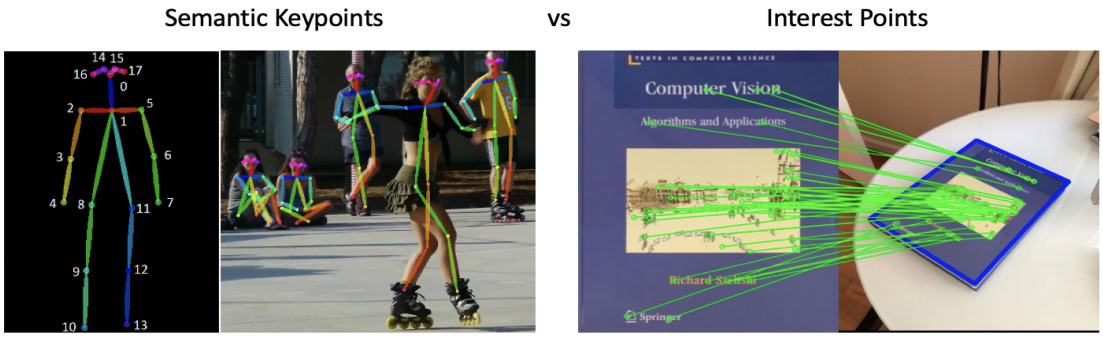

Figure 2.1: Example of different types of keypoint

setting, the model predicts a spatial probability distribution for each keypoint, often represented as a heatmap, which encodes the likelihood of the keypoint being present at each image location. The final keypoint position is then obtained by extracting a representative statistic from this distribution, such as its maximum or expected value.

3 | Related Work

There are two methods for predicting human pose: deep learning-based and traditional computer vision-based. Traditional approaches, such as SIFT [36] or HOG, use manually created features and models to identify and locate body joints. These traditional methods' features are frequently created by hand, requiring specialists to create features that can accommodate a variety of human positions. Traditional methods rely on pre-existing knowledge about the human body and the spatial relationships between its many sections. These methods use fewer computational resources to estimate the human body. Nevertheless, they are unable to capture intricate position changes or occluded, nonvisible, or overlapped joints. In recent years, the rapid advancement of deep learning has led to significant improvements across core computer vision tasks such as image classification, semantic segmentation, and object detection, with similar progress observed in HPE. Despite these advances, several challenges remain open, including occlusions, limited availability of annotated data, and depth ambiguity.

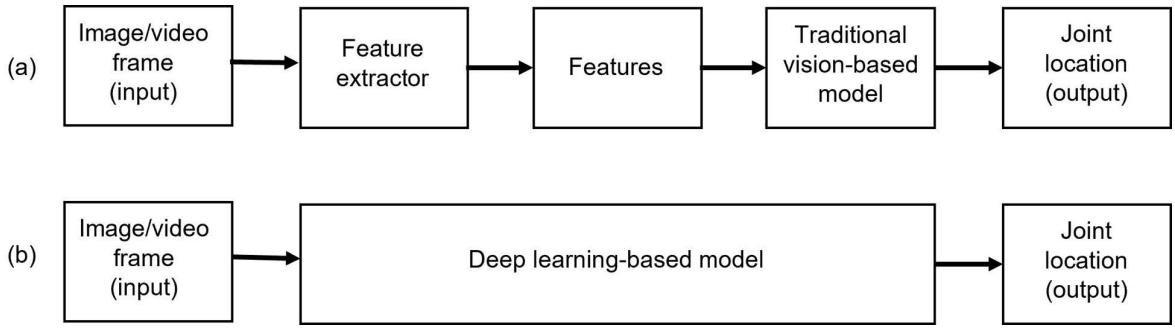

Figure 3.1: (a) Traditional vision-based flow; (b) Deep learning flow. Image taken from [57].

While 2D HPE from images and videos has reached high accuracy under supervised learning, particularly for single-person scenarios, more complex settings such as multi-person scenes with severe occlusions remain challenging. The problem becomes substantially harder in 3D HPE, where obtaining accurate ground-truth annotations is significantly more difficult. Although motion capture systems can provide precise 3D pose data in

controlled laboratory environments, they are poorly suited for in-the-wild scenarios. In monocular 3D HPE, depth ambiguity represents the primary challenge, whereas multi-view approaches must address the additional complexity of viewpoint association across cameras.

In parallel, consumer drone technology has evolved rapidly over the past decade, benefiting from advances in autonomous flight control, high-precision stabilization, onboard sensing, and edge computing. Modern drones are capable of complex operations such as autonomous navigation, subject tracking, and real-time decision-making. As a result, drone-based systems have emerged as a promising solution for markerless human pose estimation in large-scale outdoor environments. These systems typically combine lightweight aerial platforms with RGB cameras mounted on stabilized gimbals, offering new opportunities for human pose estimation beyond traditional static camera setups.

This chapter is based on the work of Zheng et al. [81], Samkari et al. [57], El Kaid et al. [9], Kalampokas et al. [26] and Han Teh et al. [63].

3.1. 2D Human Pose Estimation

Numerous methods have been developed over the years in the significant area of computer vision research known as 2D HPE. The task involves locating and identifying human joints in images or videos, which is essential for a number of applications like augmented reality, activity recognition, and human-computer interaction. Since the accuracy of 3D pose estimation greatly depends on 2D HPE, 2D keypoint extrapolation is frequently a first step in the larger field of human pose analysis.
Predicting the coordinates of keypoints that correspond to human joints in a 2D image is the aim of 2D HPE.

Figure 3.2: 2D Human Pose Estimation example from OpenPose [4]

The input for a 2D HPE model is typically an image or, in the case of a video, a sequence of images. The result is a set of keypoints that reflect the estimated pose of the person or people in the image and contain the (x, y) coordinates for each joint identified in the image.

Occlusions, differences in lighting and appearance, and the complex nature of human positions are the primary obstacles in 2D HPE. Accurate pose estimate is further complicated by issues including body size differences, overlapping people, and determining which person each joint belongs to in multi-person scenarios.

Evolution

With the advent of deep learning methods like Convolutional Neural Networks (CNNs) and the more recent Transformers [68], 2D HPE has advanced significantly. In this section, I will only talk about deep learning approaches, which have revolutionized the field since the release of DeepPose [67] in 2014.

The first tendency was to directly recover human keypoint coordinates from images using CNNs and **regression-based** techniques. However, the complex nature of human poses and differences in appearance, lighting, and occlusions presented difficulties for this method. In order to achieve more reliable localization that accounts for uncertainties in the joint position predictions, researchers turned to **detection-based** approaches, in which the model predicts a probability distribution over potential locations for each keypoint.

The latest trend in 2D HPE is the use of pretrained models and attention mechanisms, which have shown promising results in capturing long-range dependencies and improving the overall accuracy of pose estimation.

Principal approaches

When we talk about 2D HPE, we can divide the approaches into two main categories: **single pose estimation** and **multi-pose estimation**. We can also categorize them based on whether they are **image-based** or **video-based**.

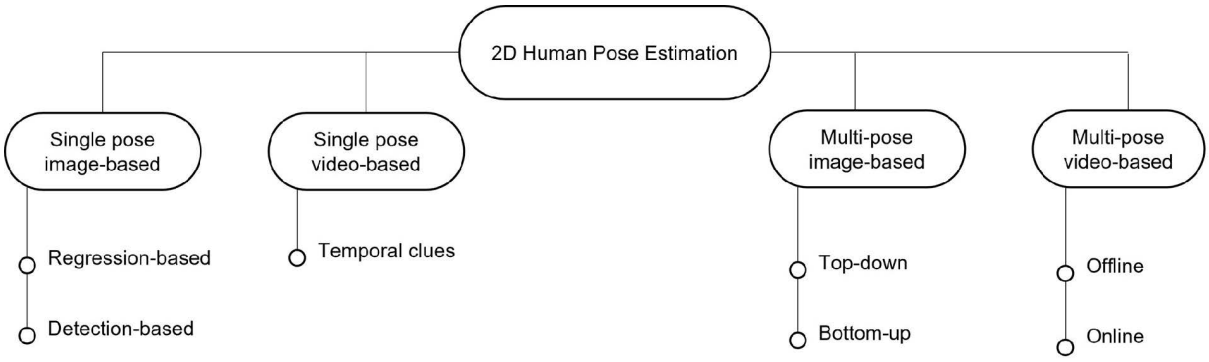

Figure 3.3: Taxonomy of 2D HPE approaches. Image taken from [57].

In single pose estimation image based, the two main approaches are **regression-based** methods and **detection-based** methods. Regression-based methods directly predict the coordinates of keypoints from the input image, while detection-based methods generate a heatmap for each keypoint, indicating the likelihood of the keypoint's presence at each pixel location.

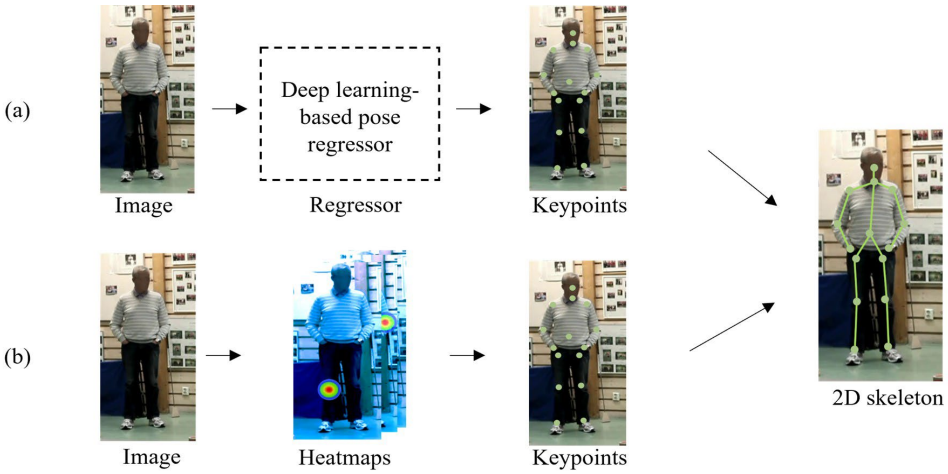

Figure 3.4: (a) Regression-based approach and (b) Detection-based approach. Image taken from [57].

Top-down and **bottom-up** methods are the two primary approaches in multi-pose estimation image-based. Top-down approaches estimate each person's pose independently after first detecting every person in the scene. In contrast, bottom-up approaches first identify each significant human keypoint in the image before grouping them into distinct poses. A top-down approach is typically more accurate but computationally costly, whereas a bottom-up approach is more efficient but may have trouble with occlusions and overlapping poses. The decision between these approaches frequently depends on the particular application and the trade-offs between accuracy and computational efficiency.

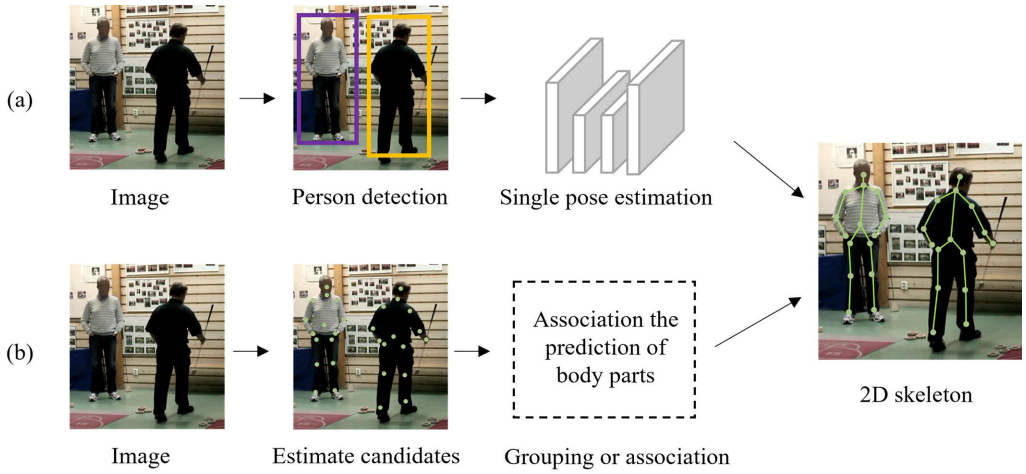

Figure 3.5: (a) Top-down approach and (b) Bottom-up approach. Image taken from [57].

In single pose estimation video based, is now common to use a single pose estimation image based model to extract 2D human keypoints from each frame of the video, and

then take into account the temporal consistency between frames to improve the accuracy and stability of the pose estimation over time. This can be achieved through techniques that use temporal clues across frames.

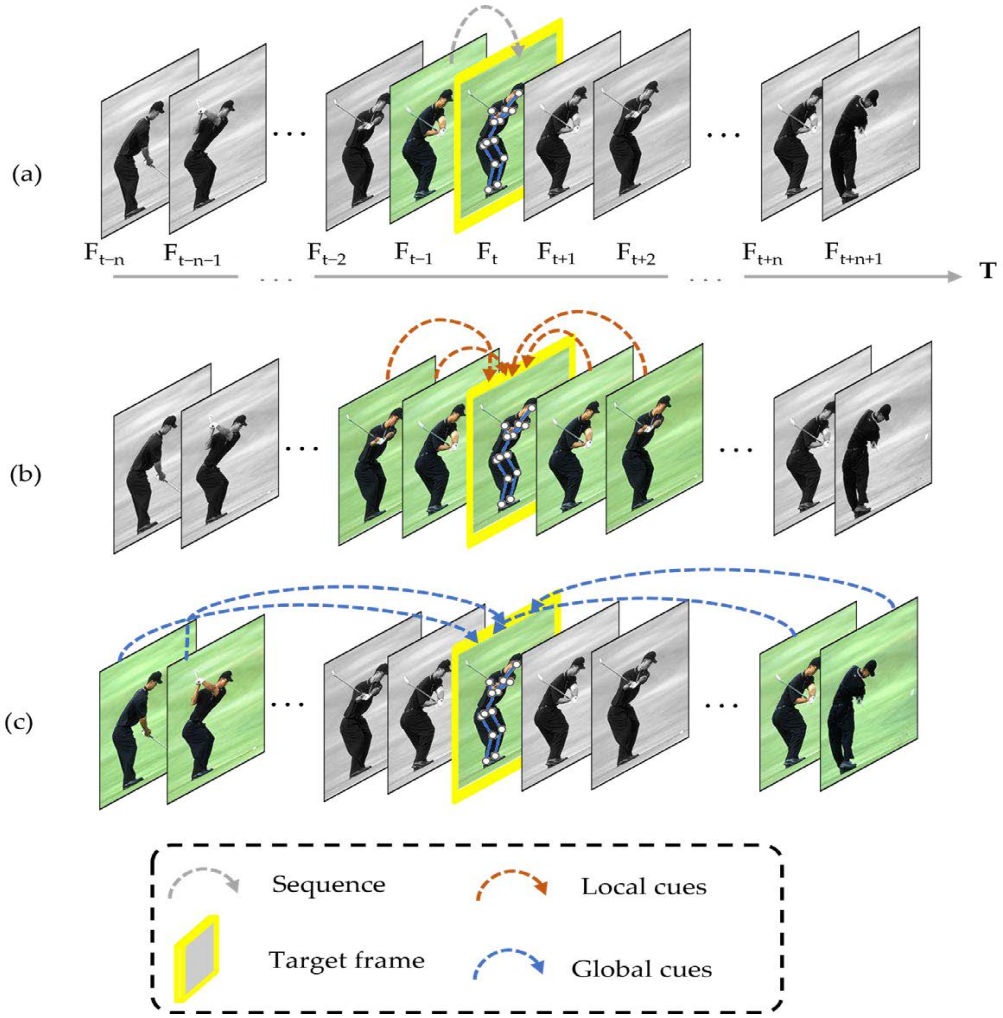

Figure 3.6: Different types of temporal clues across frames, (a) frame sequence, (b) local cues, and (c) global cues, are used to estimate the pose in the target frame. Image taken from [57].

Online and **offline** approaches are the two primary methods used in multi-pose estimation video. Online methods estimate poses in real-time without spatiotemporal interactions by processing each frame sequentially. Conversely, offline methods use data from several frames, enabling spatiotemporal interactions that can enhance pose estimation accuracy by taking surrounding frame context into account. The needed accuracy for the particular application and the requirements for computational efficiency often determine which of these approaches is best.

3.1.1. Datasets, losses and metrics

Datasets

2D HPE models are typically trained on large datasets that contain manually annotated frames with ground truth keypoint locations. They require diverse data to handle challenges such as occlusions, varying backgrounds, illumination and clothing. Some datasets also provide different pose activities, such as daily life and sports that contain complex poses and occlusions. Below are some of the most commonly used datasets for 2D HPE, the readers should be pay attention that datasets **are not all annotated in the same way**, some use different keypoint definitions and number of keypoints.

Leeds Sports Pose (LSP) and Lead Sports Pose Extended (LSPE) The LSP dataset [24] is used for single-pose estimation. It has 1000 training images and 1000 testing images of sports activities, with 14 annotated keypoints per person. The LSPE [25] is an extended version of LSP, which includes an additional 10,000 training images.

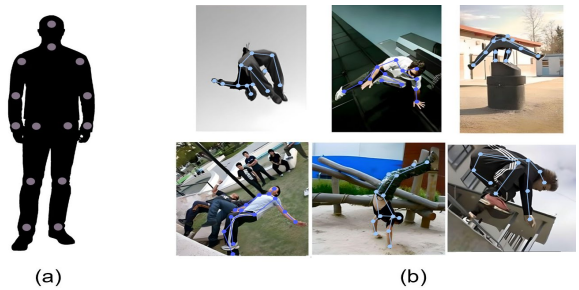

Figure 3.7: The LSP dataset, (a) 2D keypoint annotations and (b) some data. [57]

Frames Labeled in Cinema (FLIC) The FLIC dataset [58] is used for single and multipose estimation. It is split into training and testing datasets containing 4000 and 1000 images. The images are extracted from movies, and each person is annotated with 10 keypoints.

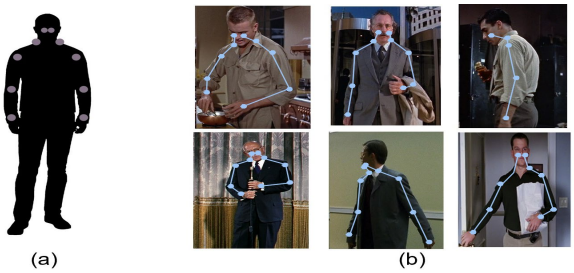

Figure 3.8: The FLIC dataset, (a) 2D keypoint annotations and (b) some data. [57]

Common Objects in Context (COCO) The COCO dataset [34] contains object detection, segmentation, and keypoint annotations. The keypoint detection subset has over 200,000 images with more than 250,000 person instances annotated with 17 keypoints each. The COCO dataset includes various human poses and objects of varying sizes and occlusions patterns.

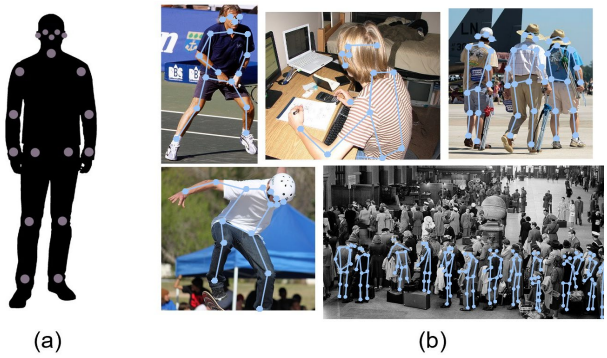

Figure 3.9: The COCO dataset, (a) 2D keypoint annotations and (b) some data. [57]

CrowdPose The CrowdPose dataset [31] is designed to address the challenges of pose estimation in crowded scenes, and it is focused on multiple poses. It contains over 10,000 training images, and 8000 testing images annotated with 14 keypoints per person.

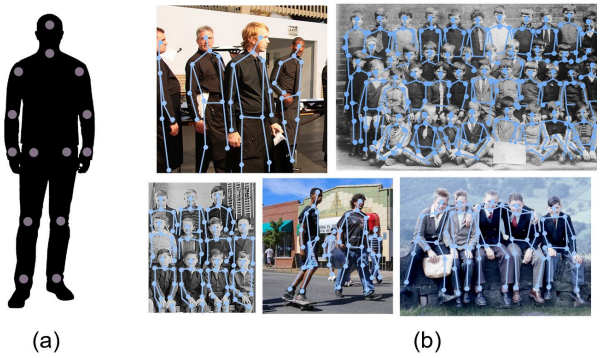

Figure 3.10: The CrowdPose dataset, (a) 2D keypoint annotations and (b) some data.

Max Plank Institute for Informatics (MPII) The MPII dataset [1] is used for single and multipose estimation and contains images and video frames. It has approximately 25,000 images depicting 410 different activities, with each person annotated with 16 keypoints. Images were collected from YouTube videos, providing different scale variations and complex poses.

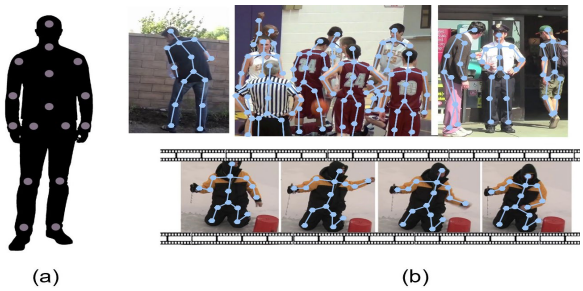

Figure 3.11: The MPII dataset, (a) 2D keypoint annotations and (b) some data. [57]

PennAction The PennAction dataset [79] is used for single-pose estimation in videos. It contains over 2000 RGB videos with annotated frames, totaling 330,000 frames with an average of 70 frames per video. Each person is annotated with 13 keypoints, and the dataset includes 15 actions such as pushups, strumming guitars, baseball, jumping rope, and tennis serve.

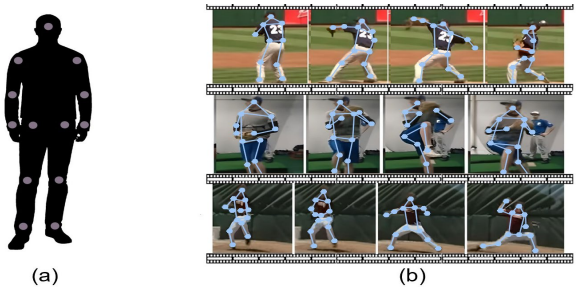

Figure 3.12: The PennAction dataset, (a) 2D keypoint annotations and (b) some data.

Joint Human Motion Database (JHMDB) The JHMDB dataset [23] is used for single-pose estimation in videos. It contains complete annotations of human actions, such as brushing hair, catching, clapping, and climbing stairs. It is used for human action recognition and human pose estimation. Each person in this subset has 15 annotated joints and around 900 video clips.

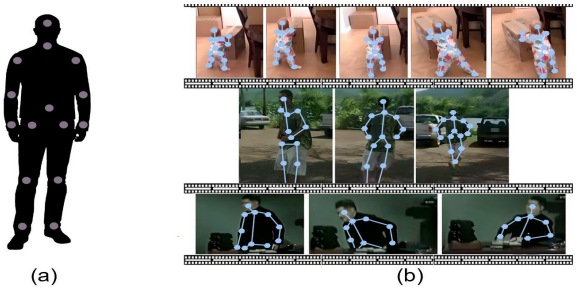

Figure 3.13: The JHMDB dataset, (a) 2D keypoint annotations and (b) some data. [57]

PoseTrack The PoseTrack dataset [2] is used for multi-pose estimation in videos. This dataset contains challenging scenarios involving highly occluded individuals in crowded environments with complex movements. It consists of approximately 1100 videos, and each person is annotated with 15 keypoints.

Figure 3.14: The PoseTrack dataset, (a) 2D keypoint annotations and (b) some data. [57]

Loss functions

The model's performance is greatly impacted by the selection of a suitable loss function. Some frequently used loss functions in 2D HPE are described in this subsection.

Mean Absolute Error (MAE) Mean Absolute Error (MAE), or L_1 loss, measure the absolute differences between the ground truth and predicted values. It is defined as:

$$\mathcal{L}_{\text{MAE}} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{3.1}$$

where y_i and $\hat{y}_i$ are the ground truth and predicted coordinates of the i-th joint, respectively, and N is the total number of joints. MAE is robust to outliers compared to L_2 loss.

Mean Squared Error (MSE) Mean Squared Error (MSE), or L_2 loss, measures the squared differences between the ground truth and predicted values. It is defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.2}$$

where y_i and $\hat{y}_i$ are the ground truth and predicted coordinates of the i-th joint, respectively, and N is the total number of joints. MSE penalizes larger errors more than MAE.

Evaluation Metrics

Metrics are not utilized to change a model's parameters during training, in contrast to loss functions. Rather, they are employed to assess the performance of the model. Some frequently used evaluation measures in 2D HPE are described in this subsection.

Percentage of Correct Keypoints (PCK) If the distance between predicted joints and ground truth joints falls inside a set of threshold constraints, the Percentage of Correct Keypoints (PCK) measure, which is defined as estimated keypoints, is considered accurate. The PCK is calculated as:

$$\text{PCK@}\alpha = |\text{Predicted Keypoints} - \text{Ground Truth Keypoints}| < \alpha \cdot \text{Threshold}_{\text{max}} \quad (3.3)$$

where α is a percentage, and $\text{Threshold}_{\text{max}}$ is considered a side with a maximum length of the outer rectangle covering truth body joints.

Percentage of Detected Joints (PDJ) If the mean distance between the ground truth joint and the predicted joint falls below a set of threshold constraints, PDJ is the detected joint that was determined to be correct. The PDJ is calculated as:

$$\text{PDJ@}\alpha = |\text{Predicted Joints} - \text{Ground Truth Joints}| < \alpha \cdot \text{Torso Diameter} \quad (3.4)$$

where α is a percentage, and the torso diameter is the distance between two opposite body joints of the torso.

Average Recall (AR) and Average Precision (AP) Average Recall (AR) and Average Precision (AP) are widely used metrics for evaluating multi-person pose estimation models. AP measures the precision of the predicted keypoints, while AR measures the recall. Both metrics depend on the Object Keypoint Similarity (OKS), which quantifies the similarity between predicted and ground truth keypoints.

$$\text{OKS} = \frac{\sum_i \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \quad (3.5)$$

where d_i is the Euclidean distance between the predicted and ground truth keypoints, s is the human scale, k_i is a per-keypoint constant that controls falloff, and v_i indicates the visibility of the keypoint.

AP measures the precision of the predicted keypoints:

$$\text{AP} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \text{AP}_{\text{OKS}=t} \tag{3.6}$$

$$\mathcal{T} = \{0.50, 0.55, \ldots, 0.95\} \tag{3.7}$$

AR measures the recall of the predicted keypoints:

$$\text{AR} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \text{Recall}_{\text{OKS}=t} \tag{3.8}$$

3.1.2. Single Pose Estimation Image Based

Single pose models are designed to estimate the pose of a single person in an image, so a single frame. The two main approaches for this task are **regression-based methods** and **detection-based methods**. Regression-based methods directly predict the coordinates of keypoints from the input image, while detection-based methods generate heatmaps for each keypoint, indicating the likelihood of the keypoint's presence.

Regression-based methods

Regression-based methods for single pose estimation involve training a model to directly predict the (x, y) coordinates of each keypoint from the input image. One of the pioneering works in this area is DeepPose [67], which utilized a deep neural network to regress joint locations. The DeepPose framework uses a cascade of stages to progressively refine the predicted joint locations.

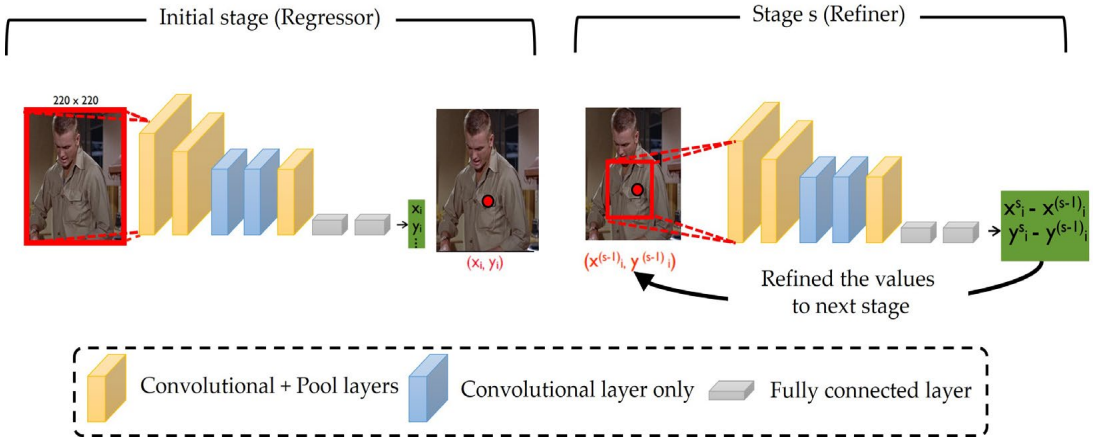

Figure 3.15: DeepPose framework [67] regresses joint locations in a cascade of stages.

Nowadays, these methods are less common due to the challenges in capturing the complex spatial relationships between joints and handling occlusions effectively.

Detection-based methods

In detection-based approaches, a 2D Gaussian distribution with its center at the joint is applied to joint locations to generate ground truth. The suggested approaches must generate a heatmap for every joint j_i with coordinates (x, y). Each joint's heatmap represents $H_1, H_2, ..., H_N$, and the entire number of heatmaps is equal to the total number of N joints. There are two challenges for the detection-based system. The first involves evaluating the probability that each pixel represents a joint in order to create a keypoint heatmap. The second entails improving the joint confidence map that is produced.

One of the most influential detection-based methods is the Stacked Hourglass Network [48], which employs a series of encoder-decoder modules (the Hourglass) to capture features at multiple scales. The network repeatedly downsamples and upsamples the feature maps, allowing it to integrate both local and global context for accurate keypoint localization. A residual connection is used to facilitate the flow of information between layers. Every module of the hourglass network produces a set of heatmaps for the keypoints, which are then refined through subsequent modules. Every module's output is supervised using intermediate supervision, which encourages the network to learn meaningful representations at multiple stages.

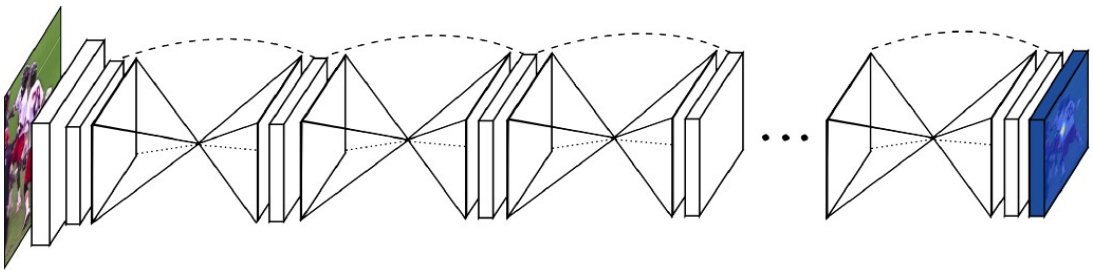

Figure 3.16: Stacked Hourglass Network architecture [48].

High-resolution networks (HRNet) [62] is another detection-based networks, that maintains high-resolution representations throughout the network, allowing for precise localization of keypoints. The network consists of multiple parallel branches that process features at different resolutions, with information exchanged between branches to enhance feature representation.

A more recent approach is the use of Transformer-based architectures for 2D HPE, such as VitPose [74]. VitPose employs plain and non-hierarchical Vision Transformers (ViTs) [8]

as backbone to extract feature maps for the given person instances, where the backbones are pre-trained with masked image modeling pretext tasks. It is noted that this framework is very scalable because of the scalable pre-trained ViT and the ability to stack multiple transformer layers; it is very simple because the design of the backbone encoder does not require any particular domain knowledge; and it is flexible in the training paradigm because it is possible to add more decoders without retraining the encoder.

3.1.3. Multi Pose Estimation Image Based

Multi-pose estimation models are designed to estimate the location of all the keypoints for all people, regardless the number of people in the image. The two main approaches for this task are **top-down methods** and **bottom-up methods**. Top-down approaches estimate each person's pose independently after first detecting the bounding box of every person in the scene. In contrast, bottom-up approaches first identify each significant human keypoint in the image before grouping them into distinct poses.

Top-Down methods

The main idea of top-down methods is to use a separate stage of features extraction and classification to recognize humans before estimating their poses. A region-proposal pipeline detector must first detect each human pose in the image before applying a single-person pose estimation method to predict the keypoints for each detected person. The strength of this framework is its ability to extract features with high accuracy due to using a pretrained model for object detection. However, its weakness is the high computational cost, especially when there are many people in the image.

The first top-down method was Region-based Convolutional Neural Networks (R-CNN) [13], which used selective search to generate region proposals and then applied a CNN to each proposal to classify it. However, one of the R-CNN's limitations was its slow speed. Fast R-CNN [12] improved upon R-CNN by introducing a region of interest (RoI) pooling layer, which allowed the network to process the entire image in a single forward pass, significantly speeding up the detection process, while Faster R-CNN [53] further improved the speed and accuracy of object detection by introducing a region proposal network (RPN) that shares convolutional features with the detection network. By extending Faster R-CNN by substituting a RoIAlign layer for the RoI pooling layer, which enhances the alignment of the extracted features with the input image, Mask R-CNN [15] focused on improving object detection accuracy by utilizing two built-in feature extractors to achieve good accuracy and efficiency. Also YOLO (You Only Look Once) [52] is a well-known

top-down method that treats object detection as a single regression problem, predicting bounding boxes and class probabilities directly from the input image in one pass, and it is used in YOLO-Pose [42] for real-time multi-person pose estimation.

Bottom-Up methods

These methods first detect all keypoints in the image and then group them into individual poses. The main advantage of bottom-up methods is their efficiency, as they can process the entire image in a single pass, regardless of the number of people present. However, they may struggle with occlusions and overlapping poses, making it challenging to accurately group keypoints.

One of the most notable bottom-up methods is OpenPose [4], which introduced the concept of Part Affinity Fields (PAF) to learn keypoint locations and their associations through a set of 2D vector fields that indicate the location and orientation of body parts.

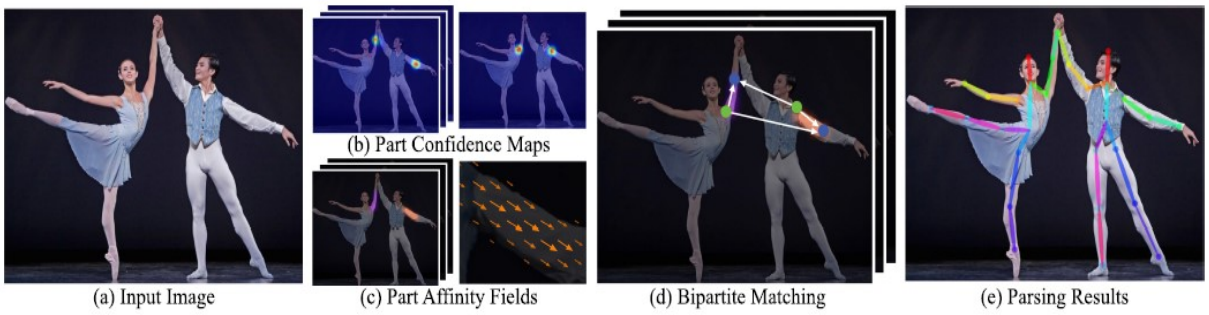

Figure 3.17: OpenPose pipeline [4] for multi-person 2D pose estimation.

OpenPifPaf [30] tries to estimate and track multiple human poses in image sequences to develop a fine-grained understanding of pedestrian behavior in the field of self-driving cars. The major challenges that need to be addressed are occlusions and prediction speed to react to real time changes in the environment. The authors proposed a Temporal Composite Association field (TCAF) which is used to form a spatio-temporal pose.

3.1.4. Single Pose Estimation Video Based

It is difficult to estimate poses from videos. Video-based estimating has a dynamic background because of subject movement, in contrast to image-based estimation, which has a static background. These difficulties are additionally exacerbated by phenomena like motion blur and changing light levels. However, because the location of occluded and invisible keypoints may be inferred from previous frames, video-based estimation makes

it easier to estimate poses with occlusion issues than image-based estimation.
The majority of video-based work on single-person pose estimation explores how to propagate temporal information across frames in order to improve results from a single frame. Temporal cues might be global, local, or sequential. Sequential temporal cues refer to the use of only the previous frame to estimate pose in the current frame because sequence model-based techniques rely on prior knowledge to predict current information. Conversely, local and global cues gather the information they need from many distant frames. I will now discuss the models that were most helpful to me in understanding the 2D HPE single pose video based problem.

Nie et al. [49] proposed the Dynamic Kernel Distillation (DKD) model, which exploits temporal continuity in videos through dynamically generated convolutional kernels. Instead of estimating the pose from scratch at each frame, DKD extracts features from the previous frame using a lightweight distillator network and uses them to generate pose-specific kernels for each keypoint. These kernels are then applied to the current frame via simple convolutions, reformulating keypoint localization as an efficient matching problem. By propagating pose kernels from one frame to the next, DKD leverages the fact that pose variations between consecutive frames are typically small. This design significantly reduces computational cost, as only a light network is used and the kernels are compact, enabling fast and efficient pose estimation in videos.

Zhang et al. [80] proposed the Key Frame Proposal Network (K-FPN) based on the same notion that variations between successive frames are usually modest. By extracting the most informative frames (key frames) from the video in an unsupervised manner and using a conventional 2D HPE model to extract poses from them, K-FPN estimates the pose more efficiently and avoids repetitive computations on comparable frames. It then uses the dynamic model of pose changes from the different key frames to recreate the additional poses from them. By avoiding performing inference on each frame, the model reduces computing overhead by concentrating on processing only the most important frames.

Following the idea of learning motion dynamics, Ma et al. [40] proposed a semi-supervised framework to alleviate the limited availability of temporally sparse annotations in videos. Their method leverages both labeled frames and unlabeled frames by exploiting temporal consistency among predicted keypoints. Specifically, they introduced the REinforced MOtion Transformation nEtwork (REMOTE), in which a Motion Transformer (MT) is designed to perform cross-frame reconstruction using paired labeled and unlabeled frames sampled from a video. The poses estimated by a pre-trained Pose Estimator (PE) are used to guide the reconstruction process. By modeling the motion dynamics between

the source and target poses, the MT learns to warp pose representations across frames, enabling the PE to benefit from supervision derived from both labeled and unlabeled data. In addition, an RL-based Frame Selection Agent, guided by a task-specific reward function, is employed to select informative frame pairs for training the MT.

Poseidon [50] is a more modern method that extends the ViTPose model [74] by incorporating temporal information for improved accuracy and robustness. It does this by using a multi-frame pose estimator architecture. Poseidon presents three major innovations: a Multi-Scale Feature Fusion (MSFF) module that combines features from various backbone layers to capture both fine-grained details and high-level semantics; an Adaptive Frame Weighting (AFW) mechanism that dynamically prioritizes frames based on their relevance, ensuring that the model focuses on the most informative data; and a Cross-Attention module that facilitates efficient information exchange between central and contextual frames, improving the temporal coherence of the model.

3.1.5. Multi Pose Estimation Video Based

Three methods are used when there are several persons in a single video frame: pose detection, estimation, and tracking. Multiperson pose tracking can be done online or offline. For reliable tracking, offline tracking techniques usually depict intricate spatiotemporal interactions over several frames, but they are computationally expensive. Graph partitioning-based techniques are frequently used in offline pose tracking techniques. Online pose-tracking techniques, on the other hand, are more effective since they do not require the modeling of intricate spatiotemporal connections.

Offline methods

The PoseTrack method [21] tracked each person's head and neck edges in a video by using a spatiotemporal graph to represent joint body detection every three frames. 3D Mask R-CNN [11] expanded Mask R-CNN [15] to incorporate temporal information as a third dimension (3D). The 3D Mask R-CNN consists of two stages: the first uses Mask R-CNN to extract posture features, and the second uses temporal information to track multiperson poses. In the same way, HRNet [62] was expanded in 3D HRNet [70] to incorporate temporal information between critical points as 3D for video pose tracking. The method combined the projected poses of the same individual using a video-tracking pipeline after using a clip-tracking network to estimate and track posture joints.

Online methods

The first online pose tracker was the PoseFlow model [73]. Three pipelines make up this model: a pose flow builder, a pose estimator based on Faster R-CNN [53], and a nonmaximal suppression pipeline to improve tracking. The JointFlow model [7] used a Siamese network to extract pose features such belief maps or Part Affinity Fields (PAF) [4]. Next, multiperson poses between two frames (previous and current) were tracked using temporal flow fields.

3.1.6. Sports Application

Sports videos have become a valuable resource for performance analysis, allowing athletes and coaches to study movements, refine techniques, and improve training outcomes. With the advent of deep neural networks, high-performance models for detection, tracking, and human pose estimation have shown great potential in automatically analyzing sports videos, enabling fine-grained motion understanding and athletic performance analysis.

In order to help with athletic training, Wang et al. [69] proposed AI Coach, a deep neural network-based system that aims to provide automatic and customized feedback on movement quality. It is divided in four modules: detection and tracking, pose estimation, pose classification and training suggestion. Rubiagatra et al. [54] provide a method to demonstrate that human pose estimation technology can be used to accurately measure the performance of athletes in Pound Fitness movements. The proposed system collect and annotate squat data, extract keypoints using pose estimation models, calculate joint angles, classify correct and incorrect techniques, and provide real-time feedback through a user-friendly application. Martinelli et al. [43] introduce a Ski Pose Estimation method, a self-supervised network that take in input human keypoints and add sky keypoints with a domain adaptation transformer.

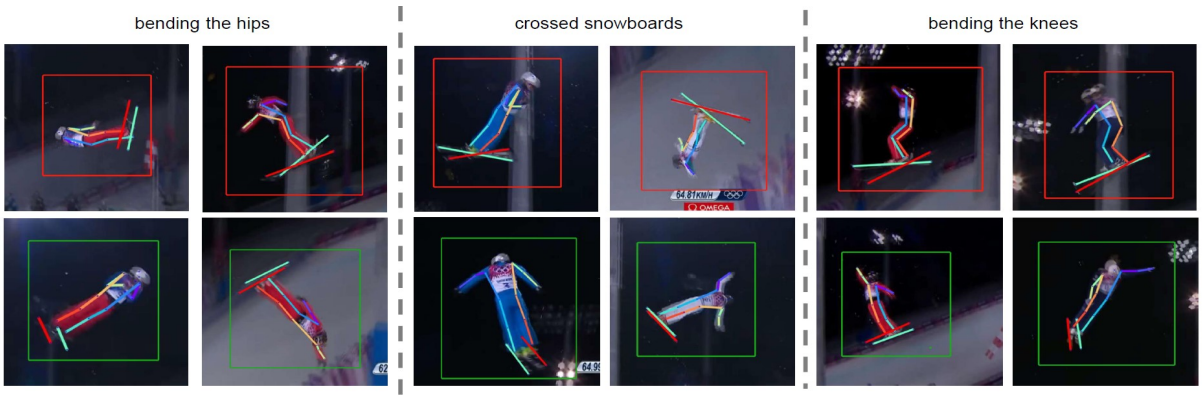

Figure 3.18: Examples of pose detection using keypoint-based features in AI Coach [69].

3.2. 3D Human Pose Estimation

Accurate pose estimation in real-world situations depends on depth estimate. Three-dimensional information helps clarify ambiguities in 2D poses, which frequently result in identical appearances for distinct stances when seen from different camera angles.
The goal of the 3D HPE is to estimate the three-dimensional configuration of the human body using visual data. Formally speaking, the objective of 3D HPE is to forecast a set of three-dimensional keypoints that define the location of the main human joints in space. These keypoints are each represented by coordinates (x, y, z).

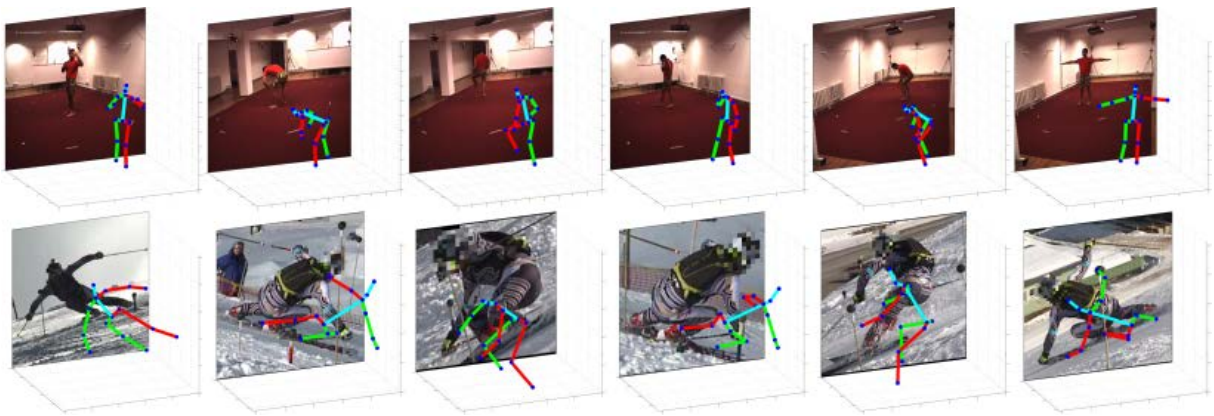

Figure 3.19: 3D Human Pose Estimation example.

Because of its close relationship to 2D HPE, 3D HPE inherits both its constraints and its representations. The 2D pose serves as an explicit or implicit intermediate in the 3D HPE in the majority of contemporary methods. A two-stage approach is used by several techniques, wherein 2D keypoints are first estimated in the image plane and then "lifted" into a three-dimensional environment. This method makes the pipeline more adaptable and modular by enabling the use of high-performance, pre-trained 2D models.
Nevertheless, relying solely on 2D estimation results in error propagation, as errors in the 2D pose are directly reflected in the 3D estimation. As an alternative, some research suggests end-to-end models that use the RGB image to directly predict the 3D position. Although these methods do not require the intermediary 2D pose phase, they are typically more difficult to train and more sensitive to data distribution.

3D HPE has far more difficult problems than 2D HPE. Since many various 3D configurations might result in comparable 2D projections, one of the primary challenges is the inherent depth uncertainty. For single-view 3D HPE in particular, this makes the problem extremely ill-posed. The lack of annotated 3D data is another issue since obtaining 3D ground truth necessitates costly infrastructure and controlled settings. As a result, many

3D datasets are restricted to indoor environments and show minimal variation in context, lighting, and clothing.

Evolution

Traditional computer vision methods, typically based on discriminative models and manually created features, were the foundation of the first methods for 3D HPE. A mapping function is learned using a set of image features that are either computed as body part information to the pose in 3D space or directly retrieved as shape context, segmentation, silhouette, HOG, and SIFT [36]. However, because these approaches relied on handcrafted features, they were limited in their capacity to generalize to new positions and viewpoints. Deep neural networks are now widely used to automatically identify important information in images due to the exceptional performance of deep learning techniques in computer vision, so also in this section I will only talk about deep learning approaches.
Initially, researchers concentrated on end-to-end learning techniques that employed CNNs to extract features and regress the 3D coordinates in order to directly predict 3D joint locations from photos. However, these approaches mostly depend on the distribution of the data as well as vast quantities of annotated 3D data, which are frequently hard to get by and costly to obtain. Researchers started investigating two-stage methods that estimate 2D keypoints from images and then lift them to 3D space in order to overcome this restriction. This method, which has become the de facto standard for 3D HPE, takes use of extremely accurate pre-trained 2D pose estimate models, reduces reliance on substantial volumes of 3D annotated data, and increases the pipeline's modularity and adaptability.
3D HPE has followed the recent rise in popularity of generative AI. The most recent methods look into using diffusion models to generate several poses, producing images or 2D keypoints from an input frame, and transforming a single-view problem into a multi-view problem.

Principal approaches

The methods for estimating 3D human poses can be broadly classified along several orthogonal dimensions, reflecting differences in acquisition setups, input modalities, and problem formulations.

A first fundamental distinction concerns the number of available viewpoints. **Single-view** methods estimate the 3D pose from a single camera and represent the most challenging setting due to the intrinsic ambiguity of depth estimation. In contrast, **multi-view** ap-

proaches exploit geometric constraints across multiple synchronized and calibrated cameras, significantly reducing depth ambiguity at the cost of increased acquisition complexity and reduced scalability.

Within the single-view setting, or **monocular**, approaches can be further categorized based on the input modality. **Image-based** methods estimate the 3D pose from a single frame, without leveraging temporal information, while **video-based** methods process sequences of frames and exploit motion continuity over time to improve robustness and accuracy. Video-based techniques are generally more resilient to noise and occlusions but introduce additional computational and modeling complexity.

From a pipeline perspective, single-view methods can follow either a **two-stage** formulation, in which 2D keypoints are first estimated and subsequently lifted to 3D, or a **single-stage** formulation, where the model directly regresses 3D joint coordinates from images or videos. Two-stage pipelines benefit from the availability of highly accurate 2D pose estimators and offer modularity, whereas single-stage approaches aim to reduce error propagation at the expense of more demanding training requirements.

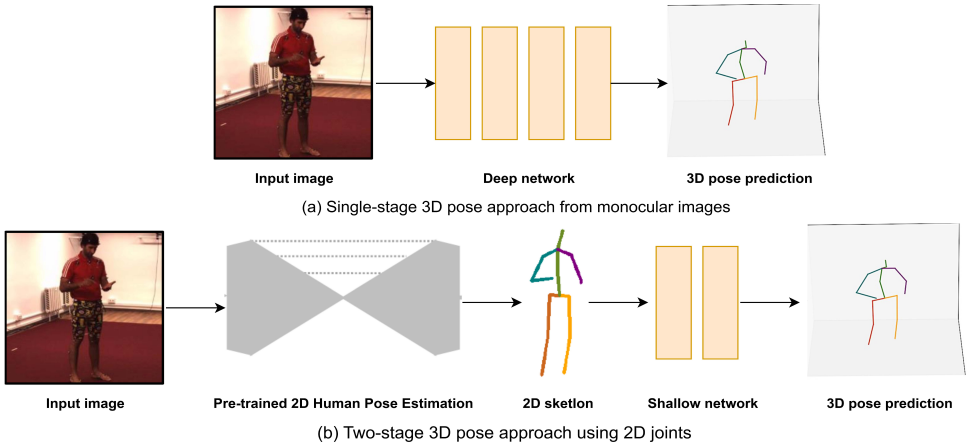

Figure 3.20: Different pipeline formulations for 3D HPE: (a) Single-stage, (b) Two-stage. Image taken from [9].

Another important dimension of classification is related to the number of subjects present in the scene. **Single-person** approaches assume a single individual and are commonly adopted in controlled environments, while **multi-person** methods address more realistic scenarios involving multiple interacting subjects. In the multi-person case, methods can be further divided into **absolute** approaches, which predict poses in a global camera-centric coordinate system, and **root-relative** approaches, which estimate each pose relative to a reference joint, typically the pelvis, and subsequently handle global localization and association.

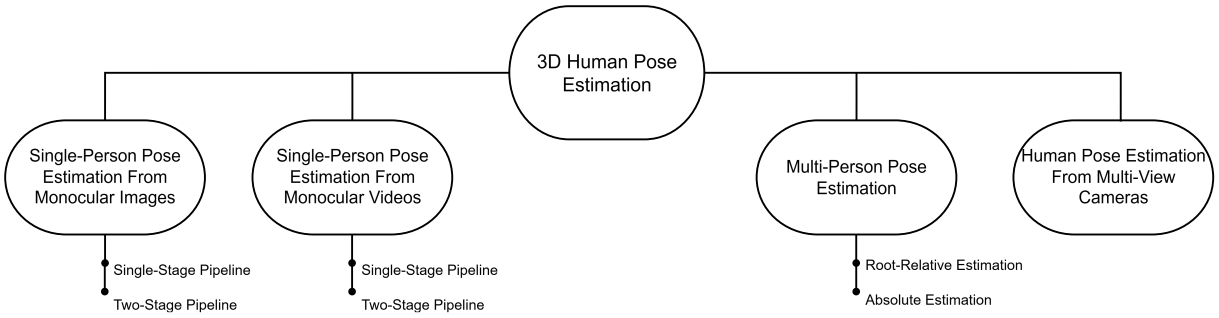

Figure 3.21: Taxonomy of 3D HPE approaches.

3.2.1. Datasets and metrics

Datasets

Like 2D HPE, 3D HPE models are typically trained on large datasets that contain frames with ground truth 3D keypoint locations. However, obtaining accurate 3D annotations is more challenging and often requires specialized equipment such as motion capture systems or multi-camera setups. For this reason, many 3D HPE datasets are collected in controlled environments, which may limit their diversity and generalization to real-world scenarios. As in 2D HPE, datasets **are not all annotated in the same way**, some use different keypoint definitions and number of keypoints. Below are some of the most commonly used datasets for 3D HPE.

Human3.6M The Human3.6M dataset [20] is the largest and most widely used benchmark for 3D human posture estimation. 11 professional actors' corresponding stances and 3.6 million interior video frames were recorded by the MoCap system from four different camera angles. It is a single-person dataset, and each person is annotated with 17 keypoints.

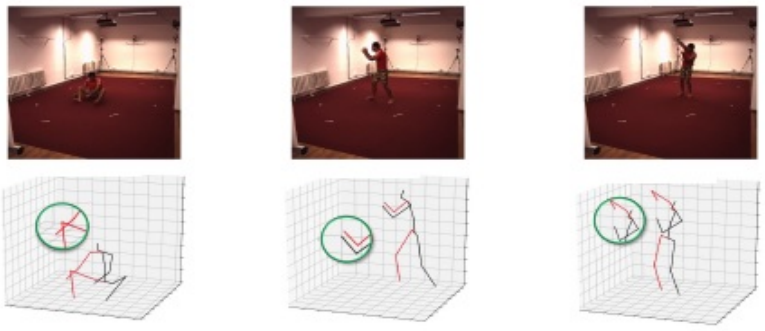

Figure 3.22: Some examples from the Human3.6M dataset.

MPI-INF-3DHP The MPI-INF-3DHP dataset [46] includes more than 1.3 million frames of both complicated outdoor and confined inside situations that were taken with markerless motion capture using 14 RGB cameras. Eight subjects are engaged in eight sets of activities. It is a single-person dataset, and each person is annotated with 17 keypoints.

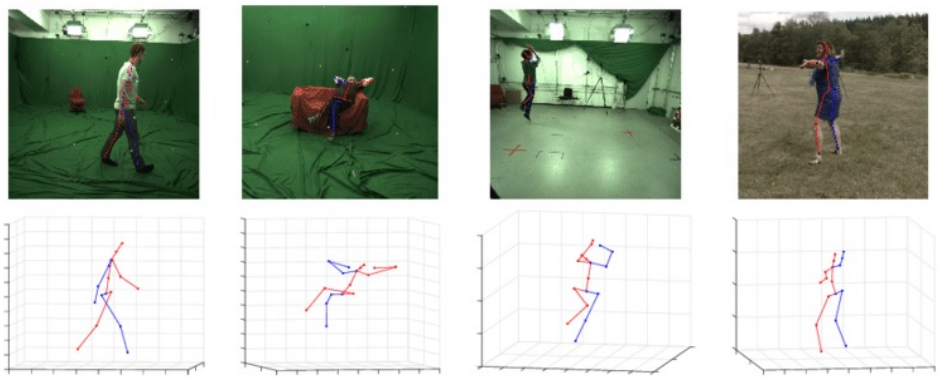

Figure 3.23: Some examples from the MPI-INF-3DHP dataset.

AMASS The AMASS dataset [41] is a large-scale dataset containing over 40 hours of motion capture data from more than 300 subjects, with each frame annotated with a detailed 3D mesh representation of the human body using the SMPL model and 24 keypoints.

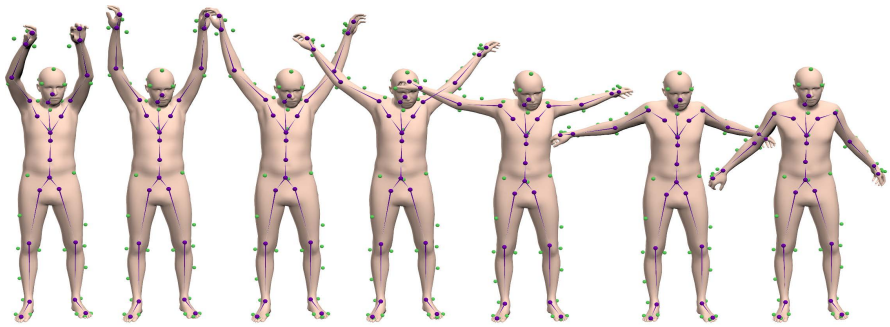

Figure 3.24: Some examples from the AMASS dataset [41].

SportsPose The SportsPose dataset [18] is a large-scale 3D human pose dataset consisting of highly dynamic sports movements. SportsPose offers a wide range of 3D poses that capture the intricate and dynamic character of sports motions, with over 176,000 poses from 24 different persons engaging in five different sports activities. It is a single-person dataset, and each person is annotated with 17 keypoints following the COCO format.

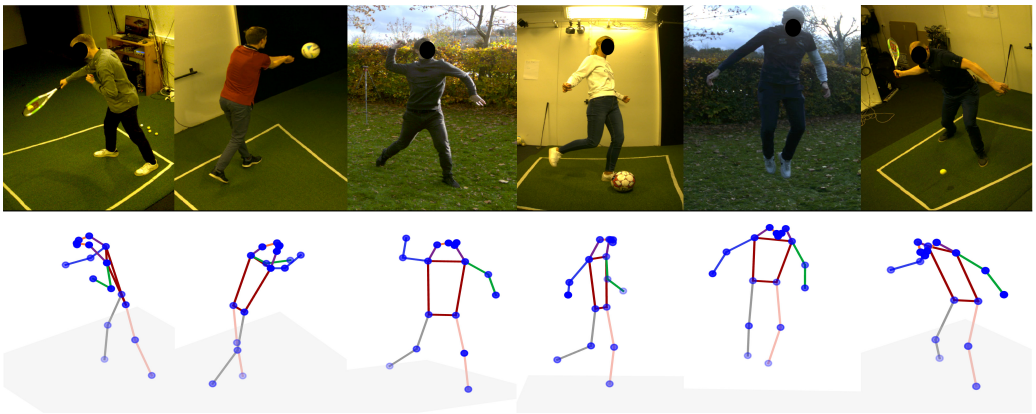

Figure 3.25: Some examples from the SportsPose dataset [18].

AthletePose3D The AthletePose3D dataset [75] includes 12 types of sports motions across various disciplines, with approximately 1.3 million frames and 165 thousand individual postures, specifically capturing high-speed, high-acceleration athletic movements. It is a single-person dataset, and each person is annotated with 17 keypoints.

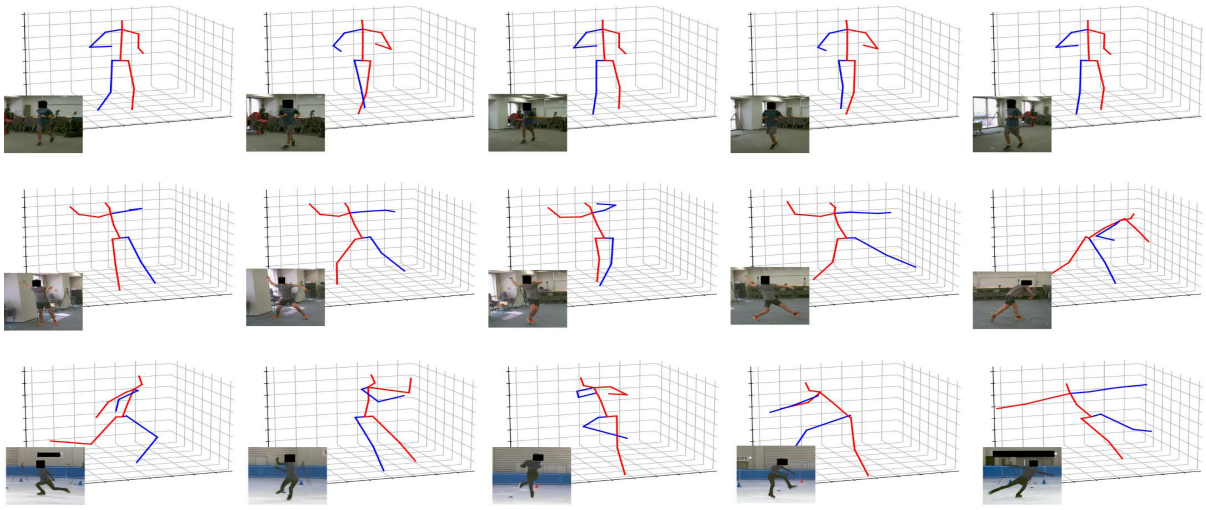

Figure 3.26: Some examples from the AthletePose3D dataset [75].

Evaluation Metrics

The evaluation metrics for 3D HPE have to account for the additional depth dimension and the inherent ambiguities in 3D pose estimation.

Mean Per Joint Position Error (MPJPE) Mean Per Joint Position Error (MPJPE) measures the average Euclidean distance between the predicted and ground truth 3D joint

positions. It is defined as:

$$\text{MPJPE} = \frac{1}{N}\frac{1}{J}\sum_{n=1}^{N}\sum_{j=1}^{J}\|\hat{\mathbf{P}}_{n,j} - \mathbf{P}_{n,j}\| \tag{3.9}$$

where $\hat{\mathbf{P}}_{n,j} \in \mathbb{R}^3$ and $\mathbf{P}_{n,j} \in \mathbb{R}^3$ are the predicted 3D and ground truth coordinates of the j-th joint in the n-th frame, respectively, N is the total number of frames and J is the total number of joints. MPJPE is an absolute error, and it is sensitive to translation, rotation, and scale errors.

Procrustes Aligned MPJPE (PA-MPJPE) is a variant of MPJPE that first aligns the predicted pose to the ground truth pose using a similarity transformation that operates on translation, rotation, and scale. This alignment helps to isolate the pose estimation error from global transformations, and it is called **Procrustes analysis**. I need to find:

$$s, R, t \quad \text{such that} \quad \min_{s,R,t}\sum_{j=1}^{J}\left\|sR\hat{P}_j + t - P_j\right\|^2 \tag{3.10}$$

where $s \in \mathbb{R}^+$ is a scaling factor, $R \in \text{SO}(3)$ is a rotation matrix, and $t \in \mathbb{R}^3$ is a translation vector. After finding the optimal s, R, and t, the aligned predicted pose is computed as:

$$\text{PA-MPJPE} = \frac{1}{N}\frac{1}{J}\sum_{n=1}^{N}\sum_{j=1}^{J}\left\|s_n R_n \hat{P}_{n,j} + t_n - P_{n,j}\right\| \tag{3.11}$$

This metric measures only the shape differences between the predicted and ground truth poses, making it more robust to global transformations.

Bone-Aligned MPJPE (BA-MPJPE) is another variant of MPJPE. The idea is to translate the skeleton and normalize the scale based on the lengths of the bones, before computing the MPJPE. This approach does not apply global rotation.
To compute BA-MPJPE, the first step is to calculate the root alignment:

$$\tilde{\hat{P}}_j = \hat{P}_j - \hat{P}_{\text{root}}, \quad \tilde{P}_j = P_j - P_{\text{root}} \tag{3.12}$$

Next, compute the bone-length scaling. Let's define a scale:

$$\alpha = \frac{\sum_k l_k^{gt}}{\sum_k l_k^{pred}} \tag{3.13}$$

where l_k is the length of bone k:

$$l_k = \|P_{i_k} - P_{j_k}\| \tag{3.14}$$

where i_k and j_k are the indices of the joints connected by bone k. Then apply the scaling:

$$\hat{P}_j^{\text{scaled}} = \alpha \tilde{\hat{P}}_j \tag{3.15}$$

Finally, compute the BA-MPJPE:

$$\text{BA-MPJPE} = \frac{1}{N}\frac{1}{J}\sum_{n=1}^{N}\sum_{j=1}^{J}\left\|\hat{P_{n,j}}^{\text{scaled}} - \tilde{P_{n,j}}\right\| \tag{3.16}$$

Mean Root Position Error (MRPE) Mean Root Position Error (MRPE) measures the average Euclidean distance between the predicted and ground truth root joint in the absolute localization. It is defined as:

$$\text{MRPE} = \frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{P}^t_{\text{root}}} - \mathbf{P}^t_{\text{root}}\| \tag{3.17}$$

3.2.2. Single-Person Pose Estimation From Monocular Images

The goal is to infer the three-dimensional coordinates of human body keypoints for a single subject from a single RGB image. In this context, the monocular setting represents one of the most challenging configurations for 3D human pose estimation. Unlike multi-view or stereo setups, which provide explicit depth cues through multiple perspectives, monocular images offer only a single two-dimensional projection of the scene. Consequently, estimation methods must rely exclusively on the visual information contained within a single image to recover the underlying 3D pose. This inherent lack of depth information makes the problem ill-posed and increases the difficulty of accurately reconstructing the human body configuration. Despite these challenges, monocular approaches present notable advantages. They require only a single camera, making data acquisition simpler and more cost-effective, and they are more easily applicable to real-world scenarios where multi-camera systems or specialized hardware are impractical. As a result, monocular 3D human pose estimation remains an active area of research, motivating the development of increasingly robust and sophisticated models capable of inferring accurate 3D poses from single-view image.

The two main approaches for this task are **single-stage methods** and **two-stage methods**.

Single-stage methods

Single-stage methods aim to directly predict the 3D joint coordinates from the input image without relying on intermediate representations such as 2D keypoints. These approaches typically employ deep CNNs to learn a mapping from image pixels to 3D joint locations using a direct regression approach.

The approach presented by Tekin et al. [64], which used an end-to-end regression architecture to achieve structured prediction by adding a pre-trained autoencoder at the top of conventional CNN networks instead of directly regressing joint coordinates, is one of the pioneering works in this field. They were able to account for joint interdependence and learn a high-dimensional latent pose representation using the autoencoder.
SSP-Net, a scalable convolutional neural network architecture created especially for real-time 3D human pose regression, was introduced by Luvizon et al. [39]. The difficulties brought on by different input sizes and model complexity are addressed by SSP-Net. In particular, multi-scale processing is made possible by its pyramid structure, which captures a variety of details and contextual information. By incorporating intermediate supervisions at various resolutions, SSP-Net improves accuracy and refines pose predictions.

Two-stage methods

Two-stage approaches to 3D human pose estimation follow a decoupled two-step paradigm. The process first estimates 2D joint locations in the image plane and subsequently lifts these keypoints into the three-dimensional space. This formulation explicitly separates 2D pose detection from 3D pose reconstruction, effectively combining advances in 2D human pose estimation with dedicated 3D inference models.

One of the main advantages of this strategy lies in its modularity. By decoupling the two tasks, each stage can be optimized independently, allowing the 3D reconstruction component to directly benefit from continuous improvements in 2D pose estimation. In particular, two-stage pipelines can leverage highly accurate 2D pose estimators pre-trained on large-scale datasets such as COCO or MPII, resulting in improved robustness and generalization in unconstrained environments.

Furthermore, two-stage methods are generally more data-efficient with respect to 3D supervision. Since the lifting stage operates on a compact representation of 2D keypoints

rather than raw images, it can be trained with fewer annotated 3D samples and can naturally incorporate geometric and kinematic constraints. This makes such approaches particularly suitable in scenarios where large-scale 3D ground truth data is scarce or difficult to acquire.

From an architectural perspective, the lifting module is often simpler and computationally lighter than end-to-end image-based models, as it processes low-dimensional pose representations instead of high-resolution visual inputs. Additionally, the explicit intermediate 2D representation enhances interpretability, as errors in the final 3D pose can be more easily traced back to inaccuracies in either the 2D detection or the 3D reconstruction stage.

Martinez et al. [44] developed and examined a straightforward yet quick neural network for 2D to 3D lifting. This experiment demonstrated that lifting 2D poses is not as difficult as previously believed.

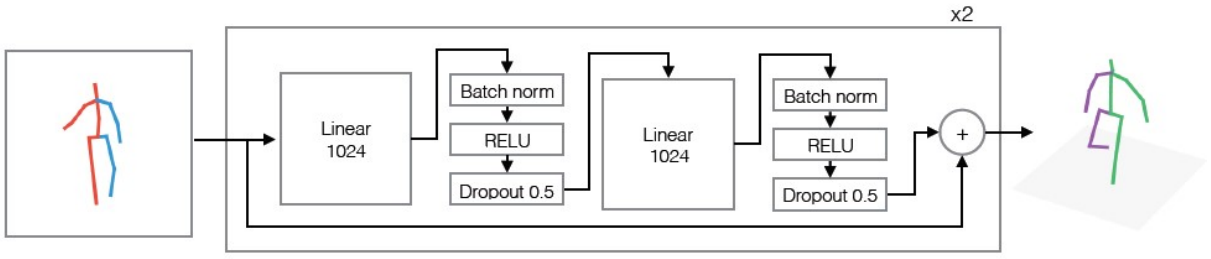

Figure 3.27: Framework of the NN proposed by [44]

A self-supervised technique for 3D human body posture prediction from a single image was presented by Sosa et al. [60] in a more recent year. A dataset of unlabeled images showing humans in common stances and a collection of unpaired 2D poses are used to train the prediction network. With just an empirical prior on unpaired 2D poses, the method simultaneously learns 2D and 3D pose representations in a largely unsupervised manner: a first CNN maps an input image to an intermediate skeleton image, a second CNN maps from the intermediate skeleton to a 2D pose representation, and a third fully connected network lifts the 2D pose to the necessary 3D pose.

In a comparable way, SelfPose [66] is regarded as self-supervised 3D egocentric pose estimation. SelfPose uses a downward-looking fish-eye camera mounted on a head's rim to evaluate a person's 3D pose. From top to bottom, the network takes an egocentric perspective. The proposed architecture for egocentric 3D HPE consisting of two modules: a 2D pose detector that predicts heatmaps from the input RGB image, and a multi-branch auto-encoder that finds a representation of poses which includes also a level of uncertainty

of prediction per joint.

KeyDiff3D [22] is a more contemporary method that reliably predicts 3D keypoints from a single image using an unsupervised monocular 3D keypoint estimation framework. This method allows monocular 3D keypoint estimate using only a collection of single-view images, whereas earlier methods relied on calibrated multi-view images or manual annotations, both of which are costly to acquire. It does this by using strong geometric priors included into a multi-view pre-trained diffusion model. This model generates multi-view images from a single image in the framework, which acts as a supervision signal to give the model 3D geometric cues.

3.2.3. Single-Person Pose Estimation From Monocular Videos

Single-person 3D human pose estimation from monocular videos addresses the problem of reconstructing the three-dimensional motion of an individual from a sequence of RGB frames captured by a single camera. Compared to image-based approaches, video-based methods can exploit temporal continuity and motion dynamics, which provide additional cues for resolving depth ambiguities inherent to the monocular setting. The monocular video scenario represents a particularly challenging yet realistic configuration, as it lacks explicit geometric depth information while reflecting acquisition conditions commonly encountered in real-world applications. In this context, temporal information becomes a key source of constraint, as human motion follows consistent kinematic patterns over time. By modeling these temporal dependencies, video-based methods can enforce smoothness, reduce frame-wise jitter, and improve the overall stability of the estimated 3D poses.

Single-stage methods

A single-stage approach is also used by some models in video-based approaches. A multi-task framework was presented by Luvizon et al. [38] to address the challenges of simultaneously estimating 2D or 3D human poses and action recognition. Instead, Honari et al. [16] suggested an unsupervised feature extraction technique to extract rich latent vectors from monocular videos by identifying and encoding subjects of interest in each frame.

Two-stage methods

Two-stage approaches represent the de facto standard in monocular video-based 3D human pose estimation. By explicitly separating 2D pose estimation from 3D pose reconstruction, these methods enable the seamless integration of highly accurate 2D pose estimators pre-trained on large-scale datasets. This decoupled formulation provides a high degree of

modularity and flexibility, allowing each stage to be optimized independently, while also reducing computational complexity by operating on compact pose representations rather than raw image data.

An encoder *seq-to-seq* with two kinds of transformer blocks—a spatial and a temporal one—is introduced by MixSTE [78]. The model alternates between a spatial block to learn the relationship between joints in the same frame and a temporal block to describe the dynamics of a single joint. For each joint representation, MixSTE projects the input 2D keypoints into a high-dimensional features space. The model alternatively learns the spatial correlation and distinct temporal motion using this latent space. Lastly, it uses a regression to get the 3D outputs for every input frame.

The objective of MotionBERT [84] is to learn the representation of human movement from a variety of data sources by focusing on the difficulty of acquiring 3D position in a real setting. There are two stages to the framework. It takes 2D skeleton sequences from several motion data sources and corrupts them with random masks and noises during the pretraining stage. The motion encoder is then trained to extract the 3D motion from the corrupted 2D skeletons. The motion encoder is made up of a Dual-Stream Spatio-Temporal Transformer (DSTFormer), which assembles the fundamental building blocks to fuse the spatial and temporal information in the flow given a spatial and temporal MHSA that records the intra-frame and inter-frame body joint interactions, respectively. In order to solve one of these three tasks—3D pose estimation, action recognition, and mesh recovery—the motion encoder that had been pretrained in the preceding stage was refined in the finetune stage.

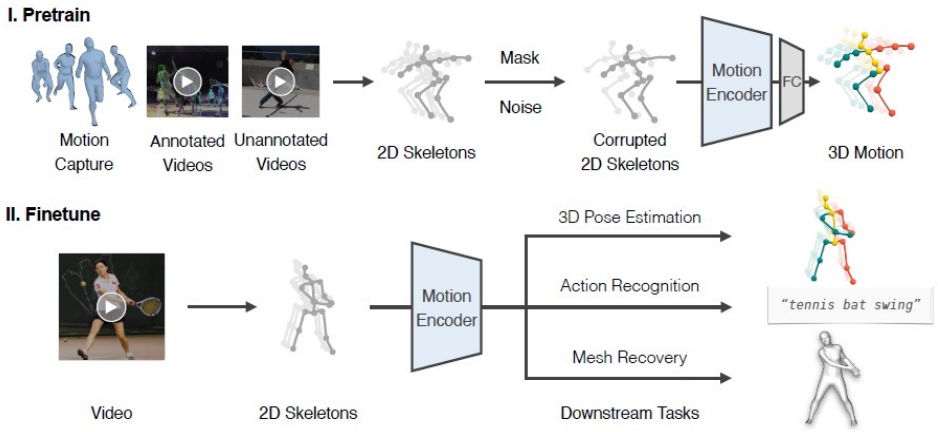

Figure 3.28: The framework of MotionBERT [84].

In comparison to MotionBERT, MotionAGFormer [45] emphasizes a lighter structure. It makes use of a hybrid transformer-graph architecture designed specifically for 3D

HPE. Graph Convolutional Networks are used to integrate local spatial and temporal relationships whereas Transformers are used to capture global information. The spatial metaformer efficiently captures intra-frame interactions within a single frame by processing individual body joints as unique tokens. The temporal metaformer captures inter-frame interactions over time by treating each frame as a single token.

TCPFormer [35] uses an implicit pose proxy as an intermediate representation. The Proxy Update Module (PUM), Proxy Invocation Module (PIM), and Proxy Attention Module (PAM) are the three main parts of the technique. In order to store representative data from the posture sequence, PUM first updates the implicit pose proxy using pose features. The motion semantics of each pose are then improved by PIM by invoking and integrating the pose proxy with the pose sequence. Lastly, PAM improves the temporal connection of the entire posture sequence by utilizing the aforementioned mapping between the pose sequence and pose proxy.

Li et al. [32] adopt a more current method. They suggested a four-stage framework called MVLift. A line-conditioned diffusion model that produces 2D pose sequences while adhering to epipolar line restrictions is trained in the first stage. Through an optimization technique that directly optimizes multi-view 2D sequences using explicit multi-view consistency objectives, they establish stronger multi-view consistency in stage two. In the third stage, optimize 2D reprojection objectives to recover believable 3D motions. In stage four, they directly produce multi-view consistent 2D sequences in a single forward pass by training a specific diffusion model.

3.2.4. Multi-Person Pose Estimation From Monocular Videos

Estimating 3D human pose from a single camera becomes significantly more difficult when multiple people are present. In addition to the challenges of single-person estimation, the model must handle occlusions, differences in body size and orientation, interactions, and complex spatial relationships between individuals. Each person can influence how others are perceived, and the difficulty grows rapidly as more people enter the scene. Despite these challenges, multi-person 3D pose estimation is crucial for applications such as crowd monitoring, team sports analysis, and studying social interactions.

PandaNet, introduced by Benzine et al. [3], is a single-shot anchor-based framework that jointly predicts bounding boxes along with 2D and root-relative 3D human poses in a single forward pass, eliminating the need for post-processing to associate body joints. To better manage overlapping individuals, it employs a pose-aware anchor selection mechanism and adjusts the optimization weights across different person scales and joint coordinates,

mitigating the imbalance caused by variations in human size within the image.

XNect [47] is an absolute 3D HPE method that follows a sequential multi-stage pipeline. In the first stage, a convolutional neural network processes the full image to predict 2D joint heatmaps along with intermediate 3D pose features for each detected joint. After grouping joints into individual persons, the second stage employs a fully connected network to reconstruct a complete absolute 3D pose for each subject. Finally, a third stage applies sequential model fitting over the predicted poses to ensure temporally consistent and smooth motion capture results.

3.2.5. Human Pose Estimation From Multi-View Cameras

Using several camera views has become a potent tactic in the field of 3D human pose estimation to get over the inherent problems caused by the loss of depth information during the projection from the 3D environment to the 2D image plane. By recording many viewpoints of the subject, multi-view techniques provide an improved understanding of spatial structure, allowing for more accurate pose estimation.
These techniques face challenges. They require particular camera setups, which might not be available or useful in many real-world situations. Additionally, information obtained from these systems may need a significant amount of processing time. For situations needing quick outcomes, like real-time tracking or rapid feedback, this slowness may make them ineffective. Another problem is spatial limitations. The locations where these systems can be employed are limited by the requirement that cameras be positioned at different angles around the target, which may not be possible in small or congested settings. Another possible barrier is cost. Due to the requirement for several cameras and their specific configuration, multi-view systems can be costly to implement, which may restrict their use to those who can afford such expenses.

A substantial amount of 3D annotated data is needed to accurately estimate 3D human pose from monocular camera images. It is difficult to collect 3D annotated data outside the lab, unfortunately. Gholami et al. [10] attempt to address this issue with a weakly-supervised approach that uses single-view cameras for inference and multi-view cameras for training. Using classical triangulation, pseudo-3D labels are obtained from the available unlabeled uncalibrated multi-view inputs. These pseudo-3D labels and multi-view re-projection loss are used to train a pose estimator.

HMVFormer [82] introduces a set of multi-view fusion modules that are integrated into the feature extraction process to enable efficient multi-level aggregation of information from multiple viewpoints. To effectively capture both discriminative and consistent cues across

views and improve pose feature representations, the method progressively incorporates cross-view fusion mechanisms within the spatial and temporal feature extraction stages of each individual view. Feature fusion between neighboring viewpoints is achieved by modeling associations between corresponding human joints using multi-head attention, allowing the integration of spatial semantic information across views. Subsequently, a multiview enhancement module aggregates view-aware spatial features from all available viewpoints through channel-wise fusion. Finally, temporal dependencies within the video sequence are exploited to further refine the representation, resulting in a spatio-temporally enhanced feature for 3D pose estimation.

Srivastav et al. [61] propose SelfPose3D, a self-supervised framework for multi-person 3D pose estimation from multi-view images. The method does not rely on annotated 2D or 3D ground-truth poses, but instead leverages calibrated multi-view inputs together with pseudo 2D poses obtained from an off-the-shelf 2D pose estimator. The approach introduces two self-supervised objectives: 3D person localization and 3D pose estimation. Person localization is learned using synthetically generated 3D root positions and their projections across views, while 3D poses are modeled through a bottleneck representation and projected back to 2D joint heatmaps in a fully differentiable manner, enabling supervision via the pseudo 2D poses.

3.3. Drone Based Human Pose Estimation

In recent years, there has been a growing interest in the use of drones for outdoor video acquisition, driven by their ability to access a wide range of viewpoints with minimal deployment effort. Unlike fixed cameras, drones can dynamically adapt their position in the environment, making them particularly well suited for covering large areas and tracking subjects in motion. As a result, drone-based video capture has found increasing adoption in several application domains, including sports analytics, surveillance, and search-and-rescue operations.

The main advantages of a drone-based setup stem from the mobility of the camera, which enables the acquisition of viewpoints that would otherwise be difficult or impossible to obtain, such as top-down or oblique views during motion. The ability to dynamically adjust the camera position allows drones to capture complex scenes and follow subjects over time, potentially reducing occlusions that commonly affect fixed, ground-based camera setups. Moreover, in environments where the installation of static cameras is impractical or infeasible, drones provide a flexible and effective alternative for video and image acquisition.

Despite its advantages, drone-based human pose estimation remains a highly challenging problem. The aerial viewpoint is often strongly oblique or near top-down, leading to extreme scale variations, pronounced perspective distortions, and an apparent reduction in the spatial resolution of the observed subjects. In addition, camera motion introduces artifacts such as jitter and motion blur, which further complicate accurate pose estimation. From a geometric perspective, depth ambiguities are significantly amplified in the single-view drone setting, where no explicit multi-view constraints are available. Furthermore, drone-acquired data exhibit a substantial domain gap compared to standard human pose estimation datasets, which are typically captured from ground-based, static cameras in controlled environments. This gap is exacerbated by the scarcity of annotated drone-based datasets, as obtaining reliable 3D ground truth in outdoor scenarios is particularly difficult. As a result, only a limited number of datasets and studies specifically address human pose estimation from drone imagery.

3.3.1. Datasets

As was already mentioned, there aren't many datasets collected by drones. A single UAV captured the video and images in the UAV-Human dataset [33]. Its tasks are 2D pose estimation, action recognition, attribute recognition, and human re-identification.

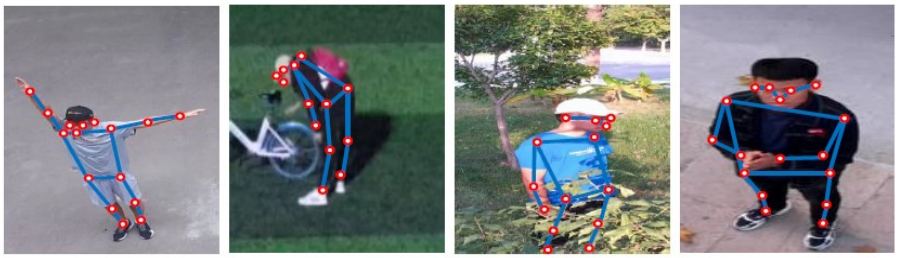

Figure 3.29: Some examples of 2D HPE from UAV-Human dataset [33].

AirPose [56] offers another dataset. This dataset, which can be utilized for 3D HPE, was gathered using two UAVs. It consists of roughly 17 minutes of video captured by two UAVs and synthetic images generated with Unreal Engine.

Figure 3.30: Some examples from AirPose dataset [56].

3.3.2. Methods

Saini et al. [55] introduce AirCap, the first fully autonomous outdoor human motion capture system based on multiple micro-aerial vehicles (MAVs). Each MAV is equipped with a monocular RGB camera, an IMU, and a GPS module, enabling markerless capture of freely moving humans in unconstrained outdoor environments. The system operates in two stages: an online acquisition phase, in which multiple MAVs autonomously and cooperatively detect and track a subject while estimating both the subject's 3D position and the camera extrinsics, and an offline reconstruction phase, where human pose and body shape are estimated using only the acquired RGB images and the self-localization information of the MAVs. AirCap addresses several key challenges of drone-based human pose estimation, including autonomous multi-drone person detection and tracking, joint estimation of camera poses and subject location, and robust fitting of a 3D body model to 2D joint detections from multiple moving aerial viewpoints. Most importantly, the work demonstrates, for the first time, the feasibility of fully autonomous human motion capture using aerial vehicles.

A few years later, the same authors proposed AirPose [56], a decentralized and distributed system of neural networks for uncalibrated moving cameras that concurrently calibrates the cameras in relation to the human and estimates human 3D pose and shape. Combine data by utilizing complimentary information from multiple viewpoints to enhance the estimation of the person's pose. An autoregressor stage comes after a ResNet50 [14] feature extractor in the network architecture. For ResNet50 input, the human bounding-box region is cropped and scaled to the fixed-size image. The cropping and scaling parameters, along with the ResNet50 features, are concatenated to represent the full-size image. The autoregressor uses this compact input in conjunction with the SMPL-X [51] parameters.

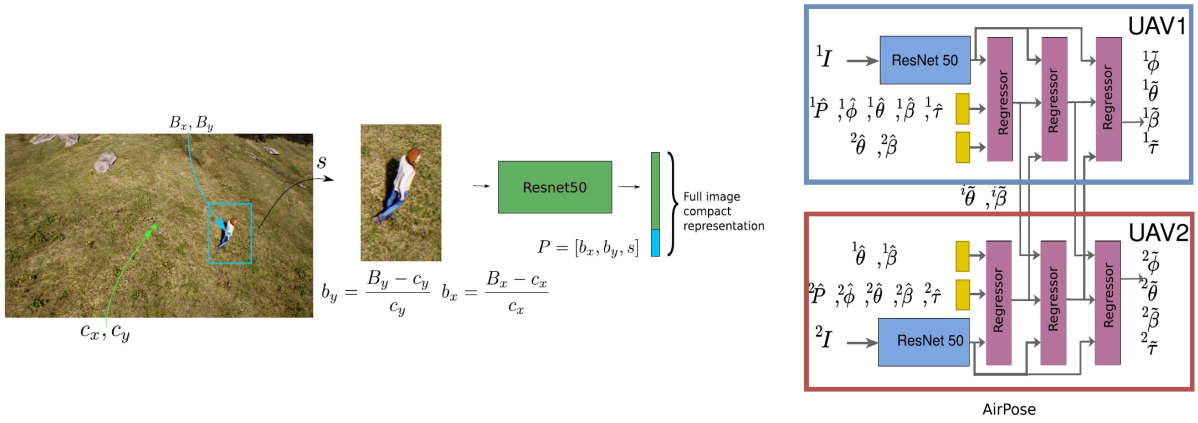

Figure 3.31: AirPose framework [56].

Aerial View [17] uses monocular images taken by a single drone to address the same 3D HPE challenge. A double Vector Quantized-Variational AutoEncoder is used by the authors. To create a codebook of discrete representation of 3D poses, a first VQ-VAE is trained on a dataset of human poses. The visual features of aerial images are encoded using a different VQ-VAE. The 3D pose codebook maps the visual representation. The system can reliably infer and optimize 3D pose thanks to this connection between visual and spatial space.

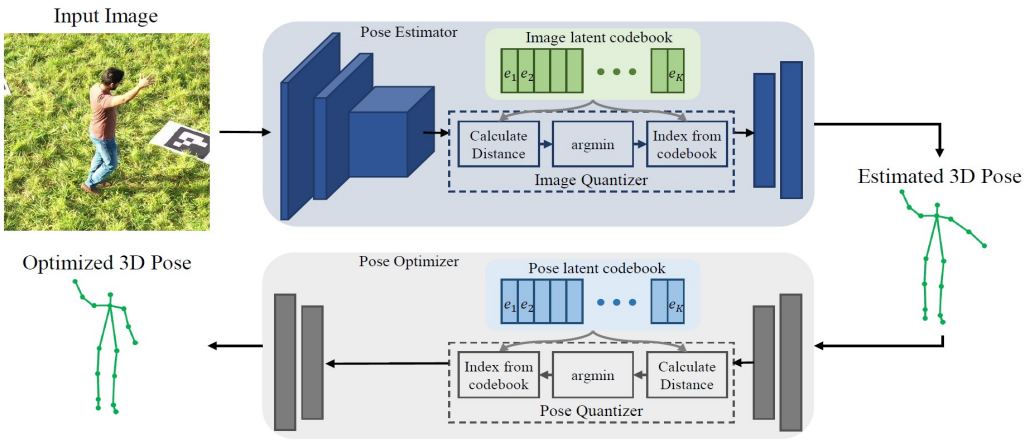

Figure 3.32: Aerial View framework [17].

In order to solve the occlusion problem in HPE in video, Chen et al. [5] propose a method where the UAV moves actively and autonomously to improve 2D HPE and gain better angle views. The system is a light network that can discover the relationship between the current HPE and the optimal subset of next view angles. The framework leverages the 3D perceptual guidance field for motion planning, and it can generate a trajectory that avoids occlusion.

Zhuge et al. [85] propose a markerless motion-capture technique to estimate the 3D pose of humans using a multi-UAV system based on visual data for uncalibrated airborne cameras. After obtaining the pose of airborne cameras, the 3D pose of humans is obtained through stereo vision intersection. They employed an algorithm to concurrently determine the sparse point cloud of the background in an overlapping view and estimate the position of aerial cameras. The accuracy of pose estimation for airborne cameras is improved by removing the wrong conjugate keypoint pairs using a 3σ rule based on reprojection error.

4 | Problem Formulation

The goal of this thesis is to estimate the 3D motion of athletes from monocular drone footage.
Specifically, the task consists in predicting the 3D HPE of an athlete over time, represented as a sequence of 3D joint coordinates. Given a video acquired from a UAV-mounted camera, the objective is to recover the spatio-temporal configuration of the human body despite challenging acquisition conditions such as motion blur, occlusions, and dynamic camera motion.

4.1. Input Representation

The input to the model is an RGB video sequence :

$$x \in \mathbb{R}^{T \times H \times W \times C} \tag{4.1}$$

where T is the temporal length of the video, H and W are the spatial dimensions of each frame and C is the number of channels.
The video is assumed to be captured by a moving drone-mounted monocular camera, resulting in non-static backgrounds and varying viewpoints.

4.2. Output Representation

The output is a sequence of 3D human poses, one for each video frame. Each pose is represented as a set of J 3D joint coordinates:

$$y \in \mathbb{R}^{T \times J \times P} \tag{4.2}$$

where T is the temporal length of the sequence and $P = 3$ represents for each joint its (x, y, z) coordinates in a root-relative reference frame, obtained by translating all joints such that the root joint is located at the origin.

4.3. Task Definition

The problem addressed in this thesis is to learn a function f_θ, parameterized by θ, that maps the input video to a sequence of 3D human poses:

$$f_\theta : x \in \mathbb{R}^{T \times H \times W \times C} \rightarrow y \in \mathbb{R}^{T \times J \times P} \tag{4.3}$$

4.4. Learning Objective

Given a training set of video sequences and corresponding ground-truth 3D poses, the learning objective is to minimize the discrepancy between predicted and ground-truth joint positions. This is typically achieved by minimizing the *MPJPE* loss (see 3.2.1) over all joints and time steps:

$$\text{MPJPE} = \frac{1}{T}\frac{1}{J}\sum_{t=1}^{T}\sum_{j=1}^{J}\|\hat{\mathbf{y}_{t,j}} - \mathbf{y}_{t,j}\| \tag{4.4}$$

4.5. Challenges and Constraints

Estimating 3D human pose from drone footage presents several challenges:

- **Motion blur and low resolution**, caused by fast camera and subject motion;
- **Self-occlusions and partial visibility** of the human body;
- **Dynamic camera motion**, which prevents the use of static background assumptions;
- **Limited availability of labeled datasets**, as only one publicly available dataset currently provides UAV-based 3D human pose annotations.

These factors make the problem significantly more challenging than standard 3D HPE from fixed ground cameras.

5 | Proposed Method

In this chapter, I present the proposed method for single-view, single-person 3D human pose estimation from aerial videos captured by a drone. The proposed approach addresses the challenging setting of in-the-wild sports scenarios, where annotated 3D pose data are scarce and difficult to acquire.

I introduce **Brancher**, a self-supervised, two-stage neural network designed to infer the 3D pose of an athlete from raw monocular video. Given an input video sequence, the pipeline first detects the target person and extracts 2D keypoints for each frame. In the second stage, the estimated 2D poses are temporally aggregated and lifted to a coherent 3D pose representation.

To overcome the limited availability of labeled 3D data, the proposed method adopts a self-supervised learning strategy. After an initial supervised training phase on a labeled dataset, the model is further optimized using unlabeled videos, allowing it to continuously adapt to real-world aerial footage captured by drones. This learning paradigm enables the model to reduce the domain gap between controlled datasets and unconstrained in-the-wild scenarios.

At the core of the proposed architecture lies a spatio-temporal module, which explicitly models both temporal dependencies across frames and spatial relationships among body joints. Building upon this shared representation, the network is organized into three specialized branches, each dedicated to optimizing a complementary aspect of human motion. This multi-branch design allows the model to disentangle different constraints of the pose estimation task and improve overall robustness and accuracy.

Before introducing the full Brancher architecture, I first describe a simple baseline model, which serves as a reference point and highlights the benefits brought by the proposed design choices.

5.1. Preprocessing

When dealing with raw RGB videos, an explicit preprocessing stage is mandatory. Directly training the proposed network on unprocessed videos would be computationally infeasible, leading to excessive RAM and GPU memory consumption and prohibitively long training times, potentially spanning several days. For this reason, all input videos are converted into a compact and structured representation before being used for training.

5.1.1. Video Resize and Truncation

Given an input RGB video of shape $[T, H, W, C]$, where T denotes the number of frames, H and W the spatial resolution in pixels, and $C = 3$ the RGB color channels, each video is first resized while preserving its aspect ratio. Specifically, the longer spatial dimension (i.e., H for vertically oriented videos or W for horizontally oriented ones) is scaled to 640 pixels, and the shorter dimension is resized accordingly to maintain the original aspect ratio. Additionally, the temporal length is truncated to a maximum of $T = 81$ frames. This operation ensures a bounded input size and prevents memory usage from scaling with the original video resolution or duration.

5.1.2. Human Detection and Cropping

To remove irrelevant background information and focus the model exclusively on the target athlete, a human detection step is applied. For each frame, a bounding box enclosing the person is extracted using a YOLOv12 detector [65], chosen for its favorable trade-off between accuracy and inference speed. The detected bounding box is then used to crop the frame around the subject, significantly reducing spatial redundancy and improving the robustness of subsequent pose estimation.

5.1.3. 2D Keypoint Extraction

A pretrained 2D human pose estimator is applied to the cropped frames in order to extract joint-level information. Before feeding the frames to the pose estimator, they are resized to the input resolution required by the model $H = 256 \times W = 192$ pixels.
Specifically, I employ ViTPose [74], which produces a heatmap for each body joint, in the shape $[T, J, H_h, W_h]$, where J is the number of joints and H_h, W_h are the height and width of the heatmaps. This 2D model uses a ViT [8] encoder, pretrained on ImageNet [6], followed by a decoder of deconvolution layer utilized to get the localization heatmaps for the keypoints, as can be seen in Figure 5.1.

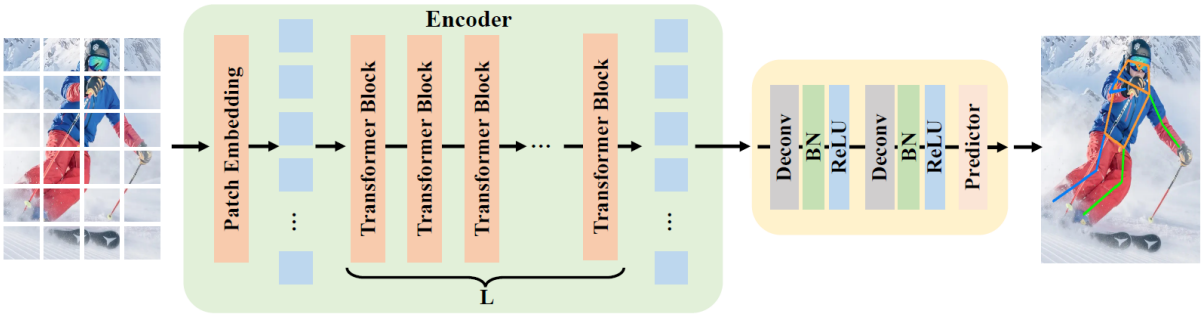

Figure 5.1: ViTPose framework [74].

The 2D coordinates of each keypoint are obtained by applying an *argmax* operation over the corresponding heatmap $\mathbf{H}$, while the peak value, that is a value in range $[0, 1]$ is used as a confidence score.

$$(u_{t,j}, v_{t,j}),\ c_{t,j} = \arg\max_{(u,v)} \mathbf{H}_{t,j}(u, v) \tag{5.1}$$

where $(u_{t,j}, v_{t,j})$ denote the spatial coordinates of the maximum response at time t for the j-th joint and $c_{t,j}$ is the corresponding confidence score.

As a result, for each frame, a set of 2D keypoints with associated confidence values is obtained with shape $[T, J, 3]$, where $J = 17$ corresponds to the number of joints defined by the COCO keypoint convention and the three channels correspond to the (x, y) coordinates and the confidence score.

However, a conversion is required because the great majority of 3D HPE models are trained using the notation established by the Human3.6M dataset [20]. Since there is currently no 2D dataset annotated with H36M notation, this conversion step from COCO notation is essential.

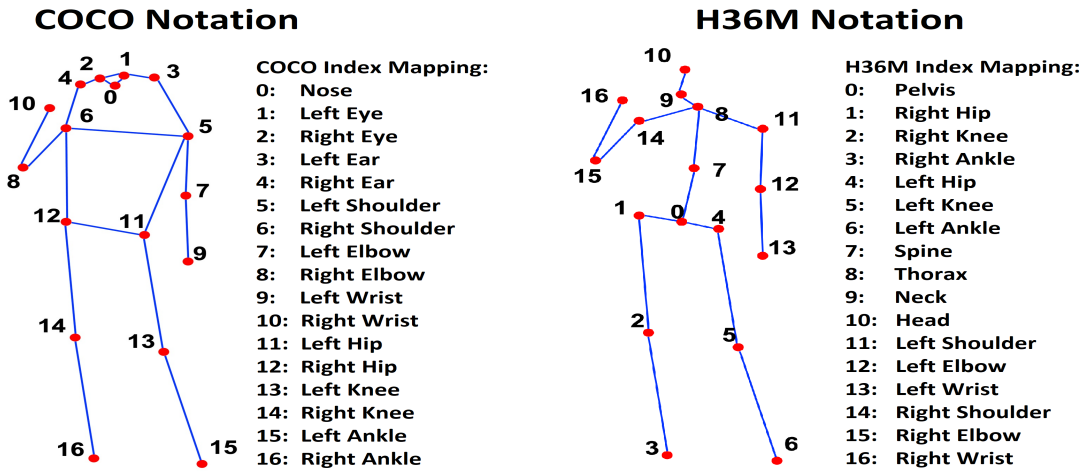

Figure 5.2: COCO notation and H36M notation for human keypoints.

5.1.4. Keypoint Normalization and Storage

To improve training stability and prevent exploding gradients, I map the extracted 2D keypoints to a camera-normalized coordinate system rather than directly using pixel coordinates. Instead of independently scaling each axis to $[-1, 1]$, I normalize both coordinates with respect to the image width, thereby preserving the original aspect ratio in the vertical direction.

Given a frame of width W and height H, I define the aspect ratio as $r = \frac{H}{W}$. For each keypoint $(u_{t,j}, v_{t,j})$ at time step t, the normalized coordinates are computed as:

$$x'_{t,j} = 2\frac{u_{t,j}}{W} - 1, \qquad y'_{t,j} = 2\frac{v_{t,j}}{W} - r \tag{5.2}$$

This transformation centers the image plane at the origin and expresses coordinates in a scale-invariant camera space. Under this formulation, the horizontal axis spans $[-1, 1]$, while the vertical axis spans $[-r, r]$, maintaining the correct geometric proportions of the image plane.

From a geometric perspective, this normalization approximates a pinhole camera coordinate system where the image plane is centered and expressed in normalized units. Empirically, I observed that this parameterization leads to more stable optimization compared to standard $[0, 1]$ scaling or independent $[-1, 1]$ normalization of both axes.

Finally, the preprocessed data are stored on disk and used as input to the proposed network. In addition to the normalized keypoints, saving the intermediate heatmaps is also beneficial, as they can be reused for further analysis or auxiliary supervision without recomputing the 2D pose estimation step.

Overall, this preprocessing pipeline transforms raw RGB videos into a compact representation of shape $[T, J, 3]$, where 3 are the coordinate (x', y') and the confidence score, enabling efficient training of the proposed model while preserving the essential spatio-temporal information required for 3D pose estimation.

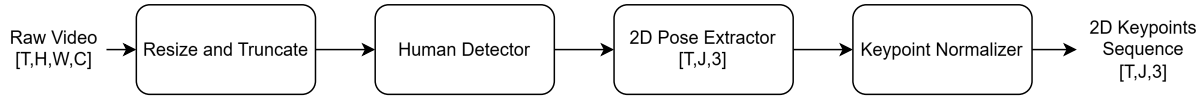

Figure 5.3: Pipeline of the preprocessing stage.

5.2. Baseline Model

A baseline model is introduced to provide a simple and efficient reference architecture for the development of the proposed method and for subsequent ablation studies. The goal of this baseline is not to achieve state-of-the-art performance, but to establish a reliable starting point against which the impact of more advanced design choices can be evaluated.

The baseline model operates on the preprocessed 2D keypoint sequences described in Section 5.1. Given an input tensor of shape $[T, J, 3]$, the model predicts the corresponding 3D human pose in root-relative coordinates.

5.2.1. Architecture

The input 2D keypoints are first **embedded** into a higher-dimensional feature space through a linear projection applied at the frame level. Specifically, the full set of joints within each frame is projected jointly, producing a tensor of shape $[T, D]$, where D denotes the embedding dimension. To encode temporal order information, a standard sinusoidal positional encoding based on *cosine* and *sine* functions [68] is added to the embedded features.

The temporally encoded features are then processed by a stack of **Transformer Encoder** layers [68], which model temporal dependencies across frames through self-attention. In this baseline configuration, the attention mechanism operates exclusively along the temporal dimension, without explicitly modeling spatial relationships between joints. This design choice intentionally limits the expressive power of the model, allowing a clear assessment of the benefits introduced by spatial and spatio-temporal attention in later sections.

Finally, a lightweight MLP-based **Pose Head** regresses the 3D joint coordinates from the transformer output. A *tanh* activation function is applied to the final layer to model nonlinearities. The final output shape is $[T, J, 3]$, where 3 is the number of 3D coordinates per joint (x, y, z).
The predicted 3D poses are expressed in root-relative coordinates, where the root joint is fixed at the origin $(0, 0, 0)$.

5.2.2. Training Objective

The baseline model is trained in a fully supervised manner using ground truth 3D annotations. The training objective is to minimize the loss Mean Per Joint Position Error

(**MPJPE**), computed as the average Euclidean distance between the predicted 3D joint positions and the ground-truth 3D poses.

$$\mathcal{L}_{MPJPE} = \frac{1}{T}\frac{1}{J}\sum_{t=1}^{T}\sum_{j=1}^{J}\|\hat{y}_{t,j} - y_{t,j}\| \tag{5.3}$$

where $\hat{y}_{t,j}$ and $y_{t,j}$ denote the predicted and ground-truth 3D coordinates of the j-th joint at time t, respectively.

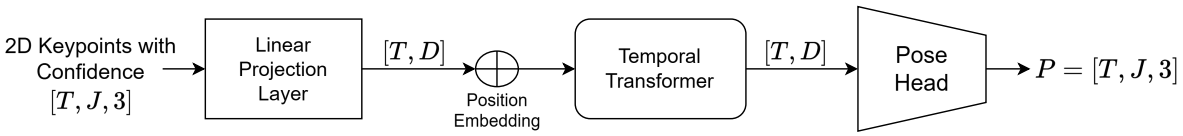

Figure 5.4: Baseline model architecture.

5.2.3. Discarded Design Choice

An alternative single-stage baseline, directly operating on raw RGB videos and relying on a pretrained ResNet encoder [14] for feature extraction, was initially explored. However, this approach resulted in prohibitively slow training and excessive computational cost, making it unsuitable for extensive experimentation. For this reason, the RGB-based baseline was discarded in favor of the more efficient keypoint-based formulation described above.

5.3. Proposed Network Architecture

In this section, I introduce **Brancher**, the proposed network architecture for single-view, single-person 3D human pose estimation from aerial videos. Brancher builds upon the baseline model described in the previous section and extends it along three main directions:

1. the explicit modeling of spatial relationships between body joints,
2. the introduction of a multi-branch architecture to optimize complementary aspects of human motion, and
3. the adoption of a self-supervised learning framework that enables training on unlabeled video data.

Unlike the baseline model, which models temporal dependencies only, Brancher employs

a spatio-temporal mechanism that jointly captures interactions across both frames and joints. This design allows the network to better exploit the structured nature of the human body and the temporal coherence of motion, which are critical in unconstrained drone-based scenarios.

Furthermore, the proposed architecture is organized into multiple specialized branches, each designed to focus on a distinct aspect of the pose estimation problem. By sharing a common spatio-temporal representation and optimizing branch-specific objectives, the model can disentangle different motion constraints and achieve more robust 3D pose predictions.

Finally, Brancher is trained using a self-supervised learning strategy. After an initial supervised pretraining stage, the network is further optimized using unlabeled videos, enabling continuous adaptation to real-world aerial footage and reducing the reliance on scarce 3D ground-truth annotations. This learning paradigm is particularly well-suited for drone-captured sports data, where obtaining accurate 3D labels is challenging and expensive.

5.3.1. Spatial and Temporal Module

To effectively model both spatial and temporal dependencies in human motion, the proposed network adopts the **AGFormer** module introduced in MotionAGFormer [45]. This choice is motivated by two main factors: the significantly lower number of parameters compared to alternative architectures such as DSTFormer [84], and the superior performance reported in benchmark evaluations. These properties make AGFormer particularly suitable for drone-based scenarios, where long temporal sequences must be processed efficiently.

The role of the spatial-temporal module is to refine the motion representation by jointly modeling interactions among body joints within each frame and motion dynamics across frames. This capability is essential for capturing coherent and physically plausible human motion from monocular aerial views.

MetaFormer Formulation

AGFormer follows the **MetaFormer** [76] design paradigm, which generalizes the Transformer architecture [68] by decoupling the token-mixing operation from the attention mechanism. Given an input feature representation $X \in \mathbb{R}^{N \times C}$, where N denotes the number of tokens and C the embedding dimension, a MetaFormer block can be expressed

as:

$$Y = \text{TokenMixer}(\text{Norm}(X)) + X, \tag{5.4}$$

where $\text{Norm}(\cdot)$ denotes a normalization operation and $\text{TokenMixer}(\cdot)$ represents a generic module that enables information exchange among tokens.

In AGFormer, the token-mixing operation is implemented using two complementary mechanisms: Multi-Head Self-Attention (MHSA) and Graph Convolutional Networks (GCNs) [29]. This dual formulation allows the model to capture both global and local dependencies in the input sequence.

Input Representation

The input to the spatial-temporal attention module consists of a sequence of embedded 2D keypoints. Each joint at each time frame is first projected into a D-dimensional feature space, resulting in an initial feature tensor $F^{(0)} \in \mathbb{R}^{T\times J\times D}$. To encode joint identity information, a learnable spatial positional embedding $P^{s}_{pos} \in \mathbb{R}^{1\times J\times D}$ is added to the input features. The resulting representation is then processed by a stack of N AGFormer blocks, producing refined features $F^{(i)} \in \mathbb{R}^{T\times J\times D}$, with $i = 1, \ldots, N$.

AGFormer Block

Each AGFormer block adopts a **dual-stream** architecture, where two parallel processing streams are employed to model motion at different granularities. Both streams consist of a **Spatial MetaFormer** followed by a **Temporal MetaFormer**, but differ in the choice of token-mixing strategy.

Transformer Stream The Transformer stream is designed to capture global dependencies using self-attention. In the spatial stage, joints within the same frame are treated as tokens, enabling the model to learn intra-frame relationships through Spatial Multi-Head Self-Attention (**S-MHSA**). The output of the spatial stage is then reshaped to form per-joint temporal sequences, which are processed by Temporal Multi-Head Self-Attention (**T-MHSA**) to model motion dynamics across frames.

Both attention modules follow the standard multi-head self-attention formulation, where queries, keys, and values are obtained through linear projections of the input features.

$$\begin{aligned} \text{MHSA}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^{(O)} \\ \text{head}_i &= \text{Softmax}\left(\frac{Q^{(i)}(K^{(i)})^T}{\sqrt{d_k}}\right)V^{(i)} \end{aligned} \tag{5.5}$$

where $W^{(O)}$ is a projection parameter matrix, h is the number of parallel attention heads, and d_k is the feature dimension of K. For computing the query matrix Q, the key matrix K, and the value matrix V, the following linear projections are used:

$$Q^i = FW^{(Q,i)}, \quad K^i = FW^{(K,i)}, \quad V^i = FW^{(V,i)} \tag{5.6}$$

where $W^{(Q,i)}$, $W^{(K,i)}$, $W^{(V,i)}$ are projection matrices.

Each MetaFormer stage is followed by residual connections, layer normalization and a feed-forward MLP ensuring stable training and efficient feature propagation.

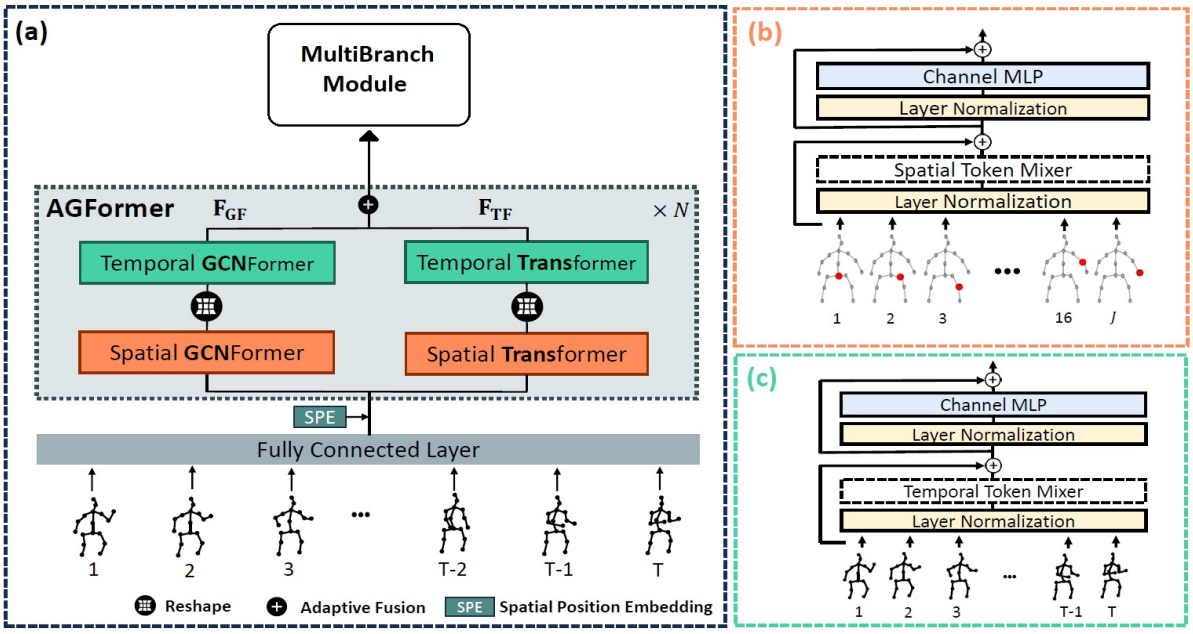

Figure 5.5: (a) AGFormer is an architecture with N dual-stream spatiotemporal blocks, using GCNFormers in one stream and Transformers in the other. (b) Spatial MetaFormer. A single human joint is represented by each input token. (c) Temporal MetaFormer. Pose sequence frames serve as input tokens. Image adapted from [45].

GCNFormer Stream In parallel to the Transformer stream, the GCNFormer stream focuses on modeling local spatial and temporal relationships using graph convolutions. While Transformers are effective at capturing global interactions, graph-based operations provide strong inductive biases that reflect the underlying skeletal structure.

The GCN module [37] aggregates information from neighboring nodes using a normalized adjacency matrix, incorporating both identity and neighborhood connections.

$$\mathrm{GCN}(F^{(i)}) = \sigma(F^{(i)} + \mathrm{Norm}(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}F^{(i)}W_1 + F^{(i)}W_2)) \tag{5.7}$$

Where $\tilde{A} = A + I_N$ represents the adjacency matrix with self-connections added, I_N stands for the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{jj}$ is defined as the sum of elements along the diagonal of $\tilde{A}$, W_1 and W_2 denote trainable weight matrices specific to each layer and $\sigma(\cdot)$ is the activation function.
The graph convolution operation is followed by residual connections, layer normalization and an MLP mirroring the structure of the Transformer-based MetaFormer.

The spatial and temporal GCNFormers differ in their graph construction. For spatial modeling, the adjacency matrix encodes the human skeletal topology. For temporal modeling, the graph structure is dynamically constructed by computing feature similarities between the same joint across different time frames, with edges connecting the K most similar temporal neighbors.

$$\mathrm{Sim}(F_T^{(t_i)}, F_T^{(t_j)}) = (F_T^{(t_i)})^T F_T^{(t_j)} \tag{5.8}$$

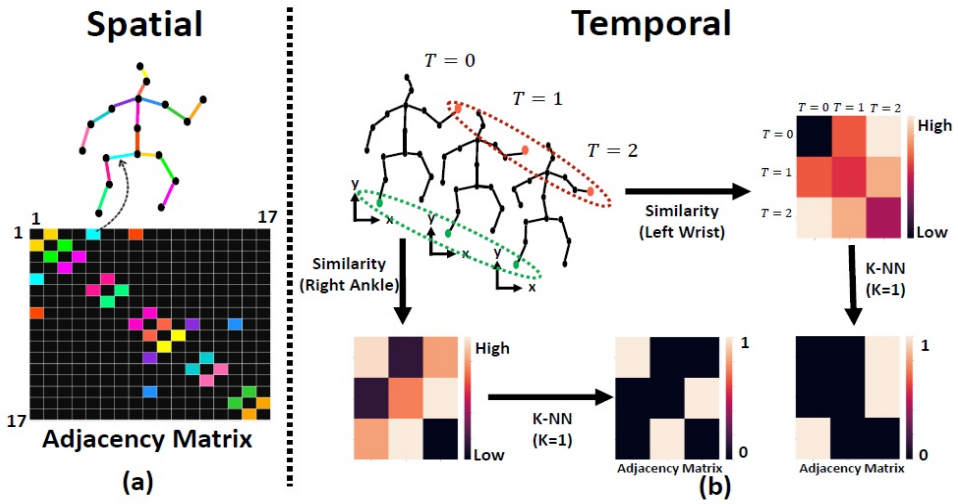

Figure 5.6: Topology of the GCNFormer module. (a) The fundamental topology of the Spatial GCNFormer is the human skeleton. (b) The Temporal GCNFormer use K-NN to identify related edges by taking into account each joint's maximum similarity throughout the course of the full time frame. Each row is linked to K columns following K-NN. Image taken from [45].

Adaptive Fusion

To combine the complementary representations produced by the Transformer and GCNFormer streams, AGFormer employs an adaptive fusion mechanism. At each depth i, the features extracted by the two streams are combined using learnable, data-dependent

weights:

$$F^{(i)} = \alpha_{TF}^{(i)} \circ F_{TF}^{(i-1)} + \alpha_{GF}^{(i)} \circ F_{GF}^{(i-1)}, \tag{5.9}$$

where $F_{TF}^{(i-1)}$ and $F_{GF}^{(i-1)}$ denote the Transformer and GCNFormer features, respectively, and $\circ$ represents element-wise multiplication.

The fusion weights are obtained via a softmax operation applied to a linear transformation of the concatenated features:

$$\alpha_{TF}^{(i)}, \alpha_{GF}^{(i)} = \text{Softmax}\big(W \cdot \text{Concat}(F_{TF}^{(i-1)}, F_{GF}^{(i-1)})\big), \tag{5.10}$$

allowing the network to dynamically balance local and global information depending on the motion context.

Overall, this spatial-temporal module enables Brancher to learn expressive and coherent motion representations, forming the foundation for the subsequent multi-branch optimization strategy.

5.3.2. Multi-Branch Architecture

Estimating 3D human pose from monocular drone footage presents unique challenges, as the subject is often partially visible, affected by strong perspective distortions, and captured from oblique and varying viewpoints. In such conditions, directly regressing 3D joint coordinates with a single prediction head can be suboptimal, as different aspects of human motion are governed by distinct constraints and sources of uncertainty.

To address these challenges, the proposed network adopts a multi-branch architecture inspired by SelfPose [66]. The core idea is to decompose the 3D pose estimation problem into multiple complementary objectives, each optimized by a dedicated branch. By explicitly modeling different factors of human motion, the network can learn more robust and physically consistent representations, particularly in unconstrained, in-the-wild drone scenarios.

All branches share a common spatio-temporal feature representation extracted by the spatial-temporal module. On top of this shared representation, each branch consists of an MLP-based head that focuses on a specific aspect of the pose estimation task and is trained using a branch-specific loss function. This design enables the network to jointly optimize multiple objectives while maintaining a clear separation of roles among the different branches.

In the proposed architecture, three branches are employed: a *Pose* branch for direct 3D

joint regression and motion consistency, a *Rotation* branch for modeling body orientation, and an *Uncertainty* branch for estimating confidence-aware pose representations.
Each branch is described in detail in the following subsections.

Pose Branch

The Pose branch is responsible for regressing the 3D joint positions of the human subject. It predicts root-relative 3D keypoints from the shared spatio-temporal features produced by the previous module. This branch consists of an MLP-based pose head followed by a *tanh* activation function, which introduces non-linearity.
The output is $\hat{y} \in \mathbb{R}^{T \times J \times 3}$, where 3 corresponds to the (x, y, z) coordinates of each joint.

Unlike a purely supervised formulation, the Pose branch is trained using a combination of supervised and self-supervised objectives. The supervised loss, namely the Mean Per Joint Position Error (**MPJPE**), provides direct geometric supervision when ground-truth 3D annotations are available and represents the most informative training signal. In contrast, the self-supervised losses enforce motion consistency constraints that are independent of 3D ground truth and can therefore be applied to unlabeled videos.

Specifically, three self-supervised losses are employed: a **bone-length consistency** loss, a **velocity smoothness** loss, and an **acceleration smoothness** loss. Together, these losses regularize the predicted motion and encourage physically plausible and temporally coherent 3D pose sequences.
Here a legend of the symbols used in the equations followed by the definition of each loss function:

- T denotes the number of frames in the input sequence;
- J denotes the number of body joints;
- O denotes the number of bones defined by the skeletal topology;
- $\hat{\mathbf{y}}_{t,j} \in \mathbb{R}^3$ represents the predicted 3D position of joint j at frame t;
- $\mathbf{y}_{t,j} \in \mathbb{R}^3$ represents the corresponding ground-truth 3D joint position;
- $\ell_o^{(t)}$ denotes the length of bone o at frame t;

$$\mathcal{L}_{MPJPE} = \frac{1}{TJ} \sum_{t=1}^{T} \sum_{j=1}^{J} \|\hat{\mathbf{y}}_{t,j} - \mathbf{y}_{t,j}\| \tag{5.11}$$

The Mean Per Joint Position Error (MPJPE) measures the average Euclidean distance

between the predicted and ground-truth 3D joint positions. It provides direct geometric supervision and represents the most informative loss when paired 3D annotations are available. Due to its reliance on ground-truth data, this loss is applied only during the supervised training phase.

$$\mathcal{L}_{bone} = \frac{1}{TO} \sum_{t=1}^{T} \sum_{o=1}^{O} \mathrm{Var}_t\left(\ell_o^{(t)}\right) \tag{5.12}$$

The bone length loss enforces temporal consistency of the skeletal structure by penalizing variations in bone lengths across frames. Since the length of human bones is invariant over time, this loss encourages physically plausible pose predictions without requiring 3D ground-truth annotations. It is therefore well-suited for self-supervised learning on unlabeled video sequences.

$$\mathcal{L}_{vel} = \frac{1}{(T-2)J} \sum_{t=2}^{T-1} \sum_{j=1}^{J} \left\| (\hat{\mathbf{y}}_{t+1,j} - \hat{\mathbf{y}}_{t,j}) - (\hat{\mathbf{y}}_{t,j} - \hat{\mathbf{y}}_{t-1,j}) \right\| \tag{5.13}$$

The velocity loss penalizes abrupt changes in joint velocities between consecutive frames. By enforcing temporal smoothness in the first-order motion dynamics, this loss helps reduce jitter and unrealistic motion artifacts in the predicted 3D pose sequence.

$$\mathcal{L}_{acc} = \frac{1}{(T-3)J} \sum_{t=3}^{T-1} \sum_{j=1}^{J} \left\| (\hat{\mathbf{y}}_{t+1,j} - 2\hat{\mathbf{y}}_{t,j} + \hat{\mathbf{y}}_{t-1,j}) - (\hat{\mathbf{y}}_{t,j} - 2\hat{\mathbf{y}}_{t-1,j} + \hat{\mathbf{y}}_{t-2,j}) \right\| \tag{5.14}$$

The acceleration loss further regularizes the motion by penalizing variations in joint accelerations over time. This second-order temporal constraint encourages smoother and more coherent motion trajectories, which is particularly important in dynamic sports actions captured from drone viewpoints.

Rotation Branch

The Rotation Branch is responsible for estimating the joint rotations of the human body. Modeling rotations is a non-trivial task due to the geometric structure of the rotation space, which does not lie in an Euclidean vector space.

Rotations in three dimensions belong to the Special Orthogonal group SO(3), defined as the set of all 3×3 orthogonal matrices with determinant equal to one:

$$\mathrm{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I},\ \det(\mathbf{R}) = 1\}. \tag{5.15}$$

This space is a smooth manifold rather than a vector space, which makes the direct regression of rotations particularly challenging for neural networks.

A common parameterization of rotations is given by Euler angles. However, Euler angle representations suffer from discontinuities and ambiguities, most notably the gimbal lock phenomenon. Small changes in the rotation matrix can correspond to large and discontinuous changes in the Euler angle parameters, violating the continuity assumptions required for stable learning in neural networks. As a result, Euler angles are not well suited for rotation regression tasks.

Unit quaternions provide an alternative and more compact representation of rotations. Nevertheless, quaternion representations are also not continuous, since each rotation corresponds to two antipodal points on the unit hypersphere, i.e. $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. This double-cover property introduces discontinuities in the representation space, which can negatively affect optimization during training.

To overcome these limitations, the Rotation Branch adopts the **6D rotation representation** proposed in [83]. Instead of regressing a minimal parameterization, the network predicts a 6D vector that corresponds to the first two columns of a rotation matrix.

Given a predicted vector $\mathbf{r} = [\mathbf{a}, \mathbf{b}] \in \mathbb{R}^6$, with $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, a valid rotation matrix is constructed using a Gram–Schmidt orthonormalization process:

$$\mathbf{r}_1 = \frac{\mathbf{a}}{\|\mathbf{a}\|}, \tag{5.16}$$

$$\mathbf{r}_2 = \frac{\mathbf{b} - (\mathbf{r}_1^\top \mathbf{b})\mathbf{r}_1}{\|\mathbf{b} - (\mathbf{r}_1^\top \mathbf{b})\mathbf{r}_1\|}, \tag{5.17}$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2. \tag{5.18}$$

The resulting matrix $\hat{\mathbf{R}} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ belongs to SO(3) by construction.

This representation is continuous and free from singularities, making it particularly suitable for neural network regression. Empirically, the 6D representation has been shown to improve convergence stability and prediction accuracy compared to Euler angles and quaternions, especially in learning-based settings. For these reasons, the Rotation Branch adopts the 6D rotation representation throughout this work.

The branch is implemented as an MLP head that takes as input the encoded features produced by the spatial-temporal attention module. Given a temporal window of length T, the Rotation Branch outputs a tensor of rotation matrices $\hat{\mathbf{R}} \in \mathbb{R}^{T \times O \times 3 \times 3}$, where O denotes the number of considered bones and each $\hat{\mathbf{R}}_{t,o} \in \mathrm{SO}(3)$ represents the rotation of

the o-th bone at time step t.

The bones are defined as the vectors connecting pairs of anatomically adjacent joints in the human skeleton. The number of bones considered in this work is fixed to $O = 16$.

Rotation Targets Construction In order to train the Rotation Branch, a suitable supervision signal is required. Since explicit ground-truth bone rotations are generally unavailable for in-the-wild drone videos, I derive rotation targets directly from the 3D pose predictions produced by the Pose Branch.

Let $\mathbf{P} \in \mathbb{R}^{T\times J\times 3}$ denote the predicted 3D joint positions in root-relative coordinates, where T the temporal length, and J the number of joints. From $\mathbf{P}$, a set of bone vectors is extracted according to a predefined skeletal topology based on the H36M keypoint convention. Specifically, I define a function

$$\mathcal{B}(\mathbf{P}) : \mathbb{R}^{T\times J\times 3} \to \mathbb{R}^{T\times O\times 3}, \tag{5.19}$$

which maps joint positions to bone vectors, where $O = 16$ is the number of considered bones.

Each bone vector is defined as the displacement between two anatomically adjacent joints:

$$\mathbf{b}_{t,o} = \mathbf{p}_{t,j_2(o)} - \mathbf{p}_{t,j_1(o)}, \tag{5.20}$$

where $(j_1(o), j_2(o))$ denotes the joint pair associated with bone o.

To define a canonical reference orientation, I introduce a T-pose skeleton with unit-length bones. Let

$$\mathbf{B}^{\mathrm{T}} \in \mathbb{R}^{O\times 3} \tag{5.21}$$

denote the set of reference bone directions, manually defined according to human anatomy under a fixed coordinate system, where the vertical axis corresponds to the Y direction and the horizontal axes span the Z-X plane.

All reference bone vectors are normalized to unit length:

$$\|\mathbf{b}_o^{\mathrm{T}}\| = 1, \quad \forall o \in \{1, \ldots, O\}. \tag{5.22}$$

It is important to emphasize that the canonical T-pose skeleton is not used in terms of absolute joint positions. Instead, I consider only the directional information of each bone. In other words, the reference vectors $\mathbf{B}^{\mathrm{T}}$ encode canonical orientations, not spatial loca-

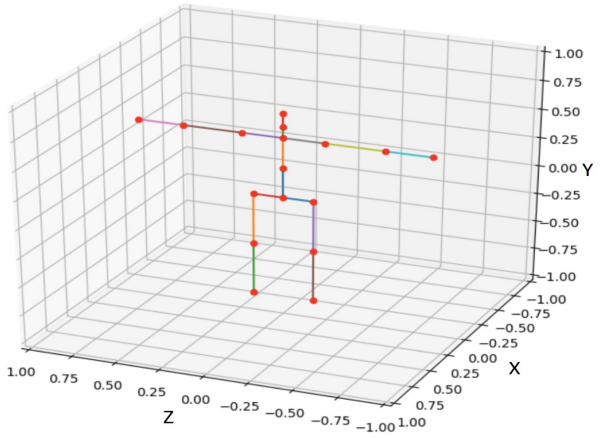

Figure 5.7: Canonical T-pose skeleton used as reference for rotation target extraction. Image taken from [72].

tions.

From an implementation perspective, each reference bone is treated as a free vector anchored at the origin of the coordinate system. Therefore, the canonical skeleton can be interpreted as a star-shaped configuration centered at the origin, where all unit-length bone direction vectors emanate from the same point. The relative spatial arrangement of joints in the T-pose is not explicitly modeled; only the orientation of each bone with respect to the fixed global axes is retained.

This formulation allows the Rotation Branch to focus exclusively on learning the rotational transformation that aligns a canonical bone direction with the corresponding predicted bone vector, independently of bone length or absolute position.

Given a bone vector from the predicted joints $\mathbf{b}_{t,o}$ and its corresponding reference direction $\mathbf{b}_o^{\mathrm{T}}$, both normalized to unit length, I compute the rotation matrix that aligns the reference bone to the predicted one, based on the work of Wu et al. [72].

To compute the relative rotation between the canonical bone direction $\mathbf{a}_o$ and the predicted bone direction $\mathbf{b}_{t,o}$, I rely on the axis-angle formulation and Rodrigues' rotation formula. Given two unit vectors

$$\mathbf{a}_o = \frac{\mathbf{b}_o^{\mathrm{T}}}{\|\mathbf{b}_o^{\mathrm{T}}\|} \in \mathbb{R}^3, \qquad \mathbf{b}_{t,o} = \frac{\mathbf{b}_{t,o}}{\|\mathbf{b}_{t,o}\|} \in \mathbb{R}^3 \tag{5.23}$$

the goal is to find the rotation matrix $\mathbf{R}_{t,o} \in SO(3)$ such that:

$$\mathbf{R}_{t,o}\mathbf{a}_o = \mathbf{b}_{t,o} \tag{5.24}$$

The relative rotation can be described by:

- a rotation axis:

$$\mathbf{k_{t,o}} = \frac{\mathbf{a_o} \times \mathbf{b_{t,o}}}{\|\mathbf{a_o} \times \mathbf{b_{t,o}}\|} \tag{5.25}$$

- and a rotation angle:

$$\theta_{t,o} = \arccos(\mathbf{a_o}^T \mathbf{b_{t,o}}) \tag{5.26}$$

The axis-angle representation of the relative rotation is obtained as

$$\mathbf{v}_{t,o} = \mathbf{a}_o \times \mathbf{b}_{t,o}, \tag{5.27}$$

$$c_{t,o} = \mathbf{a}_o^\top \mathbf{b}_{t,o} = \|\mathbf{a}\|\|\mathbf{b}\| \cos(\theta) = \cos(\theta), \tag{5.28}$$

$$s_{t,o} = \|\mathbf{v}_{t,o}\| = \|\mathbf{a}\|\|\mathbf{b}\| \sin(\theta) = \sin(\theta). \tag{5.29}$$

Given a unit axis $\mathbf{k}$ and angle θ, Rodrigues' rotation formula states that the rotation matrix is

$$\mathbf{R} = \mathbf{I} + \sin(\theta)\mathbf{K} + (1 - \cos(\theta))\mathbf{K}^2, \tag{5.30}$$

where $\mathbf{K}$ is the skew-symmetric matrix associated with $\mathbf{k}$.
Let $\mathbf{K}_{t,o}$ be the skew-symmetric matrix constructed directly from $\mathbf{v}_{t,o}$. Using the identities:

$$\mathbf{K}_{t,o} = s_{t,o}\,\mathbf{K}, \qquad \mathbf{K}_{t,o}^2 = s_{t,o}^2\,\mathbf{K}^2. \tag{5.31}$$

and the trigonometric relation

$$\frac{1 - \cos\theta}{\sin^2\theta} = \frac{1}{1 + \cos\theta}. \tag{5.32}$$

Rodrigues' formula can be rewritten as

$$\mathbf{R}_{t,o} = \mathbf{I} + \mathbf{K}_{t,o} + \mathbf{K}_{t,o}^2 \cdot \frac{1}{1 + c_{t,o}}, \tag{5.33}$$

where $\mathbf{I}$ is the 3×3 identity matrix and $\mathbf{K}_{t,o}$ is the skew-symmetric matrix associated with $\mathbf{v}_{t,o}$:

$$\mathbf{K}_{t,o} = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}. \tag{5.34}$$

This formulation is numerically stable and naturally handles the case in which the two

vectors are already aligned.

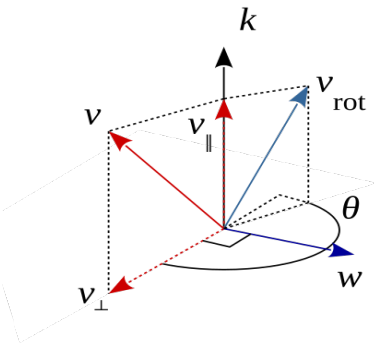

Figure 5.8: Rodrigues' rotation formula rotates v by an angle θ around vector k by decomposing it into its components parallel and perpendicular to k, and rotating only the perpendicular component [71].

The complete rotation target extraction process can therefore be summarized as a function

$$\mathbf{r}(\mathbf{P}) = \{\mathbf{R}_{t,o}\}_{t=1,o=1}^{T,O}, \tag{5.35}$$

which maps the predicted 3D joint positions to a sequence of bone rotations relative to a canonical T-pose.

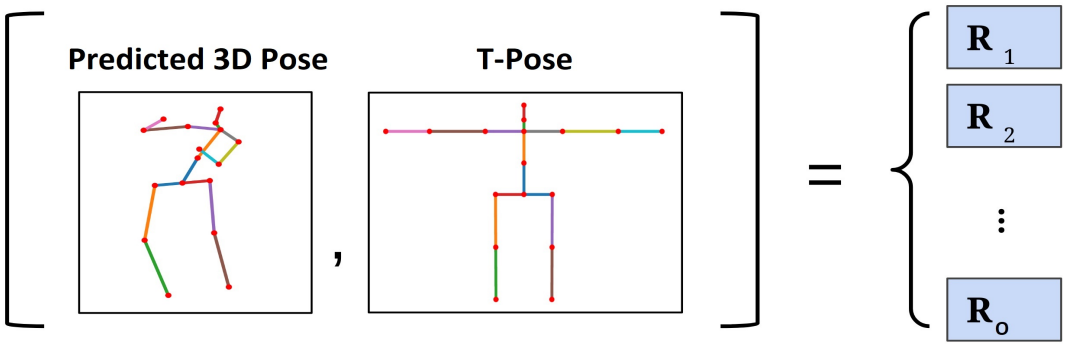

Figure 5.9: Extraction of rotation targets from predicted 3D joint positions using a canonical T-pose skeleton as reference. Image adapted from [72].

The resulting tensor $\mathbf{r}(\mathbf{P}) \in \mathbb{R}^{T\times O\times 3\times 3}$ is used as the supervision signal for the Rotation Branch, enabling the network to learn a structured and temporally coherent representation of human motion even in the absence of explicit rotation annotations.

Rotation Consistency Loss At this stage, two independent estimates of bone rotations are available. The first one is produced directly by the Rotation Branch, which predicts a sequence of rotation matrices $\hat{\mathbf{R}} \in \mathbb{R}^{T\times O\times 3\times 3}$. The second estimate is obtained indirectly from the Pose Branch by extracting bone vectors from the predicted 3D joint positions and computing their relative rotations with respect to the canonical T-pose, as described in the previous paragraph. This results in a set of target rotations $\mathbf{R} = \mathbf{r}(\mathbf{P}) \in \mathbb{R}^{T\times O\times 3\times 3}$.

To enforce consistency between these two representations, I introduce a rotation loss based on the geodesic distance on the SO(3) manifold. Unlike element-wise losses, the geodesic distance provides a geometrically meaningful measure of discrepancy between two rotations, corresponding to the minimal angle required to rotate from one orientation to the other.

$$\mathcal{L}_{geo} = \frac{1}{TO}\sum_{t=1}^{T}\sum_{o=1}^{O}\arccos\left(\frac{\mathrm{tr}(\hat{\mathbf{R}_{t,o}}\mathbf{R}_{t,o}^{T}) - 1}{2}\right) \tag{5.36}$$

This loss encourages the Rotation Branch to predict rotations that are consistent with the geometric structure implied by the 3D pose predictions, effectively coupling the Pose and Rotation Branches. Since the target rotations $\mathbf{R}$ are derived from the Pose Branch itself, the geodesic loss can be applied in a self-supervised manner and does not require additional annotations.

In addition to the geodesic rotation loss, I introduce an auxiliary constraint directly in the vector space of bone directions. While the geodesic loss enforces consistency between two rotation matrices on the SO(3) manifold, it does not explicitly constrain the rotated canonical bones to align with the predicted skeletal structure. To address this, I apply the predicted rotations to the canonical T-pose bone directions and directly compare the resulting vectors with the normalized bone vectors extracted from the Pose Branch. Let:

$$\mathbf{b}_{t,o} \in \mathbb{R}^3 \tag{5.37}$$

denote the bone vector extracted from the predicted 3D joints and normalized to unit length:

$$\tilde{\mathbf{b}}_{t,o} = \frac{\mathbf{b}_{t,o}}{\|\mathbf{b}_{t,o}\|} \tag{5.38}$$

Let $\mathbf{b}^{\mathrm{T}}o$ be the canonical unit bone direction of the T-pose. The Rotation Branch predicts $\hat{\mathbf{R}}_{t,o}$, which is applied to the canonical direction:

$$\hat{\mathbf{b}}_{t,o} = \hat{\mathbf{R}}_{t,o}\mathbf{b}_{o}^{\mathrm{T}} \tag{5.39}$$

I then define the bone direction loss as an L_1 discrepancy between the rotated canonical bone and the normalized predicted bone:

$$\mathcal{L}_{vec} = \frac{1}{TO} \sum_{t=1}^{T} \sum_{o=1}^{O} \left\| \hat{\mathbf{b}}_{t,o} - \tilde{\mathbf{b}}_{t,o} \right\|_1 \tag{5.40}$$

This loss enforces that the rotation predicted by the Rotation Branch, when applied to the canonical bone direction, reconstructs the actual bone direction inferred from the 3D pose. Unlike the geodesic loss, which operates purely in rotation space, this formulation directly constrains the induced geometric effect of the rotation.

By enforcing agreement between joint-based and rotation-based representations of motion, the proposed formulation promotes physically plausible and temporally coherent human motion, which is particularly beneficial in challenging scenarios such as drone-based, in-the-wild recordings.

Uncertainty Branch

Drone-based recordings introduce significant sources of noise, including motion blur, partial occlusions, rapid scale changes, and viewpoint variability. These factors affect joint localization reliability in a non-uniform manner: some joints remain clearly visible, while others may be poorly localized or temporarily ambiguous. For this reason, I explicitly model joint-wise prediction uncertainty through a dedicated Uncertainty Branch.

Instead of treating heatmap regression as a purely deterministic problem, I adopt a probabilistic formulation inspired by Kendall et al. [27]. However, differently from a direct heatmap-to-heatmap regression, I convert predicted heatmaps into continuous joint coordinates using a differentiable spatial *soft-argmax* operator, and apply supervision in the coordinate space.

The *soft-argmax* operation first applies a spatial *softmax* over the heatmap, transforming raw activations into a normalized probability distribution over pixel locations. Given a predicted heatmap $H_j(u, v)$ for joint j, the spatial *softmax* is defined as:

$$P_j(u, v) = \frac{\exp\left(\frac{H_j(u,v)}{\tau}\right)}{\sum_{u',v'} \exp\left(\frac{H_j(u',v')}{\tau}\right)} \tag{5.41}$$

where τ is a temperature parameter controlling the sharpness of the distribution. Lower values of τ make the distribution more peaked, approximating a *hard-argmax*.

The predicted joint coordinates are then obtained as the expected value of this distribution:

$$\mathbf{x}_j = \sum_{u,v} P_j(u,v) \begin{bmatrix} u \\ v \end{bmatrix} \tag{5.42}$$

Unlike the standard *argmax* operator, which is non-differentiable, this formulation allows gradients to propagate through the spatial localization process, enabling end-to-end training.

This design choice is primarily motivated by technical and representational constraints of the pipeline. The pseudo-label heatmaps produced by ViTPose [74] are generated on cropped human bounding boxes obtained from a detector. Although these heatmaps can be geometrically projected back to full-frame resolution, their Gaussian peaks become extremely small relative to the entire image. As a result, the spatial signal becomes highly diluted when represented at frame scale, making heatmap-to-heatmap supervision ineffective.

Predicting full-frame heatmaps in the Uncertainty Branch was also infeasible due to GPU memory limitations. I additionally experimented with a resize-based strategy, where ViTPose heatmaps were first expanded to frame resolution and then downsampled to a standard heatmap size before supervision. However, this transformation significantly degraded spatial precision, preventing the network from learning meaningful localization cues.

Using local (*bbox-level*) heatmaps in the Uncertainty Branch was not a viable alternative either. All branches share a common latent representation, and mixing coordinate systems (*frame-level* for some branches and *bbox-level* for others) would introduce spatial inconsistencies within the shared embedding space. It is worth recalling that cropping is essential in aerial footage, as full-frame inference severely degrades keypoint detection performance due to the small scale of the human subject.

For these reasons, supervising directly in the coordinate space via *soft-argmax* provides a numerically stable, memory-efficient, and geometrically consistent solution, while preserving end-to-end differentiability. In this setting, supervision does not come from manually annotated ground-truth labels, but from pseudo-labels extracted during preprocessing (Section 5.1) using ViTPose. Although this pose extractor provides high-quality 2D keypoints, its predictions cannot be considered noise-free, especially in challenging drone footage. Therefore, I explicitly model aleatoric (data-dependent) uncertainty to account for the inherent noise in these pseudo-labels.

For each joint j, the branch predicts:

- a heatmap μ_j, from which 2D coordinates $\mathbf{x}_j$ are obtained via *soft-argmax*;
- a log-variance term $s_j = \log \sigma_j^2$, modeling heteroscedastic uncertainty.

After applying *soft-argmax*, the predicted coordinates $\mathbf{x}_j$ are interpreted as the mean of a Gaussian likelihood with variance σ_j^2:

$$p(\tilde{\mathbf{x}}_j \mid \mathbf{x}_j, \sigma_j^2) = \mathcal{N}(\mathbf{x}_j, \sigma_j^2 \mathbf{I}) \tag{5.43}$$

Minimizing the negative log-likelihood (ignoring constant terms) leads to the following heteroscedastic regression loss:

$$\mathcal{L}_{hm} = \frac{1}{2} \sum_{j=1}^{J} c_j \left(e^{-s_j} \left\| \mathbf{x}_j - \tilde{\mathbf{x}}_j \right\|^2 + s_j \right) \tag{5.44}$$

where $\tilde{\mathbf{x}}_j$ denotes the pseudo ground-truth 2D coordinates extracted from ViTPose, and c_j is a confidence score associated with each joint prediction. The precision term e^{-s_j} adaptively scales the squared residual, allowing the model to attenuate unreliable supervisory signals. The regularization term s_j prevents the network from trivially inflating uncertainty.

This formulation is particularly appropriate in a self-supervised pipeline, where supervision originates from another model rather than from human annotations. By combining *soft-argmax* coordinate regression with heteroscedastic uncertainty modeling, the Uncertainty Branch improves robustness to noisy aerial data and stabilizes training in the presence of imperfect pseudo-labels.

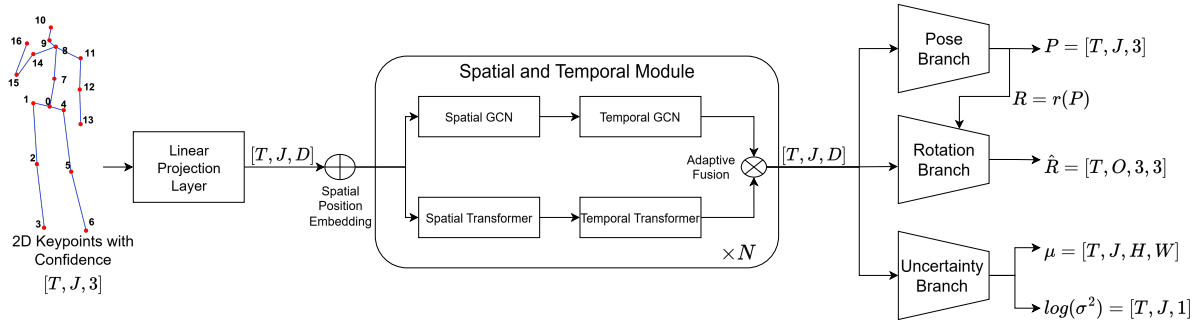

Figure 5.10: Brancher model architecture.

5.4. Loss Weighting Strategy

Training the proposed multi-branch architecture requires combining multiple objectives targeting complementary aspects of human motion, including 3D pose accuracy, temporal smoothness, bone consistency, rotational coherence, and uncertainty modeling. Since these losses operate on different physical scales and exhibit different convergence dynamics, a principled weighting strategy is required to ensure stable optimization.

Rather than adopting a single weighting scheme for all training phases, I employ two different strategies depending on the supervision regime: a structured curriculum-based weighting during supervised pretraining, and a homoscedastic uncertainty-based formulation during purely self-supervised training.

5.4.1. Supervised Pretraining

During supervised pretraining, the objective is to stabilize the learning of 3D pose reconstruction before introducing strong self-supervised constraints. In this phase, each loss term $\mathcal{L}_i$ is first normalized by a fixed scale factor a_i, compensating for differences in magnitude and physical units.

The normalized losses are then combined through adaptive importance weights derived from a learnable *softmax* parametrization, together with a temporal gating mechanism and a dynamic floor term:

$$\mathcal{L}_{\text{pretrain}} = \sum_i \Big[\text{softmax}(\alpha_i)(1 - f(t)) + \delta_{i,0} f(t)\Big]\, g_i(t)\, \frac{\mathcal{L}_i}{a_i} \tag{5.45}$$

where:

- α_i are learnable parameters whose *softmax* defines the relative importance of each task;
- a_i are fixed normalization constants;
- $f(t)$ is a dynamic floor that enforces dominance of the primary 3D pose loss (MPJPE) during early epochs;
- $\delta_{i,0}$ is the Kronecker delta ensuring that the floor is applied only to the main pose task;
- $g_i(t)$ is a temporal gate that progressively activates secondary objectives.

The gating mechanism follows a curriculum-learning rationale: auxiliary constraints such

as rotational consistency and vector alignment are gradually introduced only after the core 3D reconstruction stabilizes. This prevents unstable gradients in the early stages of training and encourages the model to first learn a reliable geometric representation before refining higher-order motion details.

5.4.2. Pure Self-Supervised Training

In the purely self-supervised regime, manual curriculum design becomes less reliable due to the indirect nature of the supervision signals. For this reason, I adopt the homoscedastic uncertainty-based formulation proposed by Kendall et al. [28].

Under the assumption that each task follows a Gaussian likelihood with variance σ_i^2 and $s_i = \log \sigma_i^2$ is the log variance, the negative log-likelihood leads to:

$$\mathcal{L}_{\text{self}} = \sum_i \frac{1}{2} \left(e^{-s_i} \mathcal{L}_i + s_i \right) \tag{5.46}$$

Here, $e^{-s_i} = \frac{1}{\sigma_i^2}$ acts as an adaptive precision term, dynamically scaling each task loss according to its estimated noise level, while the regularization term s_i prevents degenerate solutions in which uncertainty grows without bound.

This formulation allows the network to automatically balance heterogeneous objectives without manual tuning. Tasks that exhibit higher noise or instability are assigned larger uncertainty, reducing their impact on the total objective, whereas more reliable signals receive higher weight.

5.4.3. Comparison of the Two Weighting Strategies

The two weighting strategies reflect different optimization requirements in the supervised and self-supervised phases.

During supervised pretraining, it is crucial that the MPJPE loss drives the early stages of optimization. A stable and geometrically consistent 3D pose representation must first be established before introducing auxiliary objectives that depend on it, such as rotational consistency or vector alignment. Empirically, activating these secondary losses too early led to unstable gradients and degraded convergence. For this reason, a curriculum-based scheme with a dominant MPJPE floor and delayed activation of dependent tasks proved more effective, ensuring that the core 3D reconstruction task guides the learning process.

In contrast, during purely self-supervised training, the supervision signals are inherently

indirect and noisier. In this setting, manually enforcing a hierarchy among tasks becomes less reliable. Instead, the homoscedastic uncertainty formulation allows the network to autonomously estimate the relative difficulty and reliability of each objective. By learning task-dependent precision parameters, the model dynamically balances the losses based on their observed stability during training, without imposing a predefined curriculum.

In summary, supervised pretraining benefits from structured, curriculum-driven optimization centered on pose accuracy, whereas the self-supervised phase relies on adaptive uncertainty-based weighting to handle heterogeneous and noisy supervisory signals.

6 | Implementation Details

This chapter presents the implementation details of the proposed framework. The goal of this section is to provide a clear and reproducible description of all architectural choices, training strategies, and inference procedures adopted in my experiments.

I start by describing the technical details of the **preprocessing** stage, including the specific model employed for human detection and 2D pose estimation.

Next, I introduce the **Baseline** model, outlining its architecture, optimization strategy, data processing pipeline, and inference protocol.

Finally, I present the proposed **Brancher** network, highlighting its architectural modifications, integration within the Baseline framework, and the key implementation choices that distinguish it from the reference model.

All experiments are implemented in Python using the PyTorch library, and all relevant hyperparameters and settings are reported to ensure full reproducibility.

6.1. Preprocessing Implementation

For human detection, I employed the **YOLOv12-small** model from the Ultralytics library, which provides fast and accurate bounding box predictions for individual subjects in aerial footage.

For 2D pose estimation, I used **ViTPose-Huge** from the MMPose framework, which is an open-source library designed for pose estimation that offers modular implementations of state-of-the-art models, enabling flexible integration with custom pipelines.

6.2. Baseline Implementation

The Baseline model serves as the reference architecture upon which the proposed method is built. It takes as input a temporal sequence of 2D human poses represented as a tensor of shape $[B, T, J, 3]$, where B denotes the batch size (set to 64), T is the temporal

Table 6.1: Baseline model architecture. The network takes as input a temporal sequence of 2D poses and outputs the corresponding 3D pose representation.

Layer	**Input Shape**	**Output Shape**	**Params**
	$(64, 81, 17, 3)$	$(64, 81, 17, 3)$	
Linear (Input Embedding)	$(64, 81, 51)$	$(64, 81, 64)$	3,328
Positional Encoding	$(64, 81, 64)$	$(64, 81, 64)$	–
Transformer Encoder (3 layers, 8 heads)	$(64, 81, 64)$	$(64, 81, 64)$	843,456
Linear	$(64, 81, 64)$	$(64, 81, 64)$	4,160
Tanh	$(64, 81, 64)$	$(64, 81, 64)$	–
Linear (Pose Head)	$(64, 81, 64)$	$(64, 81, 51)$	3,315
Total			**854,259**

window length (81 frames), J is the number of joints (set to 17), and the last dimension corresponds to the (x, y, c) coordinates of each joint, with c indicating the confidence score.

The model predicts the corresponding 3D pose sequence, producing an output tensor of shape $[B, T, J, 3]$, where the final dimension represents the (x, y, z) coordinates of each joint in 3D space.

6.2.1. Architecture

The Baseline architecture is built upon a Transformer-based design aimed at modeling temporal dependencies across pose sequences. Each 2D pose in the sequence is first projected into a latent feature space of dimension 64 through a learnable embedding layer. This embedding step transforms the raw joint coordinates into a higher-dimensional representation suitable for temporal modeling.

The resulting sequence of embeddings is then processed by a stack of three Transformer encoder layers. Each encoder layer employs multi-head self-attention with eight attention heads, enabling the model to capture both short-term dynamics and long-range temporal correlations within the sequence.

Finally, the output features produced by the Transformer are mapped to the target 3D pose representation through a linear projection layer followed by a *tanh* activation function, ensuring bounded output predictions.

6.2.2. Training Procedure

The Baseline model is trained in a fully supervised manner using the Mean Per Joint Position Error (MPJPE) as the optimization objective. Since the model directly regresses 3D joint coordinates, it requires ground-truth 3D annotations during training.

Training is performed on the AthletePose3D dataset [75], which was selected because it contains high-quality 3D annotations and captures fast, dynamic sports movements. This makes it particularly suitable for learning complex temporal motion patterns typical of athletic actions.

It was not possible to train the Baseline model on the drone-captured dataset, as it does not provide ground-truth 3D pose annotations. Given the fully supervised nature of the Baseline architecture, the absence of 3D labels prevents direct optimization using MPJPE on that dataset.

The training process is carried out for a maximum of 100 epochs. Optimization is performed using the $AdamW$ optimizer with an initial learning rate of 5×10^{-4} and a weight decay of 0.01. The learning rate is scheduled with an exponential decay factor of 0.99 applied at the end of each epoch.

To stabilize the early stages of optimization, a warm-up phase of 2 epochs is introduced at the beginning of training. Early stopping is employed with a patience of 10 epochs, based on validation performance.

All hyperparameters were selected through a validation process conducted using the Optuna library.

6.2.3. Inference Pipeline

During inference, the system takes as input an RGB video sequence represented as a tensor of shape $[T, H, W, 3]$, where T denotes the number of frames and H and W are the spatial dimensions of each frame.

For each frame, a human detection stage is first performed using YOLOv12 [65] in order to localize the subject of interest. The detected bounding box corresponding to the person is then cropped and resized to a fixed spatial resolution of 256×192 pixels.

The cropped image is subsequently fed to a ViTPose [74] model to estimate the 2D human pose. This stage produces a set of 2D keypoints for each frame. The predicted keypoints are then normalized following the same preprocessing procedure described in Section 5.1, ensuring consistency between training and inference distributions.

The resulting normalized 2D pose sequence is provided as input to the Baseline network, which predicts the corresponding 3D pose sequence.

To further improve prediction robustness, a test-time augmentation (TTA) strategy is adopted. In addition to the original input sequence, a horizontally flipped version of the 2D poses is also fed to the model. The final 3D prediction is obtained by averaging the outputs produced from the original and flipped inputs.

6.3. Brancher Implementation

The proposed Brancher model extends the Baseline architecture by introducing a self-supervised learning framework specifically designed to optimize 3D human pose estimation from drone-captured sequences.

Similarly to the Baseline, the model takes as input a temporal sequence of 2D human poses represented as a tensor of shape $[B, T, J, 3]$, where B denotes the batch size (set to 32), T is the temporal window length (81 frames), J is the number of joints (17), and the last dimension corresponds to the (x, y, c) coordinates of each joint, with c indicating the confidence score.

The network predicts a 3D pose sequence with output shape $[B, T, J, 3]$, where the final dimension represents the (x, y, z) coordinates of each joint in 3D space.

Unlike the fully supervised Baseline, Brancher is trained in a self-supervised manner and is structured around three complementary branches, each designed to enforce geometric and temporal consistency constraints tailored to the drone acquisition setting. These branches jointly contribute to refining the predicted 3D pose without requiring ground-truth 3D annotations.

Apart from the reduced batch size, all other architectural components and temporal configurations remain consistent with the Baseline model, ensuring a controlled comparison between the supervised and self-supervised approaches.

6.3.1. Architecture

The Brancher architecture is built upon the MotionAGFormer-Small [45] backbone, which serves as the shared feature extractor for all branches. The network processes input tensors of shape $[B, T, J, 3]$ and first projects the 2D joint coordinates into a latent space of dimension $D = 64$ through a linear embedding layer.

The embedded sequence is then processed by a stack of $N = 26$ AGFormer blocks,

which model both spatial relationships among joints and temporal dependencies across frames. A final Layer Normalization layer produces a shared latent representation of shape $[B, T, J, D]$, which is fed into the three specialized branches.

Pose Branch. The Pose Branch is responsible for directly regressing the 3D joint coordinates. It is implemented as a Multi-Layer Perceptron (MLP) composed of two linear layers with an intermediate hidden dimension of 512 and a *tanh* activation function. The final output has shape $[B, T, J, 3]$ and represents the predicted (x, y, z) joint coordinates. The *tanh* activation constrains the predictions within a bounded range, stabilizing training in the self-supervised setting.

Rotation Branch. The Rotation Branch predicts bone-level rotations to enforce structural consistency of the articulated body. Instead of directly regressing rotation matrices, the network adopts the 6D continuous rotation representation for improved numerical stability. The branch first aggregates joint features and applies a linear layer followed by a *tanh* activation to produce 96 values per frame (corresponding to $O = 16$ bones $\times$ 6D representation). These are then converted into valid 3×3 rotation matrices, resulting in an output tensor of shape $[B, T, O, 3, 3]$. This branch encourages physically plausible skeletal configurations and consistent inter-joint orientations.

Uncertainty Branch. The Uncertainty Branch models joint-wise prediction confidence and spatial uncertainty. Starting from the shared latent representation, joint features are first processed by a two-layer MLP with *LeakyReLU* activations to increase representational capacity. The resulting features are reshaped and passed through a sequence of transposed convolution layers that progressively upsample the representation to generate a spatial heatmap for each joint. The final heatmap resolution is 64×48, and a *sigmoid* activation ensures values are bounded between 0 and 1. These heatmaps encode the spatial confidence distribution of each predicted joint.

In parallel, the branch also predicts a scalar log-variance value for each joint through a dedicated linear layer applied to the intermediate features. The log-variance term models heteroscedastic uncertainty, allowing the network to adaptively weight joints according to their predicted reliability during optimization.

Together, the three branches share a common spatio-temporal backbone but specialize in complementary tasks: direct 3D regression, structural consistency through rotations, and uncertainty estimation. This multi-branch design enables robust self-supervised learning tailored to drone-based 3D pose estimation.

Table 6.2: Brancher architecture. The network extends MotionAGFormer-Small with three specialized branches for 3D pose regression, bone rotations, and uncertainty estimation.

Layer	Input Shape	Output Shape	Params
Backbone (MotionAGFormer-Small)			
Linear (Input Embedding)	$(32, 81, 17, 3)$	$(32, 81, 17, 64)$	256
AgFormer $\times 26$	$(32, 81, 17, 64)$	$(32, 81, 17, 64)$	4,780,828
LayerNorm	$(32, 81, 17, 64)$	$(32, 81, 17, 64)$	128
Pose Head			
Linear $(64 \to 512)$	$(32, 81, 17, 64)$	$(32, 81, 17, 512)$	33,280
Tanh	$(32, 81, 17, 512)$	$(32, 81, 17, 512)$	–
Linear $(512 \to 3)$	$(32, 81, 17, 512)$	$(32, 81, 17, 3)$	1,539
Rotation Head			
Linear $(64 \times 17 \to 16 \times 6)$	$(32, 81, 1088)$	$(32, 81, 96)$	104,544
Tanh	$(32, 81, 96)$	$(32, 81, 96)$	–
Uncertainty Head			
Linear $(64 \to 512)$	$(44064, 64)$	$(44064, 512)$	33,280
LeakyReLU	$(44064, 512)$	$(44064, 512)$	–
Linear $(512 \to 32 \times 8 \times 6)$	$(44064, 512)$	$(44064, 1536)$	787,968
LeakyReLU	$(44064, 1536)$	$(44064, 1536)$	–
ConvTranspose2d (32)	$(44064, 32, 8, 6)$	$(44064, 32, 16, 12)$	16,416
BatchNorm2d	$(44064, 32, 16, 12)$	$(44064, 32, 16, 12)$	64
ReLU	$(44064, 32, 16, 12)$	$(44064, 32, 16, 12)$	–
ConvTranspose2d (16)	$(44064, 32, 16, 12)$	$(44064, 16, 32, 24)$	8,208
BatchNorm2d	$(44064, 16, 32, 24)$	$(44064, 16, 32, 24)$	32
ReLU	$(44064, 16, 32, 24)$	$(44064, 16, 32, 24)$	–
ConvTranspose2d (1)	$(44064, 16, 32, 24)$	$(44064, 1, 64, 48)$	257
Sigmoid (heatmap)	$(44064, 1, 64, 48)$	$(44064, 1, 64, 48)$	–
Linear (log-var)	$(44064, 512)$	$(44064, 1)$	513
Total			**5,768,401**

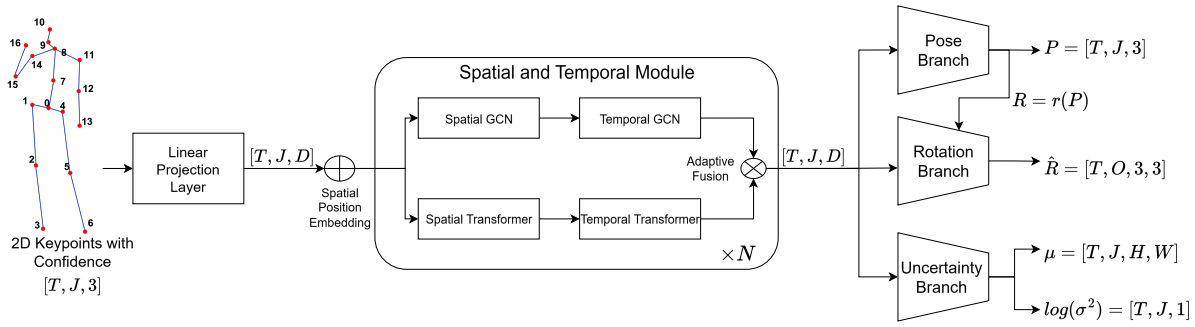

Figure 6.1: Here is reported the architecture of the proposed Brancher model. For simplicity, batches are not included in the input and output shapes.

6.3.2. Training Procedure

The Brancher model is initialized with the pretrained weights of MotionAGFormer-Small [45] trained on the Human3.6M dataset [20]. This initialization provides a strong prior for generic 3D human pose estimation under a fully supervised setting.

Training of Brancher is conducted in two sequential stages. In the first stage, the network is fine-tuned in a supervised manner on the AthletePose3D dataset [75], similarly to the Baseline 2D-to-3D lifting protocol. This stage allows the model to adapt to fast and dynamic sports movements, which are underrepresented in Human3.6M and are crucial for the target application scenario.

The second training stage is conducted entirely in a self-supervised manner using drone-based datasets, that are the UAV dataset introduced by AirPose [56] and my custom drone-captured dataset (see Appendix A.1). Both datasets consist of monocular videos acquired from UAV viewpoints and include a wide range of human motions performed under diverse environmental conditions. To further enhance variability and improve generalization, both datasets are augmented through horizontal flipping, effectively increasing pose diversity while preserving geometric consistency. Since these datasets do not provide ground-truth 3D annotations, the contribution of the MPJPE loss — the only supervised term — is explicitly set to zero. Optimization is therefore driven exclusively by the self-supervised objectives associated with the three branches. This enables the model to adapt to drone-specific viewpoints, scale variations, and motion dynamics without requiring explicit 3D ground-truth supervision. To prevent degenerate solutions during purely self-supervised training, the parameters of the PoseBranch are kept frozen. This avoids trivial minimization of the self-supervised losses through collapse of the 3D structure toward a flattened 2D configuration, ensuring that geometric consistency is preserved.

Training is carried out for 60 epochs in the supervised pretraining phase, followed by 30 epochs of purely self-supervised fine-tuning. The *AdamW* optimizer is employed with an initial learning rate of 0.0005 for pretraining and 0.000001 for the self-supervised stage, along with a weight decay of 0.01 to regularize the network parameters. The learning rate is scheduled using an exponential decay with a factor of 0.99 per epoch.

It is important to note that the multi-task loss wrapper differs between pretraining and fine-tuning, as described in Section 5.4. During supervised pretraining, losses are normalized and combined using learnable *softmax* importance weights, a temporal gating mechanism, and a dynamic floor to prioritize the primary 3D pose objective. In contrast, the self-supervised stage relies on a homoscedastic uncertainty-based weighting scheme, in which the network learns task-dependent precision parameters to adaptively balance the contributions of heterogeneous and noisy losses.

This two-stage training strategy distinguishes Brancher from the Baseline: the first stage leverages supervised signals for sports-related pose adaptation, while the second stage allows fully self-supervised fine-tuning to transfer the model to drone-captured scenarios.

6.3.3. Inference Pipeline

The inference procedure for Brancher follows the same overall pipeline described for the Baseline model (see Section 6.2.3). Input RGB frames are processed through human detection and 2D pose estimation, producing a sequence of normalized 2D keypoints. These are then fed to the Brancher network to obtain the predicted 3D poses. For improved robustness, test-time augmentation (horizontal flipping) is also applied.

7 | Experiments and Results

7.1. Evaluation Setting

Initially, I intended to evaluate the proposed method on the UAV dataset introduced by AirPose [56], as the authors provide calibration matrices for the two drones used during acquisition. In principle, this makes it possible to reconstruct 3D joint coordinates through triangulation and use them as pseudo ground truth for quantitative evaluation.

However, in practice, the triangulated 3D poses turned out to be excessively noisy. The reconstructed joints were highly unstable due to synchronization inaccuracies, imperfect detections, and sensitivity of triangulation to small 2D localization errors. As a consequence, the resulting 3D annotations were not reliable enough to produce meaningful quantitative comparisons. The evaluation metrics computed on these reconstructions were inconsistent and did not reflect the true quality of the predicted poses.

For this reason, I performed the quantitative evaluation on the test split of AthletePose3D [75], where accurate 3D ground-truth annotations are available. This allows a reliable and reproducible comparison between models under controlled conditions.

7.2. Evaluation Protocol

During evaluation, 2D keypoints are first extracted and normalized following the preprocessing procedure described in the Section 5.1. Temporal windows of 81 consecutive frames are fed to the network using a sliding-window strategy consistent with training.

Predicted 3D poses are root-centered before metric computation. When enabled, test-time augmentation (TTA) through horizontal flipping is applied, and final predictions are obtained by averaging the outputs of the original and flipped inputs.

7.3. Evaluation Metrics

To assess performance, I report Mean Per Joint Position Error (MPJPE) and Procrustes-aligned MPJPE (P-MPJPE), both measured in millimeters.

MPJPE computes the average Euclidean distance between predicted and ground-truth 3D joint coordinates. P-MPJPE applies a rigid Procrustes alignment before error computation, removing global rotation and translation differences. This metric isolates structural pose accuracy independently of global orientation.

7.4. Quantitative Results

Table 7.1: Quantitative comparison performed on the AthletePose3D test set.

Model	MPJPE (mm)	P-MPJPE (mm)
MotionAGFormer	160.6560	86.1506
MotionAGFormer Finetuned	14.1359	11.0060
Baseline	45.2764	4.3332
Brancher	30.9958	27.8366
Brancher Self-Supervised Finetuned	41.1452	32.6568

Table 7.4 reports the quantitative comparison on the AthletePose3D test set using MPJPE and P-MPJPE as evaluation metrics.

MotionAGFormer [45], originally designed and validated on Human3.6M [20], achieves strong performance on that benchmark. However, when directly evaluated on AthletePose3D, it exhibits very high error (160.66 mm MPJPE), confirming that state-of-the-art 3D pose estimators trained on generic indoor datasets do not directly generalize to fast and highly dynamic sports motions. This limitation has also been highlighted by the authors of AthletePose3D.

The same architecture, when fine-tuned in a fully supervised manner on AthletePose3D (MotionAGFormer Finetuned), achieves a drastic reduction in error (14.14 mm MPJPE), obtaining the best numerical performance among all evaluated models. This result emphasizes the importance of large-scale 3D pretraining followed by domain-specific supervised adaptation.

The proposed Baseline model achieves a significantly lower MPJPE (45.28 mm) than the original MotionAGFormer, indicating that training directly on sports-specific data

improves in-domain reconstruction accuracy even without large-scale pretraining.

Brancher, trained only in the supervised setting on AthletePose3D (without UAV adaptation), further improves over the Baseline, reducing MPJPE to 31.00 mm. This suggests that the proposed spatio-temporal modeling strategy enhances 3D pose prediction compared to a purely temporal lifting formulation.

Finally, Brancher Self-Supervised Finetuned exhibits a higher MPJPE (41.15 mm) than its supervised counterpart. This behavior is expected, as the additional self-supervised training stage on UAV datasets shifts the optimization objective away from strict minimization of the AthletePose3D benchmark error. Consequently, while in-domain accuracy slightly decreases, the model becomes adapted to aerial viewpoints and drone-specific motion characteristics.

It is worth noting that models such as AirPose [56] or AerialView [17] are designed for image-based 3D pose estimation and cannot be directly evaluated under the video-based lifting protocol adopted in this work. To the best of our knowledge, Brancher represents one of the first attempts to address self-supervised spatio-temporal 3D pose adaptation specifically for UAV video scenarios.

7.5. Ablation Study

7.5.1. Effect of Large-Scale Pretraining

Table 7.2: Impact of large-scale pretraining on AthletePose3D performance.

Model	MPJPE (mm)	P-MPJPE (mm)
MotionAGFormer	160.6560	86.1506
MotionAGFormer Finetuned	14.1359	11.0060
Baseline	45.2764	4.3332

To isolate the effect of large-scale pretraining, I compare MotionAGFormer trained on Human3.6M, its fully supervised fine-tuned version on AthletePose3D, and the proposed Baseline trained directly on AthletePose3D from scratch.

As shown in Table 7.2, MotionAGFormer performs poorly when transferred without adaptation, confirming that priors learned on controlled indoor datasets do not directly generalize to fast athletic motions. Training the Baseline directly on AthletePose3D already leads to a substantial improvement over the unadapted model, indicating that domain-

specific supervision is more critical than generic pretraining alone.

However, the fully supervised fine-tuned version of MotionAGFormer clearly outperforms both, demonstrating that large-scale 3D pretraining becomes highly effective once combined with task-specific adaptation.

Overall, this comparison confirms that generic 3D pose priors are insufficient without domain alignment, while supervised fine-tuning on sports data remains the most effective strategy when ground-truth 3D annotations are available.

7.5.2. Effect of Architectural Branches

Table 7.3: Ablation study on architectural components of Brancher.

Configuration	MPJPE (mm)	P-MPJPE (mm)
Baseline	45.2764	4.3332
Brancher with only PoseBranch	22.0248	18.5976
Brancher with only PoseBranch and RotationBranch	20.8884	17.6838
Brancher with only PoseBranch and UncertaintyBranch	35.9176	32.1069
Brancher	30.9958	27.8366

To understand the contribution of each architectural component, I progressively enable the different branches of the proposed model and evaluate their impact on reconstruction accuracy.

Introducing only the Pose Branch already leads to a substantial improvement over the Baseline, reducing MPJPE from 45.28 mm to 22.02 mm. This confirms that explicitly modeling structured pose regression within the proposed framework significantly strengthens the 2D-to-3D lifting process.

Adding the Rotation Branch further improves performance, lowering MPJPE to 20.89 mm. This suggests that explicitly supervising joint rotations provides additional geometric constraints that stabilize the 3D reconstruction and reduce ambiguity in joint orientation.

When the Uncertainty Branch is introduced, MPJPE increases (35.92 mm when combined only with the Pose Branch, and 30.99 mm in the full model). This behavior is expected, as the optimization objective is no longer driven exclusively by direct 3D coordinate regression, but instead balances multiple learning signals. The model allocates capacity to uncertainty estimation, which partially relaxes the strict minimization of positional error.

Overall, the ablation study shows that the Pose Branch is the primary contributor to performance gains, the Rotation Branch provides complementary geometric regularization, and the Uncertainty Branch introduces a trade-off between pure coordinate accuracy and multi-objective optimization.

7.5.3. Impact of Self-Supervised Fine-Tuning

Table 7.4: Ablation study on the impact of self-supervised fine-tuning on AthletePose3D performance.

Model	MPJPE (mm)	P-MPJPE (mm)
Brancher	30.9958	27.8366
Brancher Self-Supervised Finetuned	41.1452	32.6568

Finally, I evaluate the impact of the self-supervised UAV adaptation stage by comparing Brancher with its self-supervised fine-tuned version.

Brancher Self Supervised Finetuned shows an increase in MPJPE compared to the purely supervised Brancher. This behavior is expected, as the additional training phase is optimized without direct 3D supervision and shifts the objective away from strict minimization of the AthletePose3D benchmark error.

The self-supervised stage is designed to promote robustness to aerial viewpoints, scale variations, and drone-specific motion patterns. As a result, the model sacrifices a small portion of in-domain accuracy to gain improved cross-domain generalization capabilities, which are not directly reflected by benchmark metrics computed on AthletePose3D.

7.6. Qualitative Results

7.6.1. Qualitative Results on AthletePose3D

In this section I present a qualitative comparison of the different models on AthletePose3D. Rather than focusing on numerical metrics, I analyze the visual coherence, structural stability, and motion plausibility of the reconstructed 3D poses.

In all the following figures, the skeleton highlighted in red represents the 3D reconstruction produced by Brancher. The other models are shown for visual comparison under the same input sequences in blue.

MotionAGFormer (Pretrained Only)

When directly applied to AthletePose3D without fine-tuning, MotionAGFormer appears visually inadequate for reconstructing fast and complex athletic motions. As visible in the figures, the predicted skeleton frequently collapses, exhibits incorrect limb proportions, and fails to maintain coherent joint alignment across frames.

The reconstructed poses do not reflect the dynamics of sports movements, confirming that the model is not adapted to this domain not only numerically but also structurally and visually.

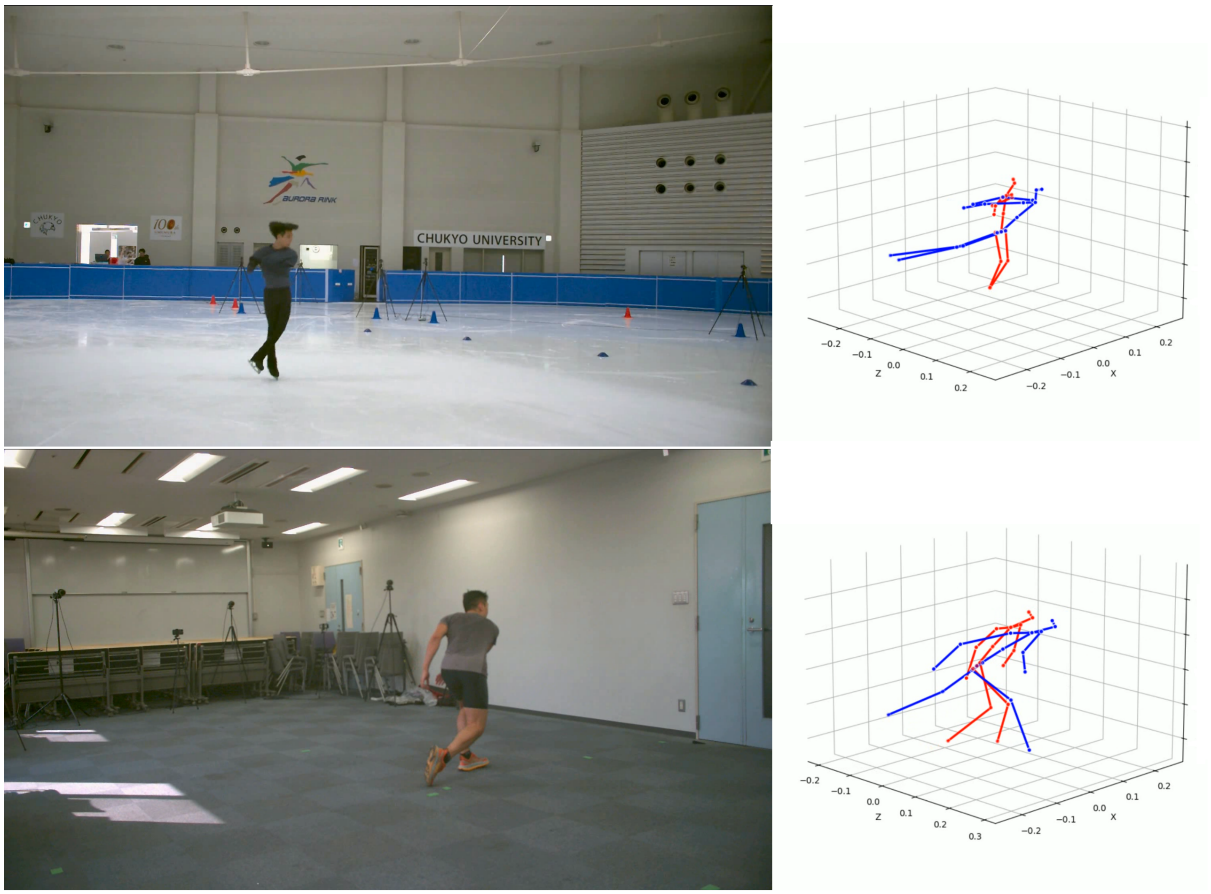

Figure 7.1: MotionAGFormer (blue) vs Brancher (red) qualitative comparison on AthletePose3D.

MotionAGFormer Fine-Tuned

The fine-tuned version provided with AthletePose3D shows a substantial qualitative improvement. The predicted poses are generally closer to the correct configuration and major structural errors are significantly reduced.

However, from the extracted frames it appears that the model tends to overfit the dataset. While single-frame poses can look accurate, the reconstructed motion lacks temporal smoothness and coherent biomechanical continuity. Rapid transitions often generate abrupt changes in joint configuration, suggesting limited motion consistency despite strong benchmark performance.

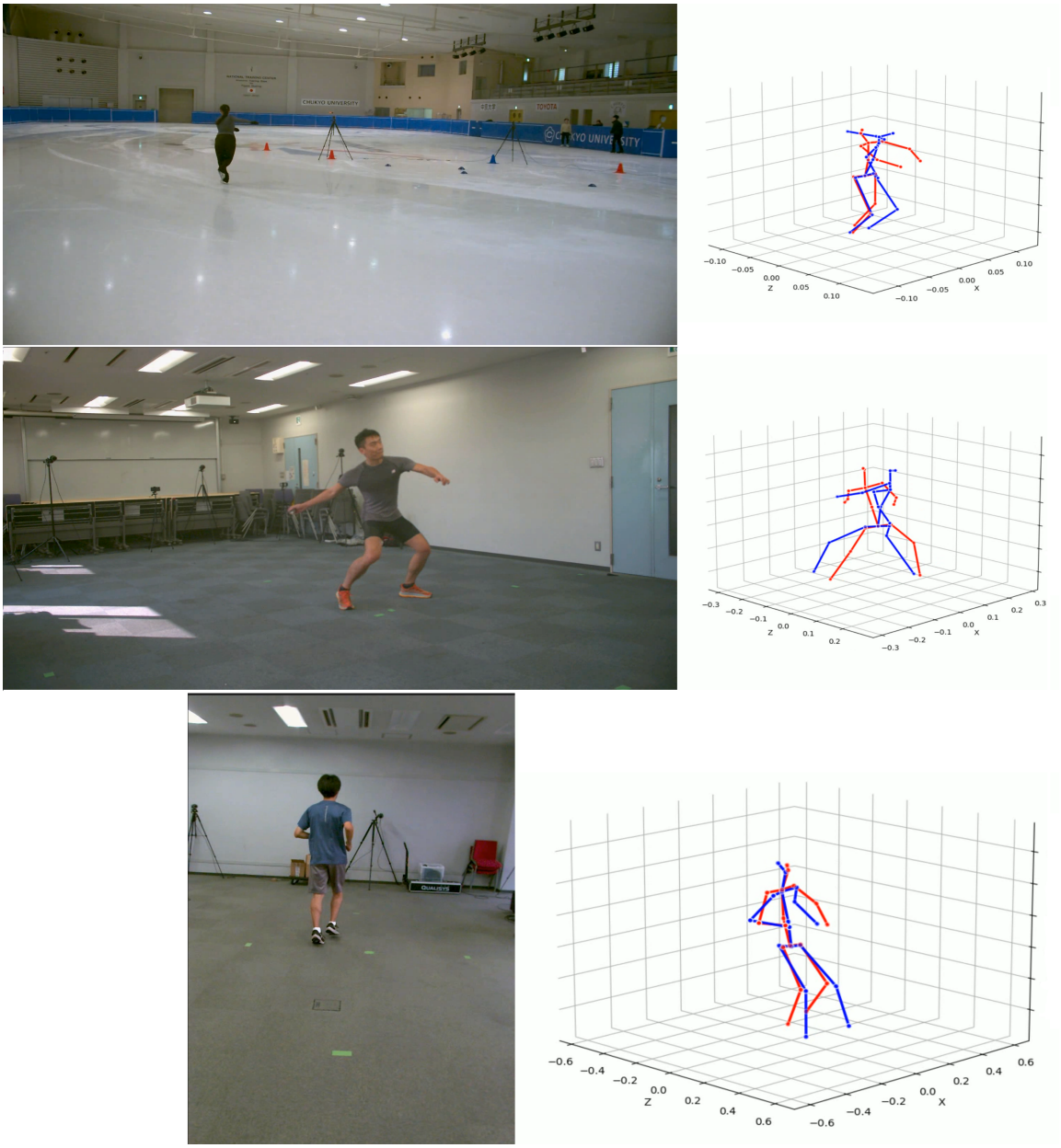

Figure 7.2: MotionAGFormer Finetuned (blue) vs Brancher (red) qualitative comparison on AthletePose3D.

Baseline

The Baseline model occasionally predicts a plausible static 3D pose, but the reconstructed motion appears unstable and unnatural across consecutive frames. Limbs may abruptly change orientation, and body alignment fluctuates even when the input motion is smooth.

Although the quantitative test score is competitive, the visual inspection reveals that the model struggles to reproduce realistic sports motion from video sequences. The benchmark performance does not fully reflect its limitations in dynamic reconstruction.

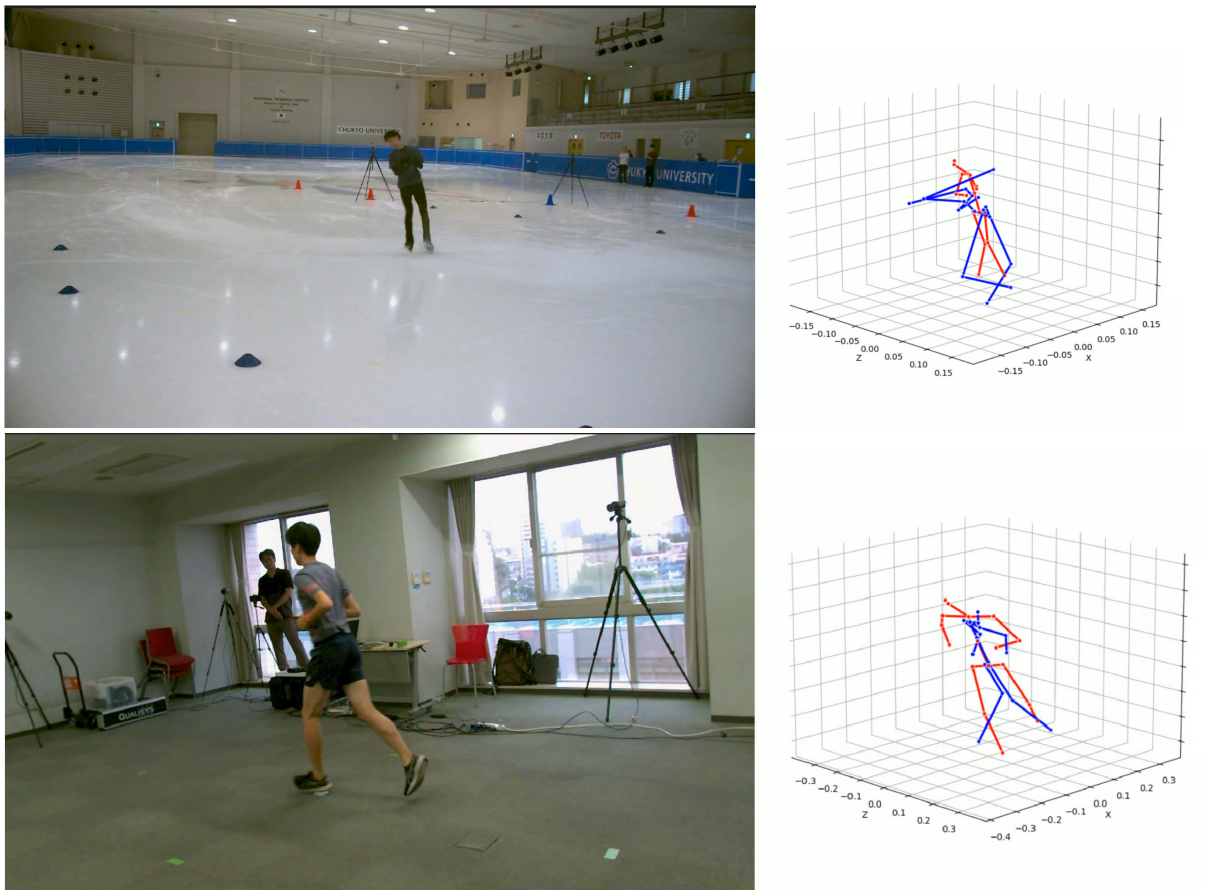

Figure 7.3: Baseline (blue) vs Brancher (red) qualitative comparison on AthletePose3D.

Brancher with Pose Branch Only

Using only the Pose Branch leads to visually improved structural stability compared to the Baseline. The predicted skeleton better preserves body proportions and overall pose configuration.

Nevertheless, during fast transitions the model occasionally produces unnatural body

"jumps" or abrupt posture adjustments. While static pose estimation is reliable, the temporal evolution of motion is not consistently smooth.

Figure 7.4: Brancher with only Pose Branch (blue) vs Brancher (red) qualitative comparison on AthletePose3D.

Brancher with Pose Branch and Rotation Branch

Adding the Rotation Branch provides slightly better geometric consistency. Joint orientations appear more constrained and anatomically plausible in several frames.

However, the qualitative difference with respect to the Pose Branch only configuration remains limited, which aligns with the marginal MPJPE improvement observed in the quantitative evaluation. Temporal smoothness issues are still present in complex dynamic segments.

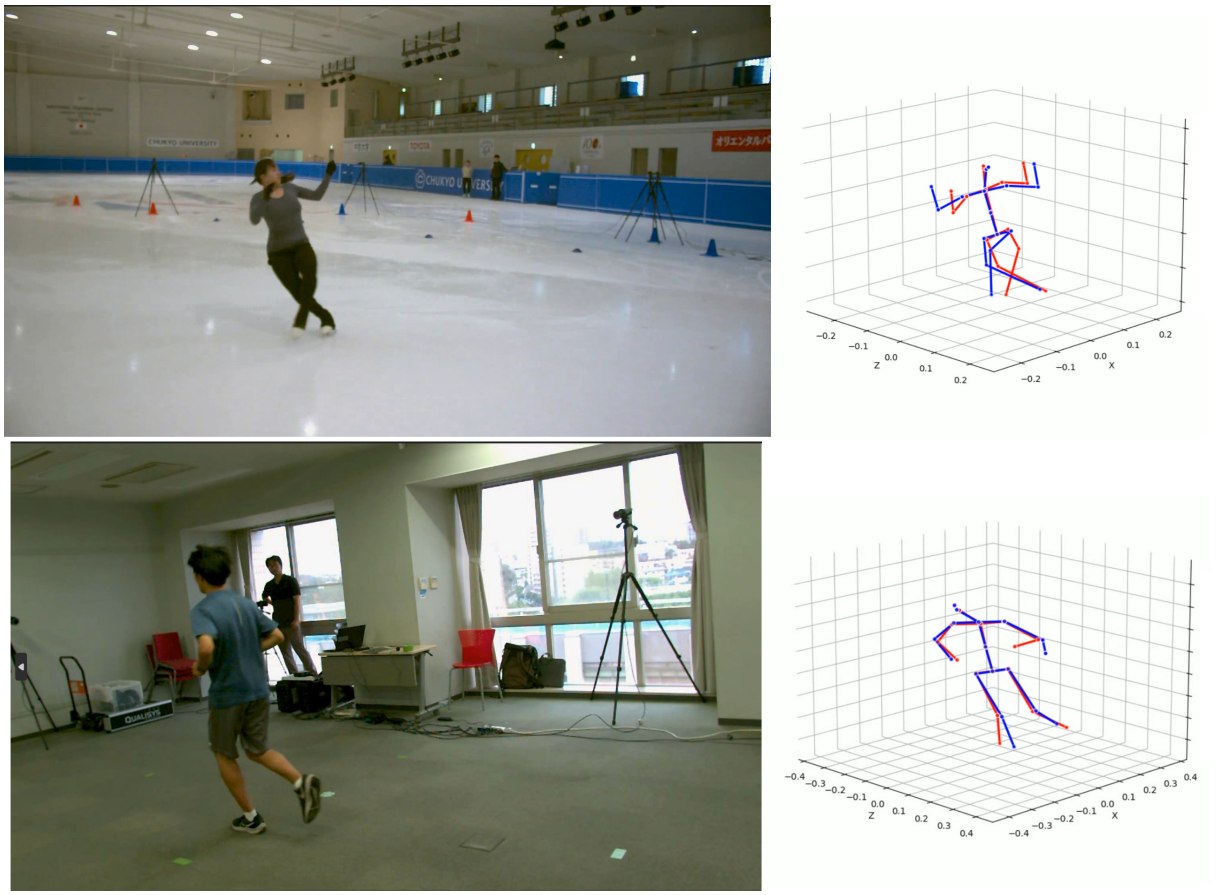

Figure 7.5: Brancher with Pose Branch and Rotation Branch (blue) vs Brancher (red) qualitative comparison on AthletePose3D.

Brancher with Pose Branch and Uncertainty Branch

When the Uncertainty Branch is introduced, minor local imperfections may appear in isolated frames. However, the overall motion becomes significantly smoother and more temporally coherent.

Although this improvement in motion fluidity cannot be fully appreciated through static images, the reconstructed sequences show a markedly more stable evolution of body dynamics. The resulting movement better reflects continuous human motion, particularly in fast athletic actions.

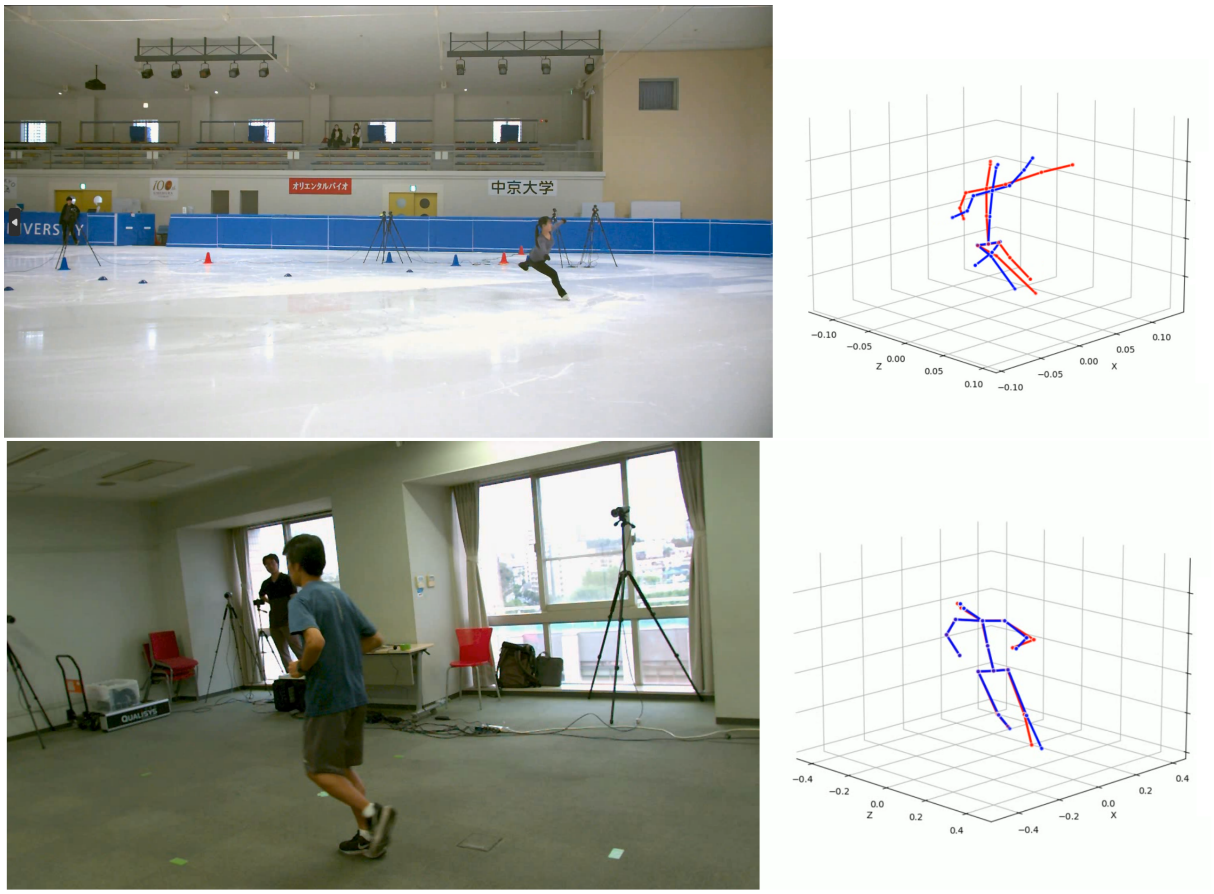

Figure 7.6: Brancher with Pose Branch and Uncertainty Branch (blue) vs Brancher (red) qualitative comparison on AthletePose3D.

7.6.2. Qualitative Results on UAV-Captured Dataset

I now evaluate the models on sequences captured from a drone viewpoint. Compared to AthletePose3D, this scenario introduces additional challenges, including aerial perspective distortion, scale variation, and increased motion complexity.

MotionAGFormer (Pretrained Only)

MotionAGFormer without fine-tuning is again visually inadequate. The reconstructed poses are unstable and frequently inconsistent with the observed motion, confirming its inability to generalize to aerial sports footage.

Figure 7.7: MotionAGFormer (blue) vs Brancher (red) qualitative comparison on UAV-captured sequences.

MotionAGFormer Fine-Tuned

The MotionAGFormer fine-tuned on AthletePose3D performs better than the original pre-trained version, but the reconstructed motion remains imprecise and exhibits noticeable temporal discontinuities. Fast actions often produce abrupt pose transitions.

Figure 7.8: MotionAGFormer Finetuned (blue) vs Brancher (red) qualitative comparison on UAV-captured sequences.

Baseline

The Baseline model appears largely unsuitable in this setting. While occasional frames may resemble plausible poses, the overall motion reconstruction lacks stability and coherence under drone viewpoints.

Figure 7.9: Baseline (blue) vs Brancher (red) qualitative comparison on UAV-captured sequences.

Brancher with Pose Branch Only

Brancher with only the Pose Branch significantly improves structural consistency compared to the previous models. However, minor motion discontinuities and precision errors are still visible during rapid movements.

Figure 7.10: Brancher with Pose Branch (blue) vs Brancher (red) qualitative comparison on UAV-captured sequences.

Brancher with Pose Branch and Rotation Branch

Adding the Rotation Branch provides a slight improvement in geometric precision, particularly in limb orientation. Nevertheless, temporal smoothness remains comparable to the Pose Branch only configuration.

Figure 7.11: Brancher with Pose Branch and Rotation Branch (blue) vs Brancher (red) qualitative comparison on UAV-captured sequences.

Brancher with Pose Branch and Uncertainty Branch

The configuration with Pose Branch and Uncertainty Branch yields a more convincing results in this setting. In addition to improved precision, the reconstructed motion becomes substantially smoother and more temporally coherent. The resulting 3D sequences better reflect continuous human dynamics under aerial observation.

Figure 7.12: Brancher with Pose Branch and Uncertainty Branch (blue) vs Brancher (red) qualitative comparison on UAV-captured sequences.

Brancher with Self-Supervised UAV Fine-Tuning

The following figures compare the full Brancher model before and after self-supervised fine-tuning on the UAV-captured dataset. In the visualizations, the skeleton highlighted in red represents Brancher without UAV adaptation, while the skeleton in green corresponds

to the self-supervised fine-tuned version.

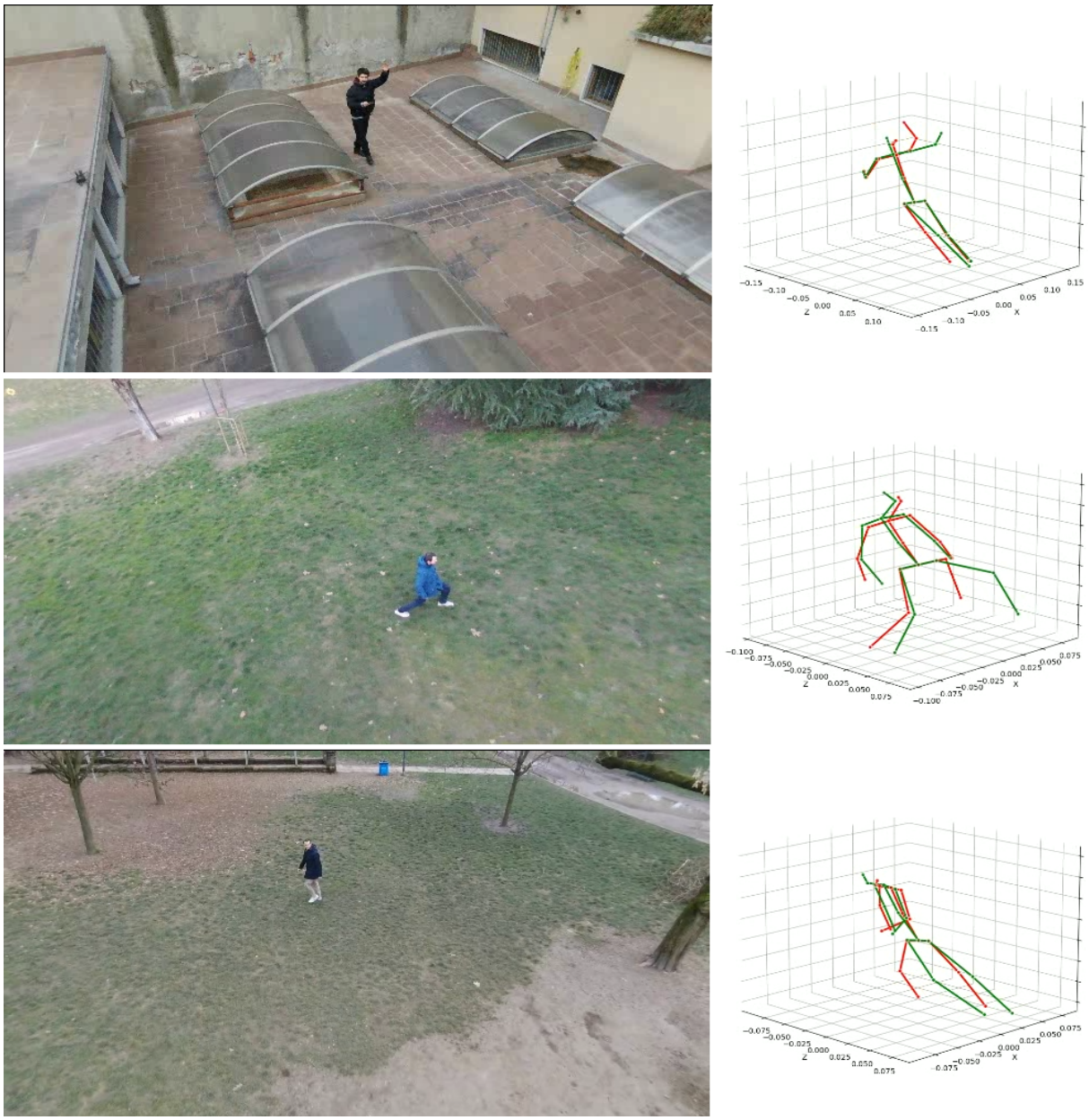

Figure 7.13: Full Brancher before (red) and after (green) self-supervised fine-tuning on UAV-captured sequences.

From a qualitative perspective, the results are mixed. In some sequences, the self-supervised adaptation produces smoother and more coherent reconstructions. In other cases, however, the fine-tuned model does not improve the structural accuracy and may even degrade the 3D pose estimation.

These observations highlight the intrinsic limitations of purely self-supervised training in

Figure 7.14: Full Brancher before (red) and after (green) self-supervised fine-tuning on UAV-captured sequences.

the context of 3D human pose estimation. As discussed in the introduction, recovering a unique 3D configuration from monocular 2D observations is fundamentally ill-posed. Without explicit 3D supervision, the optimization process is guided only by indirect constraints.

The losses introduced in Section 5.4.2 enforce consistency in bone length, temporal smoothness, velocity, acceleration, and rotational stability. However, these constraints do not guarantee geometric correctness in absolute 3D space. In particular, the Uncertainty Branch — which is intended to guide the self-supervised adaptation — may converge to degenerate but loss-minimizing solutions. For instance, partially flattening the 3D structure can reduce reprojection or smoothness-related penalties without violating bone-length consistency or temporal regularization terms.

The only constraint that would explicitly prevent such degenerate configurations is the 3D MPJPE loss, which is unavailable in this setting due to the absence of ground-truth 3D annotations for drone-captured sports sequences. As a consequence, the optimization may reach a local optimum that is consistent with the self-supervised objectives but not fully aligned with geometrically accurate 3D pose reconstruction.

This limitation reflects a broader challenge of purely self-supervised 3D human pose estimation under monocular aerial viewpoints. Nevertheless, given the lack of 3D annotations for drone footage — which would require complex and costly motion capture setups — self-supervised adaptation remains a pragmatic and realistic strategy. Within these constraints, the proposed approach represents the most effective solution achievable with the available resources.

For others qualtitative results the reader is referred to the Appendix B, where additional visual comparisons are provided.

7.7. Discussion

From both the quantitative and qualitative results, I can conclude that Brancher achieves solid performance in reconstructing 3D athletic motion from UAV-captured footage when compared to existing models. The proposed architecture improves structural consistency and motion plausibility, particularly when uncertainty modeling is incorporated.

The comparison with the Baseline also provides an important insight: large-scale pretraining on a generic dataset such as Human3.6M is fundamental for reliable 3D reconstruction of complex sports motions. Athletic movements are highly dynamic and structurally demanding, and training a model from scratch on limited domain-specific data is insufficient. A network that is already well-initialized for the 3D HPE task is significantly better equipped to handle rapid pose transitions and non-trivial body configurations.

Qualitative inspection further shows that the motions reconstructed by Brancher are generally fluid, natural, and biomechanically plausible. Compared to previous configurations, the predicted sequences better reflect continuous human dynamics rather than isolated frame-wise pose estimation.

However, the model is not without limitations. As discussed in the previous section, the absence of ground-truth 3D annotations for drone-captured sequences constrains the effectiveness of adaptation strategies. Purely self-supervised training introduces intrinsic ambiguities, as the optimization process is guided by indirect constraints rather than explicit 3D supervision. This suggests that the Uncertainty Branch, while beneficial for motion smoothness and stability, could be further improved to better prevent degenerate or geometrically suboptimal solutions.

Moreover, the model struggles in highly challenging scenarios. As shown in the following figures, reconstruction quality degrades when the subject performs extreme tricks such as backflips or interacts with external objects. These situations introduce rapid orientation changes, self-occlusions, and complex body-object interactions that are not sufficiently constrained by the current training objectives.

Overall, while Brancher represents a meaningful step forward for UAV-based 3D human pose estimation in sports scenarios, its limitations highlight the need for stronger geometric priors, improved uncertainty modeling, and more robust handling of extreme dynamic motion.

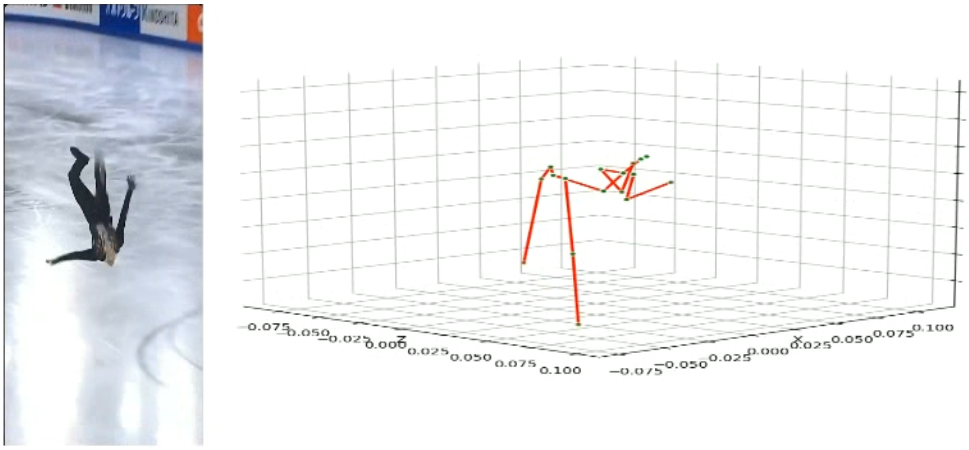

Figure 7.15: Backflip example where Brancher struggles to maintain accurate reconstruction.

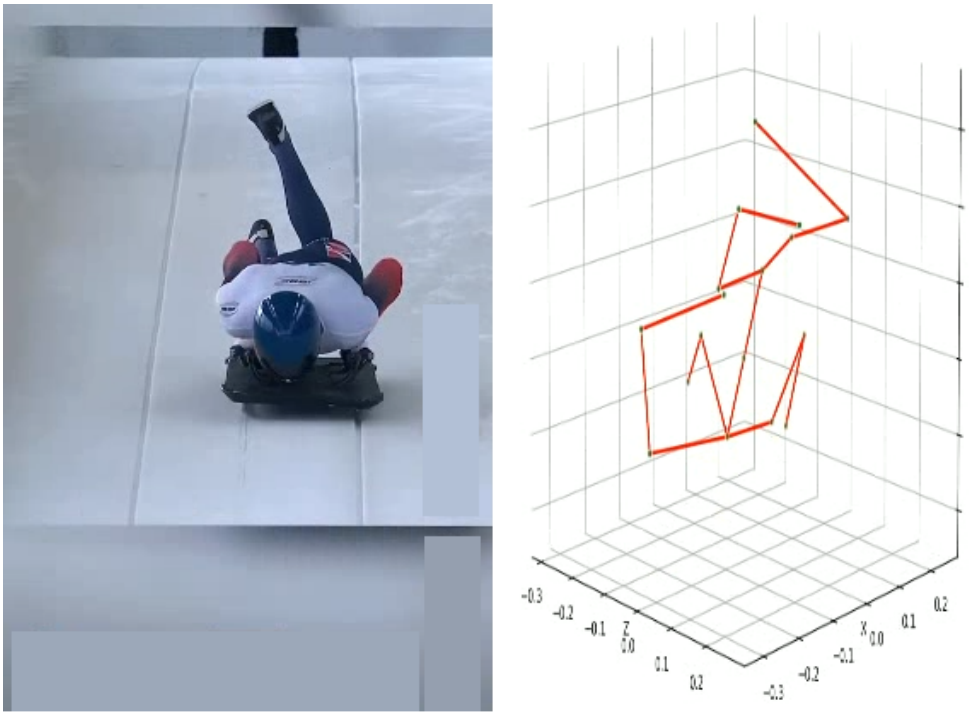

Figure 7.16: Example where Brancher fails to reconstruct the 3D HPE of the subject interacting with an object.

8 | Conclusions

The objective of this thesis was to develop a deep learning model capable of performing 3D Human Pose Estimation from sports videos captured by a drone. To address this problem, I proposed Brancher, a single-view architecture designed to operate under aerial viewpoints and challenging motion dynamics. The model was conceived to satisfy the requirements defined in the problem formulation while contributing to the current research landscape, where single-view 3D HPE methods specifically tailored to UAV footage remain limited.

Brancher integrates spatio-temporal feature manipulation with uncertainty-aware learning. By jointly modeling spatial structure and temporal dynamics, and by optimizing 2D consistency, rotational alignment, and prediction uncertainty, the model mitigates the effects of occlusions, viewpoint variations, and noisy pseudo-labels. This design enables the reconstruction of complex athletic 3D motions from challenging drone footage while maintaining structural coherence and temporal stability.

Both quantitative and qualitative evaluations demonstrate that Brancher achieves solid performance compared to existing baselines. The architecture benefits from a state-of-the-art backbone, yet extends it through task-specific modifications that improve motion plausibility and robustness under aerial observation. Importantly, the model remains computationally lightweight, with approximately 5 million parameters. This relatively compact design opens the possibility for future onboard deployment directly on UAV platforms, enabling real-time 3D motion analysis in sports scenarios.

Although the model is subject to the limitations discussed in the previous chapter, particularly those related to the absence of ground-truth 3D annotations and the intrinsic challenges of purely self-supervised adaptation, it represents a meaningful step toward practical 3D human motion reconstruction from aerial video. I developed a system capable of balancing multiple optimization objectives to address a complex and underexplored problem domain, namely 3D HPE under UAV-based acquisition. Beyond its methodological contribution, Brancher is intended as a step toward practical motion analysis in sports performance, with the long-term goal of supporting performance optimization and

injury prevention through more accessible 3D biomechanical reconstruction.

This research project evolved progressively, starting from keypoint detection and 2D HPE, extending to 3D HPE, and ultimately reaching the more demanding setting of 3D reconstruction from UAV footage. The path toward Brancher required navigating a vast body of literature, analyzing hundreds of publications to identify both methodological gaps and promising directions.

The development process involved extensive experimentation, iterative model redesign, and long training cycles performed on high-performance GPU servers. Multiple architectural configurations were tested before converging to the final formulation. This journey was not only a technical endeavor but also an exercise in critical reasoning, persistence, and methodological rigor.

Working on this problem highlighted how research is rarely a linear process. Instead, it requires continuous refinement of ideas, empirical validation, and acceptance of both progress and limitations. The transition from controlled indoor benchmarks to unconstrained aerial sports footage exemplifies the gap between theoretical performance and real-world applicability. Bridging this gap has been one of the central motivations of this work.

8.1. Future Work

A first and fundamental direction for future research is the creation of a dedicated UAV-based 3D Human Pose Estimation dataset. Such a dataset could be developed either through high-fidelity simulation environments or through real-world acquisition using multiple synchronized drones to reconstruct 3D pose with high precision.
The performance of deep learning models is strongly dependent on the quality and representativeness of the training data. In the context of aerial sports analysis, the lack of 3D-annotated UAV footage remains one of the main bottlenecks. The availability of a large-scale dataset specifically tailored to sports scenarios captured from drones would significantly enhance the training process and provide stronger supervision for models such as Brancher. This would also reduce the reliance on purely self-supervised adaptation strategies and mitigate the geometric ambiguities inherent to monocular reconstruction.

A second promising direction concerns the development of models capable of learning from datasets annotated with heterogeneous joint conventions. In practice, many 2D and 3D datasets adopt different skeletal definitions (e.g., COCO, Human3.6M, and others), which limits the possibility of jointly exploiting multiple data sources.

While Human3.6M [20] has become the de facto standard for 3D supervision due to its size and popularity, sports-related datasets are often smaller and follow alternative annotation schemes. Designing architectures that can align, map, or dynamically adapt across different skeletal representations would allow training on a broader range of datasets. In data-scarce domains such as UAV sports footage, this capability could substantially improve generalization and robustness.

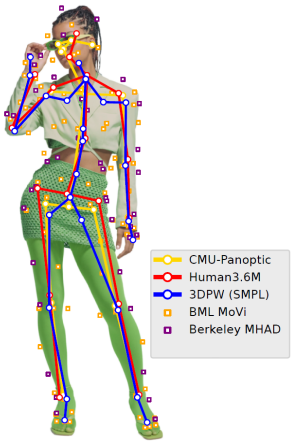

Figure 8.1: This picture taken from the work of Sarandi et al. [59], shows the diversity of joint definitions across different datasets, in a work that addresses the problem of learning from heterogeneous annotations.

Finally, in the current era of generative artificial intelligence, diffusion-based models offer an intriguing opportunity for advancing single-view 3D HPE. A possible research direction would involve training a diffusion model that takes normalized 2D joint coordinates as input and generates additional plausible viewpoints of the same pose.
By leveraging geometric constraints such as epipolar consistency or learned cross-view priors, synthetic multi-view representations could be constructed from monocular input. This would effectively transform a single-view reconstruction problem into a richer multi-view setting, where triangulation or multi-view consistency losses could be applied.
Such an approach could bridge the gap between monocular ambiguity and multi-view geometric constraints, opening new possibilities for 3D human motion reconstruction from UAV footage without requiring physically deployed multi-drone systems.

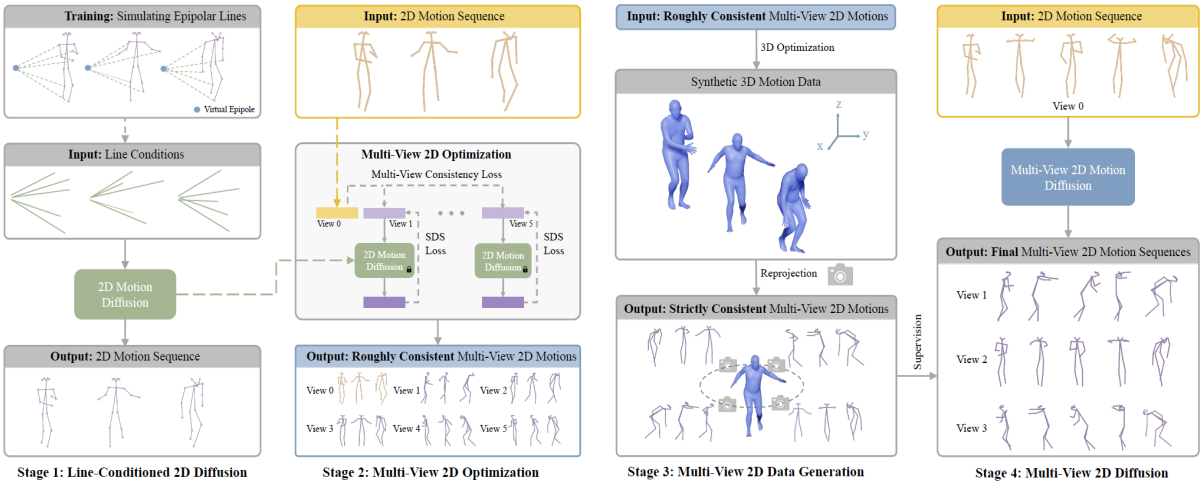

Figure 8.2: This image, taken from the work of Li et al. [32], shows a schematic representation of a diffusion model for generating multi-view 3D human poses from monocular 2D joint coordinates.

With this final section, my thesis work comes to an end, marking also the conclusion of my academic journey at Politecnico di Milano. This work represents the culmination of years of study, research, and personal growth.

Thank you for taking the time to read my Master's thesis.

Sincerely,
Angelo

Bibliography

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

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

[3] A. Benzine, F. Chabot, B. Luvison, Q. C. Pham, and C. Achard. Pandanet: Anchor-based single-shot multi-person 3d pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6856–6865, 2020.

[4] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[5] J. Chen, B. He, C. D. Singh, C. Fermüller, and Y. Aloimonos. Active human pose estimation via an autonomous uav agent. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7801–7808. IEEE, 2024.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] A. Doering, U. Iqbal, and J. Gall. Joint flow: Temporal flow fields for multi person tracking. *arXiv preprint arXiv:1805.04596*, 2018.

[8] A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] A. El Kaid and K. Baïna. A systematic review of recent deep learning approaches for 3d human pose estimation. *Journal of imaging*, 9(12):275, 2023.

[10] M. Gholami, A. Rezaei, H. Rhodin, R. Ward, and Z. J. Wang. Self-supervised 3d human pose estimation from video. *Neurocomputing*, 488:97–106, 2022.

[11] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran. Detect-and-track: Efficient pose estimation in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 350–359, 2018.

[12] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[16] S. Honari, V. Constantin, H. Rhodin, M. Salzmann, and P. Fua. Temporal representation learning on monocular videos for 3d human pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6415–6427, 2022.

[17] J. Hwang and J. Kang. Aerial view 3d human pose estimation using double vector quantized-variational autoencoders. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 341–350, 2024.

[18] C. K. Ingwersen, C. M. Mikkelstrup, J. N. Jensen, M. R. Hannemose, and A. B. Dahl. Sportspose-a dynamic 3d sports pose dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5219–5228, 2023.

[19] C. K. Ingwersen, R. Tirsgaard, R. Nylander, J. N. Jensen, A. B. Dahl, and M. R. Hannemose. Two views are better than one: Monocular 3d pose estimation with multiview consistency. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5915–5925, 2025.

[20] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[21] U. Iqbal, A. Milan, and J. Gall. Posetrack: Joint multi-person pose estimation and

tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2020, 2017.

[22] S. Jeon, I. Cho, J. Hong, and S. J. Kim. Unsupervised monocular 3d keypoint discovery from multi-view diffusion priors. *arXiv preprint arXiv:2507.12336*, 2025.

[23] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013.

[24] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *bmvc*, volume 2, page 5. Aberystwyth, UK, 2010.

[25] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR 2011*, pages 1465–1472. IEEE, 2011.

[26] T. Kalampokas, S. Krinidis, V. Chatzis, and G. A. Papakostas. Performance benchmark of deep learning human pose estimation for uavs. *Machine Vision and Applications*, 34(6):97, 2023.

[27] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[28] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[29] T. Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[30] S. Kreiss, L. Bertoni, and A. Alahi. Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):13498–13511, 2021.

[31] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10863–10872, 2019.

[32] J. Li, C. K. Liu, and J. Wu. Lifting motion to the 3d world via 2d diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17518–17528, 2025.

[33] T. Li, J. Liu, W. Zhang, Y. Ni, W. Wang, and Z. Li. Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 16266–16275, 2021.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[35] J. Liu, M. Liu, H. Liu, and W. Li. Tcpformer: Learning temporal correlation with implicit pose proxy for 3d human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5478–5486, 2025.

[36] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[37] C. Luo, S. Song, W. Xie, L. Shen, and H. Gunes. Learning multi-dimensional edge feature-based au relation graph for facial action unit recognition. *arXiv preprint arXiv:2205.01782*, 2022.

[38] D. C. Luvizon, D. Picard, and H. Tabia. Multi-task deep learning for real-time 3d human pose estimation and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2752–2764, 2020.

[39] D. C. Luvizon, H. Tabia, and D. Picard. Ssp-net: Scalable sequential pyramid networks for real-time 3d human pose regression. *Pattern Recognition*, 142:109714, 2023.

[40] X. Ma, H. Rahmani, Z. Fan, B. Yang, J. Chen, and J. Liu. Remote: Reinforced motion transformation network for semi-supervised 2d pose estimation in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1944–1952, 2022.

[41] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[42] D. Maji, S. Nagori, M. Mathew, and D. Poddar. Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2637–2646, 2022.

[43] G. Martinelli, F. Diprima, N. Bisagno, and N. Conci. Ski pose estimation. In *2024 IEEE International Workshop on Sport, Technology and Research (STAR)*, pages 120–125. IEEE, 2024.

[44] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2640–2649, 2017.

[45] S. Mehraban, V. Adeli, and B. Taati. Motionagformer: Enhancing 3d human pose estimation with a transformer-gcnformer network. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 6920–6930, 2024.

[46] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 international conference on 3D vision (3DV)*, pages 506–516. IEEE, 2017.

[47] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt. Xnect: Real-time multi-person 3d motion capture with a single rgb camera. *Acm Transactions On Graphics (TOG)*, 39(4):82–1, 2020.

[48] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[49] X. Nie, Y. Li, L. Luo, N. Zhang, and J. Feng. Dynamic kernel distillation for efficient pose estimation in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6942–6950, 2019.

[50] C. D. Pace, A. M. De Nunzio, C. De Stefano, F. Fontanella, and M. Molinara. Poseidon: A vit-based architecture for multi-frame pose estimation with adaptive frame weighting and multi-scale feature fusion. *arXiv preprint arXiv:2501.08446*, 2025.

[51] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019.

[52] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[53] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[54] D. Rubiagatra, A. D. Wibawa, and E. M. Yuniarno. Evaluating squat technique in

pound fitness through deep learning and human pose estimations. In *2024 7th International Conference on Informatics and Computational Sciences (ICICoS)*, pages 382–387. IEEE, 2024.

[55] N. Saini, E. Price, R. Tallamraju, R. Enficiaud, R. Ludwig, I. Martinovic, A. Ahmad, and M. J. Black. Markerless outdoor human motion capture using multiple autonomous micro aerial vehicles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 823–832, 2019.

[56] N. Saini, E. Bonetto, E. Price, A. Ahmad, and M. J. Black. Airpose: Multi-view fusion network for aerial 3d human pose and shape estimation. *IEEE Robotics and Automation Letters*, 7(2):4805–4812, 2022.

[57] E. Samkari, M. Arif, M. Alghamdi, and M. A. Al Ghamdi. Human pose estimation using deep learning: A systematic literature review. *Machine Learning and Knowledge Extraction*, 5(4):1612–1659, 2023.

[58] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3681, 2013.

[59] I. Sárándi, A. Hermans, and B. Leibe. Learning 3d human pose estimation from dozens of datasets using a geometry-aware autoencoder to bridge between skeleton formats. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2956–2966, 2023.

[60] J. Sosa and D. Hogg. Self-supervised 3d human pose estimation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4788–4797, 2023.

[61] V. Srivastav, K. Chen, and N. Padoy. Selfpose3d: self-supervised multi-person multi-view 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2502–2512, 2024.

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

[63] M. H. Teh, G. Lou, N. Ralston, Z. Zhao, B. Fan, and T. Li. Drone-based human motion capture: A review. *Intelligent Sports and Health*, 2(1):24–38, 2026.

[64] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016.

[65] Y. Tian, Q. Ye, and D. Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.

[66] D. Tome, T. Alldieck, P. Peluse, G. Pons-Moll, L. Agapito, H. Badino, and F. De la Torre. Selfpose: 3d egocentric pose estimation from a headset mounted camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):6794–6806, 2020.

[67] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[68] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[69] J. Wang, K. Qiu, H. Peng, J. Fu, and J. Zhu. Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance. In *Proceedings of the 27th ACM international conference on multimedia*, pages 374–382, 2019.

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

[71] Wikipedia contributors. Rodrigues' rotation formula. `https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula`.

[72] Y.-L. Wu. One pose fits all: A novel kinematic approach to 3d human pose estimation. Master's thesis, Delft University of Technology, Faculty of Mechanical Engineering, 2021. URL `https://resolver.tudelft.nl/uuid:8bcc7171-404e-4bb6-b2e2-e5877f5607d6`. Available online.

[73] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu. Pose flow: Efficient online pose tracking. *arXiv preprint arXiv:1802.00977*, 2018.

[74] Y. Xu, J. Zhang, Q. Zhang, and D. Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, 35: 38571–38584, 2022.

[75] C. Yeung, T. Suzuki, R. Tanaka, Z. Yin, and K. Fujii. Athletepose3d: A benchmark dataset for 3d human pose estimation and kinematic validation in athletic movements. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5945–5956, 2025.

[76] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.

[77] M. Zago, M. Luzzago, T. Marangoni, M. De Cecco, M. Tarabini, and M. Galli. 3d tracking of human motion using visual skeletonization and stereoscopic vision. *Frontiers in bioengineering and biotechnology*, 8:181, 2020.

[78] J. Zhang, Z. Tu, J. Yang, Y. Chen, and J. Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13232–13242, 2022.

[79] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proceedings of the IEEE international conference on computer vision*, pages 2248–2255, 2013.

[80] Y. Zhang, Y. Wang, O. Camps, and M. Sznaier. Key frame proposal network for efficient pose estimation in videos. In *European Conference on Computer Vision*, pages 609–625. Springer, 2020.

[81] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah. Deep learning-based human pose estimation: A survey. *ACM computing surveys*, 56 (1):1–37, 2023.

[82] K. Zhou, L. Zhang, F. Lu, X.-D. Zhou, and Y. Shi. Efficient hierarchical multi-view fusion transformer for 3d human pose estimation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7512–7520, 2023.

[83] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[84] W. Zhu, X. Ma, Z. Liu, L. Liu, W. Wu, and Y. Wang. Motionbert: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15085–15099, 2023.

[85] S. Zhuge, Y. He, X. Xu, S. Gan, C. Li, B. Lin, X. Yang, and X. Zhang. Markerless motion capture for humans through a multi-uav system. *IEEE Transactions on Instrumentation and Measurement*, 73:1–12, 2023.

A | Appendix A

A.1. Drone-Captured Self-Supervised Dataset

To support the self-supervised training stage of Brancher, a dedicated drone-based dataset was collected using a commercial *Parrot Bebop 1* quadcopter. The goal of this acquisition

Figure A.1: The Parrot Bebop 1 drone used for data collection.

was to obtain monocular aerial videos of human motion under realistic drone viewpoints, without relying on ground-truth 3D annotations.

The dataset consists of **698 video sequences**, each with a fixed temporal length of **81 frames**. All videos were recorded at a spatial resolution of $\mathbf{H} = \mathbf{352}$ and $\mathbf{W} = \mathbf{640}$ pixels. This fixed-length design ensures full compatibility with the temporal window used during training.

Data were collected across **five different locations** and involve **five distinct subjects**. The locations include varied outdoor environments, allowing changes in background, illumination, and scene geometry. The subjects were instructed to perform movements that generate highly diverse body configurations, covering a broad range of poses and limb articulations. This variability is crucial to avoid pose bias and to improve generalization under aerial viewpoints.

Importantly, each video focuses on a *single person* captured from a drone perspective, resulting in monocular, single-view pose sequences. No 3D ground-truth annotations were recorded. The dataset was explicitly designed for **self-supervised training**, enabling

the model to adapt to drone-specific viewpoints and motion dynamics without relying on supervised 3D labels.

This drone-captured dataset plays a central role in the second training stage of Brancher, where the network is optimized in a fully self-supervised manner to better generalize to real-world UAV scenarios.

Figure A.2: Some samples from the drone-captured dataset, showcasing the diversity of subjects, poses, and environments.

B | Appendix B

In this appendix, I present additional qualitative results that complement the quantitative findings discussed in the main body of the thesis. These results provide further insights into the behavior of the model under study, and help to illustrate key concepts and phenomena that were observed during my experiments.

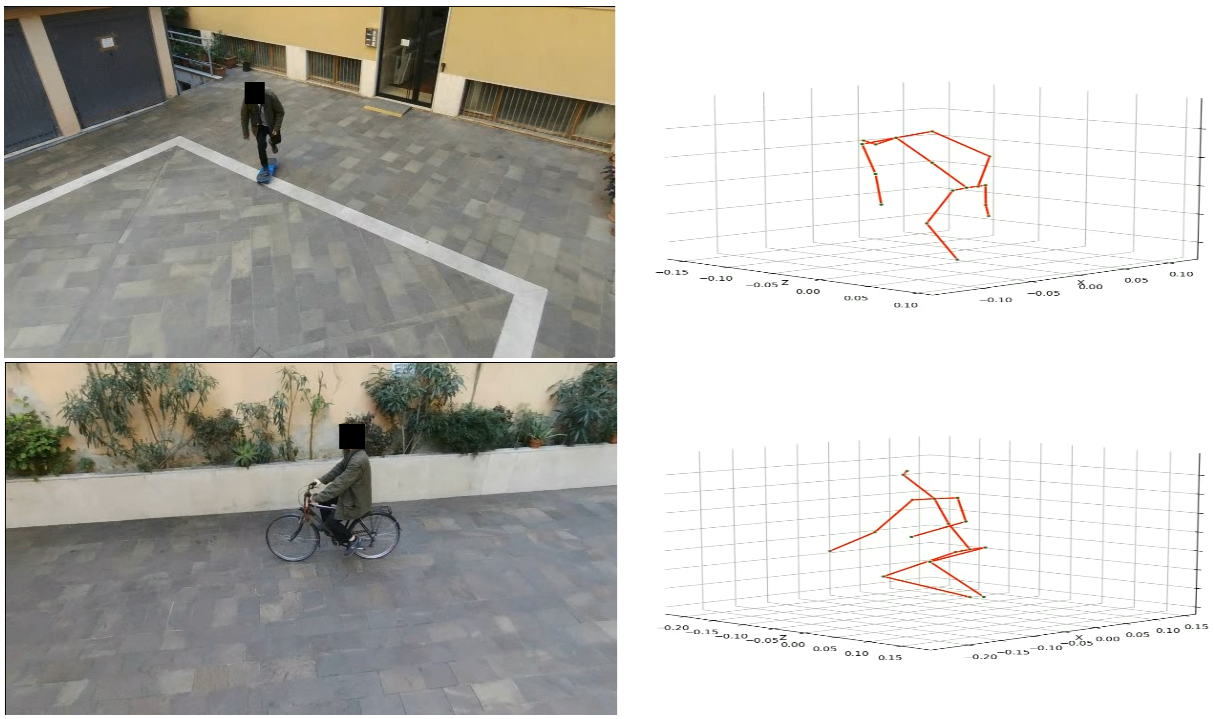

Figure B.1: Some examples where the subject interacts with some objects.

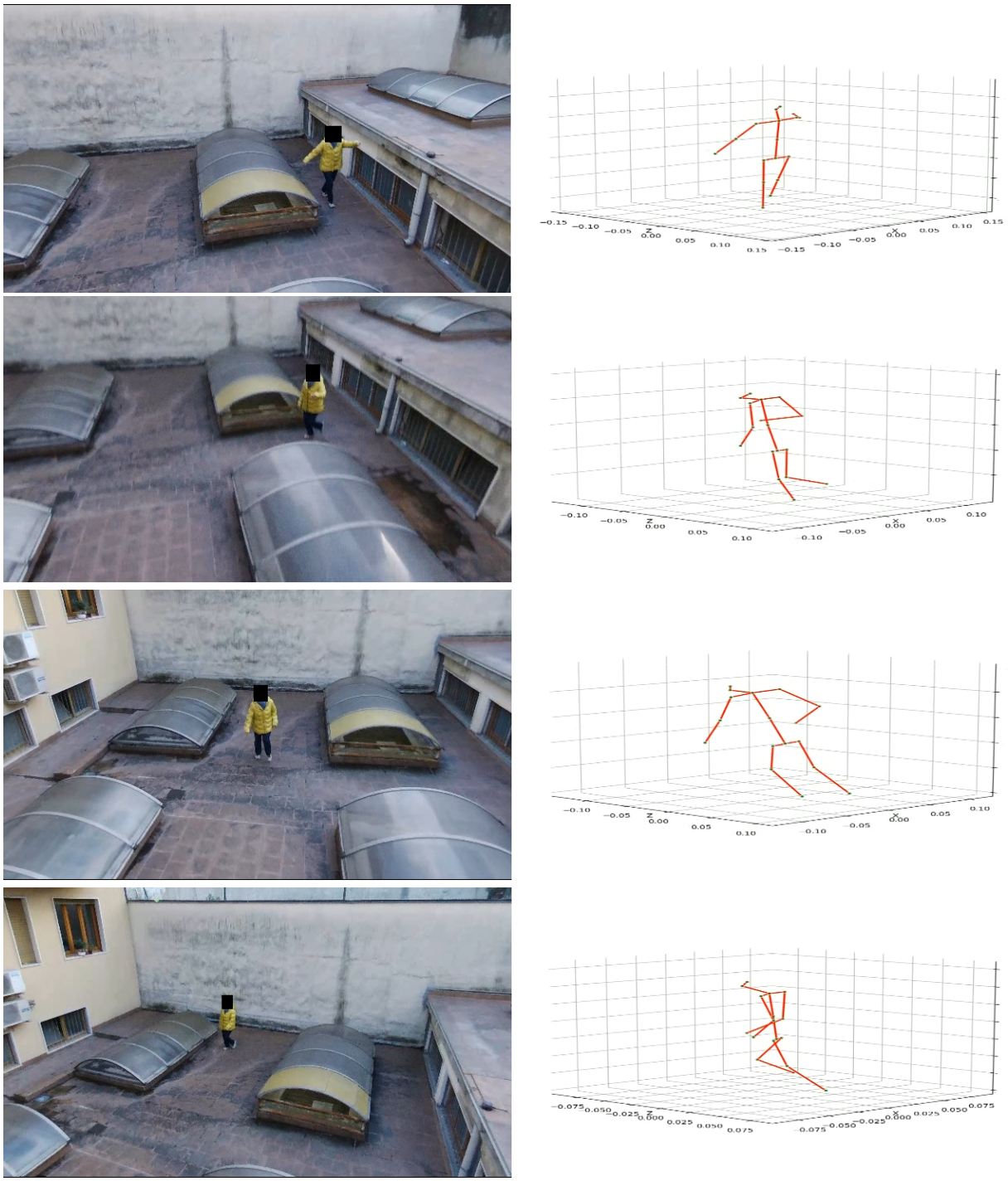

Figure B.2: Some examples from the custom dataset.

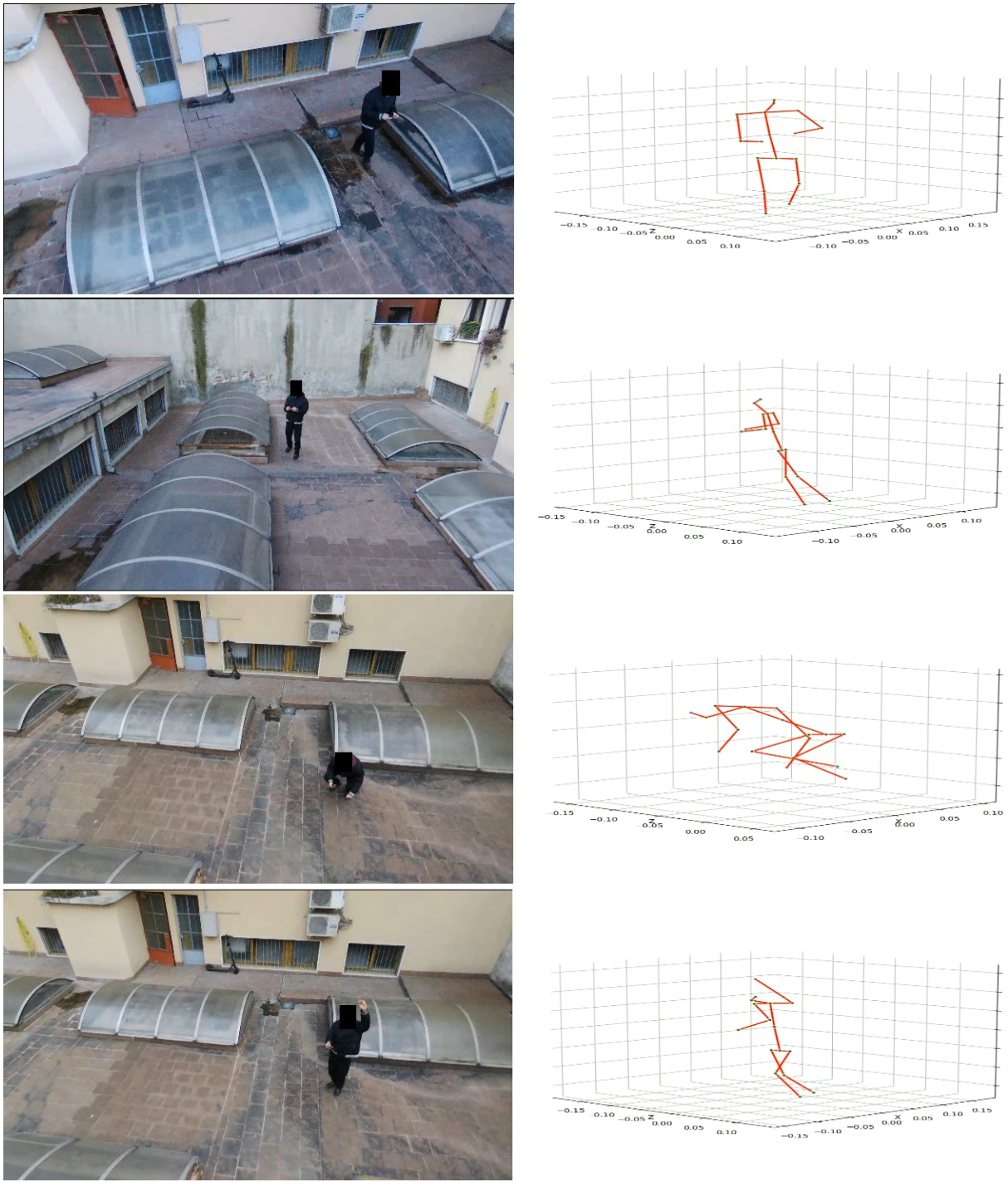

Figure B.3: Some examples from the custom dataset.

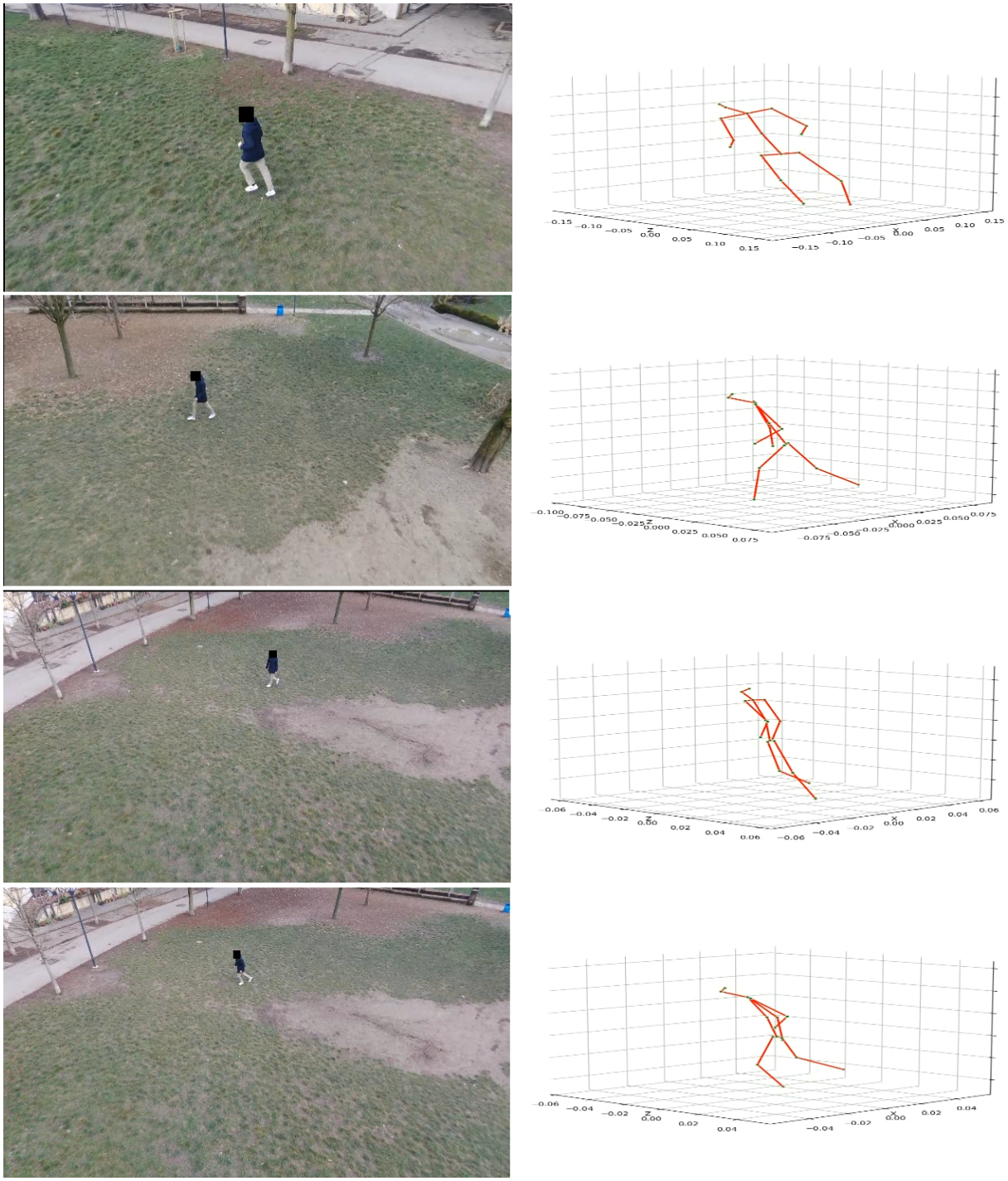

Figure B.4: Some examples from the custom dataset.

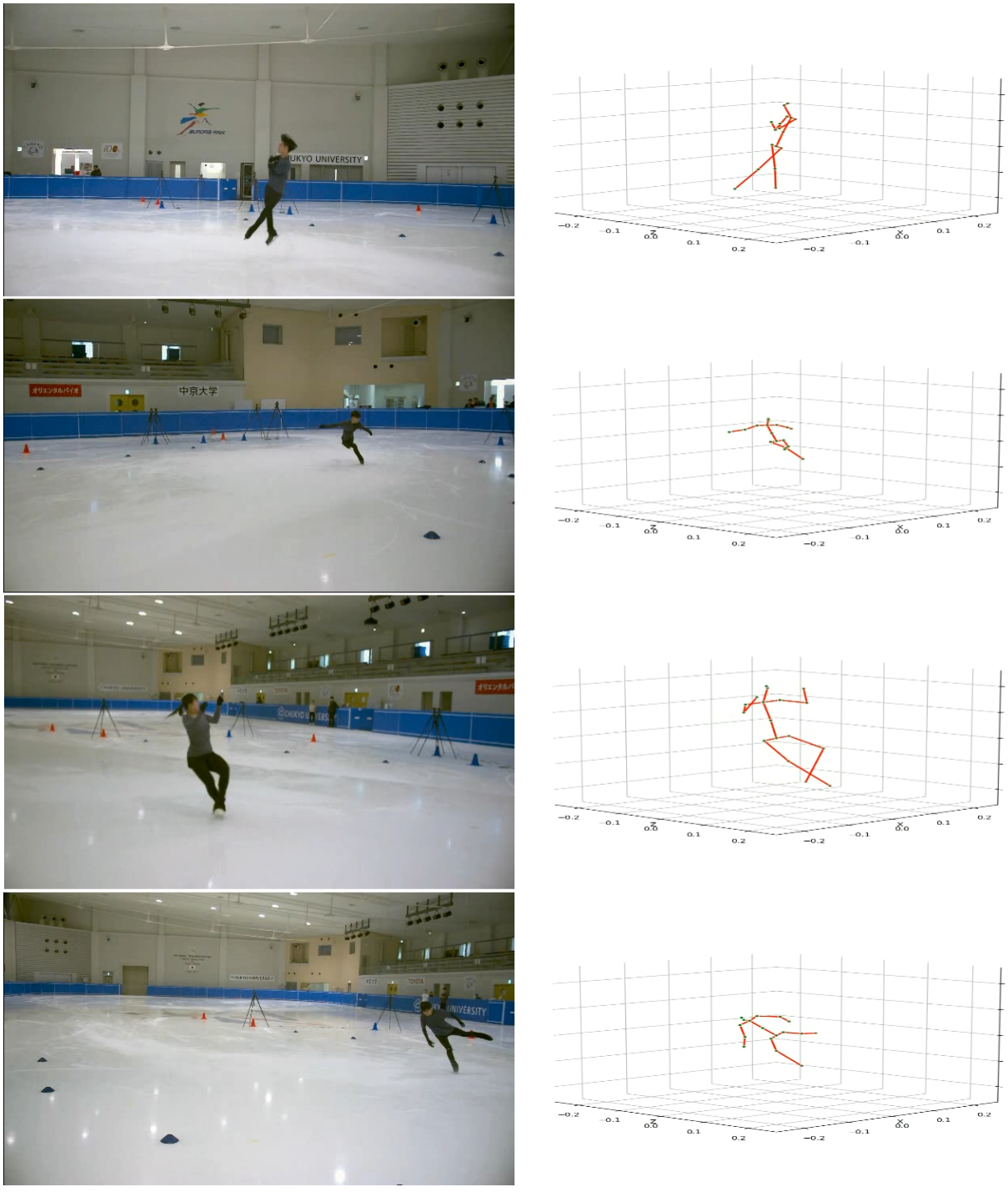

Figure B.5: Some examples of ice skating.

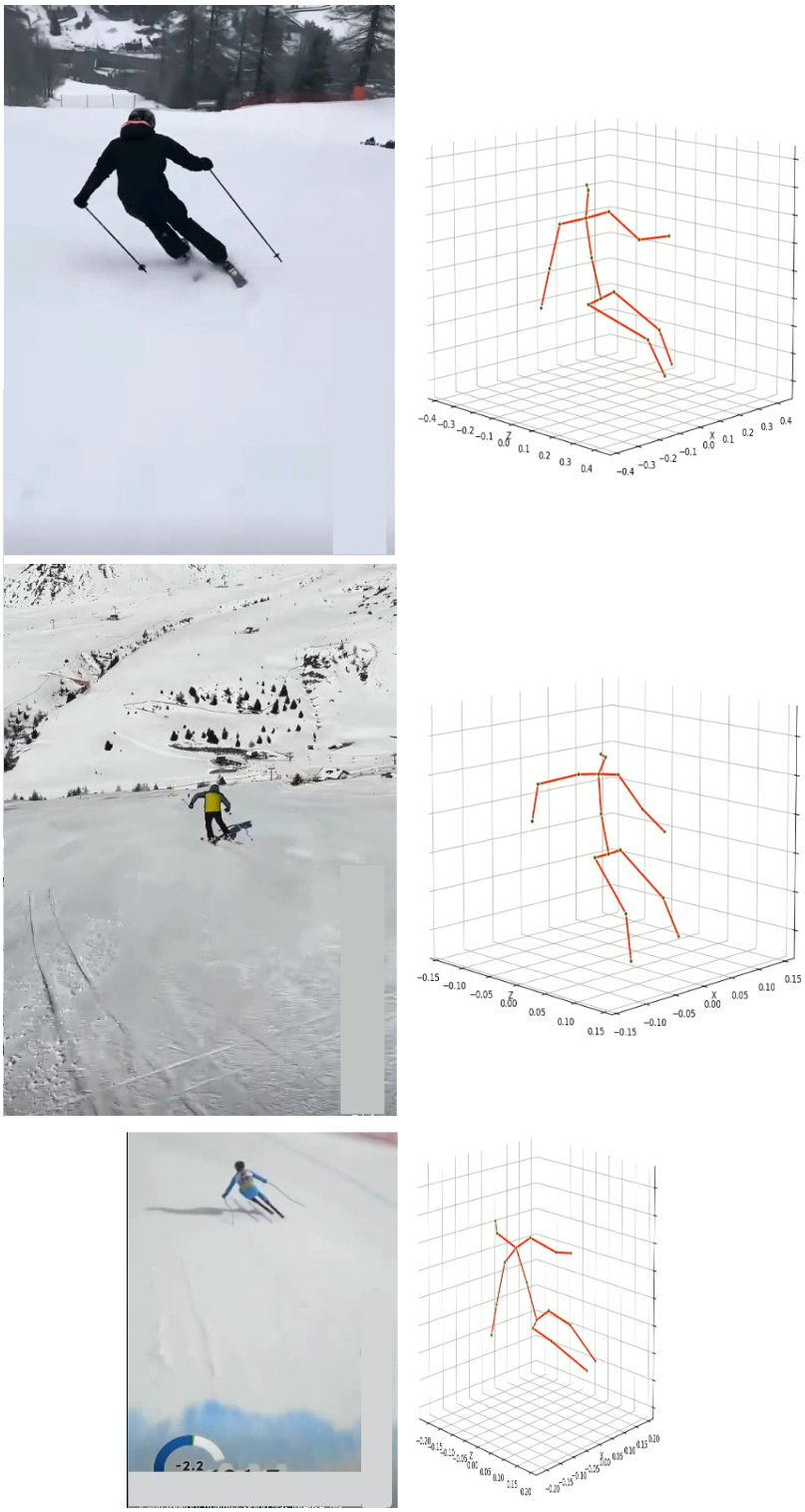

Figure B.6: Some examples of sky.

List of Figures

List of Tables