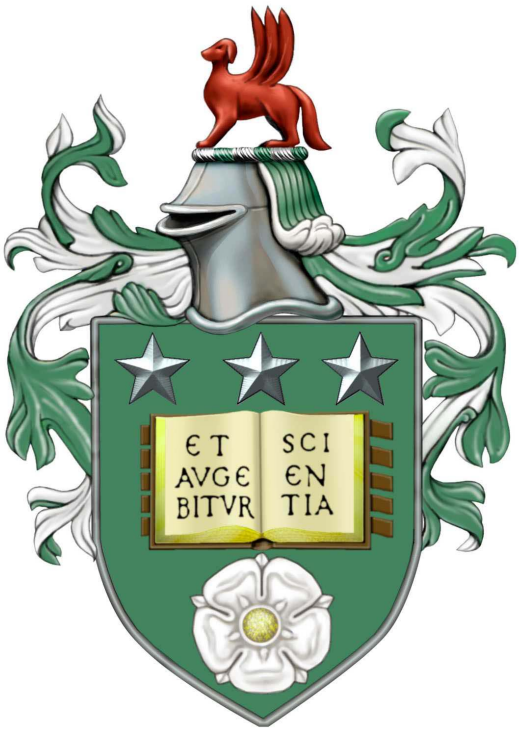
Self-supervised Pose Estimation



Jose Angel Sosa Martinez

The University of Leeds

School of Computing

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

September 2023

Declaration

The candidate confirms that the work submitted is his/her own, except where work which has formed part of a jointly authored publication has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement. The right of Jose Angel Sosa Martinez to be identified as Author of this work has been asserted by Jose Angel Sosa Martinez in accordance with the Copyright, Designs and Patents Act 1988.

©Jose Sosa, The University of Leeds, 2023.

It is seeing which establishes our place in the surrounding world; we explain that world with words, but words can never undo the fact that we are surrounded by it.

John Berger

Acknowledgements

First and foremost, I am immensely grateful to my PhD advisor, David Hogg, for his wise and patient guidance throughout the last four years. David has always motivated me to set high research standards and helped me shape and develop my ideas into realistic and high-quality work. Without his invaluable mentorship and constant support, this thesis and other achievements during my PhD would not have been possible.

During my PhD, I had the privilege of meeting and working with some fantastic people. I especially want to express my gratitude to my lab mates, Rebbeca Stone and Mohammed Alghamdi, for their constant encouragement, interesting research discussions, and enjoyable coffee talks. I will always cherish the time I spent with them.

Thanks to all the other colleagues and friends I met during my PhD journey, who were there to listen and offer encouragement during tough times. Thank you to all my previous mentors and advisors, both in industry and academia, who helped shape me into the person I am today.

Abstract

Human and non-human pose estimation has been studied within the computer vision community for many decades. The progress made within this area has permitted its application to solve multiple tasks, for example, human activity recognition, animal tracking, video surveillance, autonomous driving, and behaviour analysis. Despite the tremendous advancement in developing methods and creating datasets for pose estimation tasks, there remains a lack of tools that work with minimal assumptions about data availability. In other words, most state-of-the-art approaches for pose estimation heavily rely on large datasets containing 2D or 3D annotations used during the training phase. This could make their adaptation to other domains challenging, particularly to the animal domain, where 2D and 3D annotations are scarce.

Throughout the chapters of this thesis, we explore developing and adapting self-supervised deep learning methods for both 2D and 3D pose estimation. Our focus is on creating methods that require minimal or no annotated data for training. This approach provides flexibility in the resulting methods, allowing these to work with diverse skeletal structures with little to no effort in the adaptation process. We start working in this direction by adapting a 2D human pose estimation model to the animal domain. To achieve this, we incorporate a prior of synthetically generated 2D poses, allowing self-supervised training and eliminating the need for manual annotations of input images. We apply this method to explore unlabelled data, as demonstrated by our successful implementation using a dataset of recordings featuring genetically modified mice. Similarly, our proposal in the human domain involves developing a self-supervised method for estimating 3D poses directly from images. Unlike previous works dealing with the same task, our approach requires no 3D annotations for training. Our method builds upon ideas from recent human pose estimation literature and adopts elements from our mice pose estimator. This makes the formulation work with only unlabelled images and an unpaired prior of 2D poses for training. We further experiment with adapting this method to different conditions and body

structures. Ultimately, we demonstrate that it also works well for a different skeletal structure and when utilising a prior of 2D poses generated through synthetic data rather than relying on annotations from existing datasets.

Contents

List of Figures

List of Tables

List of abbreviations

ALS: Amyotrophic Lateral Sclerosis

AUC: Area Under the Curve

B-SOiD: Behavioral Segmentation of Open-field In DeepLabCut

CAD: Computer-Aided Design

CNN: Convolutional Neural Network

CPM: Convolutional Pose Machine

DLC: DeepLabCut

GANs: Generative Adversarial Networks

GCN: Graph Neural Network

GMM: Gaussian Mixture Model

GPU: Graphics Processing Unit

HOG: Histogram of Oriented Gradients

LSP: Leeds Sports Pose

LSTM: Long Short-Term Memory

MPJPE: Mean per Joint Position Error

NF: Normalising Flow

P-MPJPE: Procrustes Mean per Joint Position Error

PCA: Principal Component Analysis

PCK: Percentage of Correct Keypoints

SAM: Segment Anything Model

SMAL: Skinned Multi-Animal Linear Model

SMALR: Skinned Multi-Animal Linear Model with Refinement

SMPL: Skinned Multi-person Linear Model

t-SNE: t-distributed Stochastic Neighbor Embedding

Chapter 1

Introduction

Our ability to see is fundamental, like talking, walking, or touching. We acquire this skill through unconscious self-instruction during childhood, and once our brains develop it, seeing becomes automatic [1]. We can visually perceive things even before describing them adequately with words. Although we cannot fully interpret what we see, it constitutes our visual perception of the surrounding world and gives us a sense of belonging to a particular space. Understanding how our visual system works and how to replicate it within computer systems represents one of computer vision's foundations and has been heavily investigated in recent decades. Thanks to the advances in computer vision and many related areas, we have successfully imitated fundamental concepts of human vision into algorithms. As a result, machines can now efficiently recognise and categorise objects in digital images.

Modern computer vision systems have numerous cross-disciplinary applications [2, 3, 4, 5] that require them to perform more complex tasks than simply recognising and categorising objects in images. One such task is pose estimation, which involves developing methods to understand a given object's overall structure by identifying keypoints. For instance, from an image depicting a person, a machine can recognise where the hands or the head are in the image. Knowing the locations of the body parts provides insight into how an object interacts with its environment or other objects. In addition, it produces a low-dimensional kinematic representation of the object via its body keypoints, which proves beneficial for many applications [6, 7, 8, 9, 10, 11, 12, 13].

The rise of deep learning has led to a surge in 2D and 3D pose estimation methods based on this

paradigm. In particular, supervised deep learning approaches [14, 15, 16] gained popularity and rapidly established benchmarks for pose estimation. However, supervised methods assume that carefully labelled data exists in large amounts. Unfortunately, the labelled data only represents a small fraction of the vast volume of unlabelled images and videos freely available online. The challenge for pose estimation and computer vision is to develop more approaches requiring less direct supervision, such as semi-supervised, self-supervised, and unsupervised methods, to make the most of the abundant unlabelled data.

Significant progress has also been made towards self-supervised deep learning methods for 2D and 3D pose estimation. Nevertheless, several exciting questions remain in this field, some of which motivated the research conducted in this thesis. For example, to what extent can self-supervised deep learning methods learn cues that are not explicitly visible to the human eye on the images? Is it feasible to deduce 3D information solely from 2D observations? How can we learn those 2D observations exclusively from unlabelled data? What are the minimum assumptions for training self-supervised methods to learn 2D and 3D poses?

1.1 Motivation

Human and animal pose estimation has been widely studied from many perspectives since the initial days of computer vision. Early methods commonly utilised hand-crafted features [17, 18, 19, 20] and pictorial structures [21, 22, 23, 24] to represent the body. Afterwards, deep learning methods from object recognition [25] and image classification [26] were adopted to estimate poses. In addition, several datasets containing 2D pose annotations surged in the human domain [27, 28, 29], encouraging progress on supervised 2D human pose estimation [15, 30, 31, 32, 33, 14].

Similarly, supervised deep learning methods are predominant in 2D animal pose estimation, most based on human pose estimation methods. Many labelled datasets [34, 35, 36, 37, 38, 39, 40] and tools [2, 41, 42, 43] to estimate the poses of various animals, such as dogs, mice, and monkeys, are publicly available. Regardless, animal datasets are still behind compared to the enormous amount of labelled data available for humans. Two of the main reason for the discrepancy between domains are the number of different animal species and the difficulty of acquiring 2D and 3D pose data from animals in their typical environment [44]. Considering the lack of annotated data in the animal domain, deep learning methods explore learning from

different resources, such as artificially generated data (synthetic data). Synthetic data is a low-cost alternative to quickly generate data with its respective ground truth annotations. That is probably why most work that estimates poses for animals with synthetic data still follows supervised paradigms [45, 46]. However, it typically requires performing domain adaptation as a post-processing step to adjust models to actual data.

In the context of 3D human pose estimation, many datasets exist [47, 48, 49, 50, 51, 52, 53], including 2D and 3D pose annotations. Unfortunately, a considerable portion of human data needed for critical applications remains unlabelled, representing a challenge for training supervised deep learning methods. While obtaining 2D pose annotations in the image plane is relatively straightforward, annotating 3D pose is more complex and requires consideration during dataset design and acquisition. Therefore, the challenge is not generating annotations but developing better deep-learning methods to learn poses without them. Progress has been made towards this, mainly focusing on weakly supervised methods that use cues such as 2D poses [54, 55, 56], multi-view images [57, 58], volumetric models [59], video segments [60], and 3D kinematic constraints [61].

Our work aims to learn animal and human poses with as few assumptions as possible, as annotated data may not always be available. Ideally, 2D and 3D poses should be learned solely from raw images within a fully unsupervised setting. However, to what extent are deep learning methods capable of achieving this? In this thesis, we seek to develop self-supervised methods that learn to estimate human and animal poses, relying on minimal assumptions about the availability of labelled pose data to supervise the training.

Each chapter tackles the problem from a slightly different context and has particular motivations and objectives. However, in the end, it all contributes towards the primary goal of the thesis. The work presented in this thesis could be summarised within the two following perspectives:

1. **2D and 3D animal pose estimation with artificially generated data.** Recent progress on self-supervised methods for 2D human pose estimation [62, 4] demonstrates how to learn 2D poses via unpaired priors of annotated data. Given the relaxed requirements regarding data, these methods open up the opportunity to apply them to different structures. Although they do not depend on paired data for training, they still need 2D pose annotations for a considerable portion of the data (at least 50%). Assuming that we are using a new and relevant dataset that has been collected for a different analysis out of

computer vision, and therefore it lacks annotations of any type. How can we get 2D pose annotation without manually annotating that portion of the data that we need to train the model?

Synthetic data could be a suitable answer to this question, including publicly available CAD models of animals such as mice [63] and horses [45]. Furthermore, we can build upon existing ideas for estimating 3D human poses [54, 16, 64, 65] and producing 3D animal poses. Unlike most approaches that use synthetic images and annotations for supervised training, we reduce even more assumptions and only utilise part of the synthetic data (i.e. only 2D synthetic poses and no images are required) within self-supervised frameworks.

2. **3D human pose estimation from unlabelled images and 2D prior.** Most of the progress on 3D human pose estimation has been possible thanks to new datasets introducing a higher degree of variation on the poses they contain. The existence of labelled pose data directly benefits supervised methods for pose estimation. However, self-supervised approaches also gained popularity with the premise that removing the dependency on paired pose annotations for training will help to generalise and exploit the vast amount of unlabelled data available.

   Self-supervised methods are still far from learning solely from 2D images, but much progress has been made towards this objective. The dependency on paired pose annotations has been gradually replaced by multi-view images, volumetric models, and 3D kinematic constraints. We explore learning 3D human poses from unlabelled images relying solely on an unrelated small set of 2D poses (not annotations of the training images). Inspired by a method that generates 2D poses from unlabelled images [62], we extend it to estimate 3D poses by adding some elements from existing methods for human 3D pose estimation, such as geometric consistency [54] and normalising flows [55].

1.2 Thesis outline and key contributions

In this thesis, we aim to develop self-supervised methods that learn to estimate poses for humans and other animals under minimal assumptions about the availability of paired pose annotations for training. We heavily rely on unlabelled images and experiment to reduce the need for annotations by adopting small empirical priors generated from freely available synthetic models.

The thesis consists of five chapters; next, we provide a brief introduction to each of these.

In Chapter 2, we review relevant literature related to the problem of human and animal pose estimation. We focus on self-supervised and weakly-supervised methods for 2D animal and 3D human pose estimation. Additionally, the chapter includes a summary of classical and supervised approaches for estimating 2D and 3D poses.

In Chapter 3, we describe a self-supervised method for estimating mice poses in 2D (i.e., in the image plane). Our approach relies on unlabelled images and a prior on 2D poses generated from synthetic data for training. Overall, the main contributions of this chapter are:

- We adapt a self-supervised pose estimator from the human domain to the mouse domain.
- We generate an empirical prior for 2D pose automatically using a synthetic 3D mouse model, thereby avoiding manual annotation of images.
- We demonstrate promising performance in experiments with a new mouse video dataset by comparing pose predictions with ground truth. In addition, we compare pose predictions to those of a widely used state-of-the-art tool based on supervised training.
- We provide a low-dimensional representation of mouse videos in the form of 2D poses from which gait measurements can be made as required for biomedical studies.

In Chapter 4, we introduce a new self-supervised approach for predicting 3D human pose relying on minimal assumptions about the availability of labelled data. Our method simultaneously learns 2D and 3D pose representations in a largely unsupervised fashion, requiring only an empirical prior on unpaired 2D pose. We demonstrate its effectiveness on three of the most popular benchmarks for human pose estimation. We also show our method's adaptability to other articulated structures using a synthetic dataset of human hands. Overall, our method has the following advantages:

- It does not assume any 3D annotations.
- It generalises well to new datasets for human pose estimation, only requiring fine-tuning some components.
- It holds the potential for quickly adapting to 3D pose prediction for other articulated structures (e.g. animals and jointed inanimate objects).

In Chapter 5, we present a self-supervised method that learn to estimate 3D poses for horses. The method only requires unlabelled images depicting horses and a minor prior on 2D pose. We generate the 2D poses from synthetic data to reduce the requirements further. Our formulation represents a step beyond methods for animal pose estimation by having the following benefits:

- It is a straightforward method with few requirements about data availability for training.
- Compared with the enormous datasets used for training human pose estimation methods, our horse data only represents a tiny portion (around 2%).
- Moreover, the small prior of 2D poses used is only one-third of the available images for training.

Finally, in Chapter 6, we wrap up the work presented in this thesis. This chapter summarises the main conclusions and offers a critical review of the limitations of the methods developed throughout the thesis. Additionally, we address some of the remaining challenges and provide insights for future research that can enhance the proposed methods.

1.3 Relevant publications

The work conducted on this thesis is part of the following peer-reviewed publications:

- Chapter 3: **Sosa, J.**, Perry, S., Alty, J., & Hogg, D. (2023, September). Of Mice and Pose: 2D Mouse Pose Estimation from Unlabelled Data and Synthetic Prior. In International Conference on Computer Vision Systems (pp. 125-136). Cham: Springer Nature Switzerland.
- Chapter 3: **Sosa, J.**, Perry, S., Alty, J., & Hogg, D. (2022). Of Mice and Pose: 2D Mouse Pose Estimation from Unlabelled Video Frames using Synthetic Data. In CV4Animals: Computer Vision for Animal Behavior Tracking and Modeling. (Poster)
- Chapter 4: **Sosa, J.**, & Hogg, D. (2023). Self-supervised 3D Human Pose Estimation from a Single Image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.
- Chapter 5: **Sosa, J.**, & Hogg, D. (2023). A Horse with no Labels: Self-Supervised Horse Pose Estimation from Unlabelled Images and Synthetic Prior. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops.

Some parts of this thesis have also been presented at different meetings:

- Chapter 4: **Sosa, J.**, & Hogg, D. (2022). 3D Human Body Pose Estimation From a Single Image. The British Machine Vision Association (BMVA) Symposium, 2022.

Chapter 2

Related work

Pose estimation has been extensively studied since the early days of computer vision [66, 24]. Given an input image depicting an object (e.g. human, mouse, or dog), pose estimation aims to infer the spatial location of different body keypoints or joint positions in 2D or 3D. Locating human or animal body keypoints on images is challenging because of various factors, for example, occlusions, illumination conditions, clothing, and no visible landmarks of the precise joint location in the image.

The importance of pose estimation resides in its multiple applications, whether it involves humans or other animals. For example, human pose estimation has been successfully applied to action recognition [6, 7], video surveillance [67, 68], sports analysis [8, 9, 10], sign language [69, 70, 71], animation [11, 12], human tracking [72], assisted living [73, 74, 75], human-computer interaction [76], human-robot interaction [77], gaming [13], and virtual reality [78].

Estimating the poses of different non-human structures, like animals, has also gained increasing attention because of research applications in many disciplines, including biology, zoology, ecology, biomechanics, and neuroscience [79, 80, 81, 82, 83, 84, 85, 86, 87]. The variability of animal species and the need for species-specific labelled datasets makes animal pose estimation particularly challenging. Compared with the human domain, animal pose estimation is still relatively underexplored. Nevertheless, much effort has gone into developing and adapting deep learning models to estimate animal pose by leveraging similarities between many animal species.

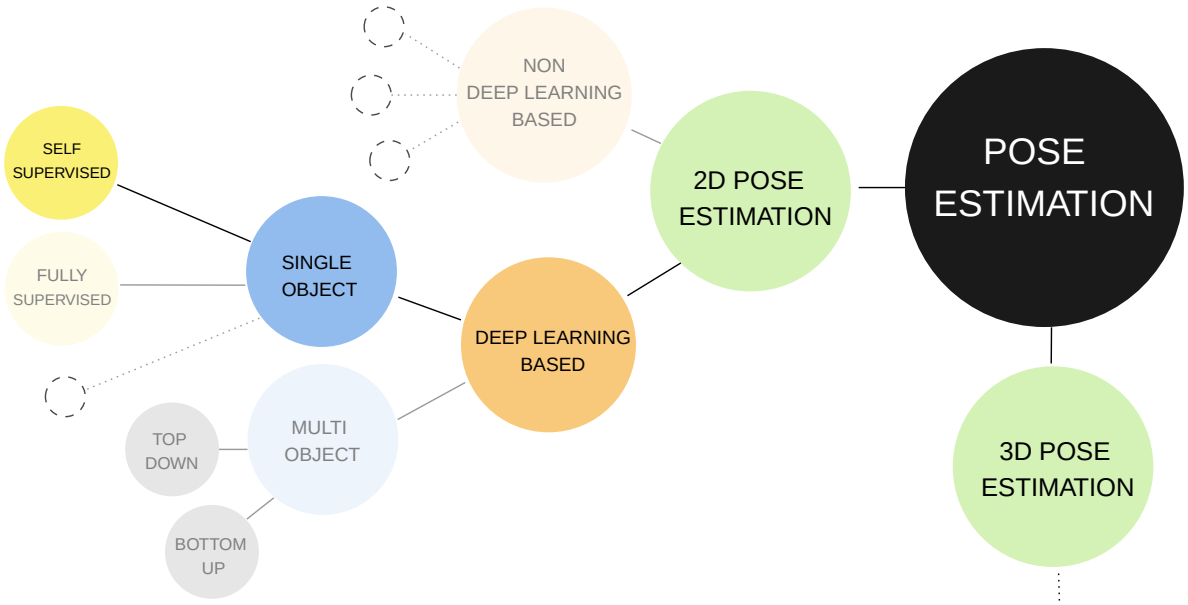


Figure 2.1: Proposed taxonomy to summarise the approaches for 2D pose estimation. We highlight the categories more related to our work.

Several surveys on pose estimation [88, 89, 90, 91, 44], particularly those related to human pose estimation, suggest different ways of breaking down and categorising the extensive domain of pose estimation. For example, a pose estimation model could be classified depending on its purpose, i.e. if it estimates 2D or 3D pose. Likewise, it is possible to categorise these given the number of subjects appearing in the input (single-object or multi-object pose estimation). Another perspective is to group the pose estimation methods depending on their adopted paradigm. Commonly this involves two groups, the approaches based on deep learning techniques and the ones that estimate pose via classical computer vision algorithms, e.g. hand crafted features or pictorial structures. Within deep learning-based methods, there are different subcategories based on their training methodology or the level of supervision needed, as illustrated by Figure 2.1.

The problem of pose estimation is closely related to body modelling. A body model representation can help to illustrate the overall structure of the 2D or 3D poses that a particular approach is expected to deliver. Most methods commonly adopt a kinematic structure as a low-dimensional representation of human and animal bodies. This structure typically involves joint locations and their connections (limbs). As shown in Figure 2.2, other approaches consider a more sophisticated representation to capture finer body shape information, employing planar or volumetric body models.

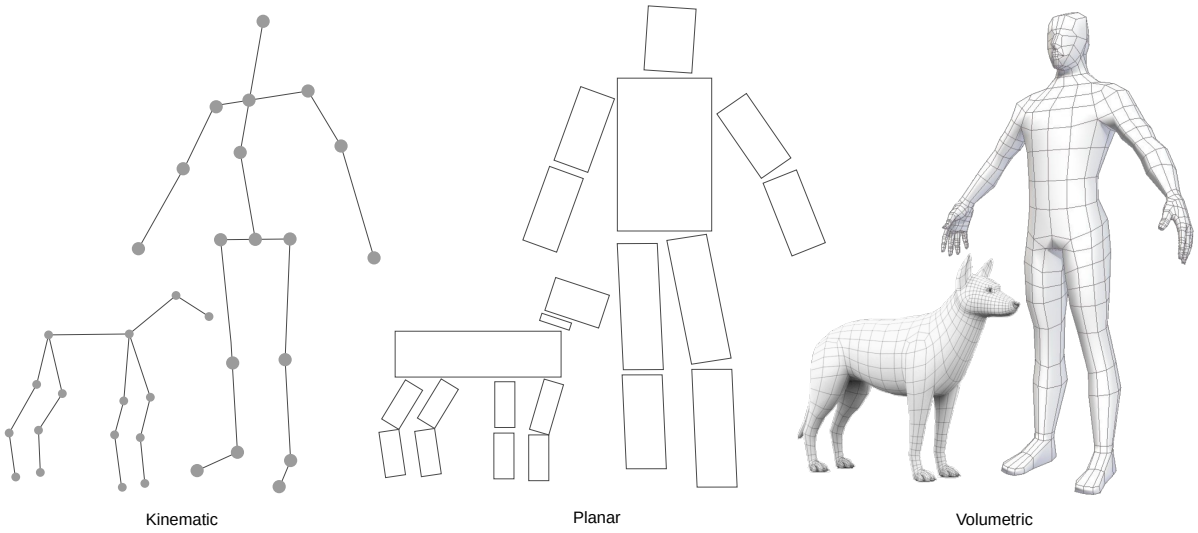


Figure 2.2: Different body models commonly adopted by 2D and 3D pose estimation approaches for humans and animals.

Since not all the existing works for pose estimation are closely related to the goal of this thesis, in this section, we will only focus on reviewing deep learning-based approaches that estimate 2D and 3D poses directly from images containing a single object, whether it is a human or an animal. We specifically focus on weakly-supervised methods for single 2D and 3D animal pose estimation using artificially generated data and image-based weakly supervised methods for single 3D human pose estimation. For convenience, we summarise all the related work on 2D pose estimation in section 2.1 and on 3D pose estimation in section 2.2. Each section reviews supervised and weakly-supervised deep learning approaches regardless of their input domain. However, note that the emphasis is mainly on the weakly-supervised methods and artificially generated data. Fully supervised settings will only be discussed in minor detail to provide more context.

2.1 2D pose estimation

2.1.1 Classical approaches

In the past, before the rise of deep learning approaches, 2D pose estimation methods [22, 23, 92, 93, 24] were commonly based on pictorial structures [21]. The core idea of these methods was to represent an object using a tree structure to express its composing parts and their spatial relationships. Although the pictorial structure framework was straightforward and popular for pose estimation, it has some limitations, such as the high number of parameters of the model

and the consequent difficulty in efficient matching [22]. Therefore, other early approaches surged, incorporating different hand-crafted features and visual cues, such as edges, silhouettes, part-templates, and Histograms of Oriented Gradients (HOG) [17, 18, 19, 20, 94, 95, 96, 97].

2.1.2 Deep learning approaches

In recent years, deep learning has gained significant popularity for its effectiveness in solving complex computer vision problems, including object detection [25] and image classification [26]. Consequently, other computer vision tasks, such as human pose estimation, have also adopted deep learning techniques. In this domain, deep learning has significantly reduced the need for manually designed body structures and labour-intensive feature engineering.

Early deep learning methods for human pose estimation [15, 30, 31, 98] solve the problem by mapping the input image to body joint coordinates, i.e. they directly produce the (x, y) positions in the image plane for a given body part. Most of these approaches adopted popular architectures for image classification and object detection as a backbone, for example, AlexNet [26], GoogleNet [99], and ResNet [100]. Since direct regression of body joint coordinates does not provide enough robustness for estimating pose, the following methods adopt a detection-based formulation, i.e. those methods define the body parts as targets for detection and typically represent them as image patches or heat maps [32, 33, 14]. In particular, a heat map represents the joint location as a probability distribution, providing more dense pixel information, which enhances the method's robustness.

New network architectures for pose estimation, such as Convolutional Pose Machines (CPM) [101], HRNet [102], and Stacked Hourglass networks [14], were inspired by detection-based approaches. The design of the Stacked Hourglass network, in particular, has gained popularity and has been widely used as a backbone for several other pose estimation methods [103, 104, 105]. The core idea of the Stacked Hourglass Network comes from fully convolutional networks [106] and other approaches that process features at different scales [32]. The hourglass design permits to consolidate information across scales of the image, which allows to better capture the relationships between the body parts.

In the context of 2D animal pose estimation, early deep learning methods that have proven to work for the human domain were adapted to estimate pose from different animal species, like farm animals [107, 108, 109, 110], dogs [36], mice [2], and monkeys [111, 112, 113]. The

foundations of some of the most popular deep learning based tools for animal pose estimation, such as DeepLabCut [2], LEAP [41], DeepPoseKit [42], and OptiFlex [43] are built on methods initially designed to estimate human pose [114, 115, 116, 14, 117]. A common attribute of these approaches is their requirement for full supervision, meaning that they need ground truth 2D pose data for training. While 2D pose data is widely available, and 2D pose estimators are very mature, some datasets, especially in the animal domain, still lack annotations. This premise became the central challenge for subsequent pose estimation methods and raised the interest in learning accurate 2D poses without having access to actual pose annotations data for training.

2.1.3 Learning 2D poses from artificially generated data

New computer vision techniques, such as Generative Adversarial Networks (GANs) [118], allow for improved training procedures for pose estimation methods. Incorporating priors on unpaired [119, 4, 65] or synthetic data [120] can make these approaches even more powerful and less reliant on fully labelled datasets. For instance, Jakab et al. [62] proposes a method that effectively learns to estimate 2D poses without requiring paired 2D pose annotations. It relies on CycleGAN-based training [121] and a generated image-based representation of the pose that serves as an intermediate representation for self-supervised training. Similarly, Schmidtke et al. [4] adopts a CycleGAN-based training but incorporates shape templates as an intermediate representation instead of an image.

Another relevant addition to the pose estimation methods is the use of synthetic data either as an intermediate representation or as a mechanism for domain adaptation. This easily generated data permits adjusting methods trained with 2D pose annotations and images from Computer-Aided Design (CAD) models to actual examples, as shown in [45, 46]. Overall, the use of synthetic data hugely benefits pose estimation. The advancements in artificial object modelling encourage the development of new techniques and the adaptation of existing approaches to different skeletal structures. Furthermore, synthetic data reduces the dependency on fully annotated datasets, allowing the exploitation of abundant unlabelled data, particularly in the animal domain.

2.2 3D pose estimation

2.2.1 Overview

It is relatively straightforward for humans to get a sense of the objects' overall 3D pose and shape solely from 2D observations. When looking at a picture, we can immediately recognise and understand subtle relationships between the objects and easily translate that into 3D space. It has also been studied how this 3D perception of 2D objects from photographs (pictorial space [122]) matches the 3D space where we exist and move [123]. Since computers intrinsically lack that understanding of space, predicting the locations in 3D from 2D observations is more challenging than learning them directly in the 2D plane. This is especially true for pose estimation problems, where it is harder for machines to learn the ambiguous locations of 3D body joints than the 2D poses on the image plane.

The study of estimating 3D pose has been ongoing in computer vision for a long time, just like 2D pose estimation. Before deep learning, most methods for inferring 3D pose were based on image features, such as edges, silhouettes [124, 125, 126, 127], and joint positions [128]. Since then, much work has been carried out, promoting the field's growth. However, three key factors have considerably impacted the development of 3D pose estimation for both humans and animals:

1. The adoption of deep learning to solve most of the computer vision tasks.
2. The surge in the creation of new datasets with 3D annotations.
3. The development and use of synthetically generated volumetric models of human and animal bodies.

In summarising the vast number of 3D pose estimation approaches, we rely on established taxonomies from the literature [91, 129, 130]. Note that the goal is not to propose a new taxonomy but rather to provide readers with a packed visual resource to get the overall idea behind the more related approaches to the work of this thesis. The classification for 3D pose estimation illustrated in Figure 2.3 is similar to that of 2D poses shown in Figure 2.1, with methods organised according to their paradigm: deep learning based or classical approaches. Additionally, there are methods for single-subject and multi-subject. Within the single-object methods, we can distinguish three groups of approaches given their input: learning 3D poses

directly from images, learning 3D poses from already available 2D poses (lifting 2D poses), and methods that rely on volumetric models like SMPL [131] for 3D pose estimation. Furthermore, these methods may also present different levels of supervision during training, such as fully supervised and self-supervised.

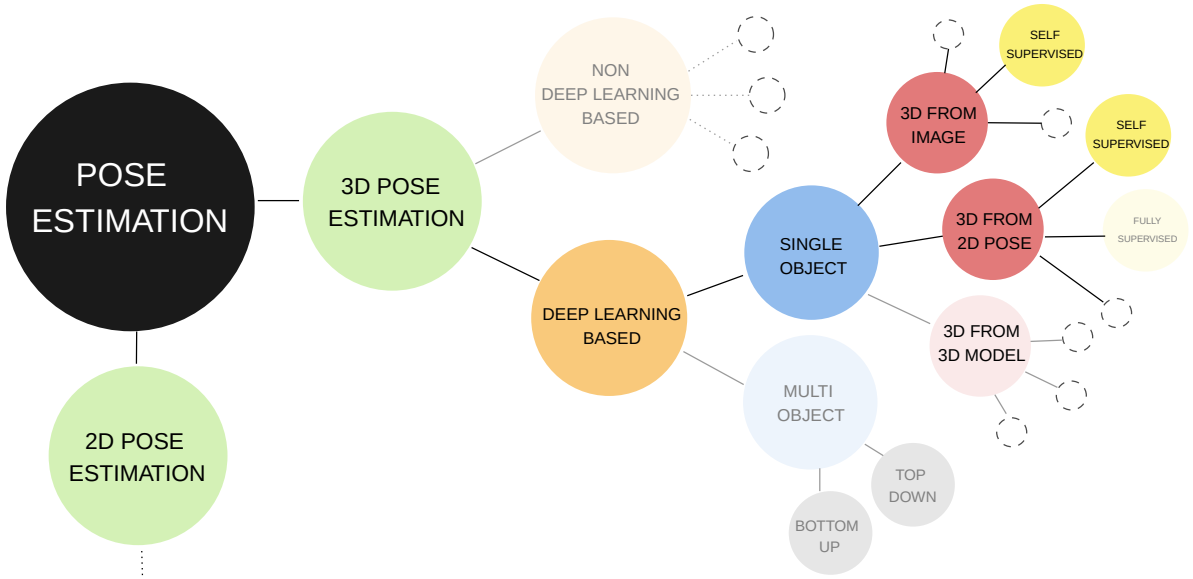


Figure 2.3: Proposed taxonomy to summarise the approaches for 3D pose estimation. We highlight the categories more related to our work.

The rest of this section summarises the available human and animal datasets used for 3D pose estimation. Furthermore, according to our primary purpose, we review self-supervised and other weakly-supervised methods that estimate 3D poses from 2D poses, and even more related to this work, the ones that estimate 3D poses directly from images. In addition, we briefly mention methods that rely on volumetric models.

2.2.2 Publicly available datasets for 3D pose estimation

In the particular context of datasets used for estimating 3D poses, many of them are concentrated on the human domain. Examples of such datasets include Human3.6M [47], CMU Panoptic [48], MPI-INF-3DHP [49], SURREAL [50], 3DPW [51], HumanEva [52], and AMASS [53]. These datasets contain millions of images, each with corresponding 3D pose annotations.

Not surprisingly, while analysing the availability of public datasets for 3D animal pose estimation, we notice a considerable discrepancy with respect to the human data available. According to Table 2.1, most animal datasets only contain some annotations on the image plane, such as 2D pose labels, segmentation masks, and bounding boxes, directly benefiting supervised 2D

pose estimation methods. This lack of 3D pose annotations for fully supervising 3D pose estimation methods has motivated the development of diverse training procedures using cues other than 3D pose labels. At the same time, the scarcity of annotated animal datasets for testing deep learning methods might be why most of the progress is within the human domain, where there are plenty of annotations to evaluate performance properly.

Year	**Animal type**	**Dataset**	2D Pose	3D Pose/Mesh
2023	Primates	OpenMonkeyChallenge [34]	✓	×
2021	Mice	Rat7M [132]	✓	✓
2021	Horses	Horse-10 Dataset [35]	✓	×
2021	Cheetahs	AcinoSet [133]	✓	✓
2020	Dogs	StanfordExtra [36]	✓	×
2019	Grevy's zebra	Synthetic Grevy's Zebra Dataset [37]	✓	×
2019	Diverse species	Animal Pose [38]	✓	×
2018	Diverse species	BADJA[39]	✓	×
2017	Diverse species	TigDog Dataset [134]	✓	×
2011	Dogs	Stanford Dogs[40]	✓	×
2004	Horses	Weizmann dataset[135]	×	×

Table 2.1: Publicly available animal datasets. List of some freely available datasets for estimating 2D and 3D animal poses. While most datasets include 2D pose annotations, 3D pose labels are scarce.

2.2.3 Learning 3D poses from 2D poses

Direct supervision

Based on the availability of many 3D pose datasets (in particular human datasets), most approaches for lifting 2D poses to 3D rely heavily on the assumption that 3D and 2D pose annotations are available to supervise training. Broadly, given a 2D pose as input, these methods learn to lift it to 3D space, i.e. for each body joint location (x, y) in the image plane, the depth z should be estimated. One of the first deep learning approaches that successfully lifted 2D poses to 3D is the work of Martinez [16]. They achieve state-of-the-art results on 3D pose estimation by using a simple network design of fully connected layers and residual blocks. Other methods incorporate different intermediate representations for the 2D poses, e.g., 2D heat-maps [136, 137], volumetric heat-maps (HEMlets) [138], and knowledge of the body configuration in the form of grammars [139]. Approaches relying on kinematic constraints [140, 141], Graph Convolutional Networks (GCN) [142], ranking networks [143], and grid convolutions [144] also have been successfully applied to lifting 2D poses to 3D.

Weaker supervision

Although supervised approaches are predominant, progress has also been made on methods relying on lower levels of supervision. Interestingly, several of these approaches [54, 56, 55] still incorporate ideas from supervised learning within their processes. For instance, Drover et al. [145] propose a weakly supervised approach to lifting 2D poses to 3D based on the work of Martinez et al. [16]. However, [145] reduces supervision by incorporating a GAN loss to assess the realism of the 2D projection of the estimated 3D pose. Later, Chen et al. [54] further develop this work by adding a symmetrical pipeline that involves a series of consecutive transformations (lifting, rotation, and projection) of the estimated 3D pose. This cycle of transformations helps to self-supervise the training while removing the dependency on any 3D correspondences. More recently, Wandt [55] incorporated two fundamental elements to the model in [54] that increase the performance of the 3D lifting process: normalising flows and a learned elevation angle for the 3D rotations. Previous methods have successfully use normalising flow to estimate 3D prior distributions given 3D human poses [146]. However, the method in [55] is the first to perform this task from 2D data.

A vast amount of data is available for pose estimation methods, but much of it lacks the necessary ground truth pose annotations for fully supervised training. While the assumption of not having 2D pose annotations might seem unlikely, it is a reality for many datasets, particularly in the animal domain. Nevertheless, weakly supervised methods have made progress using this unlabelled data (at least lacking 3D annotations), and they have demonstrated that it is possible to learn to estimate 3D poses from existing 2D poses by taking advantage of cues like rotations [54, 145, 56], normalising flows [55, 146], multi-view [147], and kinematic constraints. However, what will happen if we do not assume the availability of paired 2D poses? Is it still possible to learn 3D poses? What are the minimum assumptions to make this possible?

2.2.4 Learning 3D poses directly from images

Direct supervision

Instead of estimating 3D pose from an input 2D pose, other methods directly estimate them from images. Most of these methods assume the availability of ground truth 3D poses for supervision. For instance, the work of [148] incorporates ideas from supervised 2D pose estimation techniques. Specifically, they extend the well-known stacked hourglass network architecture

[14] to learn a volumetric pose. Other methods rely on end-to-end frameworks combining body joints localisation and regression, like [149], which employs a multi-task strategy to train a detection network jointly with a regression network to predict 3D poses from images. Similarly, [150] adopts a multi-task CNN to combine visual features and probability maps. Additional approaches tackle the 3D estimation problem from a different perspective. For example, [98] replaces the traditional body joint regression with a structure-aware regression, i.e. exploiting the connectivity between body joints (bones).

Weaker supervision

Because of the massive amount of unlabelled data publicly available, methods for 3D pose estimation from images should be able to learn with the minimum premises, ideally, just from raw 2D images. In the context of weakly-supervised pose estimation methods that learn from images, much progress has been made towards learning only with few assumptions. However, many approaches still incorporate specific priors for the 2D and 3D joint configuration or even add a small portion of actual 3D data to guide the training. For instance, [151] shows a unified multi-stage CNN architecture to estimate 2D and 3D joint locations from single images. This approach relies on a probabilistic 3D model of the human pose responsible for lifting the 2D representations. Kundu et al. [59] propose a self-supervised architecture to learn 3D poses from unlabelled images. They incorporate three assumptions: human pose articulation constraints, a part-based 2D human puppet model, and unpaired 3D poses. Other approaches explore learning without direct supervision by producing synthetic multi-views of the same skeleton [152], accessing multi-view images or videos [153, 154], relying solely on 3D kinematic constraints [61], or incorporating motion information [155].

2.2.5 Learning 3D poses with artificially generated data

Synthetic data is a cost-effective method to generate large amounts of data to train pose estimation models. Much work has been done in this area, mainly towards developing realistic artificial models of humans and animals. The key idea is to utilise those models to render images with their respective pose annotations, thereby minimising the need for manual labelling. In the human domain, many body models have been proposed [156, 157, 131]. Given its compatibility with existing rendering engines, the parametric shape model SMPL [131] has become widely adopted for 3D pose estimation and human shape reconstruction [158, 159, 160, 161, 162, 163].

In the animal domain, it is impractical to create parametric shape models for every existing animal, principally because of the variability between species and the limited cooperativeness of animals to be scanned. Nonetheless, some efforts have been made to produce body models for a few animal species. For example, Zuffi et al. [164], inspired by the success of SMPL models, propose the Skinned Multi-Animal Linear Model (SMAL). Instead of scanning living animals to learn the model, they use scans from toy figurines of different animal species, focusing on four-legged mammals that share skeletal similarities. Like SMPL, SMAL has also been incorporated into several methods for shape reconstruction and pose estimation [36, 37, 165, 39]. Unfortunately, besides the parametric shape model, these methods often require other types of annotations, such as manually extracting silhouettes and 2D poses.

Like SMAL, CAD models of animals are also helpful in generating data for training pose and shape estimation approaches. For instance, Mu et al. [45] uses CAD models to create synthetic images and their respective annotations to train a supervised pose estimation method focusing on four-legged mammals. Then, they perform unsupervised domain adaptation using small sets of actual data to reduce the domain gap. Regarding CAD models for smaller animals, Bolaños et al. [63] proposes a 3D model for mice, which can generate animations of different mouse behaviours. As mice are commonly used in medical studies [166, 167, 168], the synthetic data generated with this mouse model could be beneficial.

2.3 Summary

The abundance of unlabelled data available on the internet has encouraged the development of new deep learning-based pose estimation methods that require less supervision. Ideally, unsupervised and self-supervised approaches for estimating the poses of humans and other animals should learn solely on unlabelled data. However, this remains a distant reality. As reviewed in previous sections, several self-supervised pose estimation approaches incorporate various mechanisms, such as image generation processes, geometric constraints, and synthetic priors, to minimise the reliance on pose annotations during training. Nevertheless, it is still insufficient to eliminate the need for such annotations entirely.

Our work aims to develop and explore self-supervised deep-learning methods for estimating the poses of humans and other animals. Our approaches build upon the fact that annotated data may not always be available. Therefore we seek to exploit unlabelled images and priors of

existing and easily accessible data, such as data generated from synthetic models. By minimising the assumptions about the availability of labelled pose data to supervise training, we increase the flexibility of our methods and make them applicable to different structures without requiring too much adaptation effort.

Chapter 3

Learning to predict 2D animal pose from unlabelled images and a synthetic prior

Numerous fields, such as ecology, biology, and neuroscience, use animal recordings to track and measure animal behaviour. Over time, a significant volume of such data has been produced, but most computer vision techniques cannot explore it due to the lack of annotations. To address this, we propose an approach for learning to estimate 2D mouse body pose, relying solely on unlabelled images and a synthetically generated empirical pose prior. Our proposal is based on a recent method for estimating 2D human pose from single images utilising a GAN architecture and a set of unpaired typical 2D poses (configurations of 2D image-based joint positions) in training. We adapt this method to the limb structure of the mouse and generate the empirical prior of 2D poses from a synthetic 3D mouse model, thereby avoiding manual annotation. In experiments on a new mouse video dataset, we evaluate the performance of the approach by comparing pose predictions to a manually obtained ground truth. We also compare predictions with those from a supervised state-of-the-art method for animal pose estimation. The latter evaluation indicates promising results despite the lack of paired training data. Furthermore, we demonstrate that our approach holds the potential to be easily adapted to a different animal body structure.

3.1 Overview

The investigation of neurodegenerative human diseases, such as Alzheimer's disease [169, 170], Parkinson's disease [171], and Amyotrophic Lateral Sclerosis (ALS) [172], typically requires the use of animal models. ALS, in particular, represents a slowly progressive neurological disorder that significantly impacts motor function. The identification of alterations in motor abilities during the early stages of the disease is critical for identifying effective therapeutic targets [173]. Mice represent the preferred and most extensively used animal models for such studies, given their genomic similarity with humans and the accumulated knowledge on manipulating their DNA [174]. Prior research in the field has highlighted the value of gait analysis in mice as an effective means to identify subtle changes in the motor system related to ALS [175, 176]. Thus, the development of tools to observe, describe, and measure mouse gait has become indispensable due to the tight relationship between mice and ongoing research on this neurodegenerative human disease [86].

Some years ago, prior to the adoption of computer vision techniques, making the measurements needed for gait analysis meant considerable manual labour [177, 178]. For example, if someone wanted to measure the position of the mouse's limbs, it implies recording the animal, looking at each video frame, and manually identifying each required body part. Then, it is evident that manual inspections on large videos can be time-consuming and lead to observation errors. Early computational approaches attempt to minimise human intervention in analysing animal recordings. Some tools involve placing physical markers on the animal's body or require painting the body parts to track [179, 180]. Apparent limitations of these techniques are that the physical markers can interfere with the animal's behaviour, and the information that can be extracted is inherently limited by the positioning of the markers or the painted areas. Other approaches require costly and sophisticated equipment like infrared systems and high-speed videography to obtain data, resulting in expensive experiments and difficulties in deployment and replication [181, 182, 183].

Newer computer vision tools for tracking animals' body parts[1,2] become less dependent on physical markers. Unfortunately, these tools still needed considerable human intervention for pre-processing and post-processing video data. Supervised deep learning approaches have re-

[1] https://mousespecifics.com/digigait/
[2] https://www.noldus.com/catwalk-xt

cently become state-of-the-art for pose estimation and tracking of humans and animals [2, 41, 42]. Most of these techniques' performance depends on the amount and variability of annotated data for training, which is hard to obtain for some animal species. Thus, there remains an urgent need to develop methods for tracking animal pose that require minimal human effort in training for a new animal domain and operational use. This can be achieved by reducing the need for manual pose annotation of images.

In this chapter, we tackle the challenging task of learning to predict 2D mouse poses in unlabelled images. Different from previous deep learning approaches that generally rely on fully supervised frameworks, we adopt a self-supervised 2D pose estimator from the human domain. This method utilises a modified cyclic-GAN architecture to learn 2D human poses. During training, it assumes the availability of unlabelled images and an unpaired prior of 2D pose annotations, obtained from the same dataset. Our proposal relaxes much more the assumptions about data by building the empirical prior from synthetic 2D poses generated from a 3D model of a generic mouse. Evidently, incorporating synthetic data provides more flexibility to train the model with unlabelled datasets, which is common for many animal recordings outside of computer vision.

3.2 Background

3.2.1 Deep learning methods for animal pose estimation

Analogous to the definition of human pose estimation [184], animal pose estimation refers to the task of estimating the geometrical configuration of body parts of an animal. This problem has gained increasing attention because of research applications in many different disciplines, including biology, zoology, ecology, biomechanics [83] and neuroscience [86]. Compared with human pose estimation, it is still somewhat under-explored, mainly because of the variability of animal species and the need for species-specific labelled datasets. Nevertheless, a lot of effort has gone into developing and adapting deep learning models to estimate 2D and 3D animal pose, exploiting similarities between species. For example, monkeys [113, 112, 111] share similar skeletal structures with humans. Large quadrupeds, such as farm animals [107, 108, 110, 109] and dogs [36, 185, 186] also present similarities between their skeletal forms.

Automatic 2D pose estimation has also been applied successfully on smaller animal species such

as mice. As with larger animals, deep learning methods for pose estimation have been based mostly on supervised approaches developed for the human domain. Their performance is therefore limited by the availability and correctness of annotated data. For example, DeepLabCut (DLC) [2] adapts a pretrained ResNet with deconvolutional layers [114] to estimate the 2D pose of small animals under laboratory conditions, such as mice and flies. LEAP [41] uses an earlier model from the human pose estimation domain [32] to solve the same task. DeepPoseKit [42] employs a similar method as [41] to estimate 2D animal pose. Specifically, it uses a network architecture that improves the processing speed based on fully convolutional densenets [115, 116] and stacked hourglass modules [14]. More recently, OptiFlex [43] exploits the temporal information in video data by incorporating flowing convnets [117] into their network architecture. They report similar performance to previous methods [2, 41, 42] on estimating the pose of small animals, e.g. mice, fruit flies, and zebrafish.

Perhaps the most popular of these approaches is DeepLabCut. Many subsequent methods adopt it to estimate not only mouse pose, but also pose for a wide variety of other animal species [187, 188, 189, 190, 191, 192, 193]. A common feature of DeepLabCut, DeepPoseKit, LEAP, and OptiFlex is their reliance on manual annotation of pose in multiple video frames for training. Even though they normally provide a graphical user interface (GUI) for doing the annotation, the process is still time consuming, error prone, and requires specialised knowledge to infer pose correctly. Futhermore, the number of frames to annotate for good generalisation is hard to predict and therefore ultimately determined empirically. In contrast, through adapting a recent self-supervised approach from the human domain, we completely remove the need for manual annotation, making training and testing more straightforward.

3.2.2 Animal pose estimation with synthetic data

One alternative to avoid manual annotation for training deep learning methods for animal pose estimation is the use of synthetic data. Using an artificial animal model allows producing many synthetic images and their corresponding annotations with less time and effort than manually annotating actual data [63]. In this context, Mu et al. [45] proposes a semi-supervised pose-estimation framework trained in a supervised fashion using synthetically rendered images and ground truth pose annotations from 3D CAD (Computer-aided design) models. Then, they perform self-supervised domain adaption with a small portion of actual data to minimise the domain gap. They successfully estimate 2D poses for large animals with similar skeletal

structures, such as tigers, horses, and dogs. Some other works relying on synthetic data also focus on the domain adaptation process after learning the animal pose with synthetic data under supervised paradigms [46, 194].

We adopt a related approach using an existing 3D geometric mouse model [63], except that we do not use rendered images and require only synthetic 2D poses as a prior, i.e. we are not using synthetic images as in supervised settings. Furthermore, we use this prior on 2D poses within a GAN framework that allows our whole model to learn poses not necessarily appearing in the prior, eliminating the need for domain adaptation as in [45, 46, 194].

Synthetic data also plays a significant role in learning more complex forms of 3D animal pose. For instance, Zuffi et al. [164], inspired by the success of human shape models like SMPL [131], create toy figurines of various animals to generate data for learning statistical shape models (SMAL). Later, [165] propose SMALR, which is an extension of the previous SMAL model. It introduces a regularisation for the deformation of the animal shape to make it appear more detailed and realistic. Subsequent work [36, 195, 37] has adapted the SMAL model to particular animal species like dogs and zebras.

In contrast to learning to fit 3D shape models from 3D scans, other approaches explore the possibility of learning 3D animal models from less complex representations, like multi-view 2D images, or user-clicked 2D images [196, 197, 198]. However, the final shape representation of those models is less realistic and detailed than those produced using SMAL or SMALR. These methods have produced 3D shape models for various animal species, typically focused on large quadrupeds like tigers, dogs, and zebras. Unfortunately, creating sophisticated models for all animal species is still impractical.

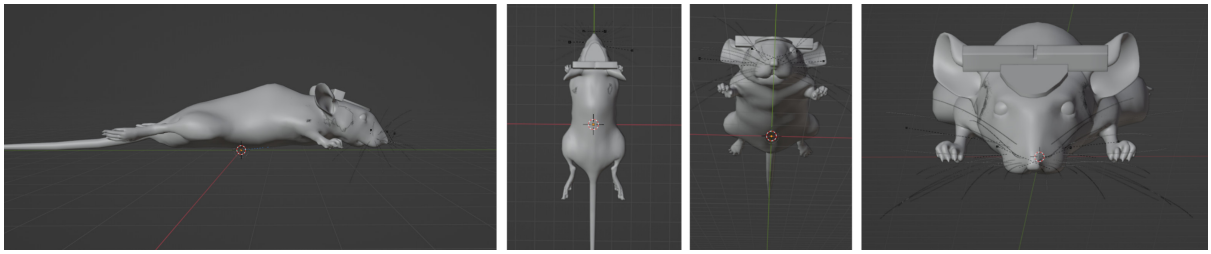


Figure 3.1: Example of the 3D mouse model. This model can simulate semi-random behavioural patterns based on accurate mice analysis. Bolaños et al. [63] release a couple of other models showing mice in different simulated environments. We use the one shown in the pictures because it is the most similar to the conditions of our dataset.

Bolaños et al. [63] has taken inspiration from previous synthetic models of large animals to

develop a similar model for mice. An illustrative example of the mouse model is in Figure 3.1. This 3D CAD model simulates semi-random behavioural patterns from real mice and incorporates the 3D structure of bones and joints. The model has successfully created training data for a well-known supervised 2D and 3D mouse pose estimation approaches [2, 199]. Nevertheless, there is still an unexplored opportunity to utilise the same model to generate data for training pose estimation models with lower levels of supervision. In particular, by adapting self-supervised methods from the human domain to learning mouse pose. We demonstrate this by relying on a recent self-supervised method that learns to estimate 2D human poses solely from unlabelled images and a prior on unpaired 2D poses. We follow the same idea, but instead of taking the unpaired pose annotations from the dataset to build the prior, we generate it using the 3D mouse model. Rather than relying on synthetic images and pose annotations like in previous works [45, 46, 194, 63], we discard the synthetic images and only use synthetic 2D poses. This means that our model is trained using actual unlabelled images and a smaller set of artificially generated 2D poses.

3.3 Method

Our method produces a mapping from full body images to the 2D pose of a mouse, as shown in Figure 3.2. The pose is represented as an articulated tree structure of 2D line segments corresponding to the parts of the body, such as the snout, tail, hind limbs, and forelimbs. The method extends the self-supervised approach of Jakab et al. [62], which estimates human 2D pose. This 2D pose estimator learns from unlabelled images and uses a set of unpaired 2D poses as an empirical prior, removing any dependence on paired annotated data. However, it does require a set of 2D pose annotations for a subset of images from the dataset, albeit the pairing is discarded. We adapt this approach by changing the topology of pose descriptions to a mouse model. We also generate an empirical prior for 2D mouse pose by projecting from an existing 3D mouse model, which removes the need for manual pose annotation altogether.

The pose estimator emulates a conditional auto-encoder that is trained to reconstruct a picture depicting a mouse. The synthesis of the reconstructed image is conditioned on an auxiliary mouse image showing a fixed pose. Additionally, the auto-encoder has a bottleneck that encodes the 2D pose as a set of joint positions, allowing it to learn the 2D poses and accomplish the reconstruction task simultaneously. Once trained, the final pose predictor is the encoding part

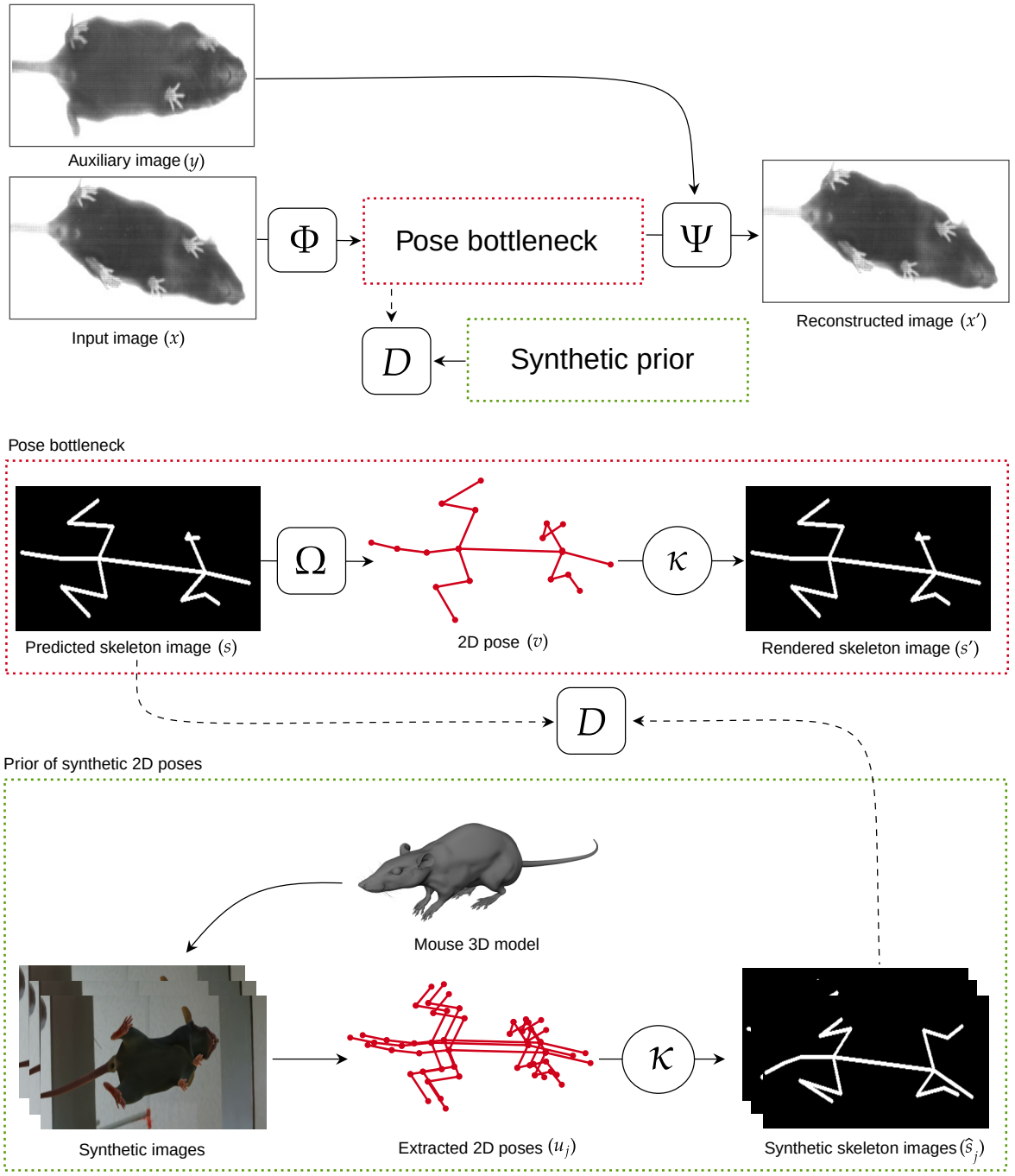


Figure 3.2: 2D mice pose estimator. We use a self-supervised 2D pose estimator from the human domain [62], which we adapt to work with mice. It uses a cyclic GAN architecture to learn from unlabelled images and a small set of typical 2D poses (i.e., 2D joint positions of a stick figure). Unlike the original implementation, we build the pose prior using synthetic data from a 3D model of a generic mouse [63]. This gives the model more flexibility to be implemented with datasets where ground truth is unavailable.

from the conditional auto-encoder, which maps from an input image to a 2D pose. The decoding stage, i.e. the mapping from the 2D pose to the reconstructed image, is just used during training.

In this scenario, the encoding part consists of two steps. Initially, a Convolutional Neural Network (CNN) Φ maps from the input image x to a skeleton image s. Then, a second CNN Ω transforms the skeleton image s to a 2D pose v. The decoding stage also involves two parts. First, a differentiable function κ maps the 2D pose v to another skeleton image s'. Then, a CNN Ψ maps from the skeleton image s' to the reconstructed image x'. Since s' does not contain enough appearance information for the reconstruction process, during the last mapping, Ψ also takes an image y as an additional input to compensate for the missing appearance information in s'.

Furthermore, introducing a dual representation of the 2D pose [62] (i.e. as a set of joints position coordinates v and as skeleton images s and s') helps to create a bottleneck that separates the geometry and appearance from the input. At the same time, it forces the auto-encoder to learn something helpful by preventing the encoder network Φ from copying the image x without learning anything. A differentiable function κ [62] allows for switching between pose representations. Its purpose is to take the x and y coordinates for the joint positions in v and produce an image by drawing lines between connected joints. Formally it is given by

$$\kappa(v)_e = \exp\left(-\gamma \min_{(i,j)\in C, r\in[0,1]} ||e - rv_i - (1-r)v_j||^2\right) \tag{3.1}$$

where C is a set of connected joint pairs (i, j), e an image pixel location, r is the value for the pixel location, and v a set of (x, y) 2D coordinates of body joint positions.

We train the model with a dataset of images depicting mice in different poses and a prior of synthetic 2D poses generated from the 3D mouse model [63]. We transform the 2D poses of the prior to skeleton images using κ (Equation 3.1). We use a similar loss function as in [62], which contains three terms. The first penalises the difference between the generated image x' and the input x via a perceptual loss. The second term is a regression loss to evaluate the mapping from the skeleton image s to the 2D joint positions in v. The third term is an adversarial loss to assess the authenticity of the skeleton images generated in the encoder. Note that the actual samples for training the discriminator are the synthetic skeleton images from our empirical prior. The following sections provide more details on the model's components, the empirical prior, loss

function, and training.

The model is an auto-encoder mapping from the input image x to output image x' in which the 2D pose v emerges as an intermediate representation. The overall formulation is as follows

$$x' = \Psi(\kappa(v) \circ \Omega(s) \circ \Phi(x), y) \tag{3.2}$$

For training the networks in Equation 3.2, a perceptual loss [200] compares each input image x with the reconstructed image x':

$$\mathcal{L}_{perc} = \frac{1}{N} \sum_{i=1}^{N} \|\Gamma(x'_i) - \Gamma(x_i)\|_2^2 \tag{3.3}$$

where Γ is a pre-trained VGG network [201] with the classification stage removed to utilise the final feature encoding.

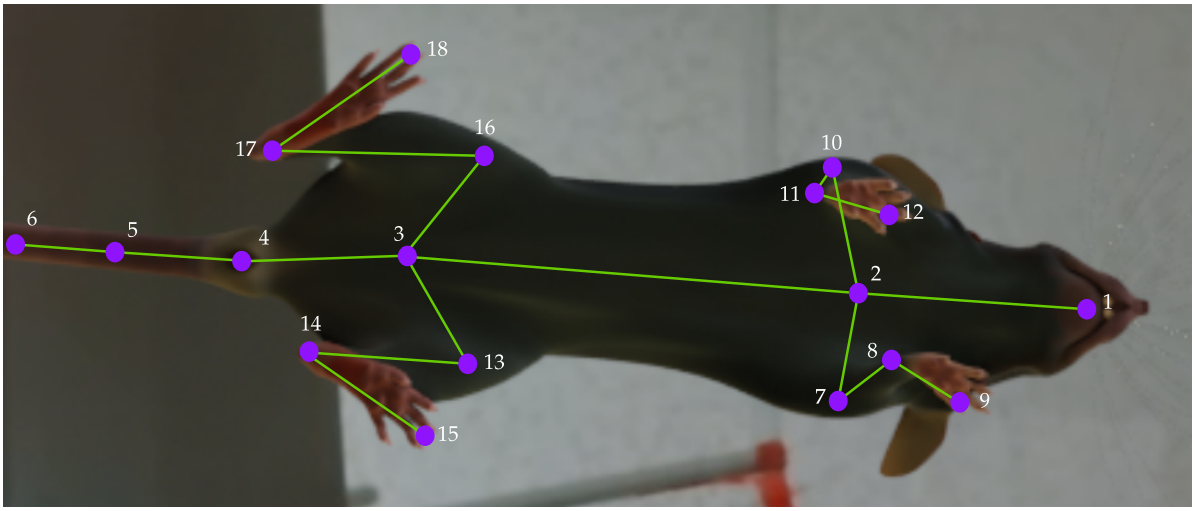
Additionally, a CNN serves as the discriminator network D, which outputs a probability that an input skeleton image comes from the prior distribution of M skeleton images. Thus, D measures the extent to which a skeleton image s looks like an authentic skeleton image from the synthetic prior distribution. Note that contrary to [62], our prior $\{u_j\}_{j=1}^{M}$ is synthesised by projecting from a 3D mouse model, which means we do not need a portion of annotated data from the training dataset. We obtain the skeleton images $\{\hat{s}_j\}_{j=1}^{M}$ via κ, i.e. $\{\hat{s}_j = \kappa(u_j)\}_{j=1}^{M}$, and then we compare this distribution $p_{data}(\hat{s})$ with the distribution $p_{data}(s)$ from the predicted skeleton images $\{s_i = \Phi(x_i)\}_{i=1}^{N}$ by means of the adversarial loss [202]:

$$\mathcal{L}_D = \frac{1}{M} \sum_{j=1}^{M} D(\hat{s}_j)^2 + \frac{1}{N} \sum_{i=1}^{N} (1 - D(s_i))^2 \tag{3.4}$$

Finally, we derive a loss from the duality of Ω and κ, which combines two terms as follows:

$$\mathcal{L}_\Omega = \|\Omega(\hat{s}) - u\|^2 + \lambda \|\kappa(\Omega(s)) - s\|^2 \tag{3.5}$$

The first term uses unpaired 2D poses from the prior, while the second one utilises the pose on the predicted skeleton image s. The last term ensures that the network learns poses that appear on the training images but not necessarily on the prior. The balancing coefficient λ is



1 - Snout	7 - Right elbow (**RE**)	13 - Right Knee (**RK**)
2 - Vertebral column base (**VCB**)	8 - Right fore paw tip (**RFP**$^{-}$)	14 - Right hind paw tip (**RHP**$^{-}$)
3 - Vertebral column end (**VCE**)	9 - Right fore paw top (**RFP**$^{+}$)	15 - Right hind paw top (**RHP**$^{+}$)
4 - Tail base (**TB**)	10 - Left elbow (**LE**)	16 - Left knee (**LK**)
5 - Tail middle (**TM**)	11 - Left fore paw tip (**LFP**$^{-}$)	17 - Left hind paw tip (**LHP**$^{-}$)
6 - Tail end (**TE**)	12 - Left fore paw top (**LFP**$^{+}$)	18 - Left hind paw top (**LHP**$^{+}$)

Figure 3.3: 2D joint positions obtained by projecting from the 3D model of the mouse. The 2D pose representation of the mouse involves 18 joints positions as indicated in the figure.

set to 0.1 in our experiments.

3.3.1 2D synthetic prior

Since we do not have pose annotations for our data, we synthetically build the 2D prior needed for training the pose estimator. This generates an unpaired set of 2D poses to be converted to skeleton images and used with unlabelled images during training. For building the 2D pose prior, we adopt a synthetic 3D model of a mouse [63]. This animated mouse model simulates synthetic behavioural data using animation and semi-random joint movements. We keep the original joint-constrained movements of the freely moving mouse model. However, we change the size of the mouse model and the camera viewpoint to match the synthetic scene to our experimental videos. We also introduce a linear path in the synthetic scene to simulate the treadmill in which the mice run in our experimental setup. We animate and render the scenes with the synthetic model and extract the 2D coordinates of 18 joints on the mouse's body, as detailed in Figure 3.3. We use Blender[3] to animate the mouse model and extract the 2D poses. See Appendix A for more details about the mouse model.

[3]https://www.blender.org/

Finally, we use those joint positions to create their respective skeleton image, as shown in Figure 3.2. We remove poses with joints outside the boundaries of the image. To introduce more variability into the poses of our prior, we extend it by randomly rotating the positions representing the tip and top of the fore paws of the synthetic poses. Our prior consists of 15,408 different 2D poses transformed into skeleton images. Figure 3.4 displays some randomly selected 2D poses from the prior.

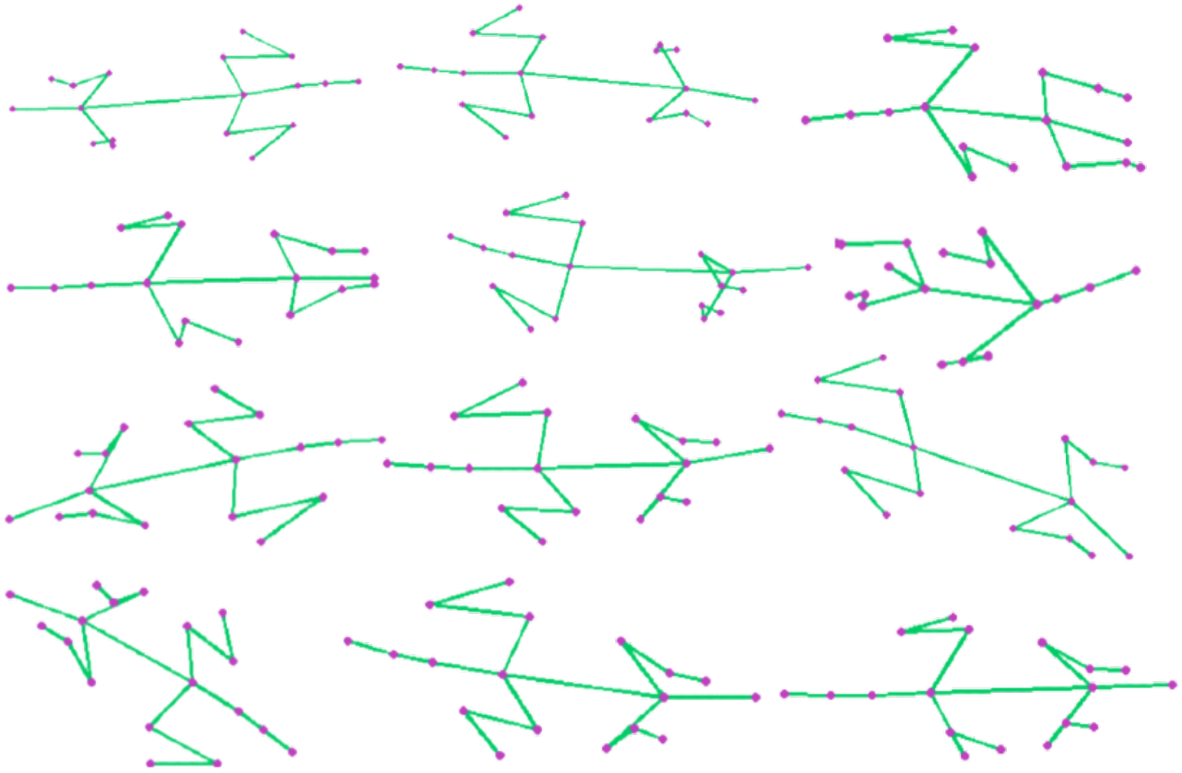


Figure 3.4: Random examples from the prior. We provide a visualisation of some items from the synthetic prior of 2D poses.

3.3.2 Training procedure

Following [62], we use a perceptual loss $\mathcal{L}_{perc}$ (Equation 3.3), an adversarial loss $\mathcal{L}_D$ (Equation 3.4), and a regression loss (Equation 3.5) in training the convolutional networks Φ, Ω, and Ψ. Note that κ is not a learnable function and λ represents a balance coefficient set to 10. The overall loss $\mathcal{L}$ is given by:

$$\mathcal{L} = \lambda \mathcal{L}_D + \mathcal{L}_\Omega + \mathcal{L}_{perc} \tag{3.6}$$

We train the pose estimator using images from the videos on the dataset featuring mice running

at different speeds: $10cm/s$, $20cm/s$, and $30cm/s$. Additionally, we include images from video segments that show the transition between speeds, where mice are not running but moving freely on the treadmill. During training, we also utilise the samples from the synthetic pose prior.

Unlike [62], who uses a pretrained Ω, we train all the neural networks Φ, D, Ω, and Ψ from scratch by optimising the loss function in Equation 3.6. In particular, each batch is formed by randomly sampling images x and y from the dataset and a random sample u from the synthetic 2D poses, which is then transformed into the skeleton image $\hat{s}$. The input images x and y were resized to 128×128 pixels. We set the batch size to 32 and use the Adam optimiser [203] with a learning rate of 2×10^{-4}, $\beta_1 = 0.5$, and $\beta_2 = 0.999$.

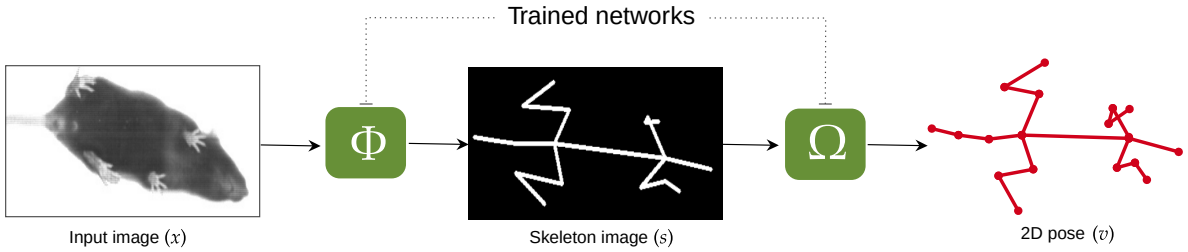


Figure 3.5: Networks used during inference for 2D mouse pose estimation. During the testing stage, we only require the trained networks (Φ, and Ω) responsible for mapping the image s to the final 2D pose v. The remaining networks and other artifacts within the approach are only necessary while training the model (see Figure 3.2 for reference).

During testing, we only rely on the trained networks Φ and Ω, to map from an input image to a 2D pose. Specifically, we resize the image x and put it through $\Phi(x)$ to obtain the skeleton image s. Then s is processed by $\Omega(s)$ to get the ultimate 2D pose v. Figure 3.5 illustrates the part of the model used for inference.

3.4 Experiments

3.4.1 Dataset

Our dataset contains images/frames taken from recordings of rodent models with Amyotrophic Lateral Sclerosis. These models have different genotypes, including $Sod1WTxSarm1WT$, $Sod1TgxSarm1WT$, $Sod1WTxSarm1TG$, and $Sod1TgxSarm1WT$, and are at the ages of four and sixteen weeks, showing different stages of the disease's progression. Overall, a minimum of 8 mice were used for each experiment, including only male mice, due to the gender-specific

variations in ALS disease development [204]. Mice for all the experimental genotypes were housed under the same standard conditions with free access to food and water. In addition, the animals were constantly monitored for changes in their overall health condition, with focus on evaluating the ALS disease progression [205]. All the mice appearing in the recordings were bred and maintained at the University of Tasmania under the guidelines outlined in the Australian Code for the Care and Use of Animals for Scientific Purposes [206]. More detailed descriptions of procedures and types of animals appearing on the dataset could be found in [207].

The recordings were made using the DigiGait^{TM} apparatus, which consists of a transparent treadmill and a camera placed underneath, as illustrated in section A of Figure 3.6. Mice at both 4 and 16 weeks of age were first acclimatised in the apparatus and then encouraged to run on the treadmill at $10cm/s$, $20cm/s$ and $30cm/s$ for a minimum of 10 seconds. The camera captures the mice on video as they move on the treadmill. Mice were gently encouraged to run by taps to their rear by the experimenter if needed. At the end of the trial, the mice were returned to their home cage. Section B of Figure 3.6 shows some images taken from one of the videos depicting different mouse poses.

We built the dataset with images from 40 videos recorded by DigiGait. Each video comprises around $13,120$ images/frames, depicting a single mouse running on the transparent treadmill at three different speeds. On average, the videos are about 80 seconds long, meaning that the mice ran for a minimum of 20 seconds at each pace, with 10 seconds of transition time between them without running. Each image has an original dimension of 658×190 pixels, and the frequency is 164 frames per second. We use images from half of the available videos to get the training set, and reserve the other half for evaluation purposes.

3.4.2 Results

Once trained, our model produces a 2D representation of the mouse pose composed of 18 joint positions for a given unlabelled image. We estimate a 2D pose for each image on the videos from the test set. Figure 3.7 and Figure 3.8 shows some of those predicted 2D poses.

Since our dataset does not contain annotations for the joint positions, we manually annotated 2D poses for 100 images randomly selected from one of the videos. These annotations provide ground truth for quantitatively measuring the model's prediction performance. Note that the images used for annotation were not included while training the model. We compare pose

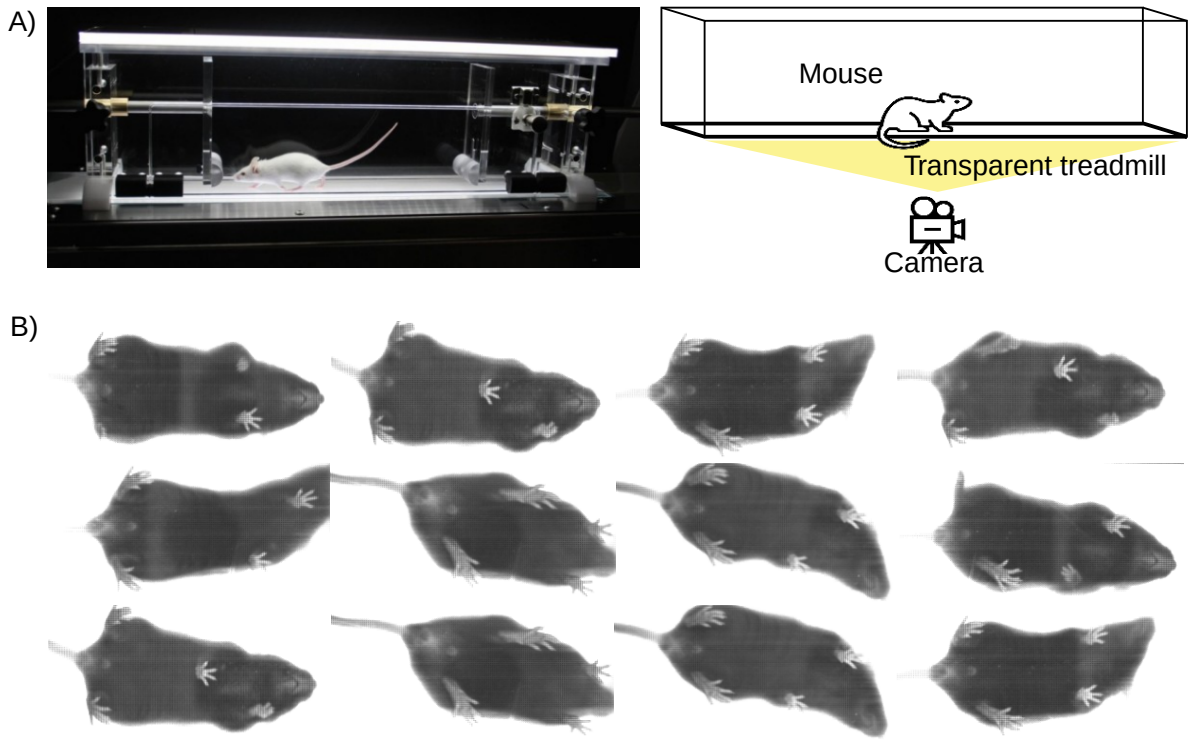


Figure 3.6: (**A**) Real and schematic example of DigiGait^{TM} apparatus, image taken from https://stoeltingco.com/Neuroscience/DigiGait-Imaging-System~9916. (**B**) Representative images from one of the videos, depicting different mouse poses. The original recordings will be made available upon reasonable request.

predictions with ground-truth on this test set using the Mean Per Joint Position Error (MPJPE) [208], which measures the mean Euclidean distance in pixels between the predicted positions for each of the 18 joints composing the mouse pose and their respective ground truth positions. Given an image x, a 2D skeleton $\bar{v}$, and a pose estimator f, the MPJPE is formally defined by

$$\text{MPJPE}(x, \bar{v}) = \frac{1}{L} \sum_{i=1}^{L} \left\| m_{f,\bar{v}}^{(x)}(i) - m_{gt,\bar{v}}^{(x)}(i) \right\|_2 \tag{3.7}$$

where L is the number of joints in $\bar{v}$, and $m_{f,\bar{v}}^{(x)}(i)$ is a function that returns the coordinates of the $i-th$ joint of a skeleton $\bar{v}$, predicted by the pose estimator f, for a given image x. Similarly, $m_{gt,\bar{v}}^{(x)}(i)$ represents the coordinates of the $i-th$ joint of the ground truth skeleton $\bar{v}$ at x.

Table 3.1 shows the average scores across images for each of the 18 joint positions. Note that the model was not trained using the ground-truth annotations; it learns solely from the synthetic pose prior and the unlabelled images from our videos. Furthermore, there are no processes for domain adaptation involved.

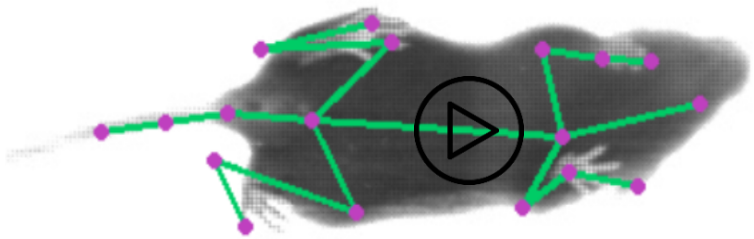


Figure 3.7: Estimated poses for consecutive images. We provide a video to visualise the predicted 2D poses for each image composing the video sequence. The figure acts as a link for accessing the video.

Joints	Snout LE	VCB LFP^-	VCE LFP^+	TB RK	TM RHP^-	TE RHP^+	RE LK	RFP^- LHP^-	RFP^+ LHP^+	**Avg.**
RI + SP	13.4 12.7	8.0 10.5	5.6 14.2	15.2 7.6	17.8 21.1	31.8 11.5	14.7 14.9	15.8 11.7	14.8 11.9	**14.1**

Table 3.1: Quantitative evaluation of predicted 2D mouse poses. We evaluate the predicted 2D poses using the MPJPE metric with corresponding ground truth. **RI + SP** indicates the use of the method trained with **R**eal **I**mages and **S**ynthetic **P**rior.

3.4.3 Semi-randomly generated prior

In an attempt to reduce the dependency on a pre-existing 3D CAD model, we experiment by generating a semi-random prior using the synthetic 2D poses. We build the new prior by randomly rotating each joint of the body extremities (fore paws, back limbs, snout, and tail) in the synthetic 2D poses while preserving the positions of the points for the mouse's spine. Subsequently, we train our model using this prior and actual images and then test it on unseen images.

If the joint rotations are not properly constrained (too random), it may result in producing unrealistic poses for the prior, as depicted in part A of Figure 3.9. Therefore, this may lead the model to produce inaccurate predictions, as shown in part B of Figure 3.9. However, designing the rotations more carefully (such as sampling from actual distributions of limb positions) would likely make it possible to create a more realistic prior of 2D poses without heavily relying on the 3D mouse model.

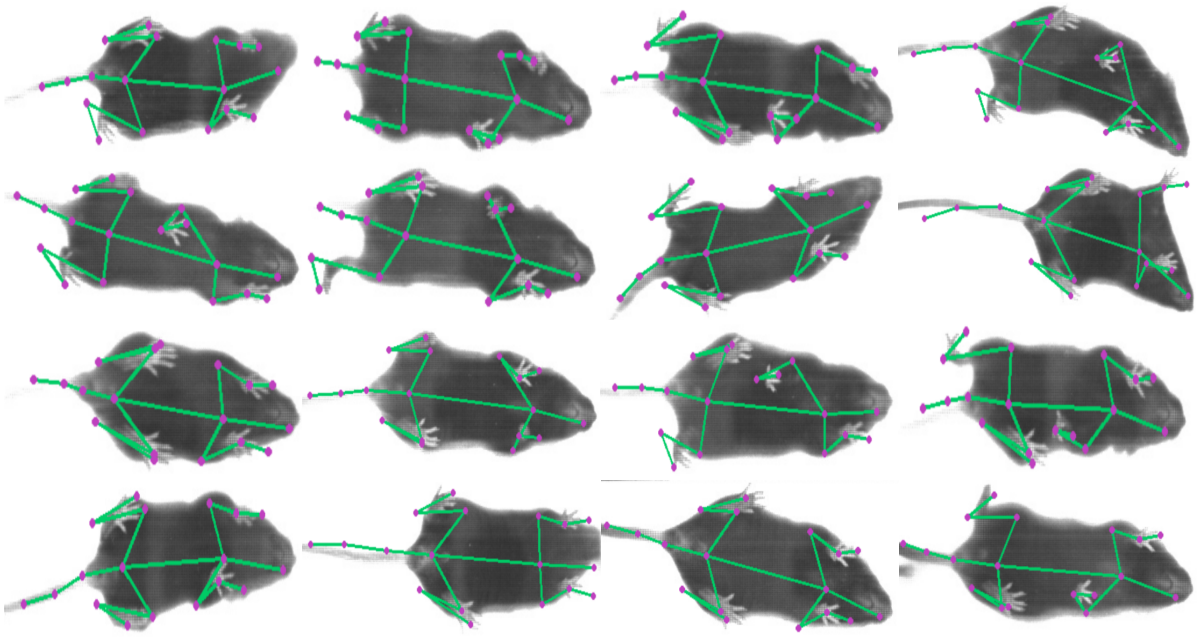


Figure 3.8: Estimated 2D poses using our method. Randomly selected images from test videos with their corresponding estimated 2D pose. Purple points represent the estimated positions for the joints.

3.4.4 Synthetic domain

In addition to the main experiment, we train and evaluate the model with a combination of images and prior from the same domain, which is closer to the original implementation. Because of the lack of pose annotations to build the prior from actual images from our dataset, we use synthetic images and synthetic 2D poses. Note that the pairing of images and poses is discarded, i.e. the pose annotations to construct the prior were obtained from images not utilised during training. We train the model with different sequences of images synthetically generated from the 3D mouse model and test it using a different set of synthetic images. We use the 2D ground truth annotations for 18 joint positions extracted from the mouse model and compare them with our model's predicted poses. We report the MPJPE for each joint position in Table 3.2, while qualitative results are shown in Figure 3.10.

Joints	Snout LE	VCB LFP^-	VCE LFP^+	TB RK	TM RHP^-	TE RHP^+	RE LK	RFP^- LHP^-	RFP^+ LHP^+	**Avg.**
SI + SP	5.9 5.9	4.0 6.9	3.0 7.0	3.7 4.1	4.3 5.2	6.2 6.0	5.9 4.0	6.6 5.1	7.1 5.0	**5.3**

Table 3.2: Quantitative evaluation of predicted 2D synthetic mouse poses. We evaluate the predicted 2D poses using the MPJPE metric with corresponding ground truth.**SI + SP** denotes use of the method trained with **S**ynthetic **I**mages and **S**ynthetic **P**rior.

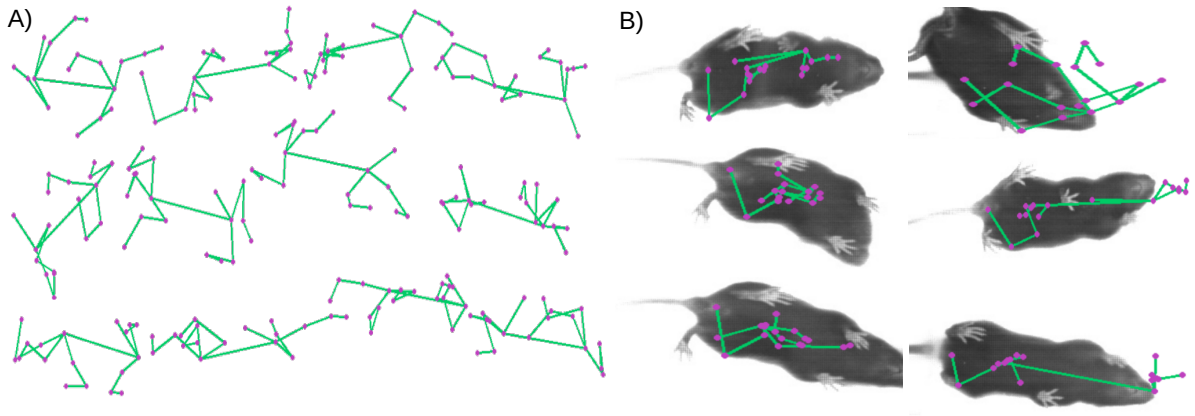


Figure 3.9: Randomly generated prior. We experiment by randomly rotating some joints based on the existing poses on the 2D synthetic prior. Part A of the figure shows examples of 2D poses after being randomly rotated. However, most of the resulting poses do not represent plausible mouse poses. Part B illustrates some predictions made with the model trained with this semi-random prior.

3.4.5 DeepLabCut comparison

Without access to a more extensive set of annotated data, evaluating all our predictions for the images on the test videos against their respective ground truth poses is impossible. Nonetheless, we also report a quantitative comparison with the predictions from a state-of-the-art supervised method for animal pose estimation, DeepLabCut [2]. The purpose of this comparison is to demonstrate that our self-supervised approach can work similarly to this supervised method, removing the requirement to annotate 2D poses for training.

In order to build the training set for DeepLabCut, we select and label a subset of 100 consecutive images from a video. We manually identify the 18 joint positions on each image, as illustrated in Figure 3.3. Then we use these images and their labelled 2D poses to train a DeepLabCut model in a supervised fashion. We follow the official implementation[4] using a ResNet-50 as a backbone and 95% of the labelled images for training and the rest for validation. With the trained DLC model, we predict the pose for unseen images from the test videos. In Figure 3.11, we can see a visual comparison of the poses estimated by DeepLabCut (represented by pink lines) versus those estimated by our method (represented by green lines). Additionally, we quantitatively compare the predictions for each joint and have summarised the results in Figure 3.12.

To generate the results shown in Figure 3.12, we utilise continuous images from a few seconds of a test video. We then estimate the 2D pose for each image using our trained method and

[4]https://github.com/DeepLabCut/DeepLabCut

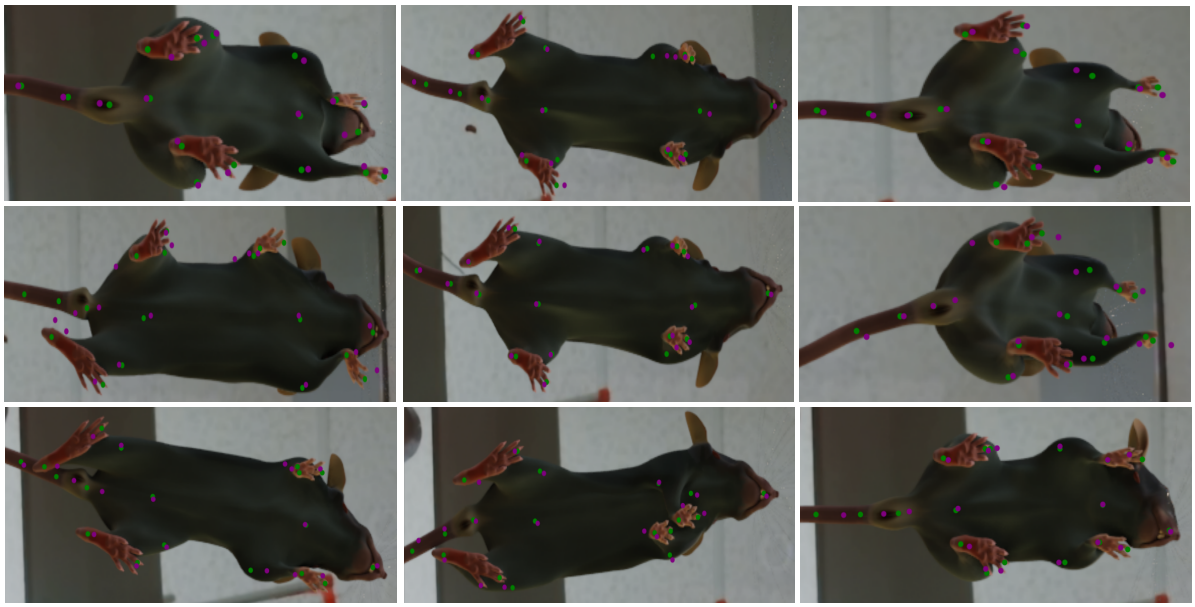


Figure 3.10: Predicted 2D poses using the model trained on synthetic images and synthetic prior. Images rendered from synthetic mouse model showing their respective predicted (purple dots) and ground truth 2D poses (green dots). Zoom-in the figure for a better visualisation.

DeepLabCut. The estimated positions of body joints are represented as pairs of coordinates (x, y) on the image frame, and we plot them over time accordingly. On each graph, we indicated our estimated positions for a given joint with solid lines, and dotted lines denote those estimated by DeepLabCut. In the inset legend, we use the label 'DLC' after the joint name to identify the predicted joint positions by DeepLabCut, while the predictions of our method simply appear indicated by the name of the joint. For specific joints like the snout, vertebral column (VCB, VCE), and tail (TB, TM, and TE), we only show the predicted x positions, which helps to better visualise the patterns in the accompanying graphs (graphs g and h). We are especially interested in estimating accurate positions for the joints representing the fore and hind paws since these joint positions are more useful for future gait measurements. Thus, we include the visualisation of both x and y predicted positions in separate graphs (a, b, c, and d) for those joints.

In particular, graphs a and b show the estimated x and y positions for the right fore paw (RFP) and left fore paw (LFP) in every image during one second of the video. The positions of these paws are given by identifying the middle point between the corresponding tips and tops of the fore paws, i.e., RFP $= (RFP^{-} + RFP^{+})/2$ and LFP $= (LFP^{-} + LFP^{+})/2$. A similar approach is taken for the right hind paw (RHP) and left hind paw (LHP) appearing on graphs c and d. Graph e groups the estimated x positions for the four paws, RFP, LFP, RHP, and LHP, while

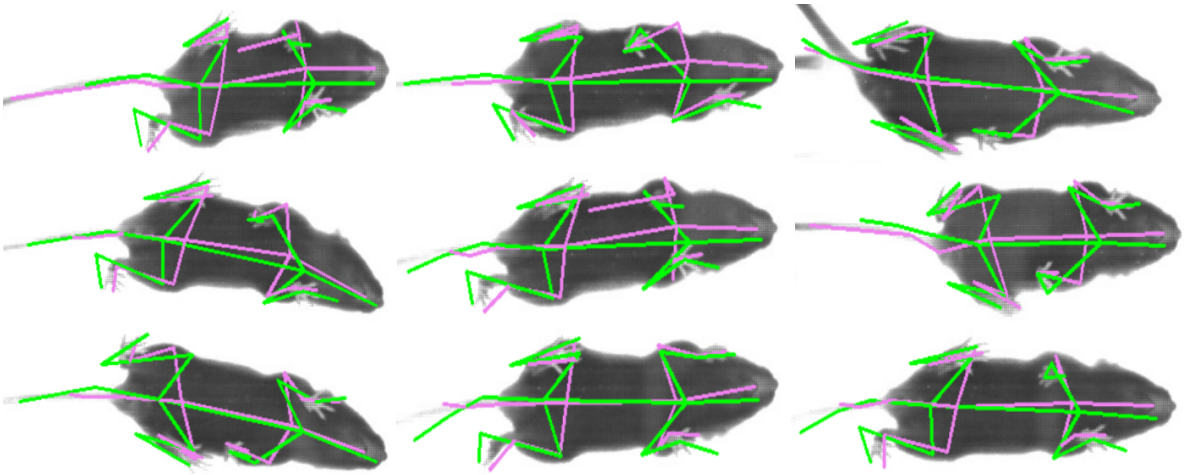


Figure 3.11: Visual comparison of predicted poses by DeepLabCut and our method. Green lines represent our predicted poses, while pink lines denote poses estimated by DeepLabCut.

graph g presents their estimated y positions. Both graphs consider an extended time window of the video (three seconds) to illustrate some abrupt changes in the pattern.

We provide the estimated positions of body joints other than the paws. For instance, graph g shows the consecutive predicted x positions for three body parts during two seconds. The top two lines correspond to the snout, the middle pair represents the vertebral column base (VCB), and the bottom two lines represent the vertebral column end (VCE). Since predictions by both methods are close, we can observe a consistent pattern for these joint positions.

Similarly, graph h illustrates the predicted x positions for different body joints. For this case, we include three joint positions corresponding to three points in the mouse's tail: tail base (TB), tail middle (TM), and tail end (TE). The differences between our predictions and those obtained with DLC are more pronounced, particularly for the tail middle and tail end joints. Our method produces smooth lines for the predicted x positions, while DeepLabCut's are noisy. To make a more detailed comparison, we focus on the predictions for the tail end (TE) and display them on graph i for three seconds of video. During manual annotation, this joint may be repeatedly located at the left border of the image frame, which is usually accurate but occasionally incorrect.

Finally, we assess some predictions of DLC quantitatively. We use the same ground truth pose annotations as those utilised for evaluating the self-supervised method (Table 3.1). Table 3.3 shows the MPJPE of DLC predictions with respect to the corresponding pose annotations. As expected, the overall MPJPE is more satisfactory for DeepLabCut. This may be partly

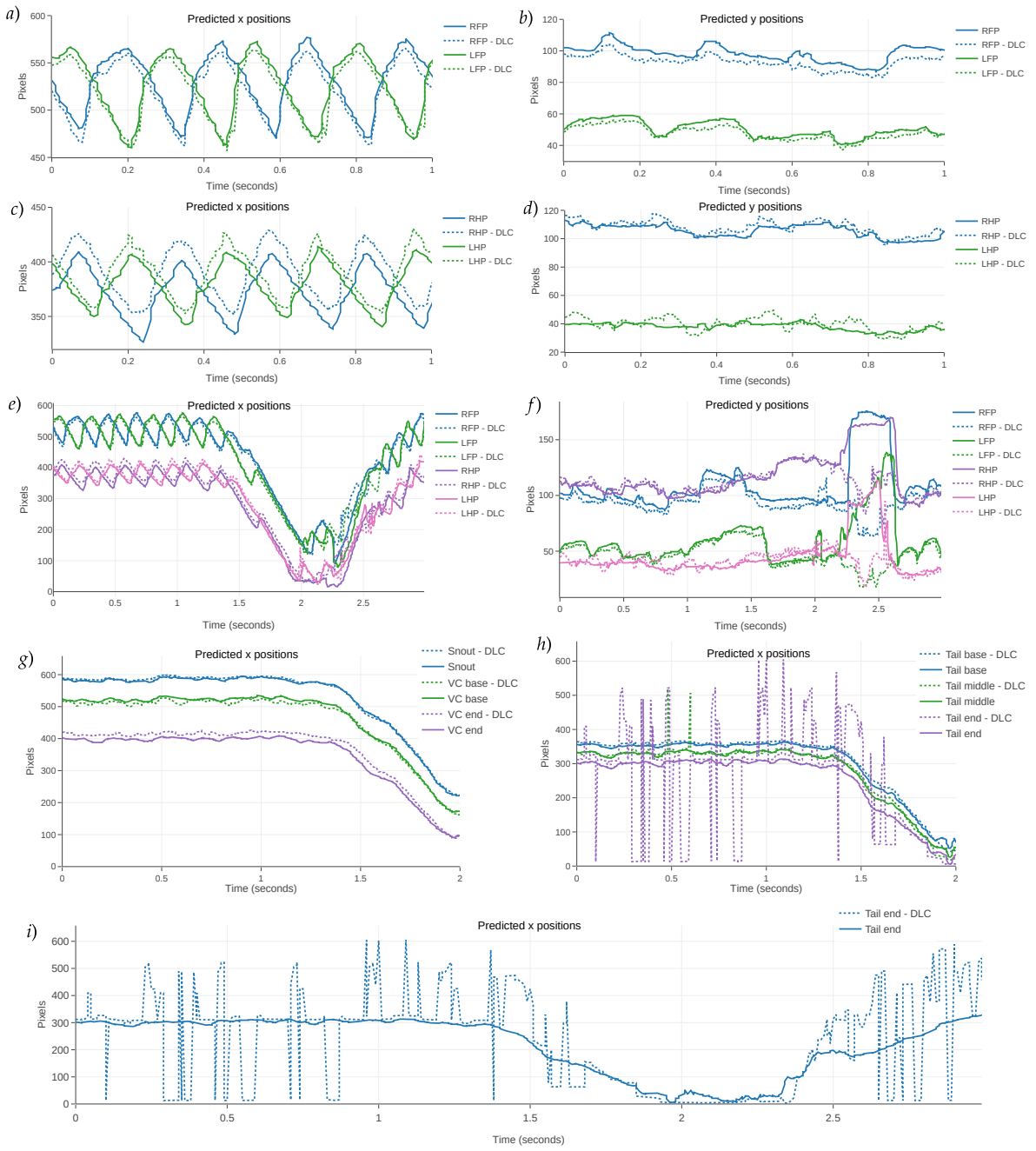


Figure 3.12: Comparison of our predicted joint positions against the ones predicted by DeepLabCut (DLC). **a)** Predictions for right fore paw (RFP) and left fore paw (LFP). **b)** Predictions for right fore paw (RFP) and left fore paw (LFP). **c)** Predictions for right hind paw (RHP) and left hind paw (LHP). **d)** Predictions for right fore paw (RHP) and left fore paw (LHP). **e)** Predictions for RFP, LFP, RHP, and LHP. **f)** Predictions for RFP, LFP, RHP, and LHP. **g)** Predictions for snout, vertebral column base (VC base), and vertebral column end (VC end). **h)** Predictions for tail base, tail middle, and tail end. **i)** Predictions for tail end.

Joints	Snout LE	VCB LFP$^-$	VCE LFP$^+$	TB RK	TM RHP$^-$	TE RHP$^+$	RE LK	RFP$^-$ LHP$^-$	RFP$^+$ LHP$^+$	**Avg.**
DLC	4.7 6.5	16.3 5.7	18.2 9.8	4.0 5.6	7.2 5.3	20.2 6.1	7.8 7.2	5.4 9.0	5.1 8.3	**8.5**

Table 3.3: Quantitative evaluation of predicted 2D mouse poses using a supervised method for animal pose estimation: DLC.

explained by using supervision in training DLC, albeit on a limited dataset. Additionally, the consistency with which joint positions were manually located during the production of ground truth for training and testing images may have contributed to these results. However, DLC's performance is lower for the base and end of the vertebral column, possibly due to challenges in consistently locating these joints during annotation.

3.5 Exploratory work for gait analysis

Utilising the predicted 2D poses generated by our model, we calculate some metrics that could be useful for performing gait analysis. Having the consecutive 2D poses corresponding to all the images in a given video, we focus on measuring the distances between the front and rear paws on the animal's left and right sides. We plot these distances against time and present the results in Figure 3.13, Figure 3.14, and Figure 3.15. Note that each second of the video corresponds to 164 images.

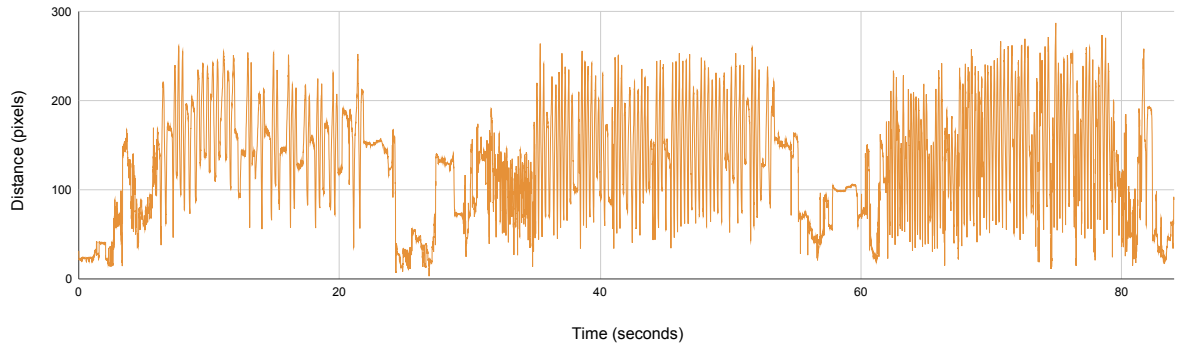


Figure 3.13: Distances between left front paw and left rear paw. We measure the distances between the left front and rear paws of each image in a video sequence. We then plot these distances for approximately 80 seconds of video.

Throughout the gait cycle, the maximum distance between each pair of paws occurs when the paws are at their farthest separation. Since each video depicts the mouse running at three different speeds, we can clearly distinguish each case in the plots. For example, Figure 3.13

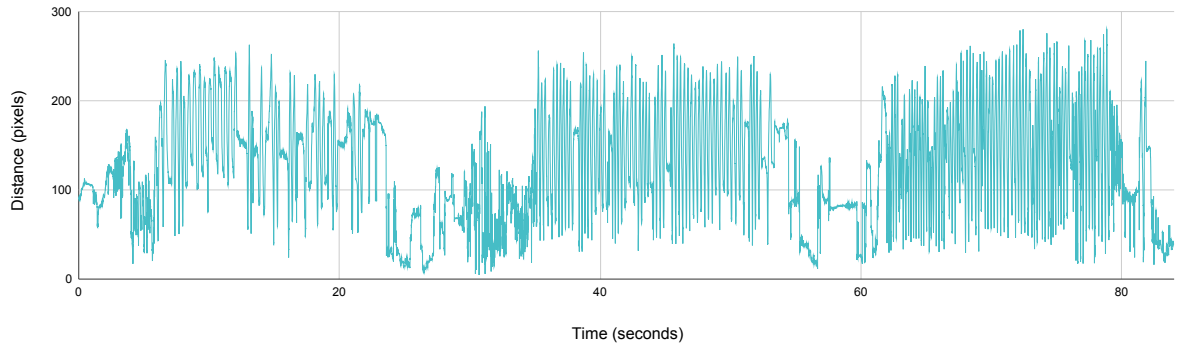


Figure 3.14: Distances between right front paw and right rear paw. We measure the distances between the right front and rear paws of each image in a video sequence. We then plot these distances for approximately 80 seconds of video.

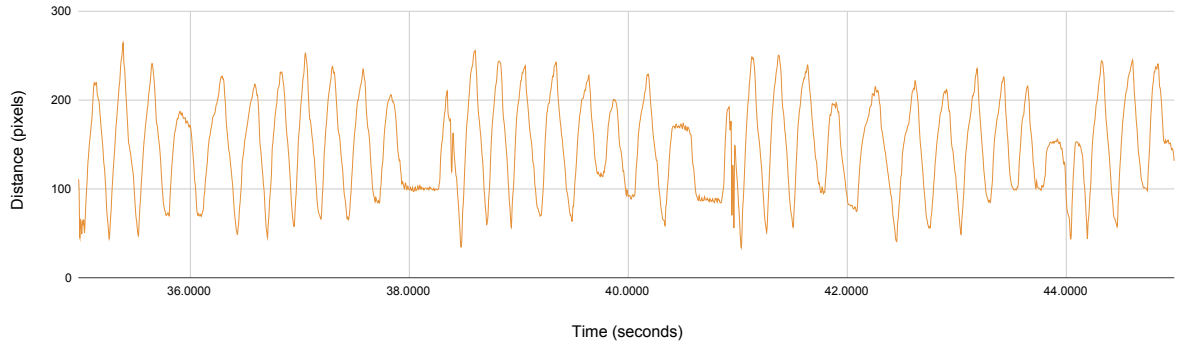


Figure 3.15: Distances between left front paw and left rear paw for 10 seconds. We provide a zoom-in into results from Figure 3.13, providing a more precise visualisation of the gait pattern during 10 seconds of video.

shows a consistent pattern between 5 and 20 seconds when the mouse runs at 10cm/s. The plot changes after that, indicating when the mouse walks freely on the treadmill but is not running. From seconds 30 to 55, the mouse starts running again, but this time at 20cm/s, which is shown as more frequent peaks in the plot. Finally, the mouse stops running again for a few seconds before running at approximately 30cm/s from seconds 65 to 75.

As an alternative approach for extracting helpful features for gait analysis, we experiment with an existing open-source method for pose clustering called B-SOiD [209]. This approach adopts ideas from unsupervised techniques to identify clusters of actions and kinematic measurements from pose data, reducing user bias and the need for manual calculations. In this case, we use B-SOiD to cluster the predicted poses by our model for a given video. It first applies t-SNE [210] for dimensionality reduction and a Gaussian Mixture Model (GMM) Expectation Maximization to group the t-SNE clusters. Then it calculates seven features for each cluster, including four distance features: relative snout to forepaws placement, relative snout to hind paws placement,

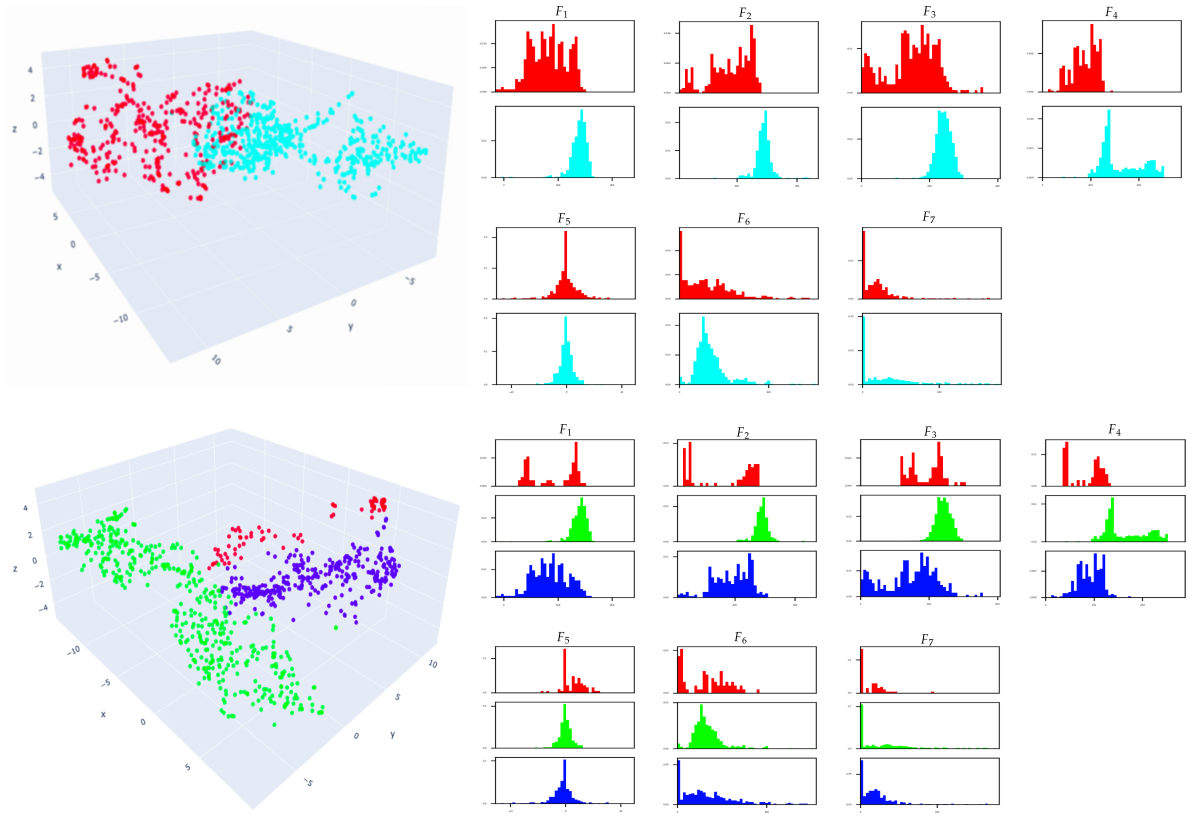


Figure 3.16: Unsupervised 2D pose clustering and features. Existing methods for unsupervised clustering, such as B-SOiD [209], permit the condensing of 2D pose data into functional clusters for activity recognition. It also produces a set of features for gait analysis. We show results using the estimated 2D poses for one video for generating two and three clusters and the corresponding seven features $(F_1, \ldots, F_7)$ for each.

inter forepaw distance, and body length (F_1, F_2, F_3, F_4); and three time-varying speed/angle features: body angle, snout displacement, and tail-base displacement (F_5, F_6, F_7). The plots in Figure 3.16 display the clusters using two and three classes (identified with different colours). The following figures indicate the corresponding features $(F_1, \ldots, F_7)$ calculated for each cluster.

Since there are no annotations for actions in the videos, we hypothesise that in the case of having two clusters, one could correspond to the poses related to the mouse in movement when it is running. In contrast, the other group could be related to the positions when the mouse is walking. A more comprehensive analysis and potential uses for these features are left for future work.

3.6 Adaptation to other animal structures

We conduct further experiments to demonstrate our method's flexibility in estimating animal pose. In particular, we use images depicting a different animal to train and test the approach, as shown in Figure 3.17. Note that this method is the same as the one described in Figure 3.2; the only differences are the input images and the synthetic prior.

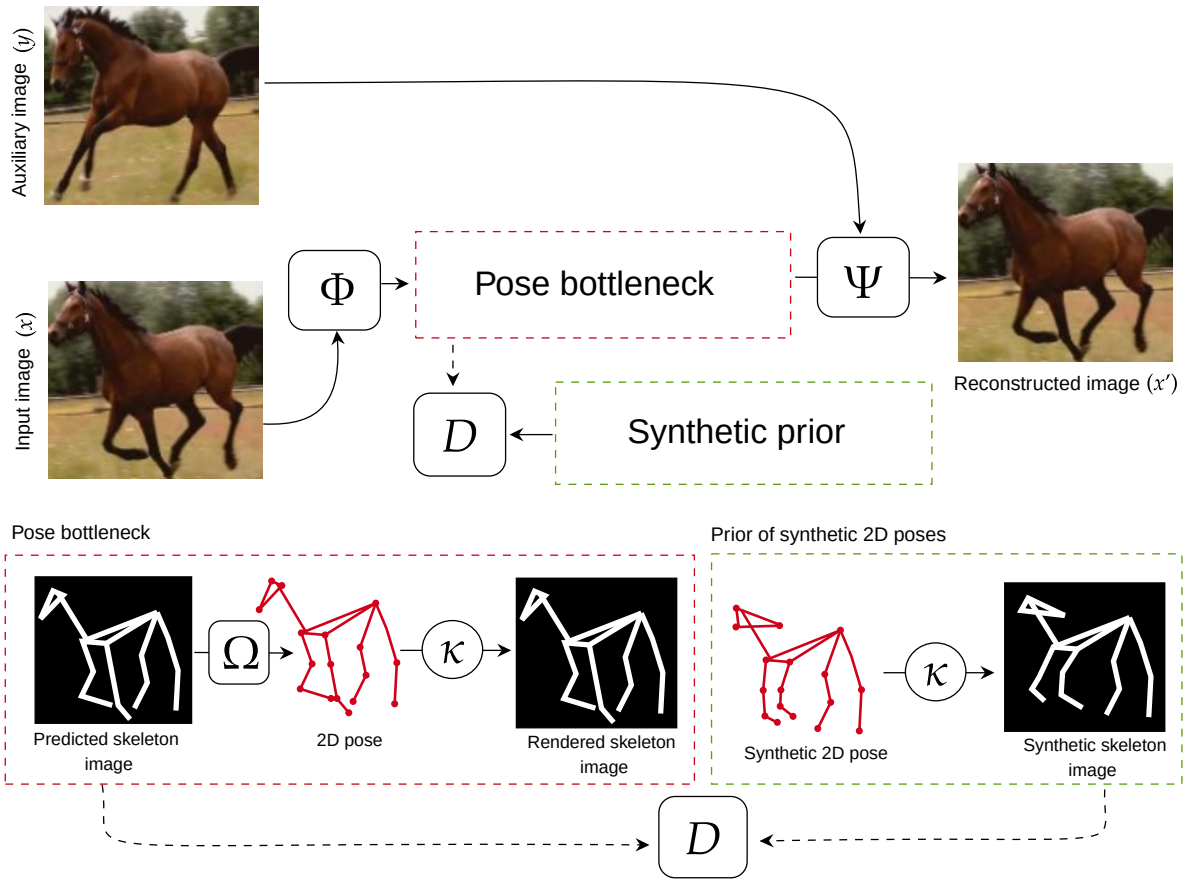


Figure 3.17: Self-supervised 2D pose estimator. The approach uses unlabelled images depicting horses and a synthetically generated prior of 2D horse poses during training. The networks are the same as those used on the main method for estimating mouse poses.

3.6.1 Data

Training

We build a dataset of images depicting full-body horses to train the model. We start by selecting the horse subset from the most recent version of the TigDog dataset [134]. We utilise images extracted from all the video sequences in the dataset, only discarding some pictures showing incomplete horses.

In addition, we automatically gather images from a manually defined collection of YouTube videos that will likely show horses for most of their frames. This enhances the training set and increases the diversity in terms of horse breeds and poses. The automatic process for collecting images consists of three main steps: Firstly, we download the videos from YouTube and split them into individual images/frames. Secondly, each image from the videos is processed using a pre-trained model [211], which identifies the horse in the frame and produces a segmentation mask. This step ensures collecting only frames containing a horse and discarding the rest. Third, we resize the frames that show horses to a predefined size of 128×128 and save them along with their respective segmentation masks.

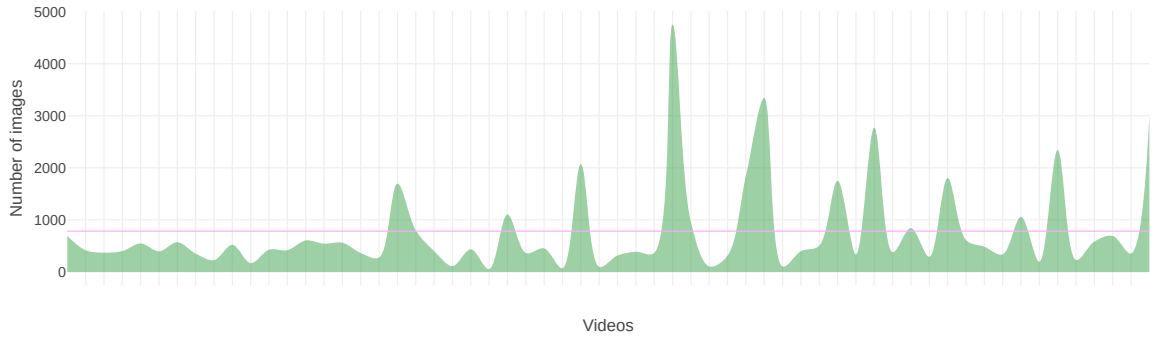


Figure 3.18: Number of images extracted from each of the YouTube videos. Most of the videos provide a similar number of images to the dataset, with a few exceptions (pink line indicates the average). We focus on collecting images from short videos to provide more variability within the data.

On average, we automatically collect 47k images depicting full-body horses from 60 videos. Figure 3.18 illustrates the distribution of video frames. Subsequent chapters provide more insights regarding the YouTube videos used to create the dataset and the collection process.

Synthetic pose prior

Similarly to the mouse case, we use a publicly available dataset of 2D poses generated from a 3D CAD model of a horse [45] as the 2D pose prior that the method needs during training. Unlike the mouse experiments, using the already generated synthetic 2D poses for the horses removed the need to manipulate the horse model manually. Figure 3.19 displays a few random examples from the synthetic prior of 2D horse poses.

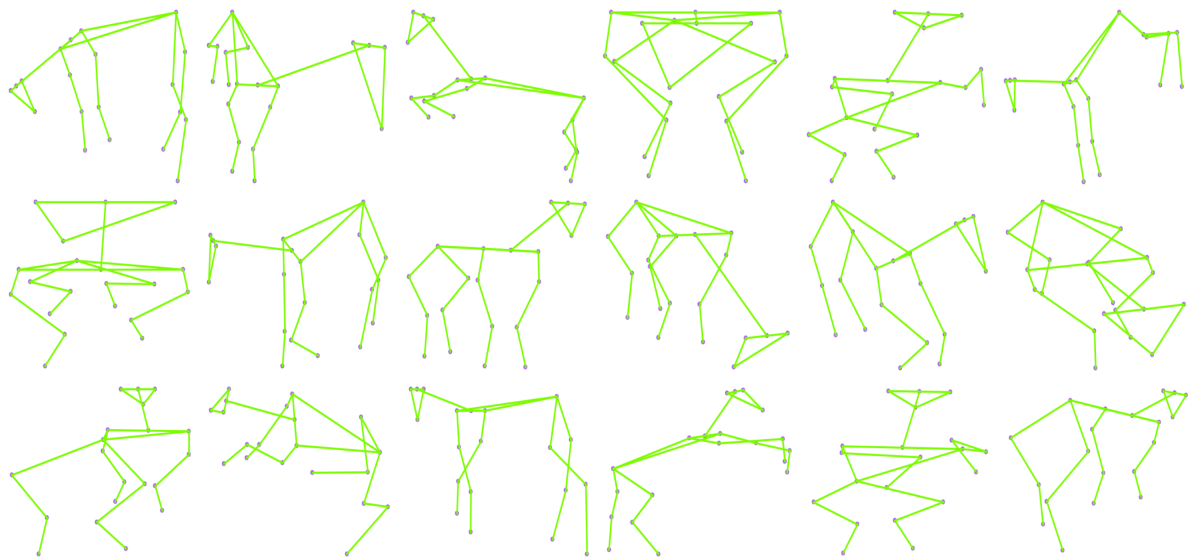


Figure 3.19: Randomly selected examples from synthetic pose prior. We use the synthetic 2D pose data from [45] to build the needed prior of synthetic 2D poses for training our approach. As demonstrated by the examples, it accurately represents the variability of horse poses.

Testing

Since the 2D pose annotations from the TigDog dataset are inconsistent, i.e. the number of labelled parts for each horse is different; and the collected video frames are not labelled. We then use images from the Weizmann dataset [135] to test the trained model. Due to the lack of pose annotations for this data, we manually labelled 2D poses consisting of 15 joint positions (three for each front and rear limb; one for the chin, and two for the eyes) for all the images on the Weizmann dataset showing full body horses (around 300 images). We utilise these pose annotations as ground truth to quantitatively evaluate the estimated 2D poses with our method.

The Weizmann dataset is diverse regarding the breed of horses, as illustrated by Figure 3.20. However, most horses are oriented to the same side; specifically, all horses face the left-hand side. We flip some of the images and pose annotations to increase the variability in orientation. Additionally, we reserve images from one of the collected videos to evaluate horses in different conditions.

3.6.2 Results

Using the trained model, we produce 2D poses for all the images in the test set. Each predicted pose comprises 20 joint positions. However, when comparing against ground truth, we only keep 15 joint positions. These include three positions for each front and rear limb, two for the

Figure 3.20: Randomly selected examples from the Weizmann dataset. In order to test our model with a different distribution of images, we use the Weizmann dataset, which contains various horse breeds that differ in appearance and pose.

eyes and one for the chin, as shown in Figure 3.21.

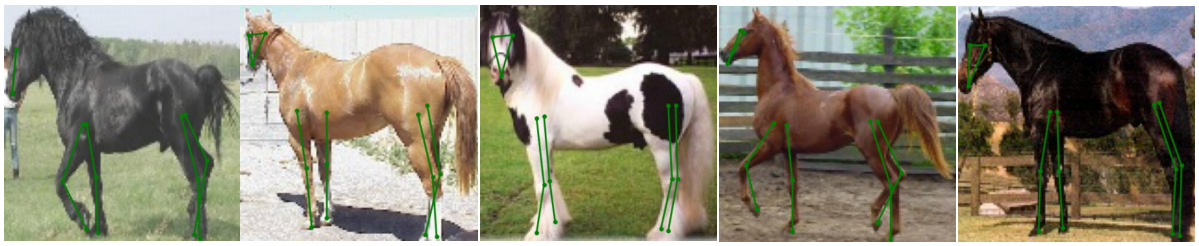


Figure 3.21: Annotated 2D poses on different images to quantitatively evaluate the model performance. We annotate 2D poses on the images from the Weizmann dataset. Each pose comprises 15 joint positions, indicated with green dots in the pictures.

To quantitatively evaluate the predicted poses by our model, we follow previous works for horse pose estimation and utilise the Percentage of Correct Keypoints (PCK@0.05) metric. This measures the alignment of the predicted 2D horse poses with their respective ground truth. In this context, the predicted keypoint is considered correct if it falls within the distance threshold (0.05). We report the results of such evaluation in Figure 3.22, which includes average PCK@0.05 scores for certain joint groups. For a more comprehensive comparison with similar methods, please refer to Chapter 5. Furthermore, Figure 3.23 shows some predicted poses for images from the Weizmann dataset and one of the YouTube videos (framed with green) excluded during training.

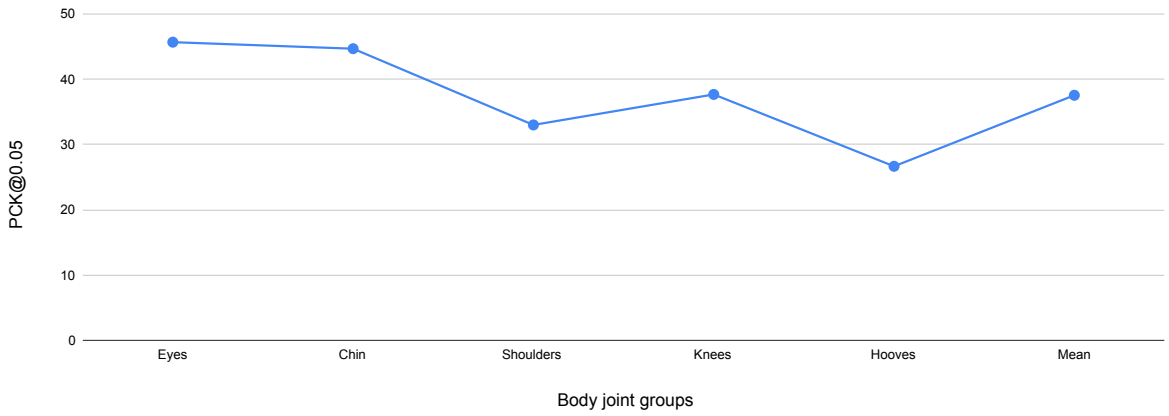


Figure 3.22: Quantitative evaluation for the estimated 2D horse poses. Each predicted horse pose was compared to its respective ground truth. The table displays the average PCK@0.05 for each group of joint positions, following standards from previous works.

We provide a more comprehensive evaluation of our model by testing it with images of wild zebras from [164]. As zebras and horses share similar skeletal structures, we anticipate that our model will generate reasonable 2D poses, despite not being explicitly trained with images of zebras. In Figure 3.24, we show some visualisations of the predictions made by the model trained with horse pictures (and horse synthetic prior) but tested on images of zebras.

3.7 Conclusion and discussion

Supervised methods learn from annotated poses on the training data, which makes them dependent on the quality of those annotations. While some joint positions are obvious to annotate, others represent a challenge and sometimes require domain specialists to locate them. Contrary to the supervised methods, our approach is not dependent on the quality of the annotations since it learns from skeleton images generated from synthetic poses. Regardless, our method produces similar 2D poses to the ones estimated with DeepLabCut (as illustrated in Figure 3.11) and is not too far off the quantitative performance of DeepLabCut in terms of MPJPE against ground-truth annotations.

According to the plots in Figure 3.12, despite some visible differences between our method and DeepLabCut for specific body parts, most graphs show smooth lines for our predictions. This trend is observable in the last graph (i), which shows a noisy dotted blue line for the DeepLabCut predictions for the tail end (TE) joint, while the predictions of our method (blue line) follow a consistent trend. When comparing both methods against ground truth annotations,

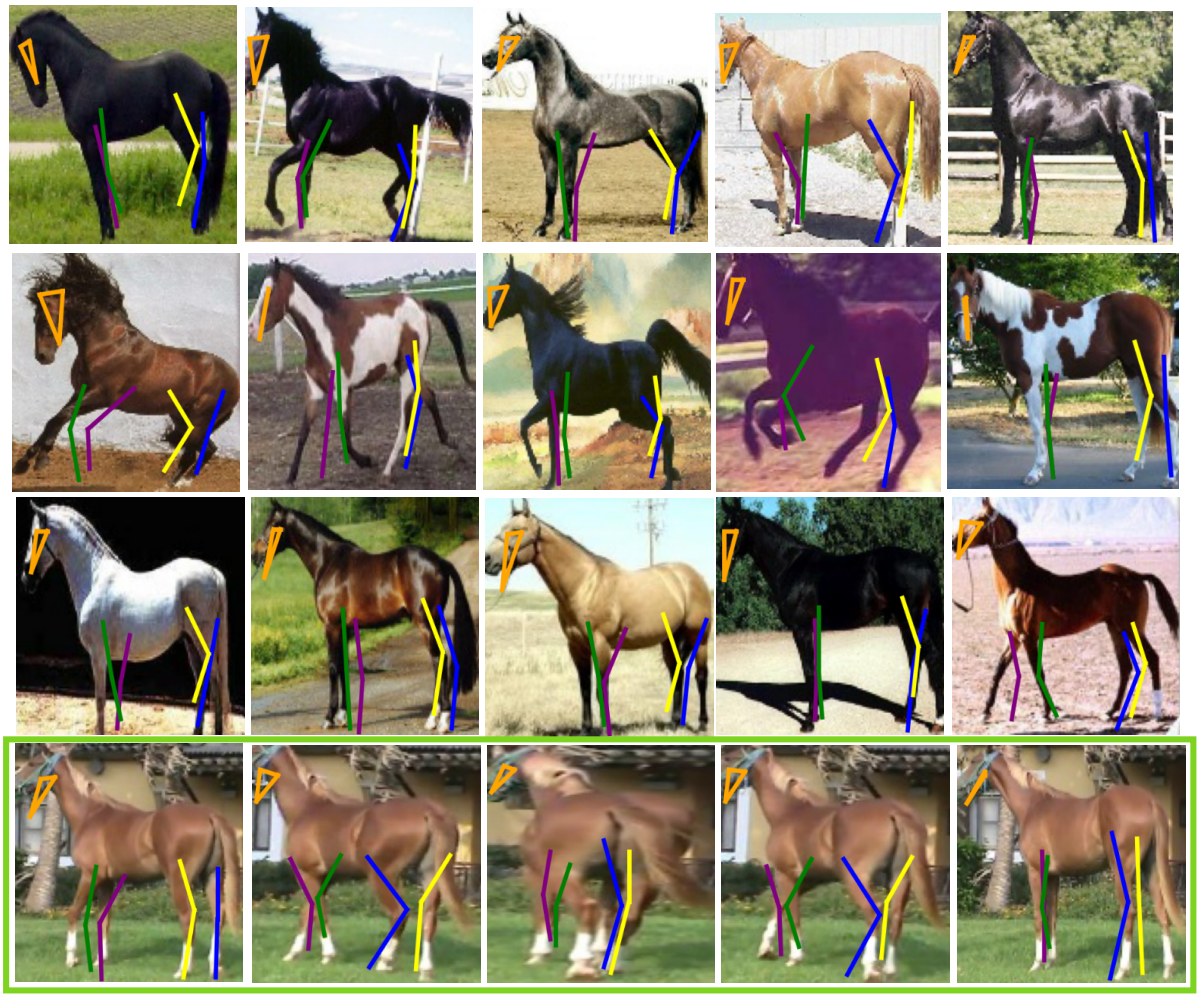


Figure 3.23: Visualisation of the predicted 2D poses for images from Weizmann dataset. Framed with green: predicted 2D poses for images from a YouTube video (excluded during training).

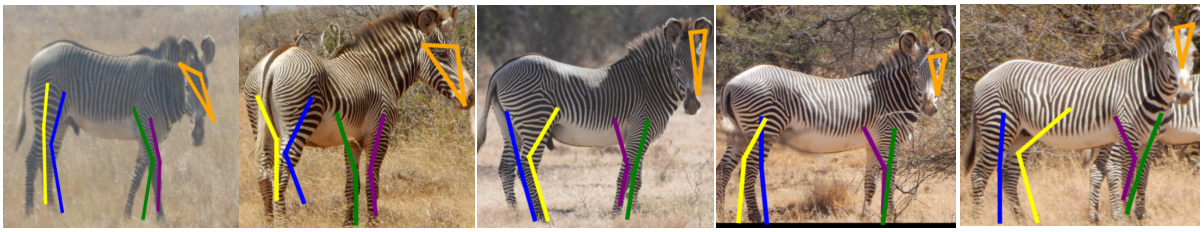


Figure 3.24: Visualisation of results using images depicting zebras [164]. Each picture in the figure represents the estimated 2D poses utilising the model trained with purely horse data; only the test data changed.

as expected, the overall performance of DeepLabCut is superior. This is probably due partly to the consistency of manual annotation of ground-truth joint locations used in training and testing DeepLabCut. On the other hand, our method has an emergent set of joint locations, which are then compared with the manual ground truth. There could be a systematic bias for each joint that, once corrected by adding joint-dependent offsets, would improve performance

further.

Our experiment training the method with synthetic images and a synthetic pose prior shows that by matching the pose prior and the image domains, the model makes accurate predictions in that domain. That experiment also shows better results than the supervised method. This may happen due to the scarce variability of the synthetic images used to train the model. Therefore the synthetic prior of 2D poses could better match their distribution of poses. In the same way, building the pose prior with unpaired annotated poses from the actual data may increase performance when training with real images. However, the latter setting does not align with the aim of this chapter.

In conclusion, we have successfully adapted a self-supervised 2D human pose estimation method to a different animal domain, replacing an empirical prior associated with actual images with a synthetic prior. We demonstrated that the approach produces promising results in relation to a state-of-the-art supervised approach in the mouse domain. Our method estimates consistent 2D locations for most body joints relying only on a few assumptions, such as unlabelled images and a small prior of synthetic 2D poses. Furthermore, the experiments with different data revealed that the method is flexible enough to work with other anatomical structures, like horses and zebras. Most significantly, our approach eliminates the need to annotate images for training, making it faster and simpler to implement with the abundant unlabelled datasets from the animal domain.

A direction of future work would be to experiment with more challenging scenarios involving more complex mouse behaviours. This could provide additional motivation to add temporal constraints to the method. An important motivation for our work has been to explore an approach that can be rapidly deployed to other animal domains without requiring extensive annotation of images. While we currently rely on a mouse model to generate the synthetic pose prior, a few manually annotated images may suffice in practice (i.e., a smaller prior with actual poses). Furthermore, a shared pose prior between species could be effective and is worth investigating. Finally, extending the work related to gait analysis could be beneficial in identifying and classifying patterns related to the development of ALS disease.

Chapter 4

Learning to predict 3D human pose from unlabelled images

We present a new method for predicting 3D human body pose from unlabelled images. The method is self-supervised and therefore has the potential for application across different domains without the need for annotated images for training. We train the prediction network using a dataset of images depicting people in a range of typical poses, along with a prior of unpaired 2D poses. Our method builds upon earlier approaches in utilising a bottleneck, involving an intermediate skeleton image and a 2D pose representation. It also critically rewards geometric consistency after randomly rotating the target 3D pose about a vertical axis, projecting to a 2D pose, lifting back into 3D and applying the inverse rotation. Unlike the current state of the art for self-supervised 3D pose prediction from images, we do not require any prior information on acceptable 3D poses, such as articulation constraints and empirical 3D pose priors. We train and test the network on images from benchmark datasets for human pose estimation, like Human3.6M, MPI-INF-3DHP, and LSP. On Human3.6M, we outperform state-of-the-art self-supervised methods that estimate 3D pose from single images, while using the MPI-INF-3DHP dataset, we obtain similar performance to the state-of-the-art. Qualitative results on a dataset of human hands show the potential for rapidly learning to predict 3D poses for structures other than the human body without the burden of collecting annotations for the training data.

4.1 Overview

Estimating 3D pose for articulated objects is a long-standing problem. Its foundations arise from the early days of computer vision with model-based approaches representing the human body as an articulated structure of parts [212, 24]. Interest in estimating 3D human pose grew within the computer vision community, partly because of the many real-world applications, for example, pedestrian detection [213], human-computer interaction [13], video surveillance [214], and sports analysis [215]. Initial work on estimating 3D pose addressed this problem by extracting hand-crafted features, such as segmentation masks [216]. Other early approaches, like exemplar-based methods, use extensive datasets of 3D poses (commonly constructed from motion data) to search for the optimal 3D pose given its 2D representation [217, 218, 219].

Deep learning methods for pose estimation have rapidly gained popularity in recent literature because of the performance improvement compared to traditional methods. These approaches also eliminate the need for manual feature extraction, resulting in faster implementation and deployment. The initial deep learning techniques for 2D and 3D pose estimation commonly describe this problem as a regression of the body joint positions [15, 16]. This means that the model must learn to map the coordinates for each joint position in an image based on the actual positions provided. Subsequent works [33, 14] improve the robustness of pose estimation methods by introducing heat-map representations for the coordinates of body joints. However, most of these pioneering approaches rely on supervised learning, implying that these require access to large annotated datasets for training, which is particularly difficult to obtain in the 3D domain.

Getting 3D joint positions is more challenging than annotating 2D joint positions on images. It is a time-consuming and error-prone process that often requires specialised equipment for data acquisition, e.g. depth sensors, multiple cameras, and wearable devices. Consequently, there is growing interest in transitioning from fully supervised deep learning methods and developing approaches for 3D pose estimation that better exploit the availability of unlabelled data to learn accurate 3D representations. Furthermore, minimising the assumptions of data availability for training deep learning models contributes to making them more flexible in terms of implementation with different datasets.

In this chapter, we describe a method that overcomes the problem of collecting 3D annotations

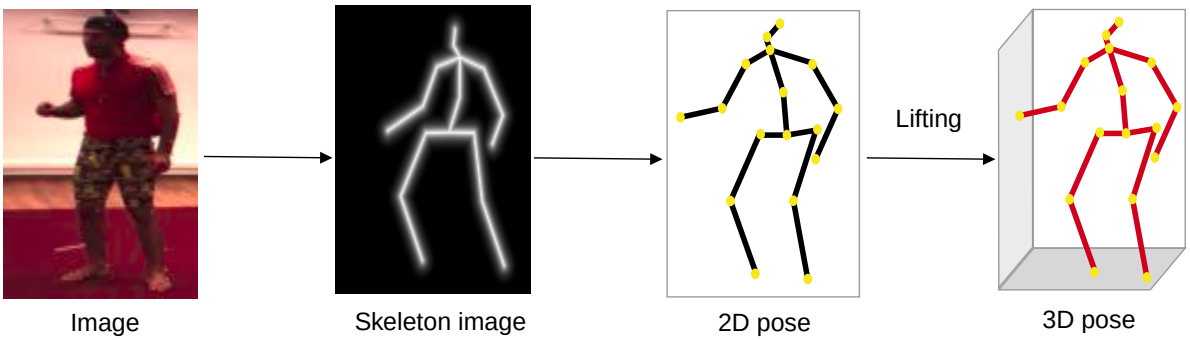


Figure 4.1: Summary of the proposed method for estimating 3D human poses. Our approach simultaneously learns to estimate 2D poses from single unlabelled images and lift them to 3D. The depicted image-to-pose mapping is embedded within a larger network for end-to-end training.

for training 3D pose estimation methods. We design a self-supervised approach for estimating 3D human pose that works under minimal data availability assumptions. In particular, it learns 3D human poses from unlabelled images and a prior on unpaired 2D poses, relying on some intermediate representations as illustrated in Figure 4.1. Our primary hypothesis is that leveraging the learning to such minimal requirements as unlabelled data and 2D unrelated pose annotations would make the method flexible enough to be rapidly implemented with data from different domains. This is especially relevant for animal datasets, which often lack sufficient annotations for both 2D and 3D domains. Note that this chapter focuses on discussing the foundations of the proposed method. For further understanding of the approach's flexibility and supporting evidence of its application to the animal domain, please refer to the last chapter of the thesis.

The proposed approach involves training a 2D pose predictor and a 3D lifting model to generate 3D joint positions from unlabelled images. This is done through an end-to-end learning framework. Our method draws inspiration from the recent state-of-the-art approaches in estimating 2D and 3D poses, including the use of skeleton images as an intermediate representation [119], the incorporation of image-to-image translation networks for learning the intermediate 2D pose representation [220, 119], and a lifting process into 3D that exploits 3D geometric self-consistency for training [54]. In addition, we use normalising flow (NF) to provide a prior on the 2D pose and estimate the elevation angle, which helps in performing rotations for geometric self-consistency [55].

In summary, our method simultaneously learns 2D and 3D pose representations in a largely

unsupervised fashion, requiring only an empirical prior on unpaired 2D poses. We demonstrate its effectiveness on Human3.6M [208], MPI-INF-3DHP [221], and Leeds Sports Pose (LSP) [29] datasets, three of the most popular benchmarks for human pose estimation. We also show our method's adaptability to other articulated structures using a synthetic dataset of human hands [222]. In experiments, our approach outperforms some state-of-the-art self-supervised methods that estimate 3D pose from images but require more supervision in training.

4.2 Background

The literature review chapter of this thesis states that deep learning methods for 3D human pose estimation are usually classified based on their input. Some take in 2D poses and lift them to 3D, while others learn the 3D poses from images. Our proposed method in this chapter falls under the second group since it inputs images. However, at the same time, it incorporates ideas from the lifting approaches, making it closely related to those methods too. More specifically, our method for estimating 3D human poses is influenced by previous works that focused on estimating 2D poses, such as the work of Jakab et al. [62]. Although this approach was not intended to work with 3D poses, it is still exciting because it does not require annotations for input images, which aligns with the purpose of our approach. Therefore, we adopt some parts of their architecture to learn a mapping from the input image to an intermediate 2D pose representation, which is then lifted to 3D. For the lifting part, we take some ideas from the work of Chen [54], such as the notion of geometric self-consistency. To further improve the accuracy of our approach, we draw inspiration from more recent literature [55], e.g. the incorporation of normalising flows. Different from related works, our approach learns both the mapping to 2D poses and the lifting to 3D in an end-to-end manner. More importantly, we do not assume access to 3D annotations or paired 2D data to supervise training.

Prior to introducing the complete architecture and components of the model, we will review the foundations of some of the core ideas that make our approach work. These include geometric self-consistency and an overview of image translation networks. We will also discuss some additions that enhance the model performance, such as normalising flow.

4.2.1 Geometric self-consistency

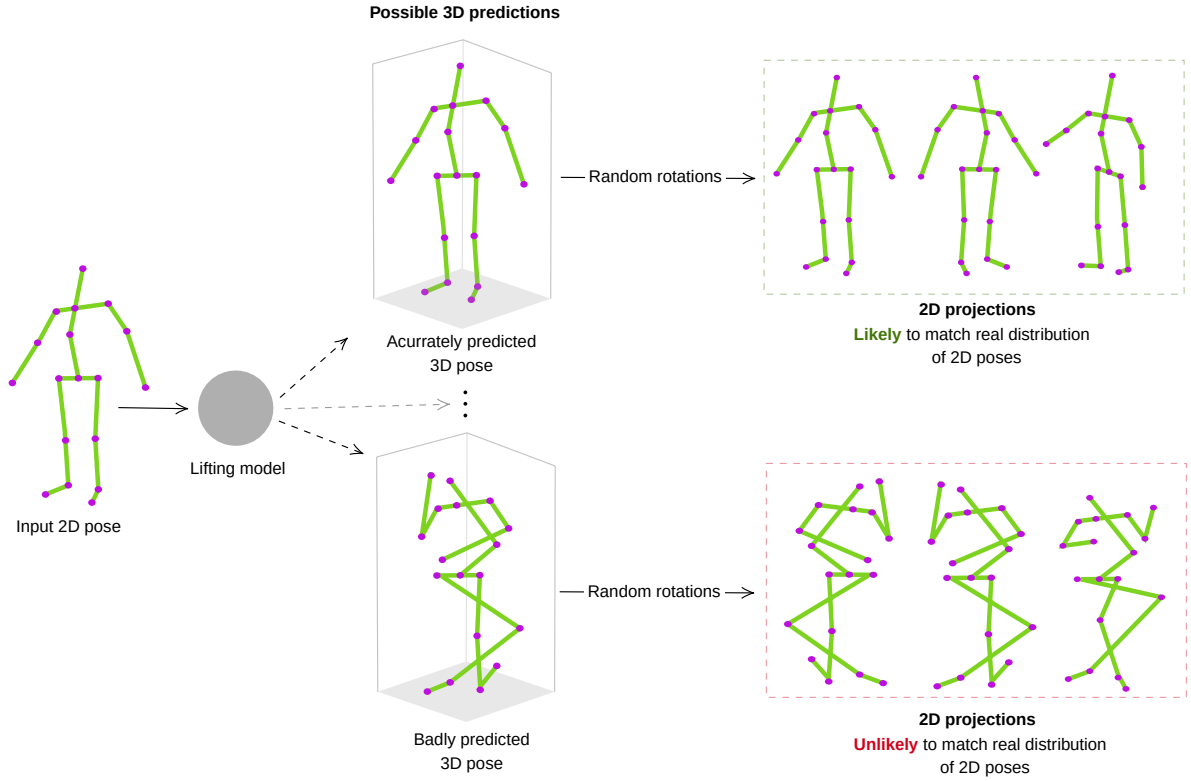


Figure 4.2: Geometry cues for pose estimation. A single 2D pose can have multiple 3D representations, but not all are plausible. When randomly rotated such 3D pose predictions, only accurately predicted 3D poses will likely project to plausible 2D poses.

As illustrated by Figure 4.2, given a 2D pose, there are infinite 3D poses whose 2D projections match the position of 2D landmarks in that view. However, an implausible 3D skeleton is unlikely to appear realistic from a different randomly chosen viewpoint. Conversely, when 3D poses are correctly estimated and then randomly projected onto a 2D plane, the resulting 2D poses are more likely to reflect the distribution of actual 2D poses regardless of the viewing direction.

The work of Drover [145] is one of the first methods to build upon the previous assumption by exploiting geometry cues, such as rotations and projections, to constrain a weakly supervised deep learning model. Their model evaluates the 2D projections of rotated 3D poses using a GAN loss, allowing it to learn realistic-looking 3D poses guided by the realism of their projections. Later, Chen [54] extends Drover's idea [145] by introducing the notion of geometric self-consistency. This involves taking the predicted 3D pose, randomly rotating it and projecting it to 2D. However, unlike [145], these steps are repeated, i.e. the 2D projection is the input

for the same prediction model, and then the resulting 3D pose is inversely rotated and projected to 2D again, as shown in Figure 4.3.

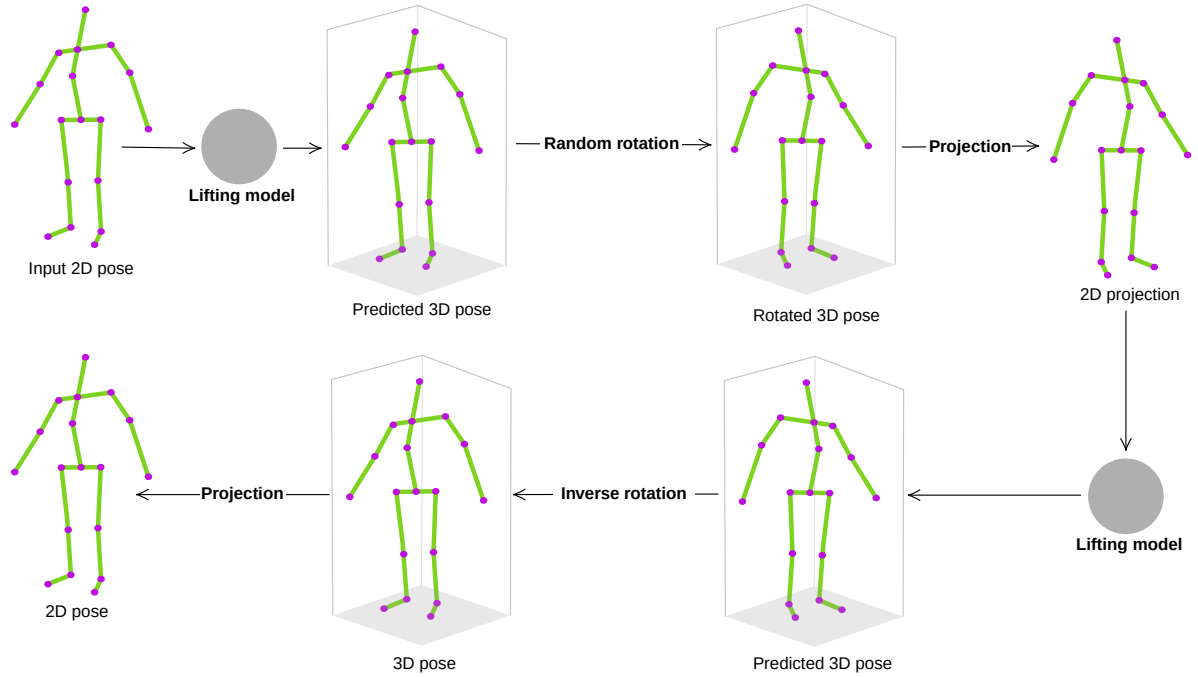


Figure 4.3: Geometric self-consistency for pose estimation. The work of Chen [54] exploits the geometric consistency for lifting a given 2D pose to 3D. After randomly rotating an accurately predicted 3D pose and projecting it to 2D, this projection will likely produce another accurate 3D pose when using the same lifting model. Comparing related pose representations from this cycle produces a strong loss term to constrain the lifting model to learn plausible 3D poses.

This cycle of geometric transformations of the pose creates a strong signal for self-supervising the learning by comparing analogous representations from the forward and backward parts of the process. This will drive the model to learn accurate 3D poses since the reasonable 2D projections from randomly rotated plausible 3D predicted poses will also produce realistic 3D skeletons. The loss terms will reward realistic-looking pose representations and penalise inaccurately estimated 3D poses.

4.2.2 Image-to-image translation

Image-to-image translation involves taking an input image from a particular domain and translating it into a corresponding output image from a different domain [220]. For instance, we can translate from aerial photos to maps, night scenes to daytime scenes, sketches to actual photographs or, more related to our method, from images to skeletons [119, 65] (Figure 4.4).

The fundamental aspect of image-to-image translation heavily relies on generative adversarial

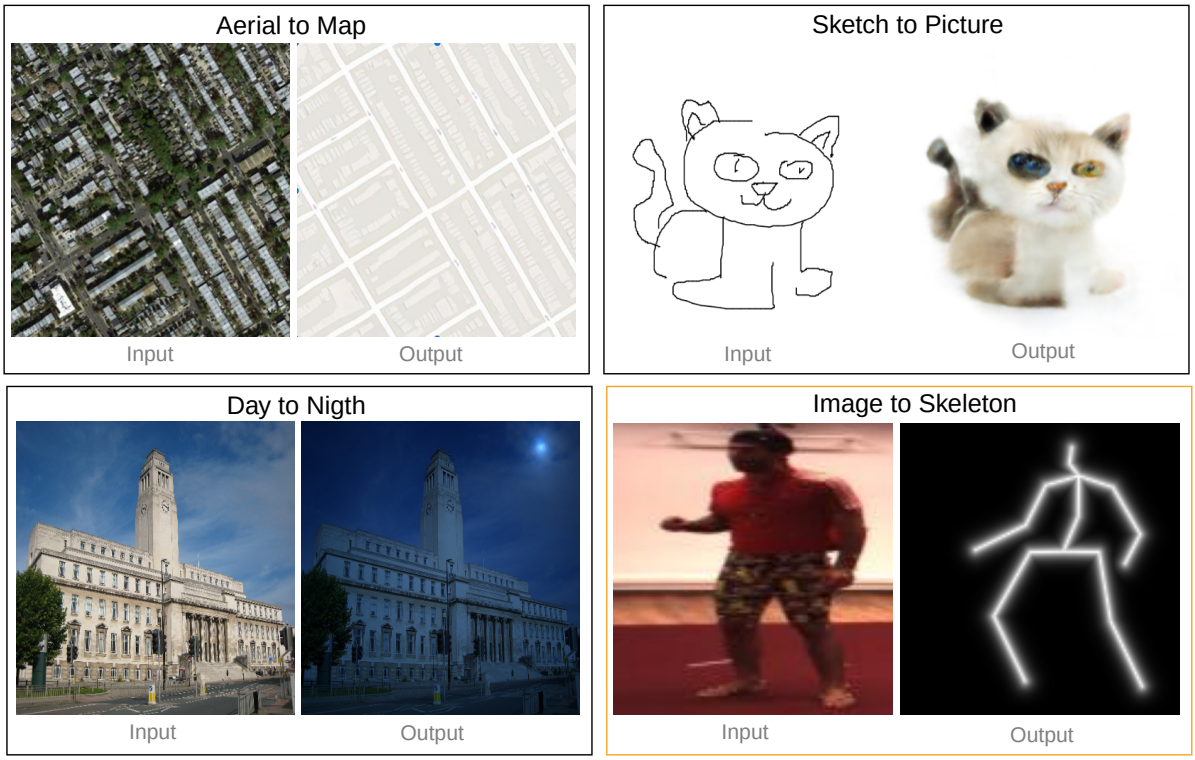


Figure 4.4: Image-to-image translation problems. Many computer vision problems can be addressed by transforming an image into another image (image-to-image translation). For example, it could be possible to map from aerial images to maps, from sketches to actual pictures, from daylight scenes to nightlight photographs, and more specifically to this chapter, from image to skeleton image. Some examples taken from [220].

networks (GANs). In particular, a type of GAN called conditional GANs is preferred as it allows for learning a conditional generative model for the data. In this scenario, the condition is based on producing an output image using an input from a distinct domain [220]. In other words, when training conditional GANs for image-to-image translation, a generator G learns to fool the discriminator D by producing realistic-looking versions of the input x. At the same time D learns to classify the predicted image $G(x)$ as real or fake. The big difference of this approach with normal GANs (or unconditional) is that both generator G and discriminator D observe the images involved in the mapping.

The setting from the original idea, as shown in Figure 4.5, requires access to paired data to learn the mapping/translation. However, further implementations show that learning this mapping is still possible when discarding the pairing [223, 224, 121]. In practice, the generator G is typically implemented as a modified version of an encoder-decoder network [225], adding skip connections and following the general shape of a U-Net [226]. While the discriminator D is often defined as a convolutional PatchGAN [220, 227]. Refer to [220, 223, 121] for a more

comprehensive review of the architectural choices.

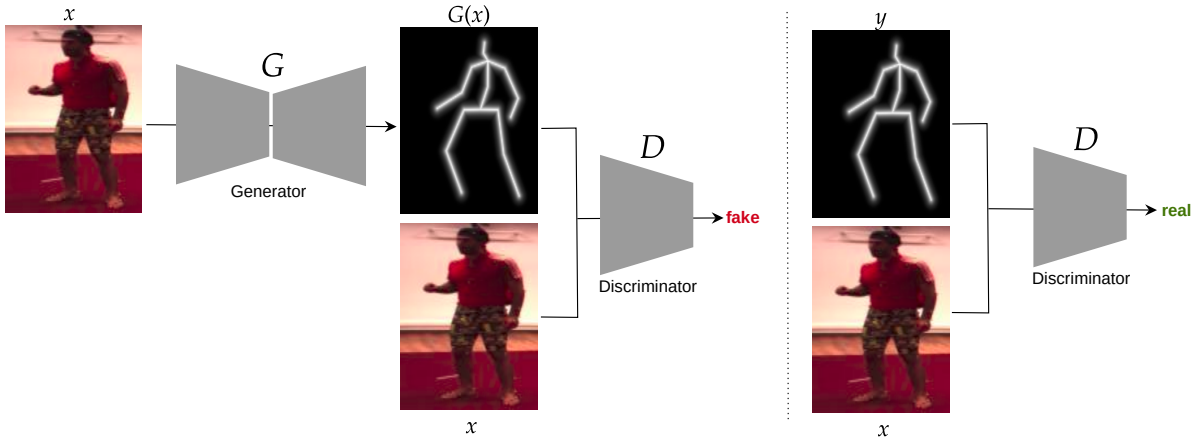


Figure 4.5: Conditional image-to-image mapping. The discriminator D learns to classify (image, skeleton) tuples as fake or real, while the generator G tries to fool the discriminator. Unlike standard GANs, the discriminator looks at both the image x and the skeleton y.

4.2.3 Normalising flows

Normalising flows (NF) are a group of generative models for learning distributions where density evaluation and sampling can be efficient and accurate. A normalising flow transforms a simple distribution (like a normal distribution) into a more complex one using a sequence of differentiable and invertible mappings [228].

According to [229], the idea of a flow-based model is to express a D-dimensional real vector $\mathbf{x}$ as a transformation T of a real vector $\mathbf{u}$ sampled from $p_u(\mathbf{u})$, i.e.:

$$\mathbf{x} = T(\mathbf{u}) \quad \text{where} \quad \mathbf{u} \sim p_u(\mathbf{u}) \tag{4.1}$$

In this context, $p_u(\mathbf{u})$ represents the base distribution of the flow-based model. To make the model work, the transformation T must be invertible, and T and T^{-1} must be differentiable, requiring $\mathbf{u}$ to be D-dimensional [230] too. Given these conditions, the density of $\mathbf{x}$ can be calculated by a change of variables:

$$p_x(\mathbf{x}) = p_u(\mathbf{u}) |\det J_T(\mathbf{u})|^{-1} \quad \text{where} \quad \mathbf{u} = T^{-1}(\mathbf{x}) \tag{4.2}$$

Similarly, we can also express $p_x(\mathbf{x})$ with respect to the Jacobian of T^{-1}:

$$p_x(\mathbf{x}) = p_u(T^{-1}(\mathbf{x}))|\det J_{T^{-1}}(\mathbf{x})| \tag{4.3}$$

Finally, the Jacobian $J_T(\mathbf{u})$ can be obtained by creating the $D \times D$ matrix consisting of all the partial derivatives of T. Usually, T or T^{-1} is implemented as a neural network while assuming $p_u(\mathbf{u})$ as a normal distribution.

NF has various applications for different tasks, including image generation [231], noise modelling [232], and improvement of reinforcement learning [233]. Flow-based methods have also been explored in pose estimation problems, mainly to learn prior distributions of 3D human poses [234, 235]. Further approaches, such as [146], incorporate images to condition the posterior distribution of the 3D poses. More recently, the method in [55] uses normalising flow to infer the probability of a reconstructed 3D pose solely from a prior distribution of 2D poses, therefore removing the need for 3D training data as in previous works. The method incorporates a flow-based model within an existing technique for lifting 2D poses to 3D [54], resulting in more accurate 3D predictions.

4.3 Method

4.3.1 Method overview

Our proposed model for estimating 3D human pose consists of a pipeline of three networks - Φ, Ω, and Λ - mapping from full body images to 3D pose. This can be seen in the blue dotted box in the upper-left part of Figure 4.6. In particular, Φ is implemented as a CNN based on image-to-image translation networks, translating from an input image x to an intermediate skeleton image s. Ω is another CNN mapping from s to a 2D pose representation y. Finally, Λ is a fully connected network that lifts the 2D pose y to the required 3D pose v. The 3D pose is depicted through an articulated structure of 3D line segments corresponding to the human body's parts, e.g., the head, torso, upper arm, and foot.

To train the three networks, we combine them into a more extensive network (as shown in Figure 4.6) and optimise it end-to-end. This network includes a loop of transformations for the 3D pose, where the degree of geometric consistency between these transformations contributes to a loss function and provides self-supervision of the training. The training starts with unlabelled images depicting people in different poses from benchmark datasets. We also assume we have a

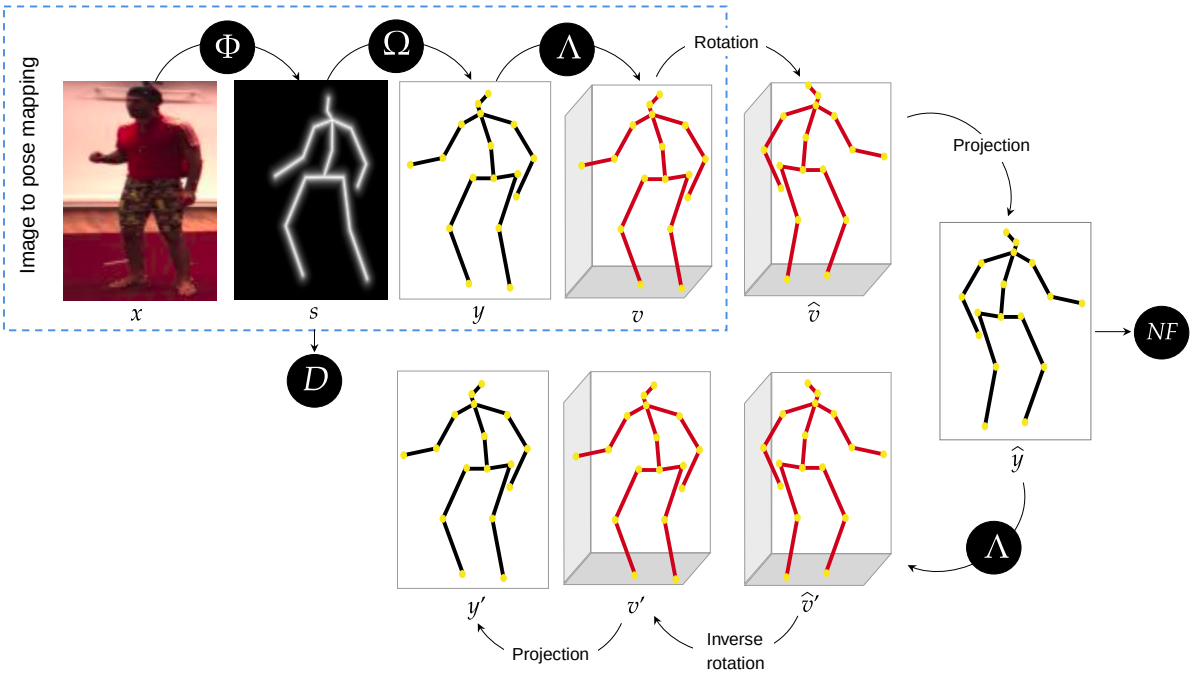


Figure 4.6: Self-supervised architecture for estimating the 3D pose of a person. Our method aims to map an image of a person to its 3D pose. It consists of an image-to-pose mapping (upper left dotted blue box) given by an image-to-image translation and a lifting process. The self-supervision comes from incorporating these networks within a bigger structure, exploiting the notion of geometric self-consistency (rotations and projections). It also uses a conditional GAN that evaluates the image-skeleton mapping. Additionally, an NF improves the geometric transformations, producing more plausible 3D representations.

(normally unrelated) dataset of typical 2D poses to be converted into skeleton images utilising a differentiable function, κ. Furthermore, our model relies on a conditional GAN framework, where the discriminator D evaluates the realism of the generated skeleton images s via the generator $\Phi(x)$ with those created by κ. In the following sections, we provide more details about the components of our model.

4.3.2 Image to 3D pose mapping

The image-to-pose mapping is the composition of the networks Φ, Ω, and Λ to map an image x depicting a person to its 3D pose representation v. The first part of the mapping is a CNN Φ, which maps from the image x to a skeleton image $s = \Phi(x)$, showing the person as a stick figure. We implement the Φ network as an image-to-image translation network, similar to the architecture of [220, 119]. Along with other components that we discuss later, this formulation can learn how to map images from one domain to another, i.e., from actual pictures depicting people to their skeleton image representation. Contrary to the standard design of

encoder-decoder networks for image-to-image translation tasks, Φ uses skip connections, and it is fully convolutional (without the bottleneck of having fully connected layers), allowing better alignment of the individual's geometry on the input x to its skeleton image representation s, which is beneficial for the further pose estimation task.

In the next part of the image-to-pose mapping, the network Ω maps the output skeleton image s from $\Phi(x)$ to a 2D pose representation $y = \Omega(\Phi(x))$. In other words, Ω learns to extract 2D joint positions $(\mathrm{x}_i, \mathrm{y}_i)$ from the skeleton image. The implementation of this network follows the one proposed in [119, 224]. During the final stage of the mapping, a fully connected neural network, Λ, lifts the given 2D pose y to the required pose v in 3D (See Appendix B for more details). In particular, $\Lambda(y)$ estimates the depth $\mathrm{z}_i = d_i + \Delta$ as the offset d_i to a constant depth Δ for each pair of $(\mathrm{x}_i, \mathrm{y}_i)$ joint positions in the input y. Then, the 3D location of joint v_i in the 3D pose v is given by

$$v_i = (\mathrm{x}_i \mathrm{z}_i,\ \mathrm{y}_i \mathrm{z}_i,\ \mathrm{z}_i) \tag{4.4}$$

where z_i is forced to be larger than one, to neutralise ambiguity from negative depths. In line with previous works [54, 56, 55], Δ is fixed to 10.

The lifting network Λ is based on the work of [16, 54] and extended following [55]. In this context, the extended version of the network estimates the depth z_i for each joint position in the input and predicts a value for the elevation angle α. This angle is useful when performing the rotations of the 3D pose v within the loop for geometric consistency. Specifically, we use α to fix the elevation angle of the vertical axis to the ground plane about which the rotation is performed.

4.3.3 Pose prior and discriminator

We encourage the generator network $\Phi(x)$ to produce realistic-looking skeleton images with the help of a prior of 2D poses $\{u_j\}_{j=1}^{M}$. To assist the image-to-image translation process, the 2D poses from the prior are rendered to skeleton images using the differentiable function κ proposed in [119]. It's important to note that these 2D poses are not annotations of the training images.

Let C be a set of connected joint pairs (i, j), e an image pixel location, and u a set of x and y coordinates of body joint positions. The skeleton image rendering function is given by:

$$\kappa(u)_e = \exp\left(-\gamma \min_{(i,j)\in C, r\in[0,1]} ||e - ru_i - (1-r)u_j||^2\right) \tag{4.5}$$

Informally κ defines a distance field from the line segments linking joints and applies an exponential fall-off to create the image. Note that this is the same as the rendering function defined in Chapter 3.

As per Jakab's approach [62], we utilise a discriminator network D to encourage the generator $\Phi(x)$ to produce plausible skeleton representations. More specifically, the task of D is determine whether a skeleton image $s = \Phi(x)$ looks like an authentic skeleton image such as those in the prior $z = \kappa(u)$. Formally, the objective is to learn $D(s) \in [0, 1]$ to match between the reference distribution $p(z)$ given by the unpaired skeleton images in the prior $\{z_j = \kappa(u_j)\}_{j=1}^{M}$ and the distribution $q(s)$ given by the predicted skeleton image samples $\{s_i = \Phi(x_i)\}_{i=1}^{N}$. A difference adversarial loss compares the unpaired samples z and the predictions s:

$$\mathcal{L}_D = \frac{1}{M}\sum_{j=1}^{M} D(z_j)^2 + \frac{1}{N}\sum_{i=1}^{N}((1 - D(s_i))^2 \tag{4.6}$$

4.3.4 Random rotations and projections

Another essential component of our model is the lifting process which helps to accurately learn a 3D pose v from the estimated 2D pose y. Since our self-supervised approach does not incorporate any 3D data for supervision, we simulate a second virtual view of the 3D pose v by randomly rotating it $\hat{v} = R * v$. Previous work [54] builds the rotation matrix R by uniformly sampling azimuth and elevation angles from a fixed distribution, usually from $[-\pi, \pi]$ and $[-\pi/9, \pi/9]$ respectively. However, [55] demonstrates that learning the distribution of the elevation angles leads to better results. Therefore, we follow their approach and utilise the network Λ to estimate the elevation angle (along with the depth predictions). We then use the distribution of predicted elevation angles to build the needed rotation matrix R for performing the rotation.

In line with [55], we predict the dataset's normal distribution of elevation angles R_e by calculating a batch's mean μ_e and standard deviation σ_e from the estimate values by $\Lambda(y)$. We then sample from the normal distribution $\mathcal{N}(\mu_e, \sigma_e)$ to rotate the pose v in the elevation direction R_e. The rotation around the azimuth axis R_a is simply chosen from a uniform distribution $[-\pi, \pi]$. Finally, the complete rotation matrix R is given by:

$$R = R_e^T R_a R_e \tag{4.7}$$

After rotating the 3D pose, we project $\hat{v}$ through a perspective projection. Then, the same lifting network $\Lambda(\hat{y})$ produces another 3D pose $\hat{v}'$ which is then rotated back to the original view. The final 3D pose v' is projected to 2D using the same perspective projection. This loop of transformations of the 3D pose helps to self-supervise the training. In this scenario, based on the notion of geometric self-consistency (see subsection 4.2.1), we assume that if the lifting network Λ accurately estimates the depth for the 2D input y, then the 3D poses $\hat{v}$ and $\hat{v}'$ should be similar. The same principle applies to y and the final 2D projection y'. This gives the following two components of the loss function:

$$\mathcal{L}_{3d} = ||\hat{v}' - \hat{v}||^2 \tag{4.8}$$

$$\mathcal{L}_{2d} = ||y' - y||^2 \tag{4.9}$$

Under the same argument, it can be assumed that 3D poses v and v' are comparable. In this case, instead of comparing the representations involved directly, we adopt [56, 55] to measure the change in the difference (or deformation) in 3D pose between two samples j and k from a batch at corresponding stages in the network. The resulting loss term is expressed as:

$$\mathcal{L}_{def} = ||(v'^{(j)} - v'^{(k)}) - (v^{(j)} - v^{(k)})||^2 \tag{4.10}$$

Similar to Wandt [55], we do not assume samples j and k are from the same sequence; these may come from different sequences and subjects.

4.3.5 Normalising flow

The notion behind normalising flow (NF) is to transform a simple distribution (e.g. a normal distribution) into a complex one so that the density of a sample under this complex distribution can be easily computed. NF has been successfully used for 3D pose estimation tasks, as discussed in subsection 4.2.3. However, Wandt [55] introduces the idea of utilising NF for learning a 3D

prior distribution solely from 2D data. We follow this approach and incorporate NF within our method to enhance its overall performance. It helps to learn the distribution of elevation angles for the rotation based on a prior distribution over 2D poses. Therefore, it also contributes towards the lifting process and the geometric self-consistency.

Let $Z \in \mathbb{R}^N$ be a normal distribution and g an invertible function $g(z) = \bar{y}$ with $\bar{y} \in \mathbb{R}^N$ as a projection of the 2D human pose vector $\hat{y}$ in a PCA subspace. By a change of variables, the probability density function for $\bar{y}$ is given by

$$p_Y(\bar{y}) = p_Z(f(\bar{y}))\left|\det\left(\frac{\delta f}{\delta \bar{y}}\right)\right| \tag{4.11}$$

where f is the inverse of g and $\frac{\delta f}{\delta \bar{y}}$ is the Jacobian of f. Following the normalising flow implementation in [55] (see Appendix B for more details), we represent f as a neural network [236] and optimise with the negative log likelihood loss:

$$\mathcal{L}_{NF} = -\log(p_Y(\bar{y})) \tag{4.12}$$

4.3.6 Additional losses

We use the same loss function as in [62] to learn the mapping between the skeleton image and the 2D pose $y = \Omega(s)$. However, we do not pre-train Ω and instead learn it from scratch simultaneously with all other networks. The loss term $\mathcal{L}_\Omega$ for learning Ω is then given by:

$$\mathcal{L}_\Omega = ||(\Omega(\kappa(u)) - u)||^2 + \lambda||(\kappa(y) - s)||^2 \tag{4.13}$$

where u represents a 2D pose from the unpaired prior, s is the predicted skeleton image, and λ is a balancing coefficient set to 0.1. The function κ is the skeleton image renderer defined in Equation 4.5. This loss term takes advantage of the duality of representing the pose as either a skeleton image or a set of 2D positions. Specifically, the second term of Equation 4.13 enables the learning of poses that appear in the training images but may not be part of the prior.

Based on the proven effectiveness of incorporating relative bone lengths into pose estimation methods [55, 237, 238], we add this to impose a soft constraint when estimating the 3D pose. Following the formulation in [55], we calculate the relative bone lengths b_n for the n-th bone

divided by the mean of all bones of a given pose v. We use a pre-calculated relative bone length $\bar{b}_n$ as the mean of a Gaussian prior. Then, the negative log-likelihood of the bone lengths defines a loss function $\mathcal{L}_{bl}$,

$$\mathcal{L}_{bl} = -\log(\prod_{n=1}^{N} \mathcal{N}(b_n|\bar{b}_n, \sigma_b)) \tag{4.14}$$

where N is the number of bones defined by the connectivity between joints. Note that this is a soft constraint that accommodates variation in the relative bone lengths between individuals. It does not establish any predetermined lengths for bones.

4.4 Experiments

4.4.1 Datasets

Human3.6M: Human3.6M [47] is a widely used large-scale pose dataset consisting of videos of eleven subjects (six male and five female) doing fifteen activities against a static background, as illustrated in Figure 4.7. These activities intend to capture a wide variety of poses, for example, when the people are walking, waiting, smoking, taking photos, eating, posing, giving directions, and sitting. The dataset contains 3.6 million images depicting the human body and corresponding 2D and 3D body pose annotations.

Two protocols have been established to train and evaluate pose estimation methods with this dataset - *protocol I* and *protocol II* [47]. In line with the standard protocol II on Human3.6M, we use images from videos of subjects S1, S5, S6, S7, and S8 for training; and testing with images from subjects S9 and S11. Unfortunately, some subjects had to be excluded due to data privacy issues. We pre-processed the video data to obtain the images by cropping the human body on each frame and removing the background, using the bounding boxes and segmentation masks provided in the dataset.

MPI-INF-3DHP: MPI-INF-3DHP [49] is another popular dataset in the human pose estimation literature. Like the Human3.6M dataset, it includes videos featuring people doing certain activities and corresponding 3D and 2D pose annotations. However, MPI-INF-3DHP incorporates recordings captured in three different settings, as shown in Figure 4.8: studio with green screen, studio without green screen, and outdoors.

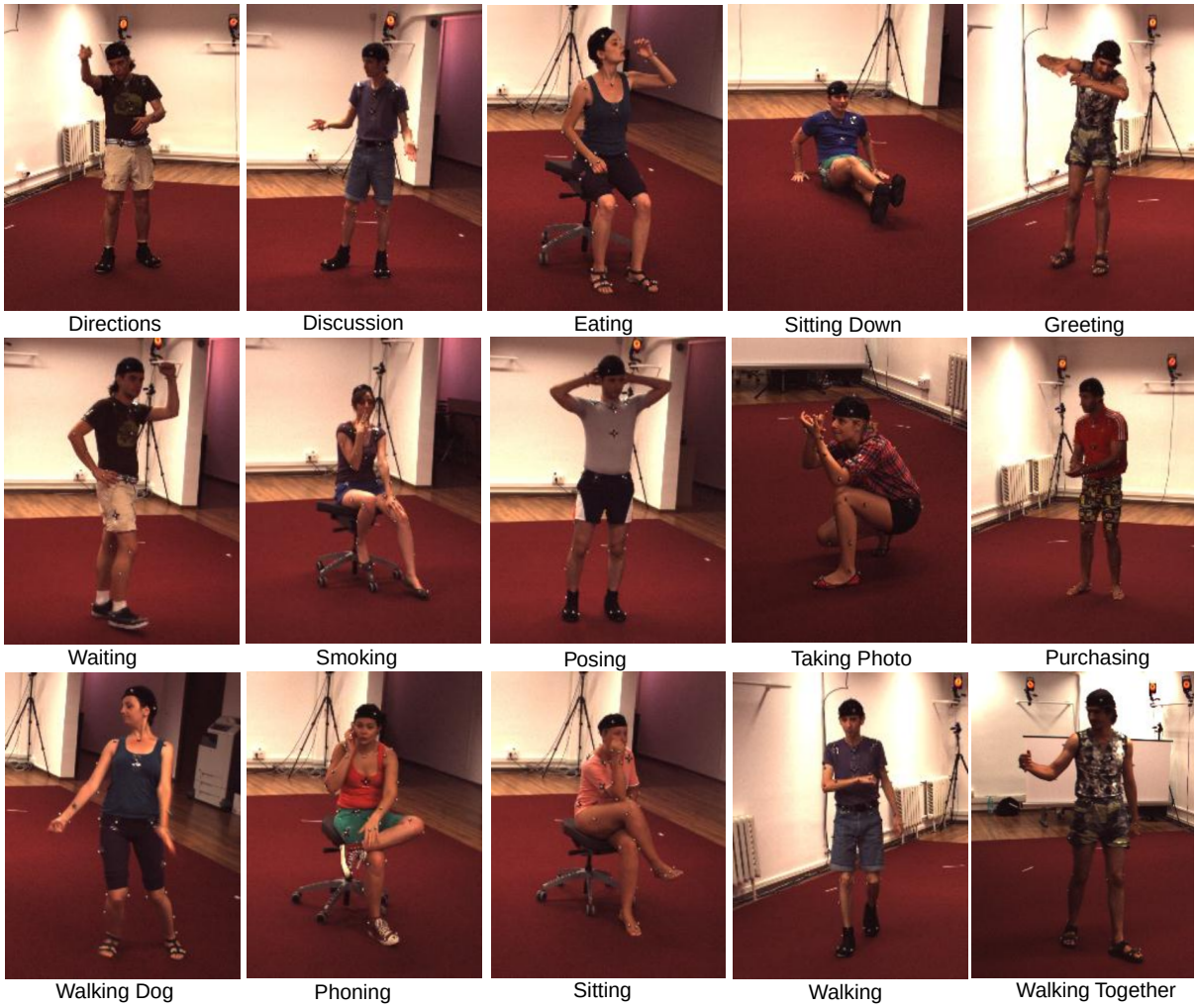


Figure 4.7: Human3.6M dataset. Representative frames for each activity on Human3.6M. Image reproduced from http://vision.imar.ro/human3.6m/description.php

Figure 4.8: MPI-INF-3DHP dataset. Representative frames from test split of MPI-INF-3DHP dataset. While the Human3.6M dataset features only studio images, the MPI-INF-3DHP dataset includes both indoor and outdoor settings.

The dataset comprises eight subjects with two video sequences for each, doing different activities, e.g. walking, sitting, exercising, and reaching. We use the same pre-processing as with Human3.6M videos. After that, we train the model using the extracted images from the videos in the train split and evaluate its performance with the provided test set.

Leeds Sports Pose Dataset (LSP): The Leeds Sports Pose dataset [29] is a widely utilised collection of images for human 2D pose estimation tasks. It is small compared with more recent data utilised for human pose estimation, such as Human3.6M or MPI-INF-3DHP datasets. It comprises only 2,000 images depicting humans in various sports, including athletics, badminton, baseball, gymnastics, parkour, soccer, tennis, and volleyball. Despite its size, the dataset offers a diverse range of non-standard human poses and appearances, as demonstrated in Figure 4.9. Note that the LSP dataset does not include any 3D annotations; only 2D pose annotations are provided, specifying the position of 14 joints in each image.

Figure 4.9: Examples of images from the LSP dataset. Although LSP is smaller compared to recent datasets for human pose estimation, it contains mostly non-standard human poses, as observed in the pictures.

HandDB: HandDB [222] is a dataset of images depicting human hands under different scenarios. Since most images show both hands or other body parts that could be counterproductive when training the model, we then decide to utilise only a portion of the subset created from synthetic data. This subset exhibits more homogenous backgrounds and hand sizes. It also includes 2D annotations for 21 key points, distributed as follows: four for each of the five fingers and one for the wrist. To train and test our model, we select two image sequences - synth2 and synth3 - out of the four available in this subset and split them 80/20, respectively. Figure 4.10

shows some random examples from two sequences of the synthetic subset of HandDB dataset.

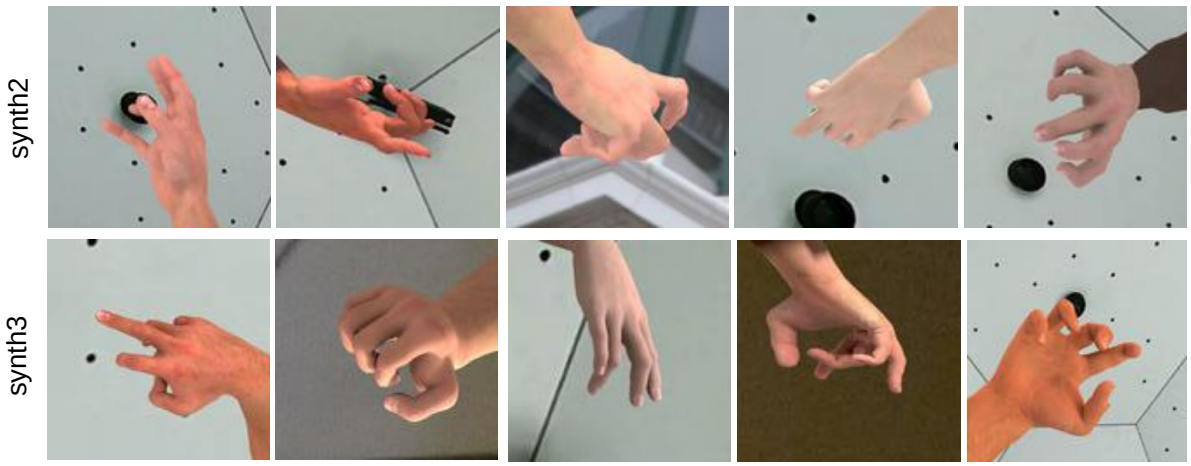


Figure 4.10: Examples from the synthetic set of HandDB dataset. We only utilise *synth2* and *synth3* sets from the available synthetic data from the HandDB dataset. As can be noticed from the pictures, this data is consistent in terms of camera perspective.

4.4.2 Evaluation metrics

Following previous methods [16, 54, 145, 152, 153], we use the standard *protocol II* to quantitatively evaluate our trained model using the test set from the Human3.6M dataset [47]. This protocol relies on the Procrustes method [239] to perform a rigid alignment between the predicted 3D pose and the 3D ground truth. Then, it calculates the Mean Per Joint Position Error (MPJPE), which takes the average Euclidean distance between the ground-truth joint positions and the corresponding estimated positions across all 17 joints [47]. For simplicity, we refer to this metric as P-MPJPE, where P stands for the Procrustes method. Assuming that we have a pose predictor f that estimates a 3D pose v given an image x and a skeleton $\bar{V}$ obtained after Procrustes alignment of v with the ground truth skeleton V', then the P-MPJPE is defined as follows

$$\text{P-MPJPE} = \frac{1}{j}\sum_{i=1}^{j} \|m_{\bar{V}}(i) - m_{V'}(i)\|_2 \tag{4.15}$$

where j is the number of joints in $\bar{V}$, and $m_{\bar{V}}(i)$ is a function that returns the coordinates of the *i-th* joint of $\bar{V}$. Similarly, $m_{V'}(i)$ get the coordinates of the *i-th* joint of the ground truth skeleton V' in x.

For a quantitative evaluation of the model performance with the MPI-INF-3DHP dataset, we

use the Percentage of Correct Keypoints (PCK) metric. This metric measures the percentage of estimated joint positions within a fixed distance from their ground truth. Formally, PCK is given by

$$\text{PCK} = \sum_{i=1}^{N} \frac{\delta(d_i <= T)}{N} \tag{4.16}$$

where d_i represents the Euclidean distance between the predicted and ground-truth skeletons for the $i\text{-}th$ joint position, T is a fixed threshold set to $150mm$, N is the number of joints in the skeleton, and $\delta(*)$ is equal to 1 when the given condition is true, and 0 otherwise. Additionally, we report the corresponding area under the curve (AUC) calculated for a range of PCK thresholds.

4.4.3 Training procedure

We train the networks Φ, Ω, D, and Λ from scratch. Only the normalising flow NF is independently pre-trained, as indicated in [55]. The complete loss function for training our model has seven components expressed as $\mathcal{L}_D$ (Equation 4.6), $\mathcal{L}_\Omega$ (Equation 4.13), $\mathcal{L}_{2d}$ (Equation 4.9), $\mathcal{L}_{3d}$ (Equation 4.8), $\mathcal{L}_{def}$ (Equation 4.10), $\mathcal{L}_{NF}$ (Equation 4.12), and $\mathcal{L}_{bl}$ (Equation 4.14). For convenience in ablation studies, we group three of these loss terms and represent them as $\mathcal{L}_{base}$

$$\mathcal{L}_{base} = \mathcal{L}_{2d} + \mathcal{L}_{3d} + \mathcal{L}_{def} \tag{4.17}$$

Thus, the final composite loss function is defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_D + \mathcal{L}_\Omega + \mathcal{L}_{base} + \mathcal{L}_{NF} + \lambda_2 \mathcal{L}_{bl} \tag{4.18}$$

Where λ_1 and λ_2 are balancing coefficients set to 10. We experiment with other different hyper-parameters to further balance the components of the loss function presented in Equation 4.18. However, none of the combinations seemed to improve results.

We train our model by optimising the loss function from Equation 4.18. The batch size is set to 96, with each batch consisting of images and random samples from the prior of unpaired 2D poses (which are then transformed into skeleton images). We utilise the Adam optimiser [203]

with a learning rate of 2×10^{-4}, and $\beta_1 = 0.5, \beta_2 = 0.999$. The model we use for reporting results through this chapter was trained for around 40 hours using one GPU from an NVIDIA DGX-MAX-Q server. When making predictions, we only keep the pipeline composed of the trained Φ, Ω, and Λ networks illustrated in Figure 4.11. Additional details on the network architectures involved in our approach can be found in Appendix B.

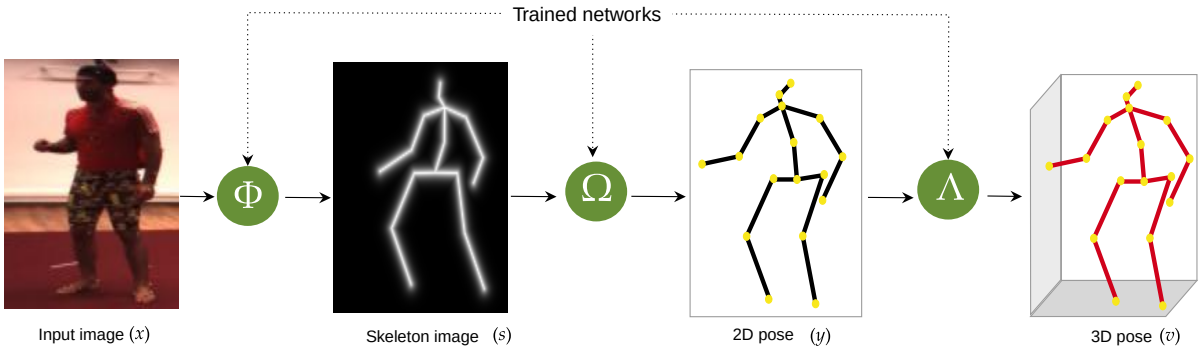


Figure 4.11: Networks used during inference for 3D human pose estimation. During the testing stage, we only require the trained networks responsible for the image-to-pose mapping (Φ, Ω, and Λ). The remaining networks (see Figure 4.6 for reference) are only necessary while training the model.

4.4.4 Results

Using our trained model with the Human3.6M dataset, we predict 3D poses consisting of 17 joint positions for all images from subjects S9 and S11 ($\sim$ 584k images). Then, we compute the average P-MPJPE across the activities in the test set. Table 4.1 compares our method with the state-of-the-art 3D pose estimation methods in terms of average P-MPJPE. We include supervised [219, 153], semi-supervised [58, 152, 57], and self-supervised [61, 59] approaches that estimate the 3D pose from images. These methods are more related to our work; however, all of these take different assumptions about data availability, especially 3D annotations. To the best of our knowledge, our method is the only one that relies on a lightweight premise, such as unpaired 2D poses, to estimate 3D poses straight from unlabelled images. For a more comprehensive comparison, we also consider supervised [16] and unsupervised [240, 54, 145, 56, 55] methods that estimate 3D pose directly from 2D poses. Note that given the nature of the inputs for those methods (an actual 2D pose is less ambiguous than an image), these exhibit better performance than the previous group (pose from images). Despite the minimum data assumptions for training our method, its performance exceeds that of previous methods that rely on 3D supervision [219], multi-view images [58, 152] or priors on 3D data [59].

Data Assumptions	Method	P-MPJPE(↓)
3D pose from 2D poses		
Full 3D	Martinez et al. [16]	52.1
Full 2D	Chen et al. [54]	68.0
Full 2D	Drover et al. [145]	64.6
Full 2D	Yu et al. [56]	52.3
Full 2D	Wandt et al. [55]	36.7
3D pose from images		
Full 3D	Chen et al. [219]	114.2
Full 3D	Mitra et al. [153]	72.5
Multi view + Parcial 3D	Rhodin et al. [58]	128.6
Multi View + Parcial 3D	Rhodin et al. [152]	98.2
Multi view + Full 2D	Wandt et al. [57]	53.0
Unpaired 3D	Kundu et al. [59]	99.2
Kinenatic 3D	Kundu et al. [61]	89.4
Unpaired 2D	Ours	96.7

Table 4.1: Comparison of average P-MPJPE (in mm's) for all activities in test set (S9 and S11) of Human3.6M dataset. We include two groups of methods: the ones that estimate 3D poses from 2D poses and those that estimate 3D poses directly from images. The methods in the second group are more related to our work. Since the assumptions about data for training are important for a fair comparison of our approach, the first column of the table indicates the data requirements of each method.

In addition, in Figure 4.12, we provide comparative per-activity quantitative results on Human3.6M. We show the P-MPJPE for each activity on the Human3.6M test set (subjects 9 and 11). As can be seen, the performance is compared with two state-of-the-art approaches for which per-activity data is available [219, 61]. Note that by only assuming unlabelled images and unpaired 2D poses for training, we achieve superior performance than [219]. Our method also outperforms [61], which incorporates 3D kinematic constraints, in 20% of the activities.

It is not common practice in the pose estimation literature to report quantitative results in terms of PCK with data from Human3.6M. However, we assess the 3D predictions with this metric to provide a more comprehensive evaluation. According to the result shown in Figure 4.13, most of the activities in the Human3.6M dataset's test set achieved high PCK scores, typically exceeding 80.0. This suggests that the estimated poses are accurate. In other words, a 3D pose with a PCK score of 80.0 implies that 80% of the joints have been accurately estimated.

We aim to demonstrate that the model is effective not just with one dataset but can also adapt to different conditions. Thus, we train and test our approach using data from the MPI-INF-3DHP dataset, also widely used in 3D pose estimation research. We create three different scenarios for performing training and evaluation with this data:

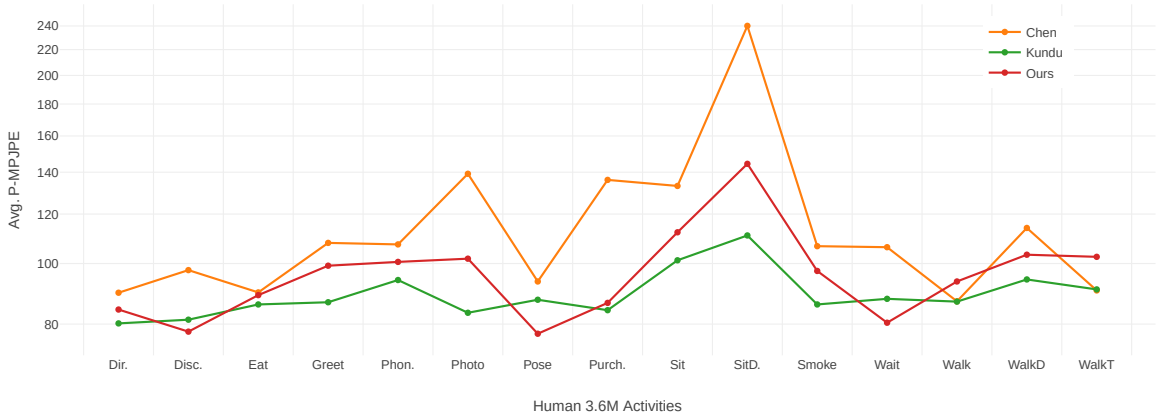


Figure 4.12: Distribution of P-MPJPE scores for each activity on the test set of Human3.6M dataset. We include only related works that provide per-activity scores.

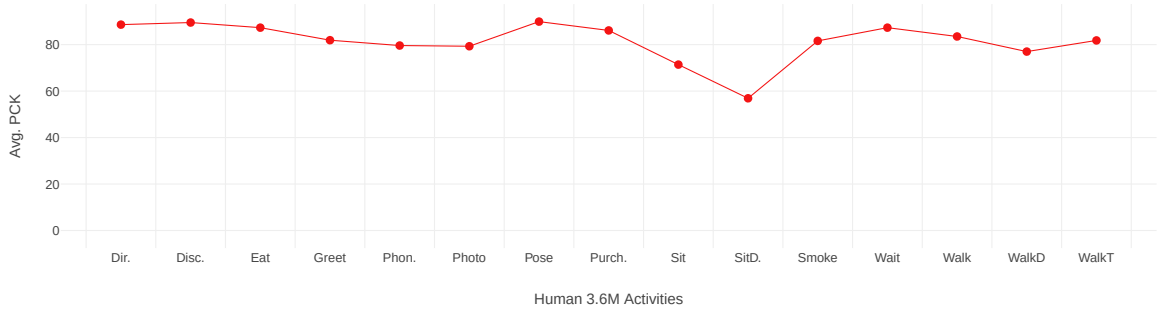


Figure 4.13: PCK scores for each activity in Human3.6M test set. Although PCK is not a standard metric in the literature to compare results with the Human3.6M dataset, we conduct an additional evaluation to provide more insights about the quantitative results of our method. PCK scores range from 0 to 100, with higher scores indicating better performance.

- Scenario #1: We train and test the model as usual, i.e., using the train and test set from MPI-INF-3DHP.
- Scenario #2: We keep the trained model with Human3.6M and evaluate it with the test set from MPI-INF-3DHP.
- Scenario #3: We combine the images in the training set of both datasets - Huma3.6M and MPI-INF-3DHP - and test with images from the MPI-INF-3DHP test set.

Regarding the unpaired prior needed for training, for all the scenarios except the second one, the unpaired prior of 2D poses is sourced from MPI-INF-3DHP. In the second case, it comes from Human3.6M. We keep the same pre-trained NF for all scenarios with data from Human3.6M. Table 4.2 compares the different evaluation conditions with the state-of-the-art (that reports

results on the MPI-INF-3DHP dataset) in terms of average PCK and AUC scores.

Data Assumptions	**Method**	**PCK(↑)**	**AUC(↑)**
Unpaired 3D	Kundu et al. [59]	83.2	58.7
Kinematic 3D	Kundu et al. [61]	79.2	43.4
Unpaired 2D	Ours (Scenario #1)	69.6	32.8
Unpaired 2D	Ours (Scenario #2)	58.7	24.3
Unpaired 2D	Ours (Scenario #3)	75.3	40.0

Table 4.2: Evaluation results on MPI-INF-3DHP dataset for the different training scenarios. First column shows the main assumption from each method. Ours (Scenario #1) represents the model trained with images from MPI-INF-3DHP. Ours (Scenario #2) indicates the model trained with Human3.6M and tested with MPI-INF-3DHP. Ours (Scenario #3) indicates that the MPI-INF-3DHP train set has been extended with images from Human3.6M.

The expectation is that the model performs better when the training images and the prior comes from similar distributions, i.e., from the same dataset as in Scenario #1. However, surprisingly our model performs best when combining images from Human3.6M and MPI-INF-3DHP datasets (Scenario #3). This suggests that increasing the number of training examples positively influences the overall performance, although the prior remains unchanged. This notion could be helpful when translating the model to other domains. For instance, in the animal domain, abundant unlabeled images are accessible, but gathering an extensive prior of 2D pose annotations may not be feasible.

Another important observation is that the model can perform decently even with data from a different dataset, such as in Scenario #2. Although we did not perform any fine-tuning on MPI-INF-3DHP for that particular case, our trained model with Human3.6M produces acceptable results for PCK and AUC metrics.

4.4.5 Qualitative evaluation

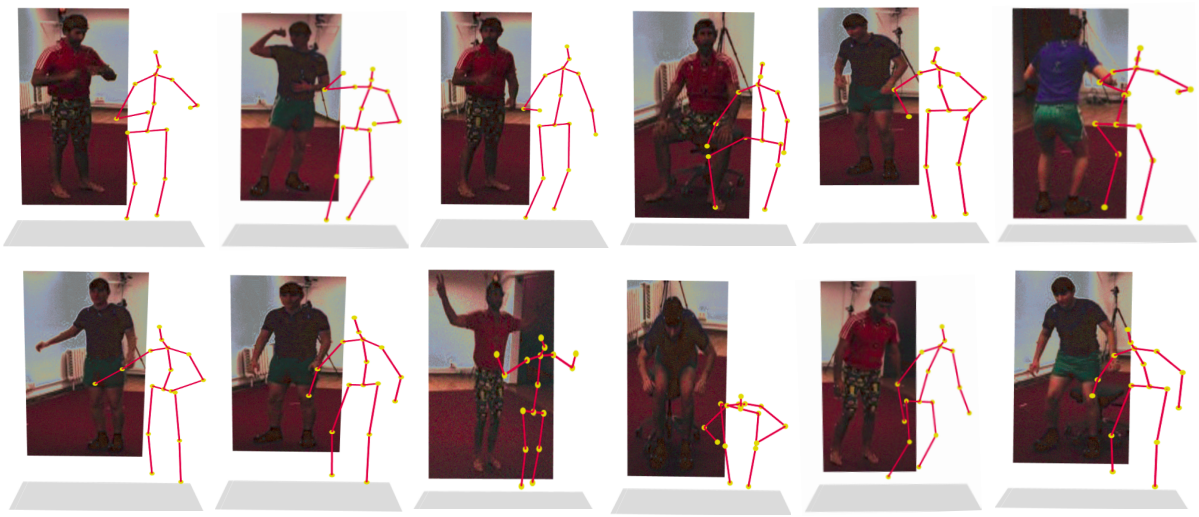


Figure 4.14: Qualitative results on images from Human3.6M dataset. Each figure contains the input image and its corresponding estimated 3D pose by our model. An extended version of qualitative results for this dataset is included in Appendix B.

We also evaluate qualitatively using the trained model with the test sets from Human3.6M and MPI-INF-3DHP datasets, respectively. Figure 4.14 and Figure 4.15 display predicted 3D poses aligned with their corresponding input images from both datasets. We choose the samples that best represent the range of data, such as various activities in Human3.6M or different recording settings in MPI-INF-3DHP.

Figure 4.15: Qualitative results on images from MPI-INF-3DHP dataset. Each figure contains the input image and its corresponding estimated 3D pose by our model. An extended version of qualitative results for this dataset is included in Appendix B.

In addition, we show a set of different visualisations of our 3D pose predictions in Figure 4.16.

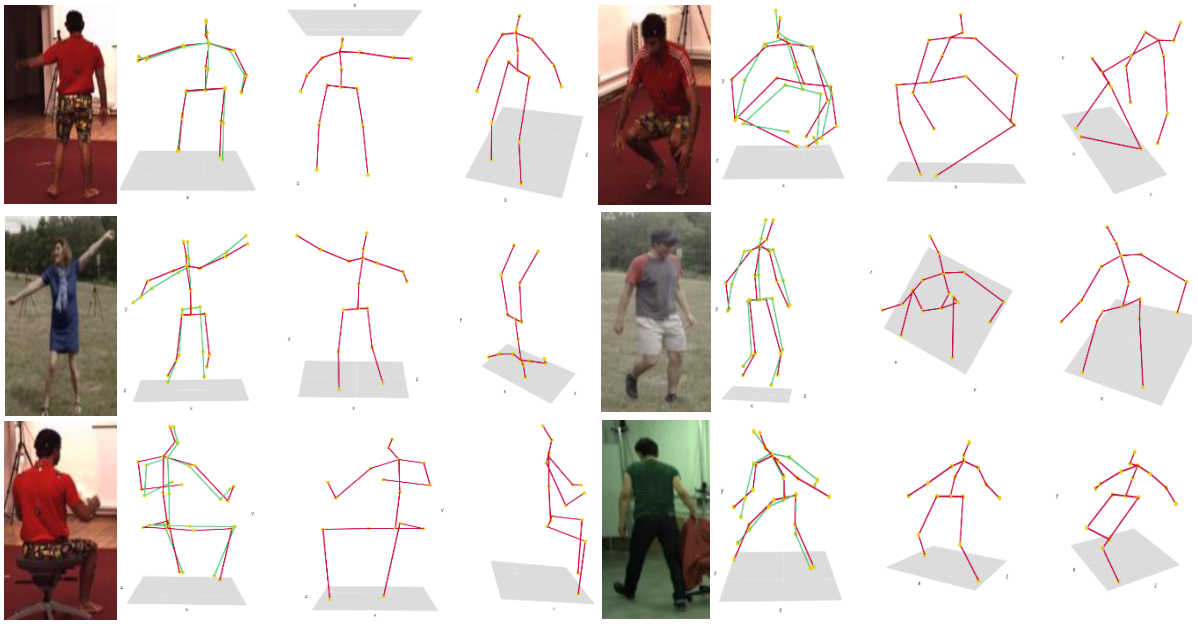


Figure 4.16: Ground truth and predictions from MPI-INF-3DHP and Human3.6M datasets. The first and fifth columns depict the input images, while the second and sixth columns show the corresponding estimated 3D poses (red) and their respective ground truth (green). The rest of the columns illustrate novel views of the 3D pose predictions.

This includes a visual comparison between the predicted 3D pose and its respective ground truth for a given image and novel views of the prediction. In Figure 4.16, columns two and five show that our model accurately predicts the 3D pose, even for challenging scenarios like sitting or outdoor environments. For more extensive visualisations, please refer to Appendix B.

4.4.6 Generalisation to unseen data

Since the Human3.6M and MPI-INF-3DHP datasets contain people depicting a similar range of poses, we use more challenging data to further demonstrate our approach's generalisation capabilities. In particular, we utilise data from the Leeds Sports Pose Dataset (LSP), which exhibits non-standard poses of people performing different sports under real-world scenarios. LSP is relatively small compared to the Human3.6M and MPI-INF-3DHP datasets, containing just 2,000 pictures.

While LSP lacks 3D pose annotations, we only provide visual representations of the estimated 3D poses. However, as the data from LSP has been annotated with 2D poses, we evaluate our intermediate 2D pose predictions against this ground truth and calculate the Percentage of Correct Points (PCK). This metric is similar to the one described in Equation 4.16, but we use a different threshold δ and consider 2D distances in this scenario. Note that we set the new δ

following the original implementation in [241].

To evaluate our model with the Leeds Sports Pose dataset, we randomly split this data into two parts. We utilise 1,000 images from LSP to test one version of our model purely trained with data from Human3.6M. Although not being trained with any related data to LSP, our method still estimates plausible 2D and 3D poses, as demonstrated in Figure 4.17. The figure shows the input images in the first and fifth columns, with the second and sixth columns displaying the 2D predictions (coloured in black) and ground truth (coloured in grey). Finally, the remaining columns exhibit the predicted 3D poses and corresponding novel views.

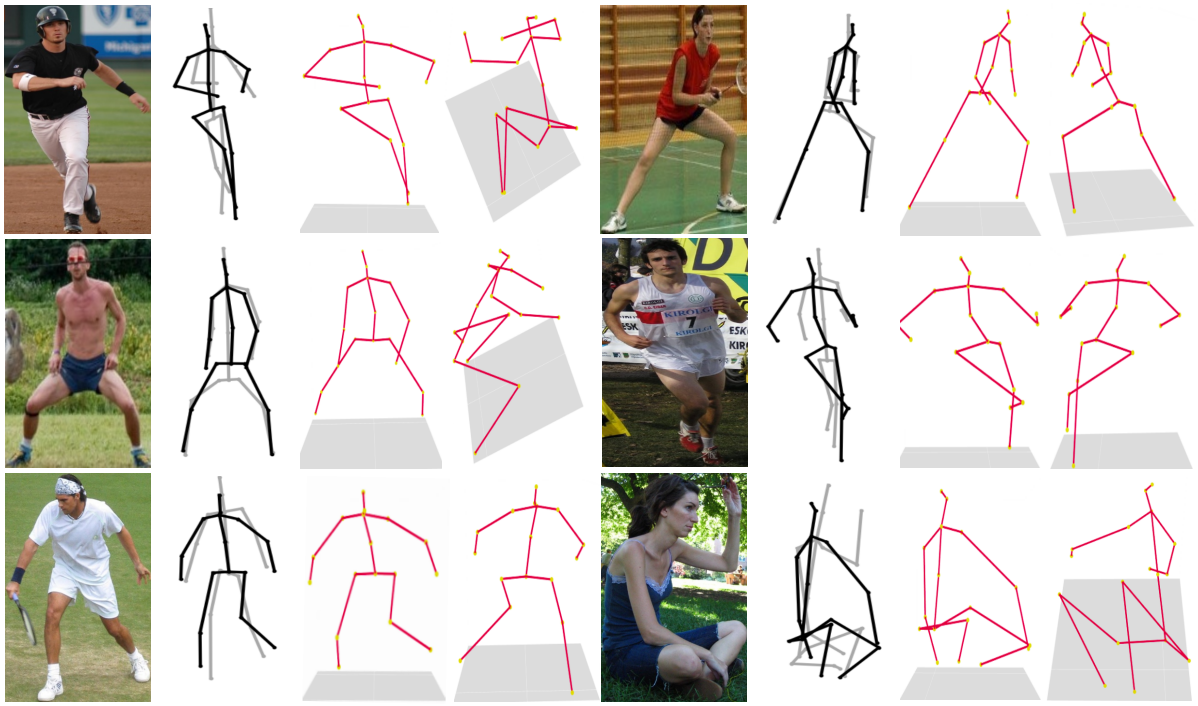


Figure 4.17: 3D and 2D pose Predictions from Leeds Sports Pose Dataset. The first and fifth columns depict the input images, while the second and sixth columns show the corresponding estimated 2D poses (black) and their respective ground truth (grey). The rest of the columns illustrate the predicted 3D poses and some novel views. More visualisations are provide in Appendix B.

Furthermore, we perform a quantitative evaluation of the predicted 2D poses and show some of the best results regarding PCK scores in Figure 4.18. The figure displays the input images, their corresponding predicted 2D pose (coloured in black), and ground truth (coloured in grey). The overall PCK score for the 1,000 estimated 2D poses is around 42%. This result is quite satisfactory, especially considering that the model has not been trained with that data distribution and, most notably when the test data comes from non-standard scenarios.

Additionally, we conduct an extra experiment where we fine-tune our model trained with Hu-

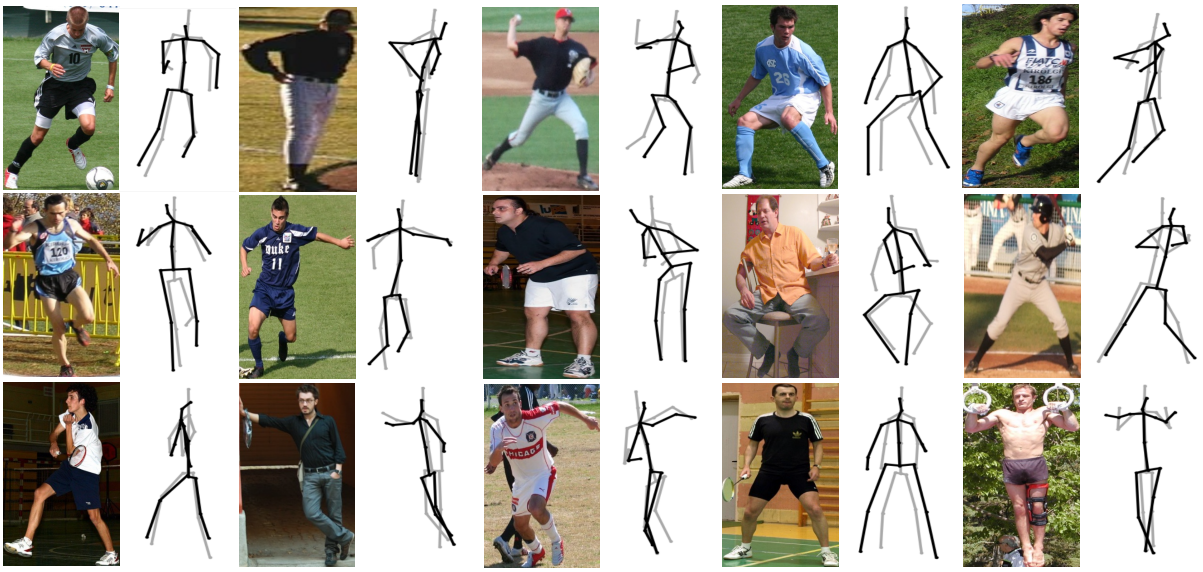


Figure 4.18: Estimated 2D poses with data from Leeds Sports Pose Dataset. Input images and their corresponding estimated 2D poses (coloured in black) and ground truth (coloured in grey).

man3.6M with the remaining 1,000 images from the LSP dataset. As expected, this data increases the performance (+6%) compared with the evaluation without fine-tuning the model. We include a comparison of the results from both evaluations in Figure 4.19. The green line represents the first evaluation, i.e. using the model solely trained with Human3.6M. The purple line corresponds to the version where the model has been fine-tuned with part of the data from the LSP dataset.

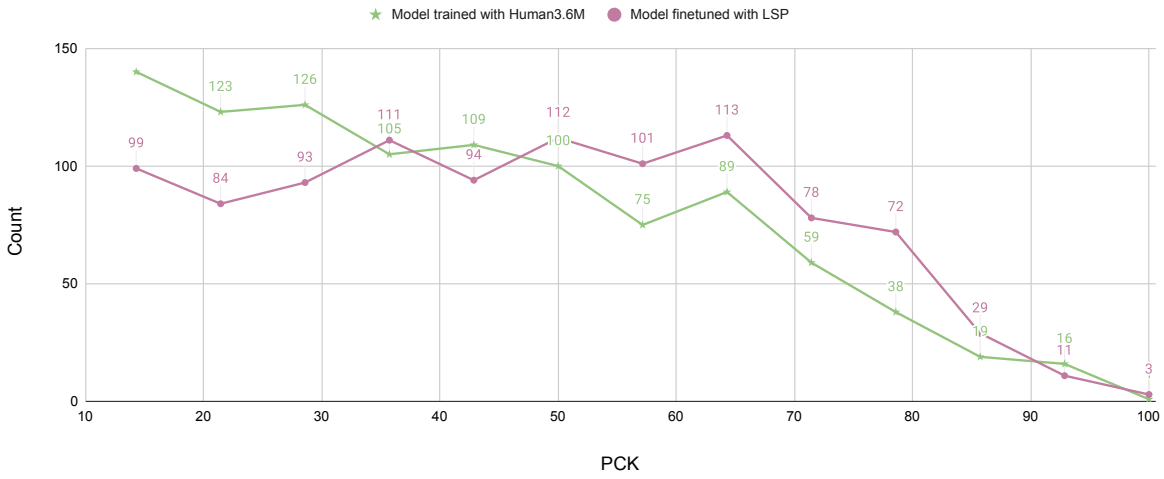


Figure 4.19: Comparison of PCK scores for the estimated 2D poses with LSP data. The green line represents the results obtained with the model trained with Human3.6M data. The purple line corresponds to the results with the fine-tuned version of the model. Fine-tuning the model with data from the LSP dataset increases the overall performance by 6% compared with the non-fine-tuned version (42%).

Evidently, training the model with close data distributions as the test set will yield better results, at least on the estimated 2D poses. Figure 4.19 illustrates that fine-tuning the model reduces the number of items with low PCK scores and increases the number of samples with high PCK scores. Finally, Figure 4.20 depicts some visual examples of the comparison. The first and fifth columns show the input images, while the second and sixth display the estimated 2D poses with the fine-tuned (purple) and non-finetuned (green) models and the corresponding ground truth (grey). We also include the PCK score for each of the estimated 2D poses. The subsequent columns exhibit the 3D pose predictions of each version of the model. Like the colours used for the 2D poses, green represents the results with the non-finetuned model, and purple corresponds to the results with the fine-tuned model. Note that in this case, we use a different set of colours for visualising the poses to differentiate both experiments easily.

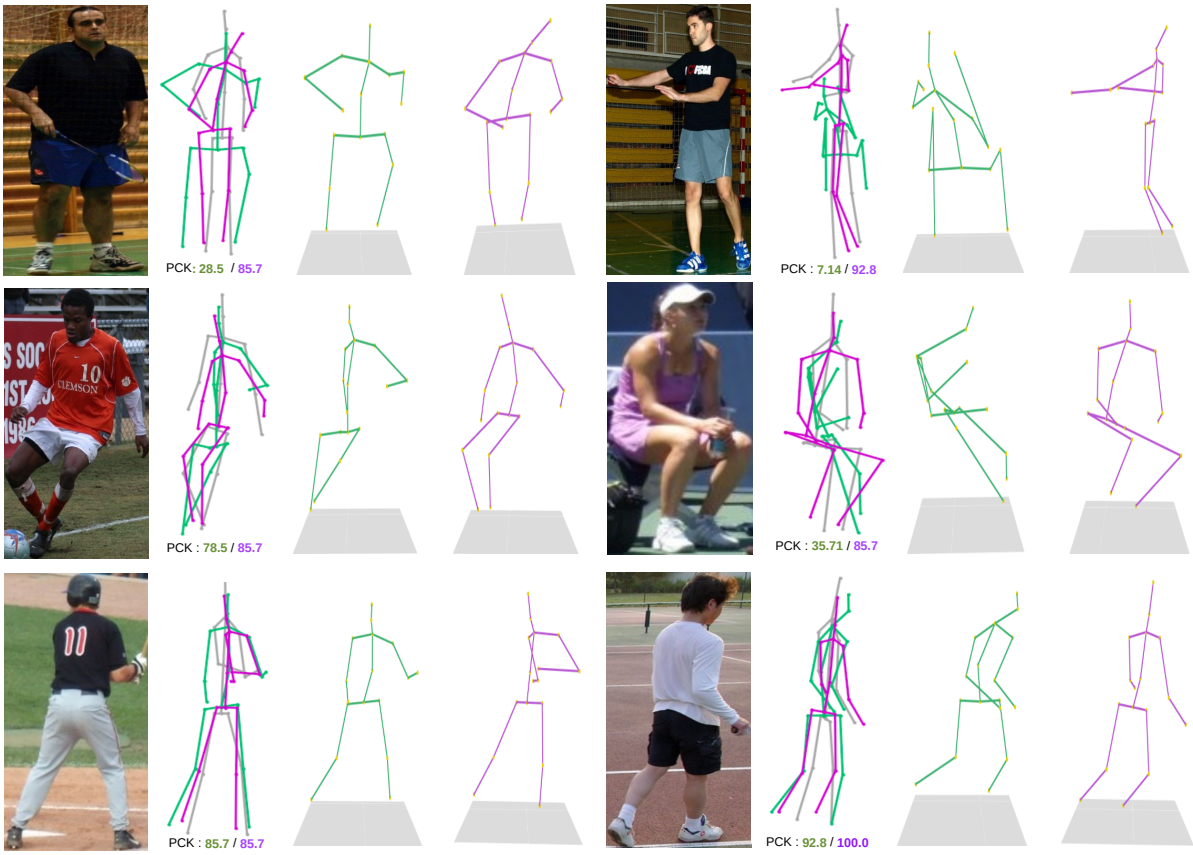


Figure 4.20: Comparison of estimated 2D and 3D poses with LSP data. The first and fifth columns depict the input images, while the second and sixth display the estimated 2D poses with the fine-tuned and non-fine-tuned models (coloured in purple and green respectively), along with the corresponding ground truth coloured in grey. The 2D predictions also include their PCK scores. The subsequent columns exhibit the 3D pose predictions of each version of the model; depicted in green the results with the non-finetuned model, and coloured purple the results with the fine-tuned model.

4.4.7 Application to different structures

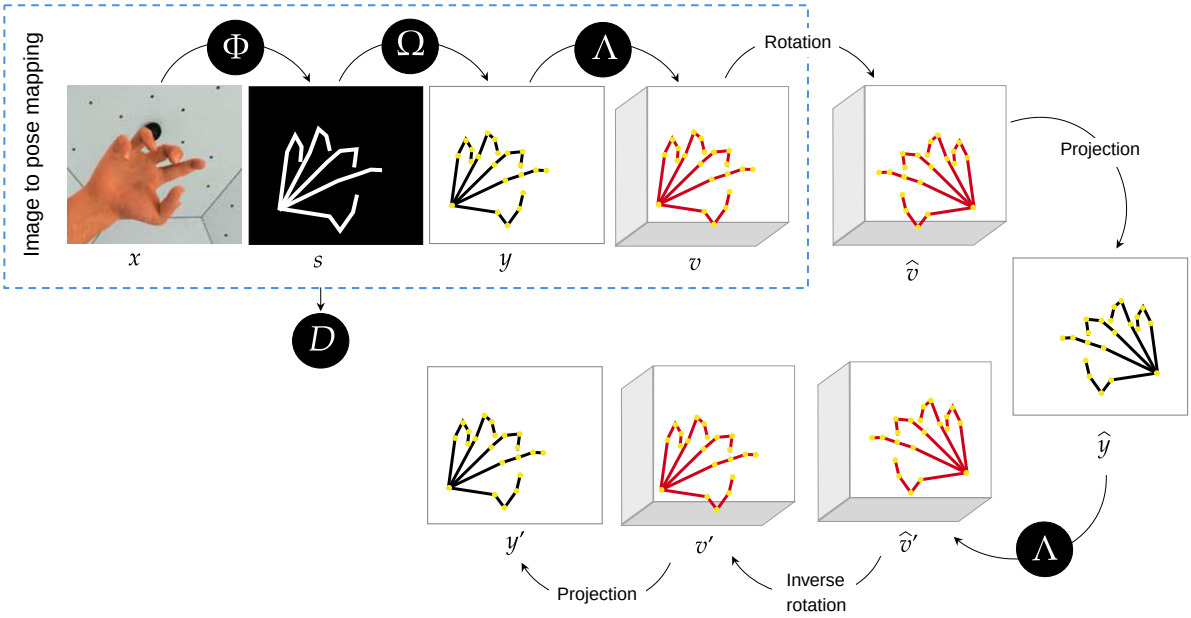


Figure 4.21: Model used for estimating 3D hand poses. We adapted our model to predict 3D hand poses. As can be noticed, the model is similar to the one used for estimating 3D human poses. Evidently, only the training images and poses in the prior are different.

Compared with previous works for 3D pose estimation that assume 3D annotations for training, our model has lower data requirements. It only needs unpaired 2D poses apart from the unlabelled images. This makes the approach flexible enough to work with structures different from the human body. We demonstrate our method's adaptability by training and testing it using data depicting human hands [222]. For building the train and test sets, we select images showing hands under similar conditions from synth2 and synth3 sequences. We further augment the training set offline by making two rotated versions of each image (45° and 90°). We use half of the 2D annotations provided with the dataset to build the prior of 2D hand poses, while the other half corresponds to images used for training. Figure 4.21 displays the components of the model. As can be noticed, all the elements are the same as those used to estimate 3D human poses, except for the NF component, which we remove to reduce the formulation's complexity even more. Furthermore, we reduce the amount of data for training; in this scenario, we use only around 5k unlabelled images and a similar number of 2D poses for the prior, representing a tiny portion when compared against the size of training sets for previous experiments.

Utilising the trained model, we estimate 3D hand poses consisting of 21 key points that represent hand joint positions. However, since the synthetic subset of HandDB does not contain 3D annotations, we report only qualitative results in Figure 4.22. This figure shows the input

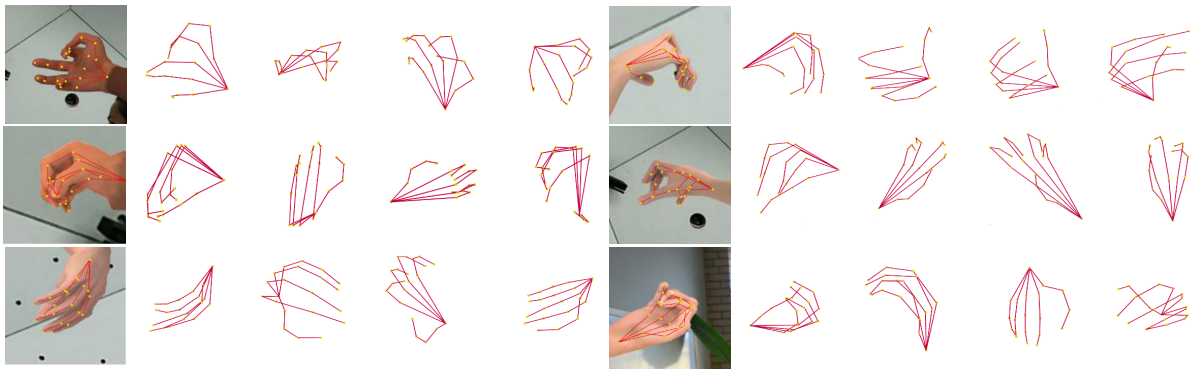


Figure 4.22: Qualitative results on HandDB dataset. First column shows the input image overlay with the 2D ground truth annotations from the dataset. Next columns display novel views of the 3D hand predictions.

images in the first and sixth columns. The second and seventh columns display the corresponding predicted 3D poses, while the remaining columns display novel views of the predicted 3D poses. Although we drastically reduce the data to train our method for this experiment, the visualisations show that the model can still accurately predict 3D structures that closely match the position of the hands in the input images.

4.4.8 Ablation study

We evaluate the effectiveness of the loss function design expressed in Equation 4.18 by progressively removing its components. We conduct the ablation studies using data from the Human3.6M dataset to train and test the model and identical hyperparameters. Since our loss function contains multiple terms, executing experiments to assess all the possible combinations is impractical. Therefore, we strategically perform three main experiments:

- Experiment #1 (E_1): Initially, we train and evaluate the model without considering the losses related to the geometric self-consistency cycle, i.e., only involving the losses $\mathcal{L}_D$ and $\mathcal{L}_\Omega$. As anticipated, even when the 2D predictions are mostly accurate, the overall performance decreases since there is nothing else to regulate the 3D predictions, and these are more susceptible to being deformed or flat.

- Experiment #2 (E_2): For the second experiment, we keep the $\mathcal{L}_D$, $\mathcal{L}_\Omega$, and $\mathcal{L}_{base}$ losses and exclude $\mathcal{L}_{NF}$ and $\mathcal{L}_{bl}$. In this scenario, the 3D representations are regulated by the loss term $\mathcal{L}_{base}$, which ensures consistency. Therefore, the model improves its performance w.r.t. the first experiment and estimates more realistic and consistent 3D poses.

- Experiment #3 (E_3): Finally, we assess the model's performance when incorporating the loss term for the NF $\mathcal{L}_{NF}$ to the loss used in Experiment #2. This change increases the performance with respect to the previous formulation. In both cases, the model produces accurate poses for most input images. However, when analysing the visualisations of the predictions, the ones trained with the NF loss term present better alignment between the upper body joints and the ground truth.

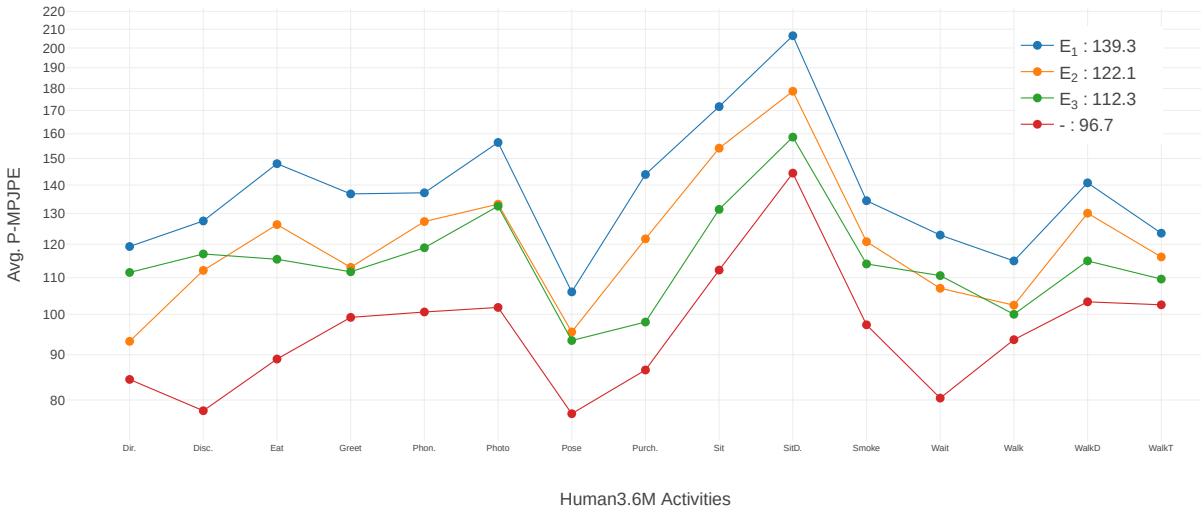


Figure 4.23: Ablation studies scores. The coloured lines in the figure display the average P-MPJPE per activity for various versions of our method (E_1, E_2, and E_3). The inset legend shows the average scores for each experiment. Note that the red line represents results obtained with the version of our model that maintains all its original components.

According to the final loss formulation in Equation 4.18 and the ablation studies, adding the combination of the loss terms for the normalising flow $\mathcal{L}_{NF}$ and relative bone length $\mathcal{L}_{bl}$ has proven to be beneficial, increasing the performance of the model by 20.8% compared to the loss function that does not contain those terms. We denote the three previous experiments, as E_1, E_2, and E_3 respectively, and present their corresponding quantitative evaluation in terms of P-MPJPE in Figure 4.23. Note that the evaluation follows the same protocol and metrics as those in subsection 4.4.2.

We perform an additional experiment to determine how the size of the prior of 2D poses affects the overall performance of our approach. We train another version of our model using half of the data initially used to build the prior 2D poses (from the Human3.6M dataset). Note that the number of training images remains the same as in the main experiment, only the size of the prior changes. We then test the model according to the protocol described earlier and compute

Figure 4.24: Experiments with different sizes for the prior of 2D poses. The figure displays coloured lines representing the average P-MPJPE per activity for different prior sizes. The blue line indicates results obtained using half of the original prior for training, whereas the red line represents the results using the complete prior. The inset legend displays the average scores for each case.

P-MPJPE. We show the results in Figure 4.24 per activity from both model versions: the one trained with the complete prior and the other with half of it. The results from Figure 4.24 show that reducing the prior size to half reduces performance for all activities, as expected. However, the average score is still decent, indicating that the model can estimate reasonable 3D poses even with a significantly reduced prior.

4.4.9 Failure cases

We have examined some instances where our model did not perform as expected. Specifically, we analysed the results of testing the model with Human3.6M data and focused on the predictions with the highest P-MPJPE scores. Not surprisingly, most failure cases on the Human3.6M dataset appear for activities such as *Sitting* and *Sitting Down*. We assume this occurs because of the self-occlusions and perspective ambiguity in these activities. However, according to the examples shown in Figure 4.25, the model can still produce plausible 3D poses for most cases, even if they do not exactly match their respective 3D ground truth. The high P-MPJPE comes from mismatches between the joints representing the body's extremities, e.g. hands and feet.

In addition, we include a visual analysis of some inaccurately predicted 3D poses using LSP data. Given the lack of 3D ground truth for this dataset, we focus on the 3D poses derived from inaccurately predicted 2D poses. Specifically, we select the 3D poses from the 2D predictions

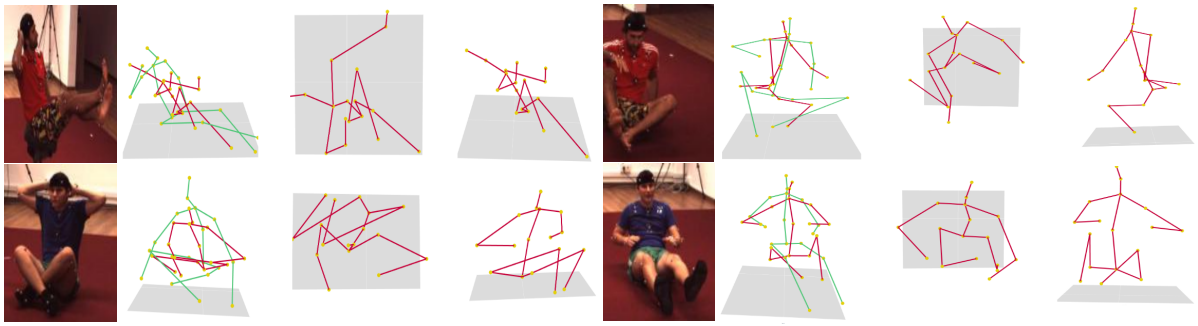


Figure 4.25: Failure cases on Human3.6M. 3D predictions with a P-MPJPE greater than 200mm. The first column shows the input images. The second column displays the predicted 3D pose (coloured in red) aligned with its respective ground truth (coloured in green). Following columns show different views of the predicted 3D pose.

with the lowest PCK scores, estimated with the model without fine-tuning. To provide a more comprehensive visualisation, we have also included results from the fine-tuned version of the model. Figure 4.26 presents the visualisation of those incorrectly predicted 2D and 3D poses. The first and sixth columns depict the input image, while the second and seventh illustrate the predicted 2D poses with each model version alongside the corresponding ground truth depicted in grey. The remaining columns show the estimated 3D poses. The green colour represents the predictions with the non-fine-tuned version of the model, while purple corresponds to those estimated with the fine-tuned version. To avoid confusion, we use the same colours as in the previous visualisation of this data.

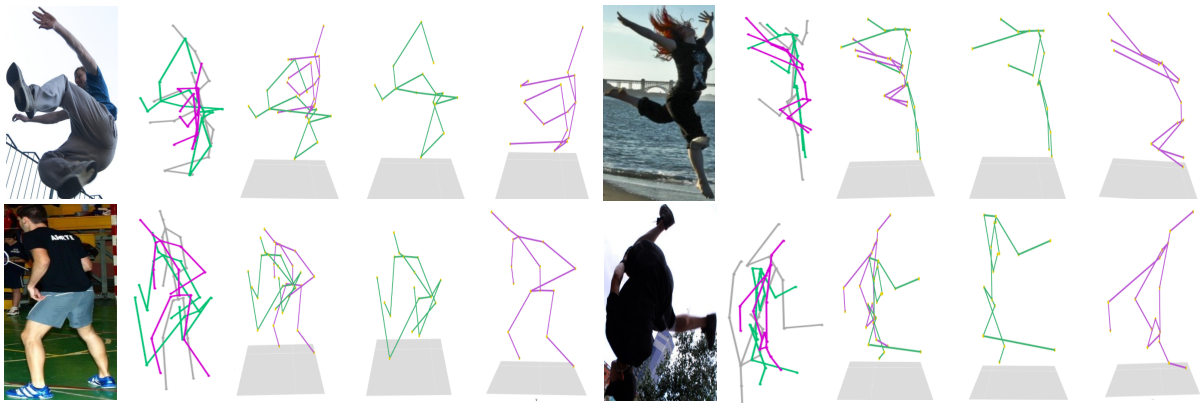


Figure 4.26: Failure cases with data from LSP dataset. The first and sixth columns show the input images. The second and seventh columns display the respective 2D predicted poses from each version of our model: fine-tuned (purple) and non-fine-tuned (green). This visualisation also includes the ground truth 2D pose depicted in grey. Subsequent columns illustrate the predicted 3D poses from both versions of the model (fine-tuned with purple and non-fine-tuned coloured with green)—Zoom in the figure to better appreciate the visualisations.

As shown in Figure 4.26, the model original model (non-fine-tuned and trained with Human3.6M

data) fails to produce an accurate 3D pose in the first image from the second row. However, its fine-tuned version produces a 3D pose representation that better aligns with the subject in the input. Unfortunately, both versions of the model fail to estimate plausible 3D poses for all other cases. The model struggles mostly with images depicting subjects performing unconventional poses in sports such as parkour or gymnastics. Moreover, the current distribution of 2D poses in the prior utilised for training the models may need more diversity to comprehensively capture these challenging poses.

4.5 Conclusions

In summary, our proposal represents a practical approach to estimating 3D poses directly from images. The fundamental benefit of our method is that it learns without the need for 3D annotations during training. Additionally, it reduces the number of 2D annotations required by leveraging the learning of the 3D poses to an unpaired prior of 2D poses. Such low data availability requirements permit adapting this approach to scenarios where collecting 2D and 3D annotations for training deep learning models is more demanding.

According to the reported quantitative results in Table 4.1 and Figure B.6, our method outperforms self-supervised state-of-the-art approaches that estimate 3D poses from images and assume unpaired 3D data for supervision [59]. It also performs better than some methods that rely on 3D supervision [219] or multi-view images [58, 152]. Moreover, its performance is similar to methods that assume 3D kinematic constraints like [61]. Specifically, we achieve superior performance than [61] in 20% of the activities in Human3.6M and comparable scores for the remaining activities.

Experiments using the MPI-INF-3DHP dataset demonstrate our method's cross-dataset generalisation (Table 4.2). Although we perform different data combinations for training and testing the model, it achieves comparable results to the state-of-the-art in each scenario. Furthermore, the augmentation of the training set with images from Human3.6M and MPI-INF-3DHP benefits the model's performance. Note that collecting unlabelled data to extend the training set is relatively straightforward since the images do not require annotations. Similarly, when testing our model with images from the LSP dataset, it performs decently in terms of 2D pose estimation despite not being trained with samples from that data distribution (trained with Human3.6M data). The 3D predictions from this experiment also show plausible poses. Further

experiments fine-tuning the model solely with images from LSP data demonstrate an increase in performance, as expected.

We demonstrate how to estimate 3D human pose with a training architecture requiring only an unpaired prior of 2D poses. We qualitatively demonstrate that our approach holds the potential for rapidly learning about the pose of articulated structures other than the human body without the need to collect ground-truth pose data, e.g. human hands (Figure 4.22). Overall, the qualitative and quantitative results suggest that our method is comparable to other self-supervised state-of-the-art approaches that estimate 3D pose from images. Furthermore, it performs better than some methods that rely on multi-view images or 3D pose annotations for supervision. Prior work has demonstrated the value of using temporal information from image sequences and domain adaptation networks. Incorporating these into our approach would be a promising direction for future work. Finally, we propose applying the method to other articulated structures (e.g., mice, dogs, horses, and other animals), exploiting the relatively light requirement for self-supervision in the form of an unpaired prior of 2D poses.

Chapter 5

Learning to predict 3D animal pose from unlabelled images and synthetic data

Obtaining labelled data to train deep learning methods for estimating animal pose is challenging. Recently, synthetic data has been widely used for pose estimation tasks, but most methods still rely on supervised learning paradigms utilising synthetic images and labels. Can training be fully unsupervised? Is a tiny synthetic dataset sufficient? What are the minimum assumptions that we could make for estimating animal pose? Our proposal addresses these questions through a simple yet effective self-supervised method that only assumes the availability of unlabelled images and a small set of synthetic 2D poses. We completely remove the need for any 3D or 2D pose annotations (or complex 3D animal models), and surprisingly our approach can still learn accurate 3D and 2D poses simultaneously. We train our method with unlabelled images of horses mainly collected from YouTube videos and a prior consisting of 2D synthetic poses. The latter is three times smaller than the number of images needed for training. We test our method on a challenging set of horse images and evaluate the predicted 3D and 2D poses. We demonstrate that it is possible to learn accurate animal poses even with as few assumptions as unlabelled images and a small set of 2D poses generated from synthetic data. Given the minimum requirements and the abundance of unlabelled data, our method could be easily deployed to different animals.

5.1 Overview

One of the main bottlenecks in developing and deploying supervised deep learning models for pose estimation is obtaining the annotations required for their training. This is particularly challenging in the animal domain, where annotated datasets are scarce compared to the available human data. Synthetic data has proven to be an attractive solution to overcome this problem, especially by relying on readily available 3D animal models to generate large amounts of annotated data. At the same time, synthetic data eliminates the need for laborious and time-consuming manual annotation, making the models more flexible for multiple applications in different scenarios.

We further extend the method described in Chapter 4, which initially learns 3D human poses from unlabelled images and a prior on 2D pose [65]. Our implementation in this chapter translates that method to the animal domain, demonstrating that it applies to different body structures. Another essential addition to the new approach is the origin of the 2D poses composing the prior. Unlike the original implementation, which uses a set of unpaired 2D poses from the training datasets, we further reduce the assumptions by using 2D poses from an existing CAD model of a horse [45]. Our model is unique in its simplicity compared with previous approaches for animal pose estimation with synthetic data. It does not require annotated training data. It uses only unlabelled images and a small set of synthetically generated 2D poses, meaning no synthetic images, pre-trained models, or complicated 3D models are required.

We build a dataset of images depicting horses by collecting data from various YouTube videos and use it to augment an existing horse dataset [134]. Altogether this provides a more realistic scenario for the training and evaluation of our approach. Additionally, we assemble the prior of 2D poses needed for training our model with synthetically generated data from [45]. By evaluating our model's 2D and 3D predictions, we demonstrate that it produces accurate pose representations of the horses without using any annotations for the input images. Due to the minimal requirements for training our model, it has the potential to be applied to a variety of body structures from other animal species.

5.2 Related work

5.2.1 Animal pose estimation

Supervised deep learning methods for human pose estimation have been widely explored and perform well under different conditions [14, 15, 16]. However, in animal pose estimation, getting the labels needed for supervision is difficult in most cases. In particular, labelling key points is more expensive and time-consuming than producing other annotations, e.g. bounding boxes. On top of this, it would be infeasible to generate labelled data for the entire diversity of animal species in the world. Since the 3D pose annotations are even more challenging to acquire than the 2D ones, many works on animal pose estimation have been focused only on estimating 2D pose [108, 2, 41, 110]. Not surprisingly, the backbones for most of these approaches are network architectures initially designed for the human domain, including stacked hourglass networks [14], ResNet [100], and OpenPose [242].

Although the problem of 3D animal pose estimation is more constrained and challenging, relevant work has also been carried out [111, 243, 244]. In this context, methods commonly rely on lower supervision levels to overcome the scarcity of labelled training data. For instance, the self-supervised approach of [245] estimates 3D pose for monkeys and dogs relying on multi-view supervision and a tiny portion of pose annotations. Dai et al. [246] propose a similar method, but instead of multi-view images, they assume the availability of actual 2D poses for each input image and lift these to 3D through self-supervision based on geometric consistency [54]. Similar to [246], our method also estimates 3D pose using self-supervision with the same geometric consistency constraint. However, we learn the 2D and 3D poses directly from images in an end-to-end manner. Most importantly, we do not require any annotations for the inputs.

5.2.2 Animal pose estimation with synthetic data

Synthetic data has been gaining attention as a cost-effective alternative for generating data with ground-truth annotations with minimum effort. Multiple works on human [131, 247, 50] and animal pose estimation [63, 45, 46, 242, 39, 36, 195, 165, 37, 120] have recently adopted synthetic data to overcome the scarcity of keypoint labels.

Focusing on the animal domain, many pose estimation methods that rely on synthetic data follow a supervised approach. This means these approaches utilised synthetically generated images

and their corresponding pose annotations for training. However, there is often a gap between synthetic and real data, so these approaches typically perform domain adaptation with samples from actual data. For example, the method in [45] learns to estimate 2D pose for animals using images and labels generated from CAD models. It also incorporates a consistency-constrained semi-supervised method to adapt the predictions to real data. Similarly, [46] concentrates on domain adaptation by generating pseudo-labels from the synthetic domain and then updating these to match the actual data. Unlike these approaches, our formulation helps to reduce the complexity and requirements for training even more. It is as simple as using unlabelled real images and a set of synthetically generated 2D poses, i.e. there is no need to generate pictures from the synthetic data. Furthermore, we include a couple of loss terms that help to adapt the synthetic poses to the actual data without additional processes to perform domain adaption.

More related to our work, [120] relies on a self-supervised method that assumes synthetic 2D poses and real images for estimating 2D mouse pose. However, we advance [120] by incorporating a cycle of geometry transformations, allowing our model to further estimate 3D poses.

Synthetic data also plays an essential role in several works that learn richer structures, such as animal shapes, mainly for different quadrupeds like dogs [36, 195, 39], tigers, lions, horses [165], and zebras [37]. However, the success of these approaches is constrained by having access to sophisticated and expensive animal models, which is not required in our approach.

5.3 Method

The method is essentially that from [65], described in Chapter 4, but with a few modifications to adapt it to the animal domain. Rather than relying on unpaired annotations of the training dataset, we utilise a prior of 2D poses from synthetic data. Furthermore, we remove the components for learning elevation angles to simplify the implementation. We reproduce the approach here in order to give the full method with the aforementioned modifications.

The main component of the approach is an image to 3D pose mapping, indicated with a dotted box in Figure 5.1. The first part of this mapping employs a CNN Φ to map the input image x to an intermediate skeleton image s. Then, another CNN Ω maps s to a 2D pose representation y. In the final stage, y is mapped to the 3D pose v by means of a fully connected network Λ. For training this set of networks, we incorporate it within a larger structure which allows for self

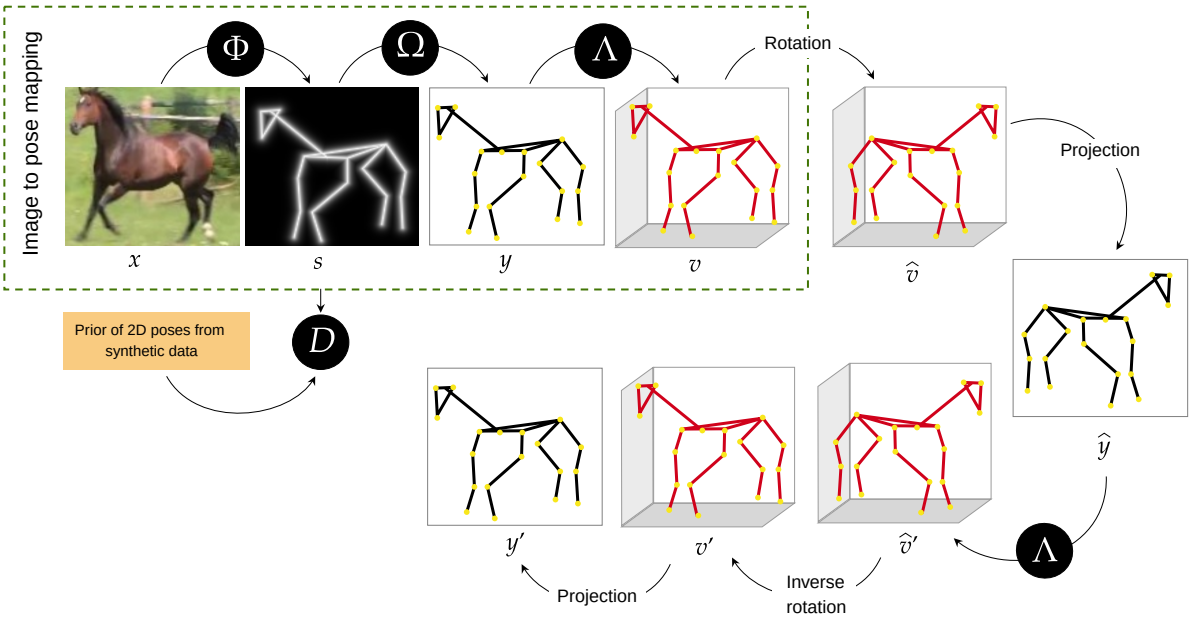


Figure 5.1: Method for predicting 2D and 3D horse poses. The approach adopted jointly learns to estimate 2D and 3D poses from unlabelled images depicting horses. Most importantly, it only requires a prior of synthetic 2D poses for self-supervision. The image-to-pose mapping is embedded within a more extensive network that permits end-to-end training.

supervision. In particular, we rely in a loop of transformations of the 3D pose v. We also use a discriminator D together with the prior on synthetic 2D poses, to ensure that the generated skeletons $\Phi(x) = s$ are realistic.

5.3.1 Main mapping

The image to pose mapping consists of three networks Φ, Ω, and Λ that allows the input image x to be mapped to its 3D pose representation v. This mapping also produces two intermediate representations of the input, a skeleton image s, and a 2D pose y. Specifically, Φ learns to align the input image with its respective skeleton image representation, i.e. $s = \Phi(x)$. Then, Ω learns to extract keypoints from s, obtaining a 2D pose as output $y = \Omega(\Phi(x))$. Finally, Λ acts as a lifter of the 2D pose y to get the 3D pose v. For each pair of joint positions $(\mathrm{x}_i, \mathrm{y}_i)$ in y, the network estimates a depth $\mathrm{z}_i = d + \Delta$, where Δ is a constant depth.

Overall, we use the same network structure as in [65] with exception of Λ. Since we are not trying to learn elevation angles for the geometry transformations like [65, 64], we opt for a simpler structure as in [54, 16].

5.3.2 Self-supervision

As illustrated by Figure 5.1, we include the main mapping within a large network structure that allows to self-supervise the training. This structure uses a discriminator network D, which relies on a prior of synthetic 2D poses to help the mapping produce skeleton images that are as realistic as possible. Furthermore, it incorporates a loop of random rotations and projections of the 3D pose v to ensure geometric consistency for the 3D predictions.

Synthetic pose prior

To create the prior of 2D poses, we use a publicly available dataset of synthetic 2D poses generated from a CAD model of a horse [45]. The prior is needed during training to ensure the estimated skeleton image looks as realistic as possible. Note that generating the prior from synthetic data and not from annotations of the dataset like [65] provides more flexibility to the method to be trained with completely unlabelled datasets, which are abundant in the animal domain. Our synthetic prior contains around 10k different 2D poses, representing approximately one-third of the available images for training. This prior is the same as the one used for some experiments in Chapter 3.

The purpose of having a prior of 2D poses is to use these as a reference distribution for the discriminator network D. Since our implementation of D works directly with images, we must first render the synthetic 2D poses to skeleton images. This is done by using the rendering function κ defined in Equation 4.5 from Chapter 4 (originally from [119]), which given a set of 2D joint positions p and their connections, can generate a skeleton image $w = \kappa(p)$. Then, the goal of D is to evaluate whether or not the predicted skeleton images by the generator $\{s_i = \Phi(x_i)\}_{i=1}^{M}$, looks like an authentic skeleton image w such as those in the prior $\{w_j = \kappa(p_j)\}_{j=1}^{N}$. Following [65, 119] we use an adversarial loss to compare w and s:

$$\mathcal{L}_D = \frac{1}{N}\sum_{j=1}^{N} D(w_j)^2 + \frac{1}{M}\sum_{i=1}^{M}((1 - D(s_i))^2 \tag{5.1}$$

Geometric consistency

We rely on the idea of geometric consistency from [54] to facilitate the learning of the lifting network Λ and, therefore, the whole mapping. Essentially this involves a series of rotations and projections of the 3D pose v. First, v is randomly rotated to $\hat{v}$ using a rotation matrix, which

is constructed by sampling azimuth and elevation angles from a fixed uniform distribution [54]. Then, $\hat{v}$ is projected to a 2D pose $\hat{y}$. Given the projection of the rotated 3D pose $\hat{v}$, the same lifting network Λ estimates its 3D representation $\hat{v}'$. Lastly, the inverse rotation is applied to the 3D pose $\hat{v}'$ to obtain v', and v' is projected to 2D to get the 2D pose y'.

After the loop of projections and rotations we expect the poses on the forward and backward parts to be as similar as possible. For example, the 3D poses v and v' should be similar, and the same with $\hat{v}$ and $\hat{v}'$. This also applies to the 2D poses y and y'. Therefore, we can derive the following loss terms:

$$\mathcal{L}_{2D} = ||y' - y||^2 \tag{5.2}$$

$$\mathcal{L}_{3D} = ||(v'^{(j)} - v'^{(k)}) - (v^{(j)} - v^{(k)})||^2 \tag{5.3}$$

$$\mathcal{L}_{r3D} = ||\hat{v}' - \hat{v}||^2 \tag{5.4}$$

Note that for Equation 5.3 we follow [64, 65] and instead of comparing the v and v' with a L_2 loss we measure the degree of deformation between 3D poses using two samples j and k in a batch. For simplicity, we refer to the sum of these three losses as $\mathcal{L}_{GC}$ given by

$$\mathcal{L}_{GC} = \mathcal{L}_{2D} + \mathcal{L}_{3D} + \mathcal{L}_{r3D} \tag{5.5}$$

For more insights about the notion of geometric consistency see subsection 4.2.1 from Chapter 4.

5.3.3 Training and additional losses

Following [119] we include an extra loss term $\mathcal{L}_\Omega$ that exploit the dual representation of the poses, i.e., as set of coordinates for joint positions and as skeleton image. This loss is designed to contribute in learning the mapping from the skeleton image s to the 2D pose y, namely $y = \Omega(s)$.

$$\mathcal{L}_{\Omega} = ||(\Omega(\kappa(p)) - p)||^2 + \lambda||\kappa(y) - s||^2 \tag{5.6}$$

where λ represents a balancing coefficient, and p is a 2D pose from the synthetic prior. The second term in Equation 5.6 helps to learn poses that potentially exist in the training images, but are not necessarily part of the prior. In other words, this formulation helps to adapt the synthetic poses in the prior to the actual data.

We train all the networks from scratch using a loss function $\mathcal{L}$ consisting of three components from Equation 5.1, Equation 5.5, and Equation 5.6:

$$\mathcal{L} = \lambda\mathcal{L}_D + \mathcal{L}_{GC} + \mathcal{L}_{\Omega} \tag{5.7}$$

where $\lambda = 10$ represents a balancing coefficient. The batch size is set to 96, with each batch consisting of images and random samples from the prior of unpaired 2D poses (which are then transformed into skeleton images). We utilise the Adam optimiser [203] with a learning rate of 2×10^{-4}, and $\beta_1 = 0.5, \beta_2 = 0.999$. At inference time, we only keep the elements from the main mapping as illustrated in Figure 5.2, i.e. the loop of rotations and projections, and D are only needed during training.

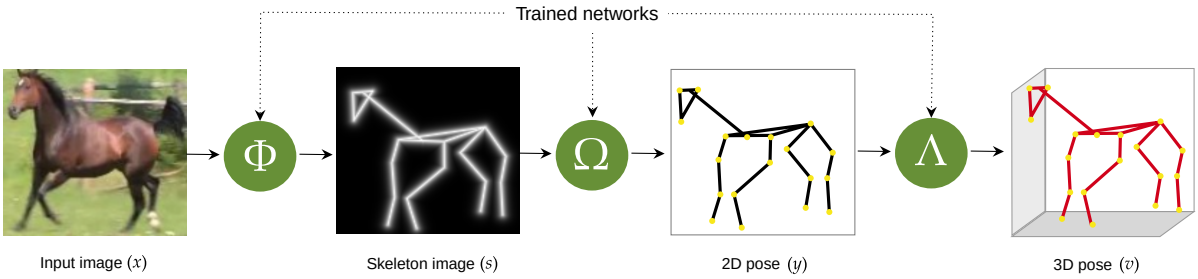


Figure 5.2: Networks used during inference for 3D horse pose estimation. During the testing stage, we only require the trained networks (Φ, Ω, and Λ) responsible for the image-to-pose mapping. The remaining networks (from the main diagram in Figure 5.1) are only necessary while training the model.

To keep our model in its simplest working version we remove the normalising flow (NF) block. Therefore the elevation angles is now sample from an uniform distribution and not learned by Λ. Apart from that all the other networks and transformations are the same that the ones described in Chapter 4.

5.4 Experiments

5.4.1 Data

We use the same dataset and synthetic prior described in section 3.6 from Chapter 3. Note that this dataset is relatively small compared to what is required for training human pose estimation models — our horse dataset is only 1.3% of the size of the Human3.6M dataset [47] and 3.6% of the size of the MPI-INF-3DHP dataset [49]. Unfortunately, we are not able to release the collected images due to copyright restrictions associated with the YouTube videos. Instead, we provide the YouTube video IDs from which the images were collected in Table 5.1 to promote the reproducibility of our data and method. In addition, Figure 5.3 provides visualisations of some images from the dataset. Regarding the data used for testing our trained model, it mostly comes from the Weizmann horse dataset [135].

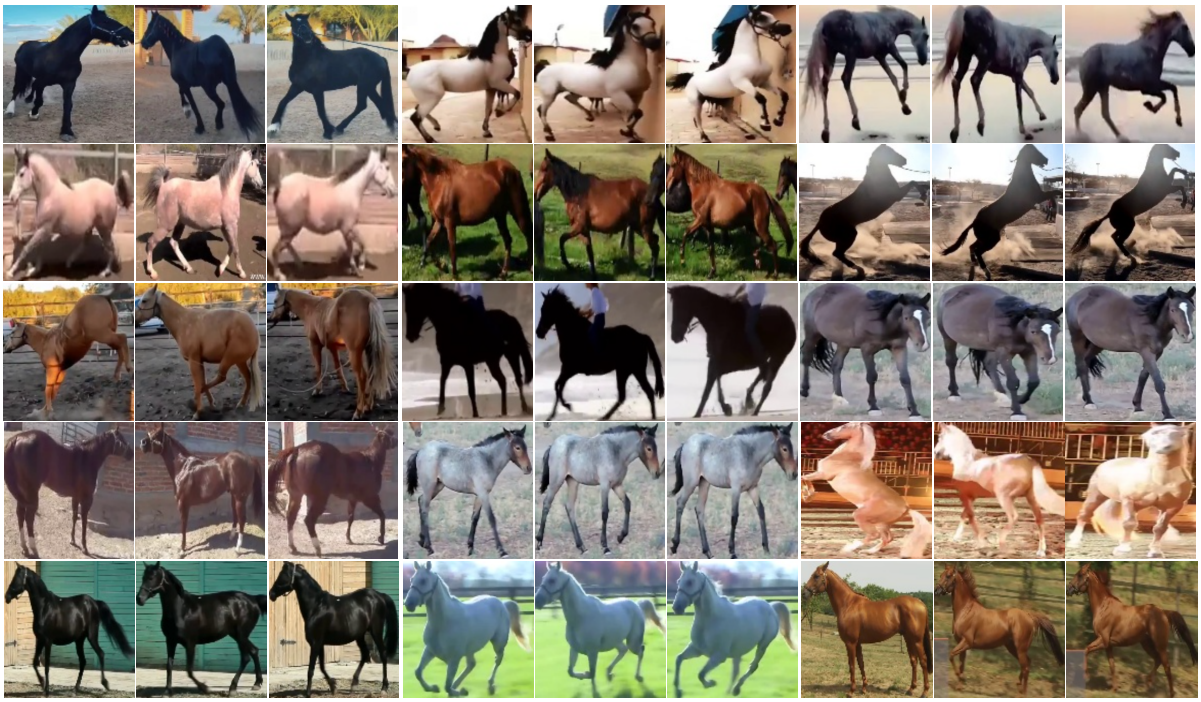


Figure 5.3: Example of images collected from YouTube videos. As can be observed from the images, the collected dataset is diverse in terms of horse appearance and poses. Note that each triplet of images belongs to the same video sequence.

5.4.2 Evaluation and metrics

Since there is a lack of available horse datasets with 3D pose annotations for a quantitative performance evaluation, we only provide a visual evaluation of the 3D predictions. While obtaining ground-truth 2D poses is more feasible than 3D poses, we evaluate the emergent 2D

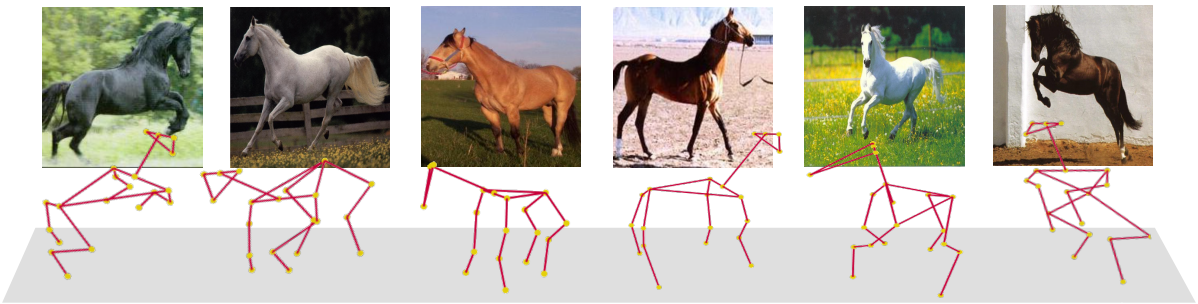
Youtube video IDs				
AoLg6aDqwUI	xbEDf9Aozuk	Socb6o6VKGE	GexrWONgj7g	74p-Lgcb1OE
t4zdTn02PWQ	wwzYvu_174I	rBdEKXVvLVY	gDIPyrrufOA	5gq7no2ZQCM
zZDzFA70fvQ	WjudHSeY8nU	r59eFzDoKyE	g0HiQN6V83I	4wuf4TTWB_U
ytAkggYAA1s	VQOLhgazzZo	pwj7cLuEj_Q	FG7_-StFaXM	3vSLOGyQanM
YSLsr9bJj6s	vqIv0USOop0	pW8zAPzayao	E35RWcryQx0	3IgU1k2wxtc
YrHe_uvcKFY	VkwF8T5czu8	poNKPf7JQ7s	Cr9Cuz4yBic	2Mv6b36LXqA
YQeiYQxjW4I	VG7Q6rzbfrE	oXqY0khS1mY	cctjggMwKDg	14VvHu0pusY
ynQjbYH-dDo	UWE2IadD6hQ	nusSGkcVWFg	bKK8KS28eKU	0thBJWe2BSA
Ym2dV2E1g4Y	uIXa9u1YWBo	N9Bur7JGl5U	AMeXRFo4axQ	_pcJnrCc-Lw
YADgTfBYGB4	TnRfBjd3E38	KYRqoNXQxGQ	aJhrHCidwZw	4k3MNxjtM0
xzj149ACvSQ	t8dNtmClGmk	hj8_SjStNmg	a_vSLBTHoQQ	8aO9HrWzj7Y
xUAdF9hBzI	SQgrdNpzJh4	gPZf1MRfUQg	9Ej2iVe1Vec	8WPzgJfKbcw

Table 5.1: YouTube video IDs. Each item in the table corresponds to one video from which images have been collected. Appendix C provides more details about the process for collecting and processing the video data.

pose predictions y quantitatively and qualitatively. Note that although the goal of the model is predicting 3D poses, the emergent 2D pose representations are also worth evaluating. We assume that if the 2D poses are reasonable, it is very likely that the 3D poses will also be accurate.

In line with previous works for 2D animal pose estimation, we use the Percentage of Correct Keypoints (PCK@0.05) to quantitatively evaluate our 2D predictions. Our predicted poses are composed of 20 joint positions. However, we use only 15 in order to compare with the ground truth 2D poses from the Weizmann dataset.

5.4.3 Results on 3D predictions



We train our model using images from the YouTube videos, the horse subset of [134], and the synthetic prior from [45]. For evaluation purposes, we utilise images from the Weizmann dataset [135]. Given an unlabelled image from this data, we use the trained model to predict the 2D and 3D poses of the horse appearing in the input. The Weizmann dataset contains no ground truth 3D pose data so we provide only a qualitative evaluation of the 3D poses estimated by

our trained model.

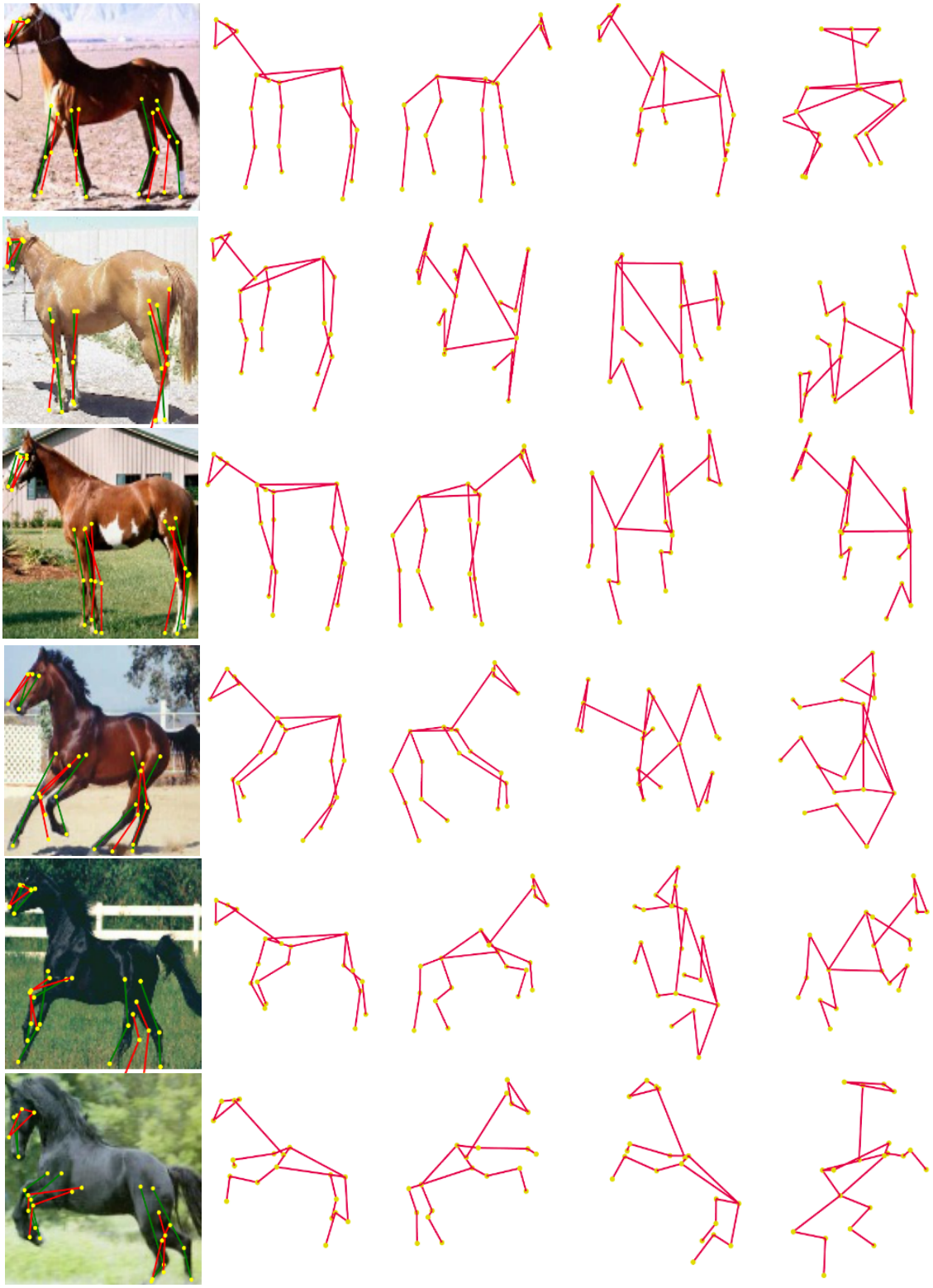


Figure 5.4: 3D poses estimated by our method. The initial column exhibits a set of images, each with their estimated (red) and ground truth (green) 2D poses. The second column provides the corresponding estimated 3D pose for each image. The remaining columns display various novel views of the predicted 3D poses.

We visualise some 3D predictions in Figure 5.4. The first column of the figure represent the input images, while subsequent visualisations correspond to the predicted 3D pose and its novel views. Furthermore, each input image shows its corresponding ground truth and estimated 2D poses, coloured green and red, respectively.

Additionally, we evaluate the generalisation capability of our model by testing it on a dataset of zebras [37]. Without ever seeing a zebra during training, the trained model with horse data still managed to estimate plausible 3D poses for zebras. This may be due to the anatomical similarities between the two species, despite some slight differences, such as zebras having wider chests and shorter legs. Overall, our model demonstrates a reasonable degree of robustness in making predictions from both domains. Figure 5.5 displays some of the 3D pose predictions for images depicting zebras.

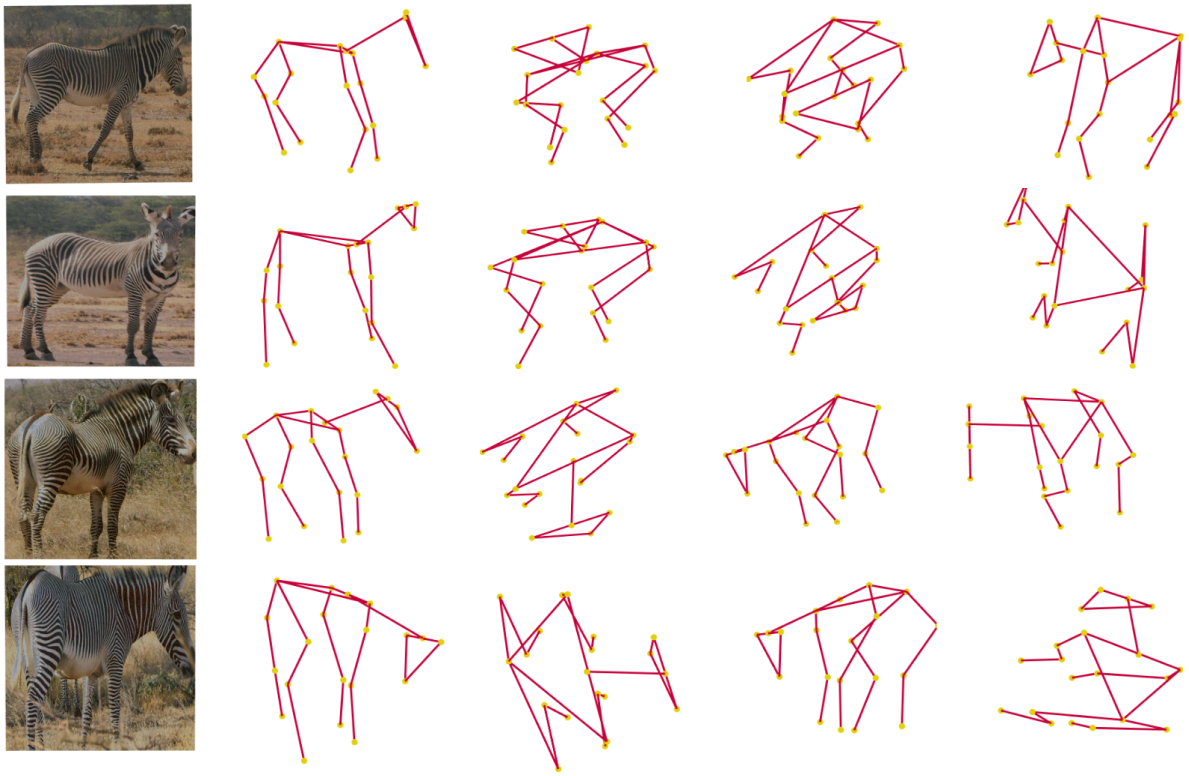


Figure 5.5: 3D pose predictions for images depicting zebras. The input image is shown in the first column, while the second column displays the estimated 3D pose. The remaining columns show novel views of the 3D prediction. The model was trained only with horse data.

5.4.4 Results on 2D predictions

Apart from predicting 3D poses, we also evaluate the intermediate 2D poses estimated by the trained model for all images in the test set. Each pose prediction consists of 20 joint positions.

Figure 5.6: Predicted 2D poses. We visualise the intermediate 2D pose predictions from the trained model. For each input image, we superimposed the corresponding predictions represented with red lines and the ground truth depicted with green lines.

However, when comparing against ground truth, we only keep 15 joint positions to match the annotations. Figure 5.6 shows some predicted 2D poses by our model compared with their respective ground truth. Each image appearing in this figure contains the estimated 2D pose coloured with red and its corresponding ground truth 2D pose annotation, coloured with green.

In addition, we reproduce the method from Chapter 3 that originally estimates 2D poses for mice. We train it with the same assumptions as our method, i.e. our same horse dataset and synthetic 2D poses. We use the Weizmann dataset to evaluate and compare their predictions with the ones obtained with our 3D method. As illustrated by Figure 5.7, our model for 3D poses can produce more accurate 2D pose representations than the 2D pose estimator from Chapter 3. This demonstrates that incorporating the geometric consistency cycle for lifting 2D poses to 3D can effectively substitute the image reconstruction process required by the 2D pose estimator from Chapter 3. Simultaneously, this addition reduces the training load, eliminating the need for a second image as input without negatively impacting the method's performance.

We use the PCK@0.05 metric to quantitatively compare the predicted 2D poses against their respective ground truth. The outcomes from the evaluation are shown in Table 5.2, which also includes results for approaches that work under similar conditions. Following previous works, we report the average scores for the standard groups of joints. Table 5.2 includes various methods that have been evaluated using different horse datasets. However, the purpose of comparing

Figure 5.7: Visual comparison of predicted 2D poses. We visually assess our 2D predictions against the ones estimated with the model from Chapter 3. Note that both models were trained under the same conditions, i.e. the same images, prior, and hyper-parameters. For each triplet of images, the green poses represent the ground truth; the orange poses are for predictions made with the 2D pose estimator from Chapter 3 and the red poses represent the 2D predictions with our model for 3D poses.

these methods is to demonstrate that our approach remains competitive despite its minimal data requirements during training. It is worth noting that all other methods on the table utilise some form of supervised learning during training, such as domain adaptation. Only method [120] shares the same approach, with no direct supervision involved.

Furthermore, we experiment by training our method on synthetically generated images of zebras [37] and utilising the same synthetic 2D horse poses as prior. We then test the trained model with the same dataset of real zebras [37] as in previous experiments (model trained with horse images and synthetic 2D poses as prior). Despite the differences between the two domains, the model trained with purely synthetic data, i.e. with synthetic images of zebras and synthetic 2D poses of horses, produces similar 2D poses as the model trained with actual horse images and synthetic 2D horse poses. Figure 5.8 displays a visual comparison of 2D pose predictions from both configurations. Precisely, images in block A depict 2D pose predictions by the model

Method	Evaluation Data	Eyes	Chin	Shoulders	Knees	Hooves	**Mean**
Syn - Mu et al. [45]	TigDog dataset	46.08	53.86	20.46	24.20	17.45	25.33
Sosa [120]	Weizmann dataset	45.67	44.67	33.00	37.67	26.67	37.54
CycleGAN [121]	TigDog dataset	70.73	84.46	56.97	49.91	35.95	51.86
Ours	Weizmann dataset	49.3	58.3	34.2	44.7	31.2	43.50

Table 5.2: Horse 2D pose estimation accuracy. We calculate the accuracy of our predicted 2D poses using the PCK@0.05 metric. For each image in the Weizmann dataset, the predicted 2D pose is compared against its respective ground truth. We also list some works that estimate 2D poses using synthetic data. Following the standard used in previous works, we average the scores for some groups of joints.

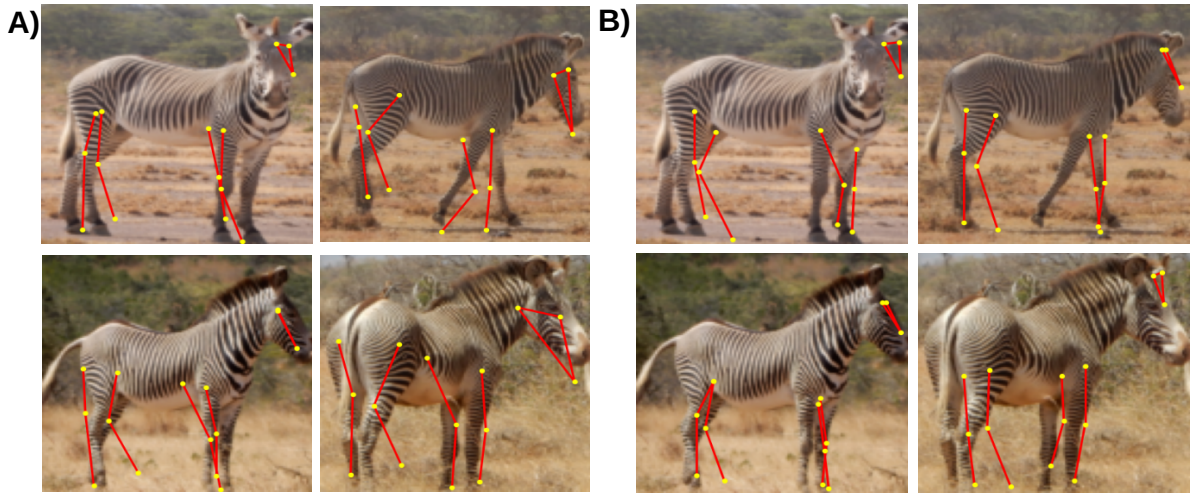


Figure 5.8: Predicted 2D poses for zebras. Block A shows 2D poses predicted by the model trained with images of horses and the prior of synthetic 2D poses. Block B displays 2D predictions using synthetic images of zebras and the same prior of synthetic 2D poses from horses.

trained with images of horses and the prior on synthetic 2D poses. In contrast, images in block B show the 2D predictions using the model trained with synthetic images of zebras and the same prior on synthetic prior.

5.4.5 Failed cases

Following the assumption that the quality of the emergent 2D pose estimations influences the accuracy of the final 3D pose predictions, we then select some inaccurately estimated 2D poses, i.e. with the lowest PCK@0.05 score, and inspect their corresponding 3D pose estimated by the model. We expect such 3D poses derived from inaccurate 2D predictions also to be inaccurate. Surprisingly, even for some non-accurate 2D predictions, our model can still recover a convincing 3D horse pose. It may not completely align with the horse's pose in the input image, but the overall 3D pose makes sense and keeps a horse-like structure. Figure 5.9 illustrates this scenario, where the first and fourth columns show the input images with their estimated 2D pose (coloured

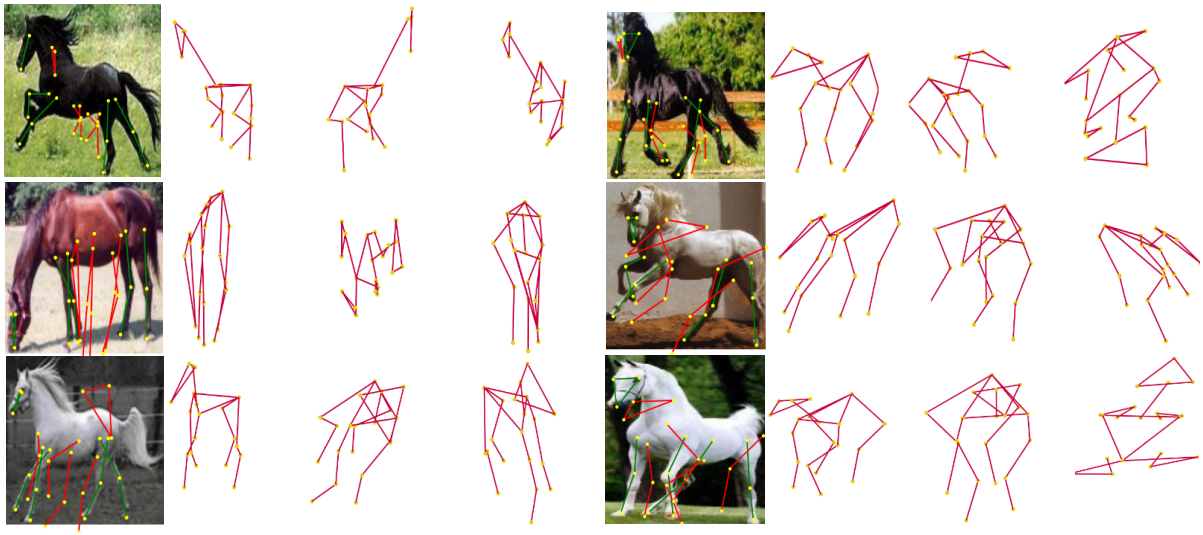


Figure 5.9: Failed cases. We select the 2D poses with lower accuracy (PCK@0.05) to asses their respective 3D predictions. The first and fifth columns of the figure display the input images, along with their respective 2D predictions (in red) and ground truth (in green). The second and sixth columns show the predicted 3D poses, while the remaining columns illustrate novel views of the 3D predictions.

in red) and ground truth (coloured in green), and the rest display novel views of the estimated 3D poses.

5.5 Conclusion

We have successfully adapted a method originally designed to estimate 3D human poses to the animal domain. We further reduce its requirements by generating the needed prior from synthetic data. We demonstrate that with only unlabelled images and a small set of synthetic 2D poses, it is possible to learn actual 2D and 3D representations. By reducing the data requirements for training to a minimum, our proposal is more flexible to be applied to many unlabelled datasets without collecting annotations typically required for supervised training.

Our approach outperforms similar methods that include more complex processes, such as image reconstruction [120, 119, 121], in the task of 2D pose estimation. We achieve this by relying solely on a cycle of geometric transformations of the pose, reducing computational costs during model training. Furthermore, the qualitative and quantitative evaluations show that our model can learn animal 2D and 3D poses without needing a large amount of data for training, unlike models that perform pose estimation with human data. However, extending the dataset and the diversity of the poses in the prior could enhance the overall performance.

From our results, there is clearly room for further improvement; we suggest exploring three ideas in the future:

- Integration of temporal information.
- Fine-tuning with small amounts of actual data to bridge the gap between the synthetic and real domains.
- Utilising image features to learn texture and shapes, not just poses.

Additionally, we plan to develop resources that allow to quantitatively evaluate the 3D pose predictions from our model. Finally, we propose utilising inpainting techniques with stable diffusion models to automatically generate datasets of different animal species from an existing one. This would simplify the process of deploying our approach and eliminate the need for image collection.

Chapter 6

Conclusions

In this thesis, we develop different self-supervised deep-learning approaches for estimating pose. Overall, we explore learning 2D and 3D poses and extend our methods to work with human and non-human body structures. Our primary focus is reducing the data requirements for training deep learning models, which is crucial for scenarios where annotated data is scarce, such as in the animal domain.

This final chapter aims to summarise the research presented through the chapters of this thesis, highlighting their main contributions. Furthermore, it addresses our work's limitations and provides insights about future work. The latter includes improvements for previous approaches, new ideas to extend our research, and scenarios where our methods could be helpful.

6.1 Summary

- Chapter 2 summarises and describes the existing literature around pose estimation. We particularly focus on reviewing deep learning self-supervised approaches to estimate both 2D and 3D poses for human and non-human structures.
- Through Chapter 3, we explore the use of unlabelled images and a synthetically generated prior of 2D poses for estimating mouse 2D poses. Our results contribute towards exploration of a new dataset of genetically modified mice, without requiring to produce annotations for this data. Additionally, we experiment with different body structures and successfully adapt the method to estimate 2D poses of horses.

- In Chapter 4, we shift to the human pose estimation domain by proposing a new self-supervised method to estimate 3D human pose. Again we rely solely on unlabelled data from well-know benchmarks and a prior of unpaired 2D poses. We compare our results against related state-of-the-art and outperform some methods with heavy requirements of data for training.
- Finally, in Chapter 5, we take inspiration from our previous approaches and extended the self-supervised method from Chapter 4 to the animal domain. Most importantly, we demonstrate that it can still learn if we create the prior of 2D poses from synthetic data (same as Chapter 3), eliminating the need to use unpaired annotations from the dataset. Given the reduced requirements in terms of data, the method is more flexible to be applied to multiple body structures. In addition, we collect a dataset of images for training our method from YouTube videos to provide a more real-world data scenario.
- Additionally we include an appendix for Chapter 3, Chapter 4, and Chapter 5 respectively. Each appendix (Appendix A, Appendix B, Appendix C) contains extra details for some experiments or implementations mentioned in the chapters, for example, network details, links to code, and extra visualisation of results.

6.2 Limitations and considerations

Learning to predict 2D animal pose from unlabelled images and synthetic prior

It can be challenging to obtain annotations for datasets not initially intended for computer vision tasks, especially when dealing with datasets created for clinical purposes. A major challenge we faced while implementing our model was the nonexistence of pose annotations to test our approach. Although we annotate a small set of poses to produce qualitative and quantitative performance comparisons, the ultimate aim is to build a more robust annotated set mainly for more extensive testing purposes. Creating such a robust test set will involve hiring experts to produce the pose annotations and compare the discrepancies between measurements made by different individuals.

While testing our approach with the mice data, we use a dataset that lacks diversity regarding the animals' body structures and poses. This means that the recordings depict animals performing similar activities under comparable conditions. Although we demonstrate that the

method works well with a different body structure (horse data), introducing more variability within the mice dataset, such as images from different perspectives and different breeds of mice, will potentially harm the model's performance without proper adjustments. Neutralising this effect might imply extending the synthetic prior to better represent the distribution of poses in the training data.

Our approach does not rely on manual annotations for input images, but it assumes the availability of a CAD model to extract synthetic poses for building the prior. Although there are numerous freely accessible CAD models of animals on the internet, there may be some animals for which no model exists, which could restrict our implementation. Nevertheless, generating synthetic animal models offers greater flexibility than creating dataset-specific annotations, as these models can be adapted and reused for various scenarios, whereas annotations cannot.

One last consideration regarding the proposal in this chapter is the requirements of the model in terms of memory to fit the training batches. Since we need two images as input, the batch size is then dependent on the size of these images. Therefore, training with high-resolution images may slow down the training process. However, using high-resolution images may also result in better geometry representations (skeleton images), which could benefit the overall performance.

Learning to predict 3D human pose from unlabelled images

Our approach for this chapter does not rely on 3D pose annotations or paired 2D ground-truth data. However, it does need an unpaired set of 2D pose annotations to build the needed prior for training. While in specific scenarios, like working with human data, obtaining this data does not represent a challenge, in other cases is difficult to access even small sets of 2D pose annotations. This limitation could restrict the use of our methods in other scenarios.

Regarding the components of our model, the intermediate representation of the pose as a skeleton image introduces more complexity to the model as it prevents the process from being a straightforward mapping between the input image and its 3D pose as illustrated by Figure 6.1. Thus, we have experimented with some alternatives to remove it from our implementation, but none have succeeded. This is likely because mapping from an image to its 3D pose is highly ambiguous and requires introducing support from 3D ground truth annotations while learning it, which will compromise the claims of our self-supervised method. The intermediate mapping, i.e., from image to skeleton image and then to 2D pose, seems significant for our process. Us-

ing robust image-to-image translation networks and conditional GANs allows the approach to ensure that the learned pose structures, such as 2D, skeleton image, and the ultimate 3D pose, are as realistic as possible.

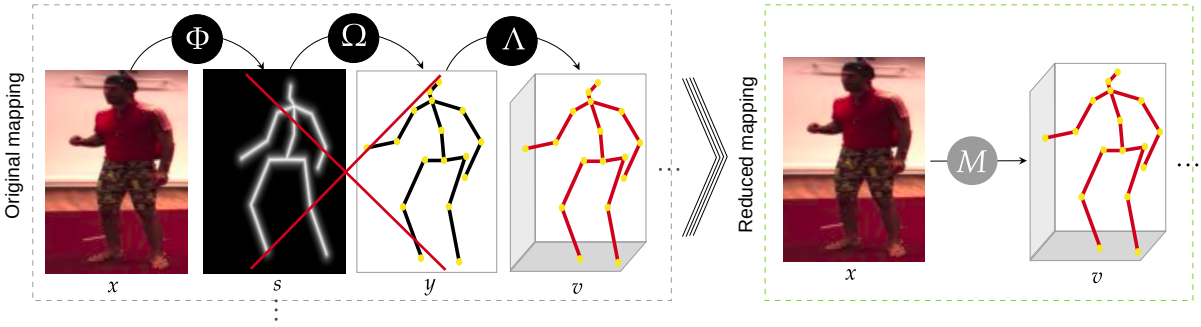


Figure 6.1: Removing intermediate representations. We plan to remove the intermediate skeleton image s and 2D pose representation y to reduce the image-to-pose mapping further. However, designing M efficiently is highly ambiguous since we are not considering any 3D data for training.

When comparing the model in this chapter with the one from Chapter 3, it could be observed that we have eliminated the requirement for the conditional image during training. Instead of relying on a reconstruction stage as in the 2D mice pose estimator, we substitute it with a set of geometric transformations of the pose that lifts the 2D predictions to 3D in an end-to-end manner. This modification makes our model more efficient during training since it only needs one image as input, allowing for larger batches.

Learning to predict animal 3D poses from unlabelled images and synthetic data

In this chapter, we have addressed some concerns from our previous models, particularly the dependency on unpaired 2D pose annotations from the dataset to construct the prior of 2D poses as in Chapter 4. However, like the approach from Chapter 3, we rely on an existing synthetic model of a specific animal, but we do not directly manipulate the CAD model in this case. Instead, we use the existing pose annotations generated with this model [45]. Since this set contains highly diverse horse poses, our model produces plausible results using these annotations as a prior. Nevertheless, having access to the actual CAD model and animating it differently could produce a richer set of poses for the prior, which will better match the pose distribution from the actual data.

6.3 Future Work

Learning to predict 2D animal pose from unlabelled images and synthetic prior

The content in Chapter 3 is part of a larger project that seeks to calculate gait parameters from videos depicting genetically modified mice without the need for significant human intervention. One of the primary objectives of this project is based on the assumption that producing measurements for these parameters may help to categorise animals with varying degrees of ALS disease and assess its impact on locomotion.

Since calculating gait parameters involves using the positions of particular body parts of the animals, having the estimated 2D poses with our method provides a suitable low-dimensional representation to make these measurements, as show in Figure 6.2. We have conducted exploratory research using this approach, but further refinement and validation with an extensive test set are required. Alternatively, using existing methods for assessing gait and inputting the 2D pose estimated with our method for a given video segment could also be interesting.

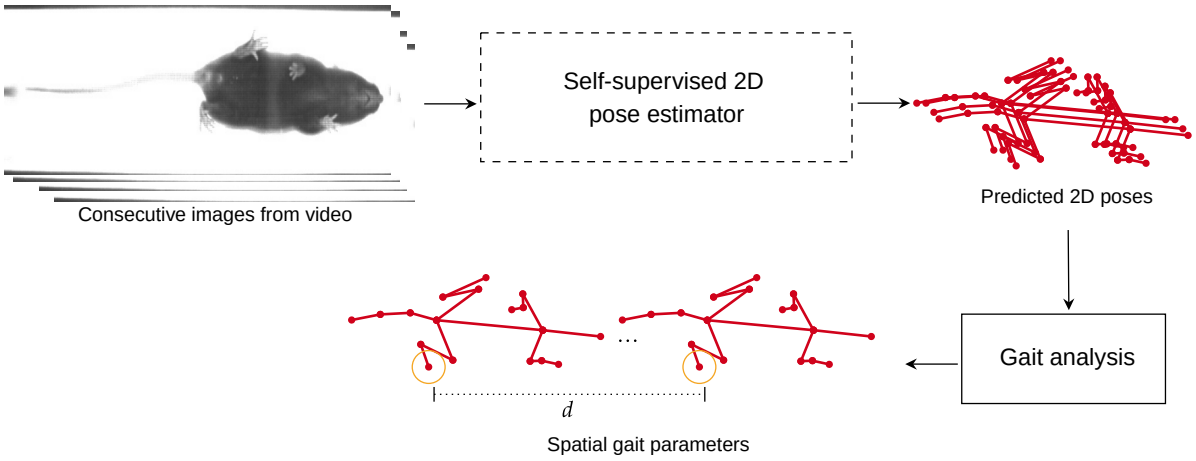


Figure 6.2: Gait analysis. Our model produces 2D pose data that can be valuable features for gait analysis. We can easily calculate spatial measurements from the predicted 2D poses. Alternatively, we can use the 2D pose representations for a given video as input to recent unsupervised clustering methods, like B-SOiD [209], to identify features without any user bias.

Future work for this research direction includes closer collaboration with our partners at the University of Tasmania to create a larger dataset of mice recordings. We aim to incorporate more challenging scenarios and poses. We will then use this data to run our model and generate pose annotations. If required, the annotations will be reviewed and corrected by experts. Our plans also involve making the model available to the public. This will provide a tool for people working with similar animal recordings to perform more comprehensive analyses with minimal

effort and, most importantly, without investing in expensive commercial equipment.

Learning to predict 3D human pose from unlabelled images

Previous work has demonstrated the advantages of incorporating temporal information into pose estimation methods. Including this feature would be a valuable addition to future versions of our model for estimating 3D human poses. We can exploit the assumption that it is possible to learn pose by modelling differences between consecutive video frames. Quantifying these differences could provide a robust consistency measure that can be incorporated within a loss function, potentially leading the model to better accuracy.

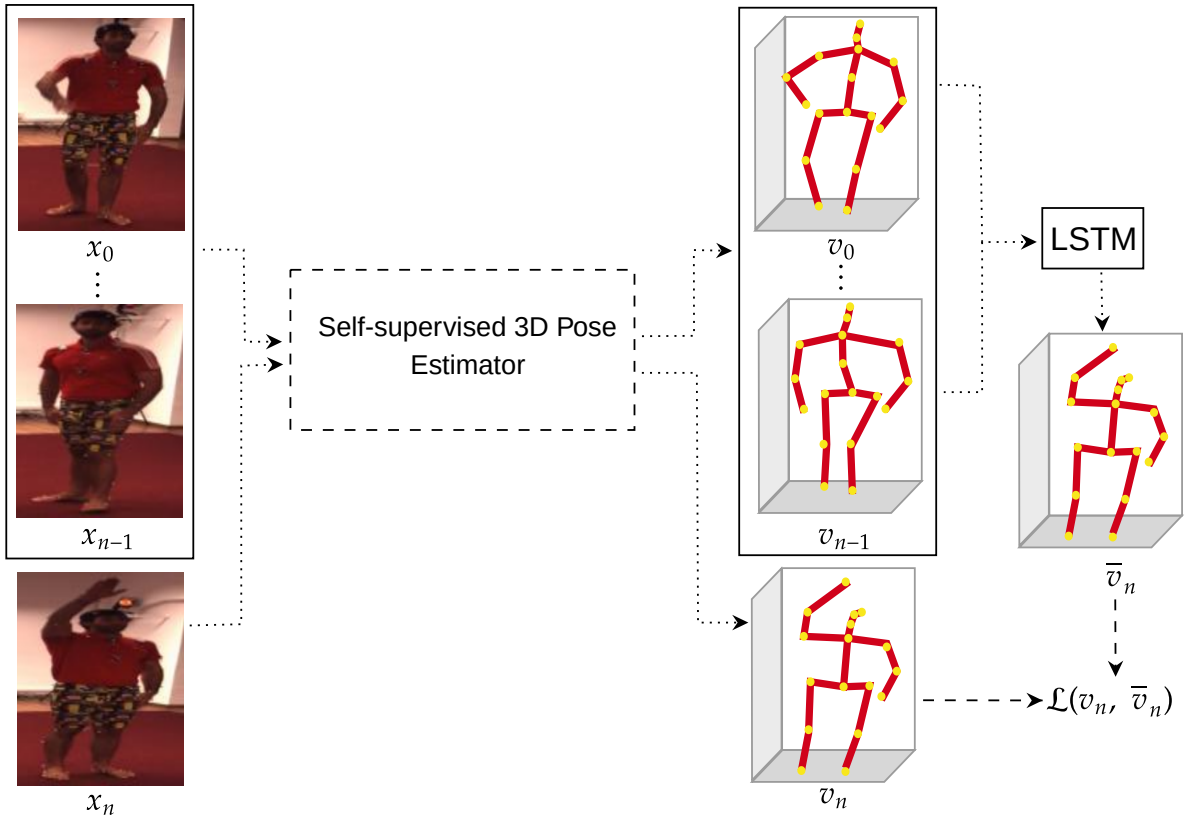


Figure 6.3: Integrating temporal information. Exploiting the temporal information from multiple consecutive images $(x_0, \ldots, x_n)$ from a video sequence could improve the model performance. It is also possible to incorporate a network that can handle temporal data, such as LTSM, and train it with the model to predict the following pose representation $\bar{v}_n$ for a given sequence $(v_0, \ldots, v_{n-1})$. This pose and the model's prediction v_n can then be compared using a loss term.

As illustrated in Figure 6.3, we could input a set of n consecutive images from a video sequence and introduce a new component of the loss function that simply measures the differences between their corresponding predicted 3D poses $(v_0, \ldots, v_n)$. Alternatively, we could rely on network architectures that handle sequential data, such as LSTM (Long short-term memory), to predict

a future 3D pose $\bar{v}$ based on a previous sequence of 3D predictions. Then, comparing the estimated pose from the LSTM with the corresponding estimation by the model could provide a useful loss term. However, this design would constrain the inputs to belong to the same video sequence, making it impossible to train the model with isolated images. Furthermore, building each batch item with a sequence of images will reduce the number of batches for training, potentially increasing the use of computational resources.

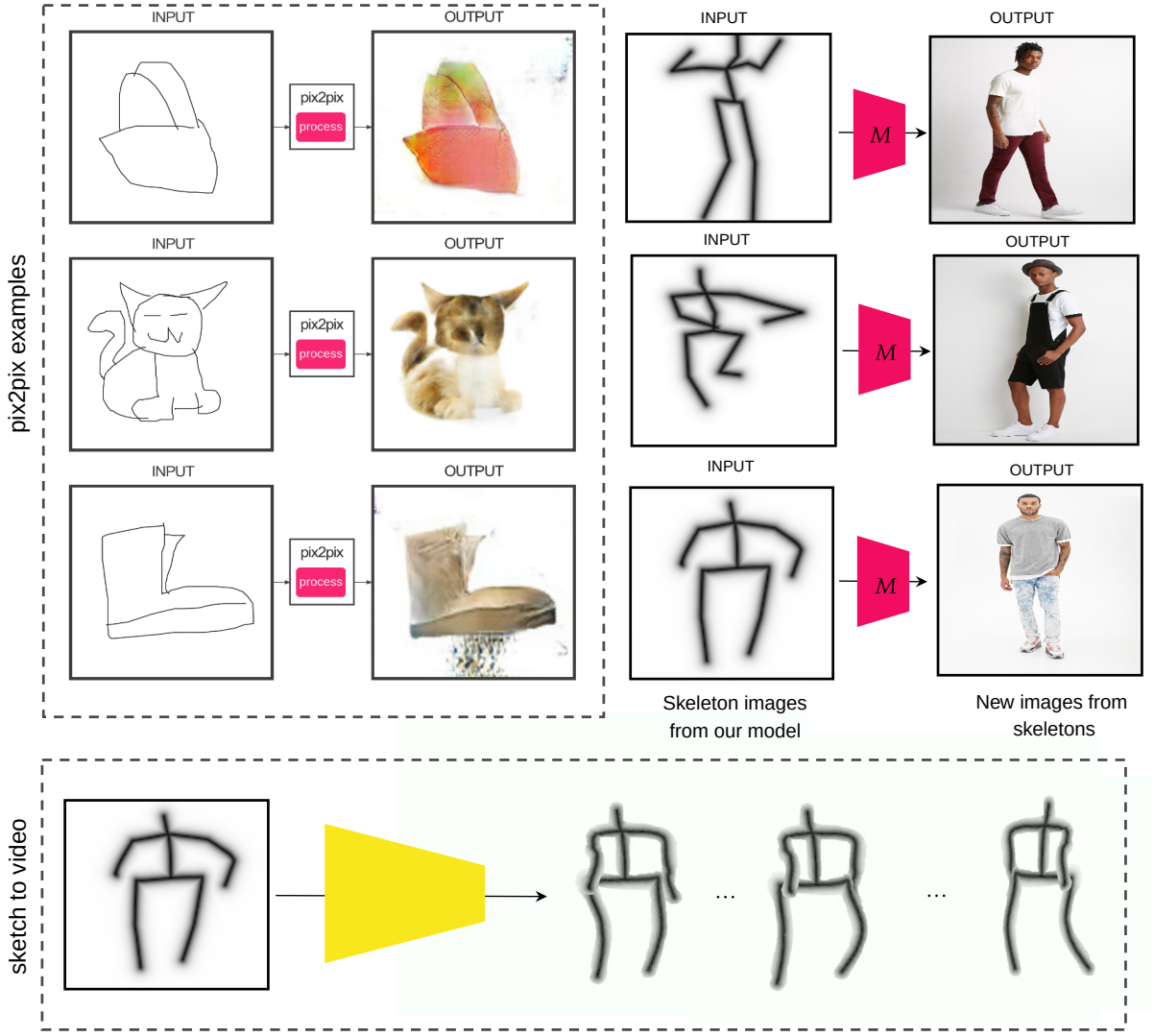


Figure 6.4: Skeletons to images. Potential approaches for exploiting the intermediate skeleton representations. pix2pix examples obtained from https://affinelayer.com/pixsrv/. Examples with human images are just for illustrative purposes. Only the skeleton images correspond to real examples produced by our model. Animation done with https://sketch.metademolab.com/canvas.

It would also be advantageous to experiment with building the prior of 2D poses from synthetic data, as in Chapter 5. This would eliminate the requirements for unpaired pose annotations from the current dataset. Besides, adding a combination of synthetic and actual data might

benefit the model's performance.

Another potential future development for our research within 3D human poses would be including SMPL models [131] within our framework. Instead of constraining the model to learn 3D joint positions, it would learn the parameters of the SMPL model. This would enhance the flexibility of our approach, making it applicable to a broader range of scenarios where more complex body shapes are required. Given the popularity of SMPL models, it would be a valuable addition to our framework.

Finally, it is possible to use the intermediate skeleton image representations to create a dataset that could benefit works that depend on sketches [220, 248, 249, 250, 251]. For example, we could rely on methods like pix2pix [220] to generate realistic-looking images of animals and objects from sketches, as demonstrated in Figure 6.4. By utilising techniques like this, we can create new images of people with different appearances based on the skeleton images from our approach. This can enhance the dataset and provide a broader range of scenarios for training models. Additionally, Smith [252] has recently released a method for animating drawings (`https://sketch.metademolab.com/canvas`), which could also be helpful for using our skeleton images. We provide an example of the animation produced with this technique, accessible by clicking on Figure 6.4.

Learning to predict 3D animal pose from unlabelled images and synthetic data

One potential approach to quickly get data for training our model with different animals without directly collecting images is to rely on recent techniques based on stable diffusion models, such as inpainting. Figure 6.5 demonstrates how we can transform the images in our dataset to show a different animal through the use of powerful vision models. For instance, we can segment the horse in our images and replace it with a zebra, cow, or any other animal using SAM (Segment Anything Model) and a stable diffusion model. Alternatively, if we have the segmentation mask for the object of interest, we can use stable diffusion and a text prompt to replace it with another object. Additionally, other stable diffusion-based techniques generate multiple variations of a given image or skeleton representations, as depicted in the *image variations* section of Figure 6.5, which increases the variability of data. In the case of the skeleton representations of the horses, it is also possible to use pix2pix [220] or some other similar models to produce realistic-looking images from them.

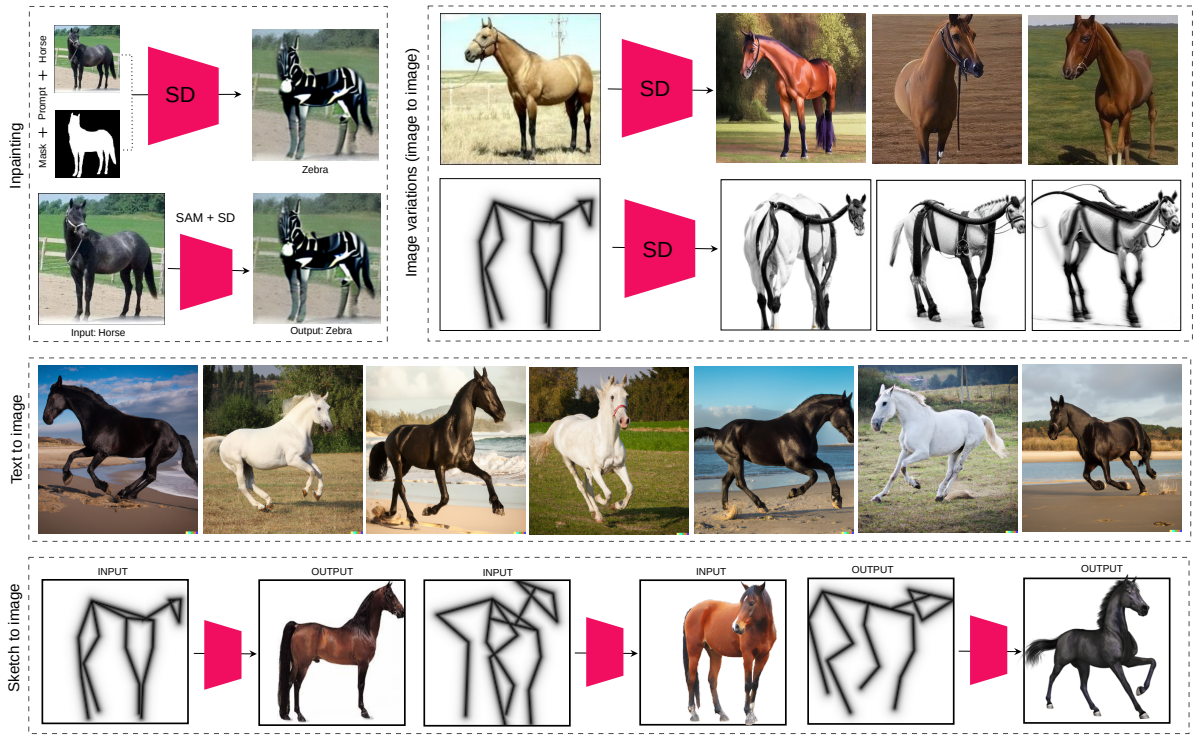


Figure 6.5: Potential directions of future work by incorporating stable diffusion based tools to easily generate new data. Inpainting examples were generated using the online demo from: https://replicate.com/andreasjansson/stable-diffusion-inpainting. Image variations for images and skeleton made by using: https://huggingface.co/spaces/lambdalabs/stable-diffusion-image-variations and https://imagevariations.com/ respectively. Text to image generated by DALL-E https://openai.com/dall-e-2, using the text prompt: *"realistic full body photo of a ... horse running in the ..."*. Sketch to image examples made for illustrative purposes only.

When it comes to generating images from text, easily accessible methods such as DALL-E [253] can provide a quick solution for creating test or training data for our model. By providing a text description of desired objects and their traits, these techniques can produce multiple images, as demonstrated in the text-to-image section of Figure 6.5. With the appropriate tools, generating new images and substituting animals in our dataset can be a simple process. This would provide more flexibility in implementing our model with various animals, eliminating the need for gathering data from each one individually.

Lastly, extending the model's capabilities to recognise animal shapes might be interesting rather than solely 3D joint positions.

References

[1] A. L. Huxley, *The art of seeing*. DigiCat, 2022.

[2] A. Mathis *et al.*, "Deeplabcut: Markerless pose estimation of user-defined body parts with deep learning", *Nature neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018.

[3] M. T. Chiu *et al.*, "Agriculture-vision: A large aerial image database for agricultural pattern analysis", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2828–2838.

[4] L. Schmidtke, A. Vlontzos, S. Ellershaw, A. Lukens, T. Arichi, and B. Kainz, "Unsupervised human pose estimation through transforming shape templates", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2484–2494.

[5] A. Esteva *et al.*, "Deep learning-enabled medical computer vision", *NPJ digital medicine*, vol. 4, no. 1, p. 5, 2021.

[6] H. Kim, S. Lee, D. Lee, S. Choi, J. Ju, and H. Myung, "Real-time human pose estimation and gesture recognition from depth images using superpixels and svm classifier", *Sensors*, vol. 15, no. 6, pp. 12 410–12 427, 2015.

[7] S. Kim, K. Yun, J. Park, and J. Y. Choi, "Skeleton-based action recognition of people handling objects", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, pp. 61–70.

[8] G. A. Thomas, "Real-time camera pose estimation for augmenting sports scenes", 2006.

[9] Y. Su and Z. Liu, "Position detection for badminton tactical analysis based on multi-person pose estimation", in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, IEEE, 2018, pp. 379–383.

[10] S. Park, J. Yong Chang, H. Jeong, J.-H. Lee, and J.-Y. Park, "Accurate and efficient 3d human pose estimation algorithm using single depth images for pose analysis in golf", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, 2017, pp. 49–57.

[11] C.-Y. Weng, B. Curless, and I. Kemelmacher-Shlizerman, "Photo wake-up: 3d character animation from a single photo", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5908–5917.

[12] N. S. Willett, H. V. Shin, Z. Jin, W. Li, and A. Finkelstein, "Pose2pose: Pose selection and transfer for 2d character animation", in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 2020, pp. 88–99.

[13] Z. Zhang, "Microsoft kinect sensor and its effect", *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[14] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 483–499.

[15] 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*, 2014, pp. 1653–1660.

[16] 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*, 2017, pp. 2640–2649.

[17] C. Sminchisescu and A. C. Telea, "Human pose estimation from silhouettes. a consistent approach using distance level sets", in *10th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG'02)*, vol. 10, 2002.

[18] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation", in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.

[19] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts", in *CVPR 2011*, IEEE, 2011, pp. 1385–1392.

[20] J. Wu, C. Geyer, and J. M. Rehg, "Real-time human detection using contour cues", in *2011 IEEE international conference on robotics and automation*, IEEE, 2011, pp. 860–867.

[21] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures", *IEEE Transactions on computers*, vol. 100, no. 1, pp. 67–92, 1973.

[22] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition", *International Journal of Computer Vision*, vol. 61, pp. 55–79, 2005.

[23] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 1014–1021.

[24] D. Hogg, "Model-based vision: A program to see a walking person", *Image and Vision computing*, vol. 1, no. 1, pp. 5–20, 1983.

[25] 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*, vol. 28, pp. 91–99, 2015.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[27] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context", in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[28] 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*, 2013, pp. 3674–3681.

[29] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation.", in *BMVC*, Aberystwyth, UK, vol. 2, 2010, p. 5.

[30] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, "Deep convolutional neural networks for efficient pose estimation in gesture videos", in *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part I 12*, Springer, 2015, pp. 538–552.

[31] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4733–4742.

[32] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation", *Advances in neural information processing systems*, vol. 27, 2014.

[33] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648–656.

[34] Y. Yao *et al.*, "Openmonkeychallenge: Dataset and benchmark challenges for pose estimation of non-human primates", *International Journal of Computer Vision*, vol. 131, no. 1, pp. 243–258, 2023.

[35] A. Mathis *et al.*, "Pretraining boosts out-of-domain robustness for pose estimation", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1859–1868.

[36] B. Biggs, O. Boyne, J. Charles, A. Fitzgibbon, and R. Cipolla, "Who left the dogs out? 3d animal reconstruction with expectation maximization in the loop", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 195–211.

[37] S. Zuffi, A. Kanazawa, T. Berger-Wolf, and M. J. Black, "Three-d safari: Learning to estimate zebra pose, shape, and texture from images" in the wild"", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5359–5368.

[38] J. Cao, H. Tang, H.-S. Fang, X. Shen, C. Lu, and Y.-W. Tai, "Cross-domain adaptation for animal pose estimation", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9498–9507.

[39] B. Biggs, T. Roddick, A. Fitzgibbon, and R. Cipolla, "Creatures great and smal: Recovering the shape and motion of animals from video", in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V 14*, Springer, 2019, pp. 3–19.

[40] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs", in *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, Citeseer, vol. 2, 2011.

[41] T. D. Pereira *et al.*, "Fast animal pose estimation using deep neural networks", *Nature methods*, vol. 16, no. 1, pp. 117–125, 2019.

[42] J. M. Graving *et al.*, "Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning", *Elife*, vol. 8, e47994, 2019.

[43] X. Liu *et al.*, "Optiflex: Multi-frame animal pose estimation combining deep learning with optical flow", *Frontiers in cellular neuroscience*, vol. 15, 2021.

[44] L. Jiang, C. Lee, D. Teotia, and S. Ostadabbas, "Animal pose estimation: A closer look at the state-of-the-art, existing gaps and opportunities", *Computer Vision and Image Understanding*, p. 103 483, 2022.

[45] J. Mu, W. Qiu, G. D. Hager, and A. L. Yuille, "Learning from synthetic animals", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 386–12 395.

[46] C. Li and G. H. Lee, "From synthetic to real: Unsupervised domain adaptation for animal pose estimation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1482–1491.

[47] 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*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[48] H. Joo *et al.*, "Panoptic studio: A massively multiview system for social motion capture", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3334–3342.

[49] D. Mehta *et al.*, "Monocular 3d human pose estimation in the wild using improved cnn supervision", in *2017 international conference on 3D vision (3DV)*, IEEE, 2017, pp. 506–516.

[50] G. Varol *et al.*, "Learning from synthetic humans", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 109–117.

[51] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 601–617.

[52] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion", *International Journal of Computer Vision*, vol. 87, no. 1-2, p. 4, 2010.

[53] 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*, 2019, pp. 5442–5451.

[54] C.-H. Chen, A. Tyagi, A. Agrawal, D. Drover, S. Stojanov, and J. M. Rehg, "Unsupervised 3d pose estimation with geometric self-supervision", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5714–5724.

[55] B. Wandt, J. J. Little, and H. Rhodin, "Elepose: Unsupervised 3d human pose estimation by predicting camera elevation and learning normalizing flows on 2d poses", *arXiv preprint arXiv:2112.07088*, 2021.

[56] Z. Yu, B. Ni, J. Xu, J. Wang, C. Zhao, and W. Zhang, "Towards alleviating the modeling ambiguity of unsupervised monocular 3d human pose estimation", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8651–8660.

[57] B. Wandt, M. Rudolph, P. Zell, H. Rhodin, and B. Rosenhahn, "Canonpose: Self-supervised monocular 3d human pose estimation in the wild", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 294–13 304.

[58] H. Rhodin *et al.*, "Learning monocular 3d human pose estimation from multi-view images", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8437–8446.

[59] J. N. Kundu, S. Seth, V. Jampani, M. Rakesh, R. V. Babu, and A. Chakraborty, "Self-supervised 3d human pose estimation via part guided novel image synthesis", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6152–6162.

[60] X. Hu and N. Ahuja, "Unsupervised 3d pose estimation for hierarchical dance video recognition", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 015–11 024.

[61] J. N. Kundu, S. Seth, M. Rahul, M. Rakesh, V. B. Radhakrishnan, and A. Chakraborty, "Kinematic-structure-preserved representation for unsupervised 3d human pose estimation", in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 11 312–11 319.

[62] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi, "Self-supervised learning of interpretable keypoints from unlabelled videos", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8787–8797.

[63] L. A. Bolaños *et al.*, "A three-dimensional virtual mouse generates synthetic training data for behavioral analysis", *Nature methods*, vol. 18, no. 4, pp. 378–381, 2021.

[64] B. Wandt, J. J. Little, and H. Rhodin, "Elepose: Unsupervised 3d human pose estimation by predicting camera elevation and learning normalizing flows on 2d poses", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6635–6645.

[65] 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 (CVPR) Workshops*, 2023, pp. 4788–4797.

[66] J. O'rourke and N. I. Badler, "Model-based image analysis of human motion using constraint propagation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 522–536, 1980.

[67] H. Hattori, N. Lee, V. N. Boddeti, F. Beainy, K. M. Kitani, and T. Kanade, "Synthesizing a scene-specific pedestrian detector and pose estimator for static video surveillance: Can we learn pedestrian detectors and pose estimators without real data?", *International Journal of Computer Vision*, vol. 126, pp. 1027–1044, 2018.

[68] A. Lamas *et al.*, "Human pose estimation for mitigating false negatives in weapon detection in video-surveillance", *Neurocomputing*, vol. 489, pp. 488–503, 2022.

[69] A. Moryossef *et al.*, "Evaluating the immediate applicability of pose estimation for sign language recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3434–3440.

[70] J. Charles, T. Pfister, M. Everingham, and A. Zisserman, "Automatic and efficient human pose estimation for sign language videos", *International Journal of Computer Vision*, vol. 110, pp. 70–90, 2014.

[71] A. Moryossef, I. Tsochantaridis, R. Aharoni, S. Ebling, and S. Narayanan, "Real-time sign language detection using human pose estimation", in *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Springer, 2020, pp. 237–248.

[72] M. Andriluka *et al.*, "Posetrack: A benchmark for human pose estimation and tracking", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5167–5176.

[73] J. Kondragunta, A. Jaiswal, and G. Hirtz, "Estimation of gait parameters from 3d pose for elderly care", in *Proceedings of the 2019 6th International Conference on Biomedical and Bioinformatics Engineering*, 2019, pp. 66–72.

[74] A. Raj, D. Singh, and C. Prakash, "Active human pose estimation for assisted living", in *2021 Thirteenth International Conference on Contemporary Computing (IC3-2021)*, 2021, pp. 110–116.

[75] S. Liu and S. Ostadabbas, "Seeing under the cover: A physics guided learning approach for in-bed pose estimation", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 236–245.

[76] S. Salti, O. Schreer, and L. Di Stefano, "Real-time 3d arm pose estimation from monocular video for enhanced hci", in *Proceedings of the 1st ACM workshop on Vision networks for behavior analysis*, 2008, pp. 1–8.

[77] C. Xu, X. Yu, Z. Wang, and L. Ou, "Multi-view human pose estimation in human-robot interaction", in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2020, pp. 4769–4775.

[78] X. Liu, X. Feng, S. Pan, J. Peng, and X. Zhao, "Skeleton tracking based on kinect camera and the application in virtual reality system", in *Proceedings of the 4th International Conference on Virtual Reality*, 2018, pp. 21–25.

[79] D. Tuia *et al.*, "Perspectives in machine learning for wildlife conservation", *Nature communications*, vol. 13, no. 1, p. 792, 2022.

[80] M. Hahn-Klimroth, T. Kapetanopoulos, J. Gübert, and P. W. Dierkes, "Deep learning-based pose estimation for african ungulates in zoos", *Ecology and Evolution*, vol. 11, no. 11, pp. 6015–6032, 2021.

[81] F. Suo, K. Huang, G. Ling, Y. Li, and J. Xiang, "Fish keypoints detection for ecology monitoring based on underwater visual intelligence", in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, IEEE, 2020, pp. 542–547.

[82] J. Mei, J.-N. Hwang, S. Romain, C. Rose, B. Moore, and K. Magrane, "Absolute 3d pose estimation and length measurement of severely deformed fish from monocular videos in longline fishing", in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 2175–2179.

[83] K. Sheppard *et al.*, "Stride-level analysis of mouse open field behavior using deep-learning-based pose estimation", *Cell reports*, vol. 38, no. 2, p. 110 231, 2022.

[84] Z. Gong, Y. Zhang, D. Lu, and T. Wu, "Vision-based quadruped pose estimation and gait parameter extraction method", *Electronics*, vol. 11, no. 22, p. 3702, 2022.

[85] C. G. Lecomte, J. Audet, J. Harnie, and A. Frigon, "A validation of supervised deep learning for gait analysis in the cat", *Frontiers in Neuroinformatics*, vol. 15, p. 712 623, 2021.

[86] M. W. Mathis and A. Mathis, "Deep learning tools for the measurement of animal behavior in neuroscience", *Current opinion in neurobiology*, vol. 60, pp. 1–11, 2020.

[87] C. Fang, T. Zhang, H. Zheng, J. Huang, and K. Cuan, "Pose estimation and behavior classification of broiler chickens based on deep neural networks", *Computers and Electronics in Agriculture*, vol. 180, p. 105 863, 2021.

[88] P. Kumar, S. Chauhan, and L. K. Awasthi, "Human pose estimation using deep learning: Review, methodologies, progress and future research directions", *International Journal of Multimedia Information Retrieval*, pp. 1–33, 2022.

[89] S. Kulkarni, S. Deshmukh, F. Fernandes, A. Patil, and V. Jabade, "Poseanalyser: A survey on human pose estimation", *SN Computer Science*, vol. 4, no. 2, p. 136, 2023.

[90] G. Lan, Y. Wu, F. Hu, and Q. Hao, "Vision-based human pose estimation via deep learning: A survey", *IEEE Transactions on Human-Machine Systems*, 2022.

[91] S. Dubey and M. Dixit, "A comprehensive survey on human pose estimation approaches", *Multimedia Systems*, vol. 29, no. 1, pp. 167–195, 2023.

[92] B. Sapp, A. Toshev, and B. Taskar, "Cascaded models for articulated pose estimation", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2010, pp. 406–420.

[93] F. Wang and Y. Li, "Beyond physical connections: Tree models in human pose estimation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 596–603.

[94] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[95] I. Haritaoglu, D. Harwood, and L. S. Davis, "Ghost: A human body part labeling system using silhouettes", in *Proceedings. Fourteenth international conference on pattern recognition (cat. no. 98EX170)*, IEEE, vol. 1, 1998, pp. 77–82.

[96] C. Bregler and J. Malik, "Tracking people with twists and exponential maps", in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, IEEE, 1998, pp. 8–15.

[97] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human pose estimation using body parts dependent joint regressors", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3041–3048.

[98] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2602–2611.

[99] C. Szegedy *et al.*, "Going deeper with convolutions", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[100] 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*, 2016, pp. 770–778.

[101] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[102] 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*, 2019, pp. 5693–5703.

[103] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1831–1840.

[104] W. Tang, P. Yu, and Y. Wu, "Deeply learned compositional models for human pose estimation", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 190–206.

[105] W. Tang and Y. Wu, "Does learning specific features for related parts help human pose estimation?", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1107–1116.

[106] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[107] C. Chen, W. Zhu, and T. Norton, "Behaviour recognition of pigs and cattle: Journey from computer vision to deep learning", *Computers and Electronics in Agriculture*, vol. 187, p. 106 255, 2021.

[108] A. Mathis *et al.*, "Pretraining boosts out-of-domain robustness for pose estimation", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1859–1868.

[109] M. Riekert, A. Klein, F. Adrion, C. Hoffmann, and E. Gallmann, "Automatically detecting pig position and posture by 2d camera imaging and deep learning", *Computers and Electronics in Agriculture*, vol. 174, p. 105 391, 2020.

[110] H. Russello, R. van der Tol, and G. Kootstra, "T-leap: Occlusion-robust pose estimation of walking cows using temporal information", *Computers and Electronics in Agriculture*, vol. 192, p. 106 559, 2022.

[111] P. C. Bala, B. R. Eisenreich, S. B. M. Yoo, B. Y. Hayden, H. S. Park, and J. Zimmermann, "Automated markerless pose estimation in freely moving macaques with openmonkeystudio", *Nature communications*, vol. 11, no. 1, pp. 1–12, 2020.

[112] S. B. Negrete, R. Labuguen, J. Matsumoto, Y. Go, K.-i. Inoue, and T. Shibata, "Multiple monkey pose estimation using openpose", *bioRxiv*, 2021.

[113] Y. Yao, Y. Jafarian, and H. S. Park, "Monet: Multiview semi-supervised keypoint detection via epipolar divergence", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 753–762.

[114] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 34–50.

[115] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[116] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, 2017, pp. 11–19.

[117] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1913–1921.

[118] I. Goodfellow *et al.*, "Generative adversarial networks", *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[119] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi, "Self-supervised learning of interpretable keypoints from unlabelled videos", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8787–8797.

[120] J. Sosa, S. Perry, J. Alty, and D. Hogg, "Of mice and pose: 2d mouse pose estimation from unlabelled data and synthetic prior", in *Computer Vision Systems*, Springer Nature Switzerland, 2023, pp. 125–136, ISBN: 978-3-031-44137-0.

[121] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[122] J. J. Koenderink, "Pictorial relief", *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 356, no. 1740, pp. 1071–1086, 1998.

[123] E. Marinoiu, D. Papava, and C. Sminchisescu, "Pictorial human spaces: How well do humans perceive a 3d articulated pose?", in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1289–1296.

[124] A. Mittal, L. Zhao, and L. S. Davis, "Human body pose estimation using silhouette shape analysis", in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, IEEE, 2003, pp. 263–270.

[125] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images", in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 3108–3113.

[126] A. Agarwal and B. Triggs, "3d human pose from silhouettes by relevance vector regression", in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, IEEE, vol. 2, 2004, pp. II–II.

[127] C. Sminchisescu and B. Triggs, "Kinematic jump processes for monocular 3d human tracking", in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, IEEE, vol. 1, 2003, pp. I–I.

[128] H.-J. Lee and Z. Chen, "Determination of 3d human body postures from a single view", *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 2, pp. 148–168, 1985.

[129] M. Ben Gamra and M. A. Akhloufi, "A review of deep learning techniques for 2d and 3d human pose estimation", *Image and Vision Computing*, vol. 114, p. 104 282, 2021, ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2021.104282. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885621001876.

[130] C. Zheng *et al.*, "Deep learning-based human pose estimation: A survey", *ACM Computing Surveys*, 2020.

[131] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model", *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[132] T. W. Dunn *et al.*, "Geometric deep learning enables 3d kinematic profiling across species and environments", *Nature methods*, vol. 18, no. 5, pp. 564–573, 2021.

[133] D. Joska *et al.*, "Acinoset: A 3d pose estimation dataset and baseline models for cheetahs in the wild", in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 13 901–13 908.

[134] L. Del Pero, S. Ricco, R. Sukthankar, and V. Ferrari, "Behavior discovery and alignment of articulated object classes from unstructured video", *International Journal of Computer Vision*, vol. 121, pp. 303–325, 2017.

[135] E. Borenstein, E. Sharon, and S. Ullman, "Combining top-down and bottom-up segmentation", in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, IEEE, 2004, pp. 46–46.

[136] B. Tekin, P. Márquez-Neila, M. Salzmann, and P. Fua, "Learning to fuse 2d and 3d image cues for monocular body pose estimation", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3941–3950.

[137] D. Mehta *et al.*, "Vnect: Real-time 3d human pose estimation with a single rgb camera", *Acm transactions on graphics (tog)*, vol. 36, no. 4, pp. 1–14, 2017.

[138] K. Zhou, X. Han, N. Jiang, K. Jia, and J. Lu, "Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2344–2353.

[139] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3d pose estimation", in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[140] J. Wang, S. Huang, X. Wang, and D. Tao, "Not all parts are created equal: 3d pose estimation by modeling bi-directional dependencies of body parts", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7771–7780.

[141] J. Cai, H. Liu, R. Ding, W. Li, J. Wu, and M. Ban, "Htnet: Human topology aware network for 3d human pose estimation", *arXiv preprint arXiv:2302.09790*, 2023.

[142] H. Choi, G. Moon, and K. M. Lee, "Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose", in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, Springer, 2020, pp. 769–787.

[143] E. Jahangiri and A. L. Yuille, "Generating multiple diverse hypotheses for human 3d pose consistent with 2d joint detections", in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 805–814.

[144] Y. Kang, Y. Liu, A. Yao, S. Wang, and E. Wu, "3d human pose lifting with grid convolution", in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[145] D. Drover, C.-H. Chen, A. Agrawal, A. Tyagi, and C. Phuoc Huynh, "Can 3d pose be learned from 2d projections alone?", in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[146] T. Wehrbein, M. Rudolph, B. Rosenhahn, and B. Wandt, "Probabilistic monocular 3d human pose estimation with normalizing flows", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 199–11 208.

[147] U. Iqbal, P. Molchanov, and J. Kautz, "Weakly-supervised 3d human pose learning via multi-view images in the wild", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5243–5252.

[148] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7025–7034.

[149] S. Li and A. B. Chan, "3d human pose estimation from monocular images with deep convolutional neural network", in *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12*, Springer, 2015, pp. 332–347.

[150] D. C. Luvizon, D. Picard, and H. Tabia, "2d/3d pose estimation and action recognition using multitask deep learning", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5137–5146.

[151] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3d pose estimation from a single image", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2500–2509.

[152] H. Rhodin, M. Salzmann, and P. Fua, "Unsupervised geometry-aware representation for 3d human pose estimation", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 750–767.

[153] R. Mitra, N. B. Gundavarapu, A. Sharma, and A. Jain, "Multiview-consistent semi-supervised learning for 3d human pose estimation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6907–6916.

[154] M. Kocabas, S. Karagoz, and E. Akbas, "Self-supervised learning of 3d human pose using multi-view geometry", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1077–1086.

[155] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3d human pose estimation from monocular video", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4966–4975.

[156] F. Scheepers, R. E. Parent, W. E. Carlson, and S. F. May, "Anatomy-based modeling of the human musculature", in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 163–172.

[157] B. Allen, B. Curless, Z. Popović, and A. Hertzmann, "Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis", in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Citeseer, 2006, pp. 147–156.

[158] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2252–2261.

[159] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5253–5263.

[160] G. Pavlakos *et al.*, "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*, 2019, pp. 10 975–10 985.

[161] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7122–7131.

[162] R. A. Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7297–7306.

[163] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image", in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, Springer, 2016, pp. 561–578.

[164] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black, "3d menagerie: Modeling the 3d shape and pose of animals", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6365–6373.

[165] S. Zuffi, A. Kanazawa, and M. J. Black, "Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3955–3963.

[166] E. B. Contreras, R. J. Sutherland, M. H. Mohajerani, and I. Q. Whishaw, "Challenges of a small world analysis for the continuous monitoring of behavior in mice", *Neuroscience & Biobehavioral Reviews*, p. 104 621, 2022.

[167] J. D. Marshall, T. Li, J. H. Wu, and T. W. Dunn, "Leaving flatland: Advances in 3d behavioral measurement", *Current Opinion in Neurobiology*, vol. 73, p. 102 522, 2022.

[168] A. Monsees *et al.*, "Estimation of skeletal kinematics in freely moving rodents", *Nature Methods*, vol. 19, no. 11, pp. 1500–1509, 2022.

[169] E. Heuer, R. F Rosen, A. Cintron, and L. C Walker, "Nonhuman primate models of alzheimer-like cerebral proteopathy", *Current pharmaceutical design*, vol. 18, no. 8, pp. 1159–1169, 2012.

[170] T. Saito *et al.*, "Single app knock-in mouse models of alzheimer's disease", *Nature neuroscience*, vol. 17, no. 5, pp. 661–663, 2014.

[171] Y. Lee, V. L. Dawson, and T. M. Dawson, "Animal models of parkinson's disease: Vertebrate genetics", *Cold Spring Harbor perspectives in medicine*, vol. 2, no. 10, a009324, 2012.

[172] T. Philips and J. D. Rothstein, "Rodent models of amyotrophic lateral sclerosis", *Current protocols in pharmacology*, vol. 69, no. 1, pp. 5–67, 2015.

[173] I. Allodi, R. Montañana-Rosell, R. Selvan, P. Löw, and O. Kiehn, "Locomotor deficits in a mouse model of als are paralleled by loss of v1-interneuron connections onto fast motor neurons", *Nature Communications*, vol. 12, no. 1, p. 3251, 2021.

[174] E. M. Fisher and D. M. Bannerman, "Mouse models of neurodegeneration: Know your question, know your mouse", *Science Translational Medicine*, vol. 11, no. 493, eaaq1818, 2019.

[175] T. G. Hampton and I. Amende, "Treadmill gait analysis characterizes gait alterations in parkinson's disease and amyotrophic lateral sclerosis mouse models", *Journal of motor behavior*, vol. 42, no. 1, pp. 1–4, 2009.

[176] C. M. Rostosky and I. Milosevic, "Gait analysis of age-dependent motor impairments in mice with neurodegeneration", *JoVE (Journal of Visualized Experiments)*, no. 136, e57752, 2018.

[177] L. P. Noldus, R. J. Trienes, A. H. Hendriksen, H. Jansen, and R. G. Jansen, "The observer video-pro: New software for the collection, management, and presentation of time-structured data from videotapes and digital media files", *Behavior Research Methods, Instruments, & Computers*, vol. 32, pp. 197–206, 2000.

[178] R. F. Olivo and M. C. Thompson, "Monitoring animals' movements using digitized video images", *Behavior Research Methods, Instruments, & Computers*, vol. 20, pp. 485–490, 1988.

[179] J. A. Bender, E. M. Simpson, and R. E. Ritzmann, "Computer-assisted 3d kinematic analysis of all leg joints in walking insects", *PloS one*, vol. 5, no. 10, e13617, 2010.

[180] A. Spink, R. Tegelenbosch, M. Buma, and L. Noldus, "The ethovision video tracking system—a tool for behavioral phenotyping of transgenic mice", *Physiology & behavior*, vol. 73, no. 5, pp. 731–744, 2001.

[181] G. Card and M. H. Dickinson, "Visually mediated motor planning in the escape response of drosophila", *Current Biology*, vol. 18, no. 17, pp. 1300–1307, 2008.

[182] T. Kirkpatrick, C. W. Schneider, and R. Pavloski, "A computerized infrared monitor for following movement in aquatic animals", *Behavior Research Methods, Instruments, & Computers*, vol. 23, no. 1, pp. 16–22, 1991.

[183] M. Gershow *et al.*, "Controlling airborne cues to study small animal navigation", *Nature methods*, vol. 9, no. 3, pp. 290–296, 2012.

[184] Z. Liu, J. Zhu, J. Bu, and C. Chen, "A survey of human pose estimation: The body parts parsing based methods", *Journal of Visual Communication and Image Representation*, vol. 32, pp. 10–19, 2015.

[185] M. Shooter, C. Malleson, and A. Hilton, "Sydog: A synthetic dog dataset for improved 2d pose estimation", *arXiv preprint arXiv:2108.00249*, 2021.

[186] S. Kearney, W. Li, M. Parsons, K. I. Kim, and D. Cosker, "Rgbd-dog: Predicting canine pose from rgbd sensors", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8336–8345.

[187] Y. Sato *et al.*, "Markerless analysis of hindlimb kinematics in spinal cord-injured mice through deep learning", *Neuroscience Research*, 2021.

[188] P. Karashchuk *et al.*, "Anipose: A toolkit for robust markerless 3d pose estimation", *Cell reports*, vol. 36, no. 13, p. 109 730, 2021.

[189] S. Raman, R. Maskeliūnas, and R. Damaševičius, "Markerless dog pose recognition in the wild using resnet deep learning model", *Computers*, vol. 11, no. 1, p. 2, 2022.

[190] R. Labuguen, D. K. Bardeloza, S. B. Negrete, J. Matsumoto, K. Inoue, and T. Shibata, "Primate markerless pose estimation and movement analysis using deeplabcut", in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, IEEE, 2019, pp. 297–300.

[191] Y. Wang, J. Li, Y. Zhang, and R. O. Sinnott, "Identifying lameness in horses through deep learning", in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021, pp. 976–985.

[192] H. Liu, A. R. Reibman, and J. P. Boerman, "Video analytic system for detecting cow structure", *Computers and Electronics in Agriculture*, vol. 178, p. 105 761, 2020.

[193] F. Farahnakian, J. Heikkonen, and S. Björkman, "Multi-pig pose estimation using deeplabcut", in *2021 11th International Conference on Intelligent Control and Information Processing (ICICIP)*, IEEE, 2021, pp. 143–148.

[194] L. Jiang, S. Liu, X. Bai, and S. Ostadabbas, "Prior-aware synthetic data to the rescue: Animal pose estimation with very limited real data", *arXiv preprint arXiv:2208.13944*, 2022.

[195] N. Rüegg, S. Tripathi, K. Schindler, M. J. Black, and S. Zuffi, "Bite: Beyond priors for improved three-d dog pose estimation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8867–8876.

[196] T. J. Cashman and A. W. Fitzgibbon, "What shape are dolphins? building 3d morphable models from 2d images", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 232–244, 2012.

[197] S. Goel, A. Kanazawa, and J. Malik, "Shape and viewpoint without keypoints", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 88–104.

[198] A. Kanazawa, S. Kovalsky, R. Basri, and D. Jacobs, "Learning 3d deformation of animals from 2d images", in *Computer Graphics Forum*, Wiley Online Library, vol. 35, 2016, pp. 365–374.

[199] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge, and M. W. Mathis, "Using deeplabcut for 3d markerless pose estimation across species and behaviors", *Nature protocols*, vol. 14, no. 7, pp. 2152–2176, 2019.

[200] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution", in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 694–711.

[201] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[202] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[203] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[204] S. R. Pfohl, M. T. Halicek, and C. S. Mitchell, "Characterization of the contribution of genetic background and gender to disease progression in the sod1 g93a mouse model of amyotrophic lateral sclerosis: A meta-analysis", *Journal of neuromuscular diseases*, vol. 2, no. 2, pp. 137–150, 2015.

[205] T. Hatzipetros, J. D. Kidd, A. J. Moreno, K. Thompson, A. Gill, and F. G. Vieira, "A quick phenotypic neurological scoring system for evaluating disease progression in the sod1-g93a mouse model of als", *Journal of visualized experiments: JoVE*, no. 104, 2015.

[206] W. T. B. A. H. AUSTRALIA, "Australian code of practice for the care and use of animals for scientific purposes", 2011.

[207] J. M. Collins, R. A. Atkinson, L. M. Matthews, I. C. Murray, S. E. Perry, and A. E. King, "Sarm1 knockout modifies biomarkers of neurodegeneration and spinal cord circuitry but not disease progression in the msod1g93a mouse model of als", *Neurobiology of Disease*, vol. 172, p. 105 821, 2022.

[208] 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*, vol. 36, no. 7, pp. 1325–1339, 2013.

[209] A. I. Hsu and E. A. Yttri, "B-soid, an open-source unsupervised algorithm for identification and fast prediction of behaviors", *Nature communications*, vol. 12, no. 1, p. 5188, 2021.

[210] L. van der Maaten and G. Hinton, "Visualizing data using t-sne", *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html.

[211] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, *Detectron2*, https://github.com/facebookresearch/detectron2, 2019.

[212] R. Nevatia and T. O. Binford, "Description and recognition of curved objects", *Artificial intelligence*, vol. 8, no. 1, pp. 77–98, 1977.

[213] W. Kim *et al.*, "Pedx: Benchmark dataset for metric 3-d pose estimation of pedestrians in complex urban intersections", *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1940–1947, 2019.

[214] L. Zheng, Y. Huang, H. Lu, and Y. Yang, "Pose-invariant embedding for deep person re-identification", *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4500–4509, 2019.

[215] K. Rematas, I. Kemelmacher-Shlizerman, B. Curless, and S. Seitz, "Soccer on your tabletop", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4738–4747.

[216] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images", *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 1, pp. 44–58, 2005.

[217] H. Jiang, "3d human pose reconstruction using millions of exemplars", in *2010 20th International Conference on Pattern Recognition*, IEEE, 2010, pp. 1674–1677.

[218] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham, "3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2601–2608.

[219] C.-H. Chen and D. Ramanan, "3d human pose estimation= 2d pose estimation+ matching", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7035–7043.

[220] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[221] D. Mehta *et al.*, "Monocular 3d human pose estimation in the wild using improved cnn supervision", in *3D Vision (3DV), 2017 Fifth International Conference on*, IEEE, 2017. DOI: 10.1109/3dv.2017.00064. [Online]. Available: http://gvv.mpi-inf.mpg.de/3dhp_dataset.

[222] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1145–1153.

[223] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks", *Advances in neural information processing systems*, vol. 30, 2017.

[224] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi, "Unsupervised learning of object landmarks through conditional image generation", *Advances in neural information processing systems*, vol. 31, 2018.

[225] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[226] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[227] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks", in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, Springer, 2016, pp. 702–716.

[228] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods", *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.

[229] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference", *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 2617–2680, 2021.

[230] J. Milnor and D. W. Weaver, *Topology from the differentiable viewpoint*. Princeton university press, 1997, vol. 21.

[231] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design", in *International Conference on Machine Learning*, PMLR, 2019, pp. 2722–2730.

[232] A. Abdelhamed, M. A. Brubaker, and M. S. Brown, "Noise flow: Noise modeling with conditional normalizing flows", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3165–3173.

[233] P. N. Ward, A. Smofsky, and A. J. Bose, "Improving exploration in soft-actor-critic with normalizing flows policies", *arXiv preprint arXiv:1906.02771*, 2019.

[234] B. Biggs, D. Novotny, S. Ehrhardt, H. Joo, B. Graham, and A. Vedaldi, "3d multi-bodies: Fitting sets of plausible 3d human models to ambiguous image data", *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 496–20 507, 2020.

[235] A. Zanfir, E. G. Bazavan, H. Xu, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "Weakly supervised 3d human pose and shape reconstruction with normalizing flows", in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, Springer, 2020, pp. 465–481.

[236] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp", *arXiv preprint arXiv:1605.08803*, 2016.

[237] Q. Nie, J. Wang, X. Wang, and Y. Liu, "View-invariant human action recognition based on a 3d bio-constrained skeleton model", *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3959–3972, 2019.

[238] Z. Li, X. Wang, F. Wang, and P. Jiang, "On boosting single-frame 3d human pose estimation via monocular videos", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2192–2201.

[239] C. Goodall, "Procrustes methods in the statistical analysis of shape", *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 53, no. 2, pp. 285–321, 1991.

[240] C. Sun, D. Thomas, and H. Kawasaki, "Unsupervised 3d human pose estimation in multi-view-multi-pose video", in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 5959–5964.

[241] 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*, 2014, pp. 3686–3693.

[242] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.

[243] D. Joska *et al.*, "Acinoset: A 3d pose estimation dataset and baseline models for cheetahs in the wild", in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 901–13 908. DOI: 10.1109/ICRA48506.2021.9561338.

[244] A. Gosztolai *et al.*, "Liftpose3d, a deep learning-based approach for transforming two-dimensional to three-dimensional poses in laboratory animals", *Nature methods*, vol. 18, no. 8, pp. 975–981, 2021.

[245] Y. Zhang and H. S. Park, "Multiview supervision by registration", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 420–428.

[246] X. Dai, S. Li, Q. Zhao, and H. Yang, "Unsupervised 3d animal canonical pose estimation with geometric self-supervision", in *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, IEEE, 2023, pp. 1–8.

[247] C. Doersch and A. Zisserman, "Sim2real transfer learning for 3d human pose estimation: Motion to the rescue", *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[248] C. Gao, Q. Liu, Q. Xu, L. Wang, J. Liu, and C. Zou, "Sketchycoco: Image generation from freehand scene sketches", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5174–5183.

[249] W. Chen and J. Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9416–9425.

[250] A. Voynov, K. Aberman, and D. Cohen-Or, "Sketch-guided text-to-image diffusion models", in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–11.

[251] S.-I. Cheng, Y.-J. Chen, W.-C. Chiu, H.-Y. Tseng, and H.-Y. Lee, "Adaptively-realistic image generation from stroke and sketch with diffusion model", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 4054–4062.

[252] H. J. Smith, Q. Zheng, Y. Li, S. Jain, and J. K. Hodgins, "A method for animating children's drawings of the human figure", *ACM Transactions on Graphics*, vol. 42, no. 3, pp. 1–15, 2023.

[253] A. Ramesh *et al.*, *Zero-shot text-to-image generation*, 2021. arXiv: 2102.12092 [cs.CV].

[254] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 221–230.

Appendix A

Chapter 3

Overview

This supplemental material provides extended descriptions of the work done in the corresponding chapter of this thesis. In section A.1, we include more details about the data acquisition process and some insight related to previous data analyses. section A.2 provides more information related to the 3D mouse model used for obtaining the synthetic 2D poses. Finally, section A.3 includes implementation details. To ensure the reproducibility of our approach, we have added links to the publicly available models utilised. Our code will be made accessible shortly after the publication of this thesis.

Note that most of the figures in these sections are of high quality. If something is not completely visible on the current figure size, simply zoom in for better visualisation. Some visualisations also serve as links to videos or other visual resources. This information is indicated in the captions of the respective figures.

Project website: https://josesosajs.github.io/micepose/

Paper (ICVS 2023): https://tinyurl.com/y3thhbay

Presentation (ICVS 2023): https://tinyurl.com/mryj5rvb

Pre-print: https://arxiv.org/abs/2307.13361

Poster (CVPR-W 2022): https://tinyurl.com/48enkhu8

A.1 Data acquisition and previous analysis

We use a commercial tool called DigiGait (https://mousespecifics.com/digigait/) to acquire recordings for training our models. DigiGait includes a treadmill and camera to record animals while running or walking. Figure A.1 works as a link to one of the videos from our dataset. The complete recordings will be made available under reasonable request.

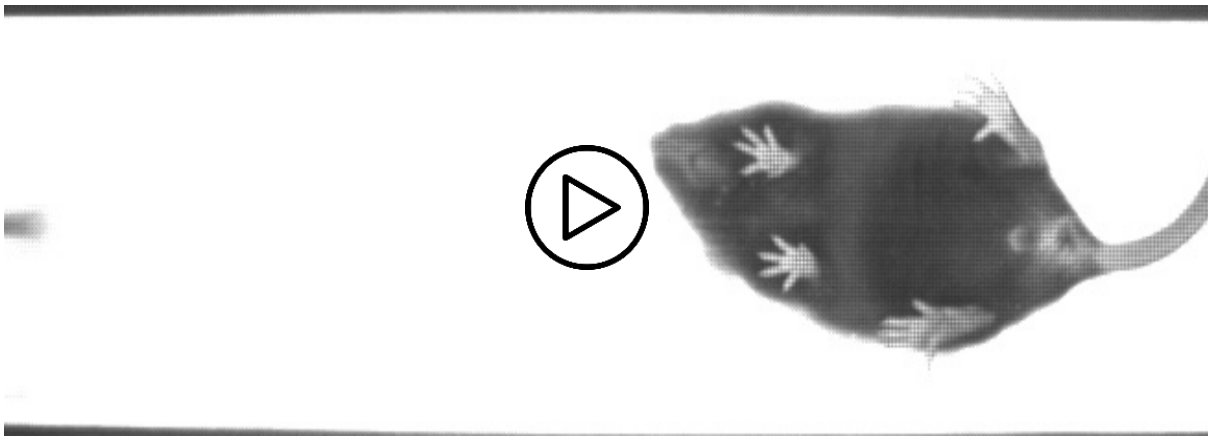


Figure A.1: Example of recordings. The dataset includes recordings that have a similar structure. Most show a mouse running at three different speeds for about 20 seconds each, with 10-second intervals in between. During these intervals, the mouse is simply walking on the treadmill. Note that the figure acts as a link for accessing the video.

DigiGait provides both equipment and software for data acquisition and analysis. However, we do not have access to the software as we are not involved in the data acquisition stage. The DigiGait software potentially relies on some elementary computer vision/image processing techniques to identify the regions of the mouse's paws in each frame and measure their areas at different times. From this process, the software generates a set of 16 gait parameters and plots illustrating the areas of the paws, as presented in Figure A.2. As the software is not open-source, we lack specific technical details on how it works.

The gait parameters that DigiGait produces are as follows: stride left, stride right, step width front, step width back, step length right, step length left, step length front, step length back, stride width left, stride width right, stride left, stride right, step length right, step length left, step length front, and step length back. Figure A.3 shows an example of the values for each parameter averaged for a set of video sequences A, B, C, D, and E.

DigiGait involves much manual work to revise the estimated positions of the paws and can only manage short video sequences. To address this, we began exploring the use of deep learning

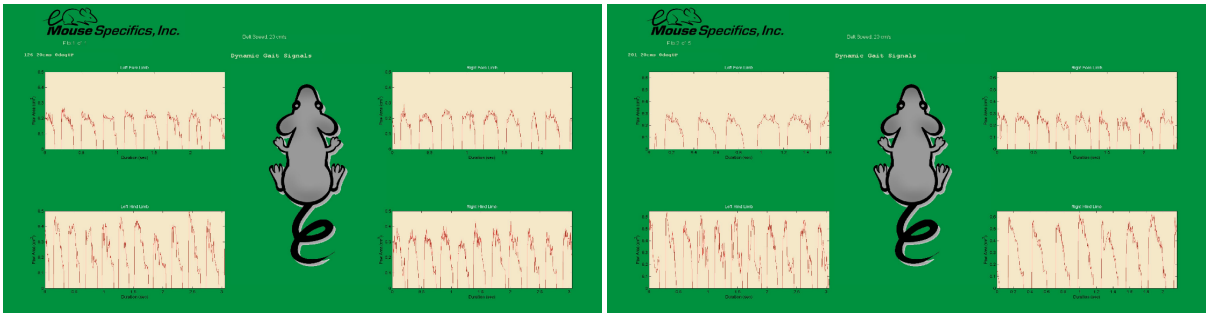


Figure A.2: DigiGait's output example. The DigiGait software generates plots showing the paws' areas over time. However, it is unclear which processes the Digigait software uses for the videos. We hypothesise that it employs basic image processing techniques.

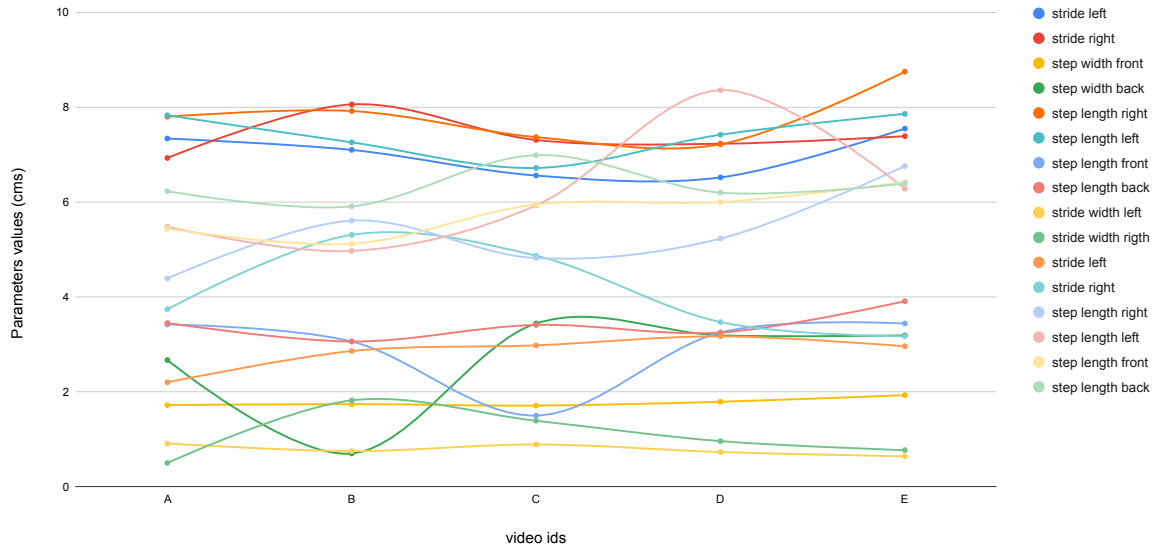


Figure A.3: Example of gait parameter values for different videos. In addition to producing the plots, DigiGait estimates values for a set of metrics helpful for gait analysis. The inset legend denotes these metrics.

and advanced computer vision techniques to replicate the functions of DigiGait. By doing so, we aim to decrease the need for human involvement in estimating gait parameters.

We experiment with a simple pipeline illustrated in Figure A.4, relying on existing methods for image segmentation (OSVOS) [254] and bounding box detection (Region Proposal Network) [25]. This permits estimating segmentation masks and bounding boxes for the mouse's paws in each video frame. However, we later discarded this approach since it limits the body structure to only four joint positions and required manually producing some segmentation masks to fine-tune the image segmentation model.

Using that primitive approach, we estimate the paw areas for every image in a given video

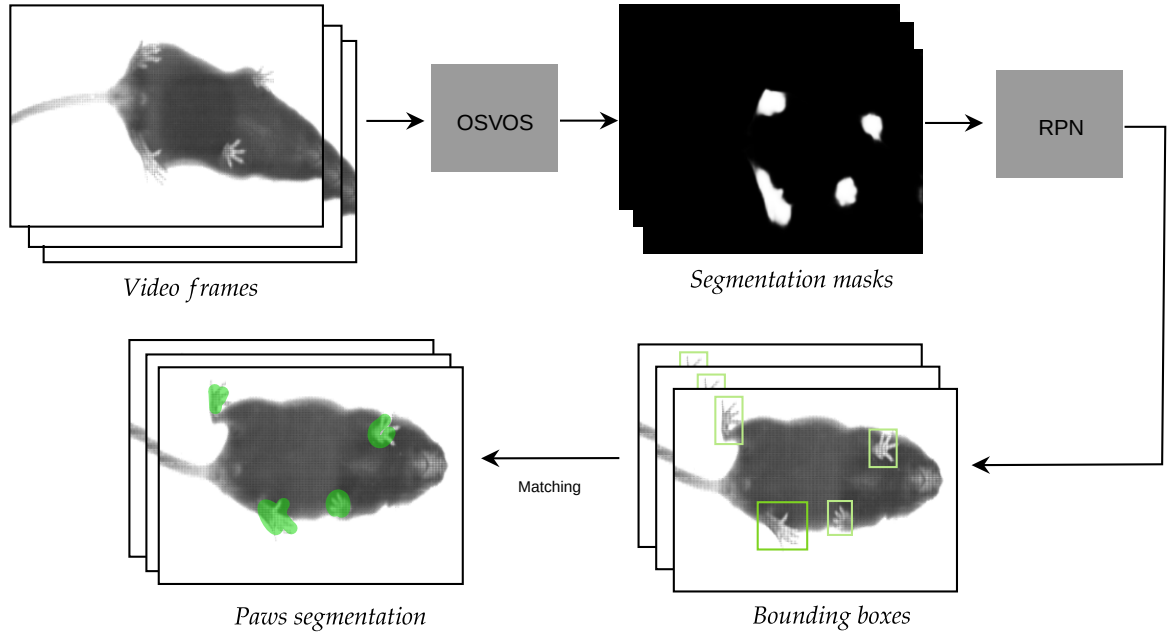


Figure A.4: Preliminary computer vision approach for estimating paw areas. We rely on deep learning approaches, such as OSVOS and Region Proposal Network, to detect paws in each frame of a given video and estimate the corresponding segmentation masks and bounding boxes.

sequence. In Figure A.5, we show some of the plots, including the areas of each paw for consecutive images corresponding to three seconds of video. We use videos showing the mice running at three speeds, 10cm/s, 20cm/s and 30cm/s. Note that each second of the video involves the predictions for 164 images/frames. The first two plots in each row correspond to the left and right front paws, respectively, while the third and last plots correspond to the left and right back paws. The red crosses on each plot correspond to the peaks, indicating when that specific paw is probably in full contact with the treadmill.

Furthermore, in Figure A.6 we provide links to videos illustrating the results of tracking all four paws using the previous method. Note that Figure A.6 acts as a link to the folder containing the videos. Additionally, links for individual videos are provided in the figure's caption.

A.2 Synthetic mouse model

To manipulate the 3D mouse model, we utilise Blender, which is a free, open-source 3D computer graphics software for creating animations. Blender can be downloaded directly from its website: https://www.blender.org/download/. In the work done throughout the chapter, we rely on

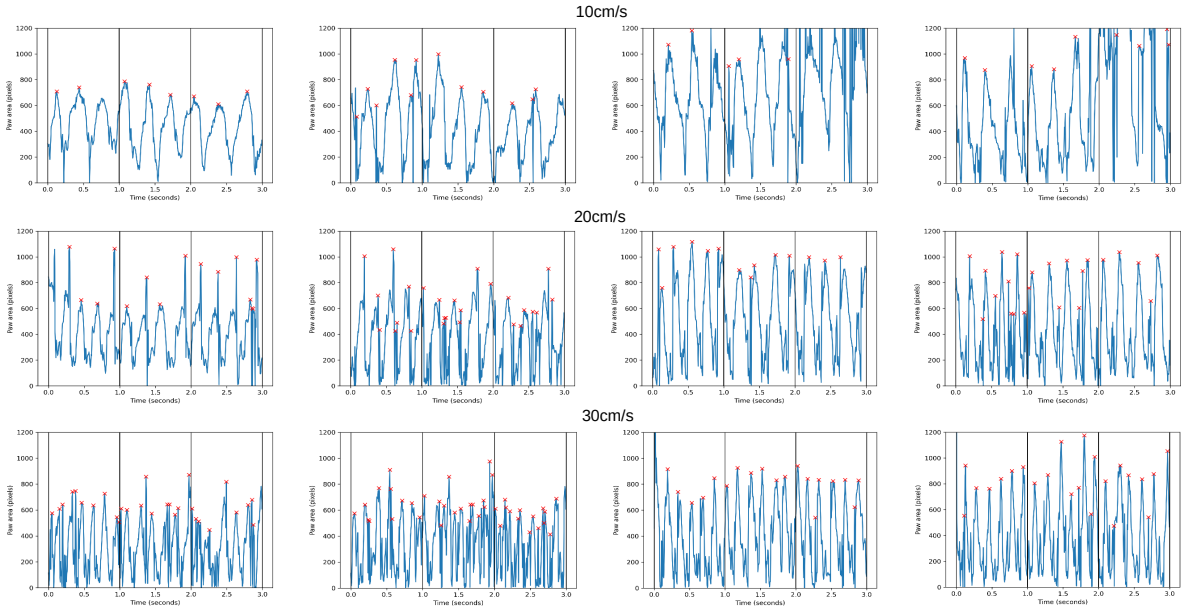


Figure A.5: Estimated paw areas over time. The plots show the estimated paw areas using the approach depicted in Figure A.4. Each paw is plotted for three seconds, corresponding to 492 images, at three different speeds. The first and second columns depict the areas for the front paws, while the remaining columns are for the rear paws. On the plots, the maximum area is marked with a red cross, indicating when the paw is in complete contact with the treadmill.

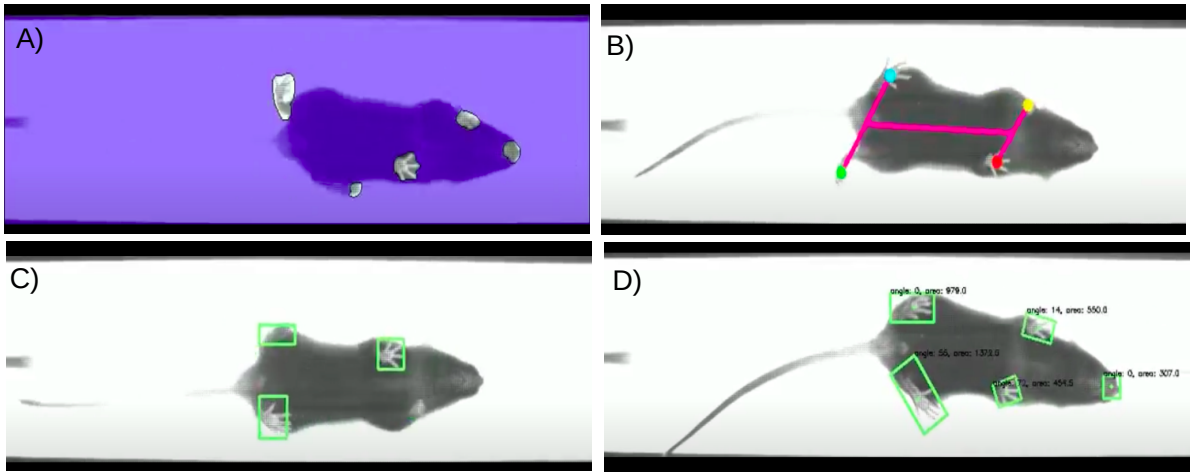


Figure A.6: Different video visualisations for the paw tracking. The figure acts a link for the shared folder containing the videos. A) https://tinyurl.com/paw-tracking-seg, B) https://tinyurl.com/paw-tracking-pose, C) https://tinyurl.com/paw-tracking-bb, D) https://tinyurl.com/paw-tracking-angle.

the existing models from [63]. These models are freely available, and could be downloaded from this website: https://osf.io/h3ec5/. In particular, we use the model showing the mouse freely moving in an open field: *SyntheticData_DemoScene_Openfield.blend*. We introduce some changes to make the mouse follow a similar moving trajectory as in the videos from our dataset, i.e. to simulate the mouse running on the transparent treadmill. Figure A.7 displays examples

of rendered images from the mouse model.

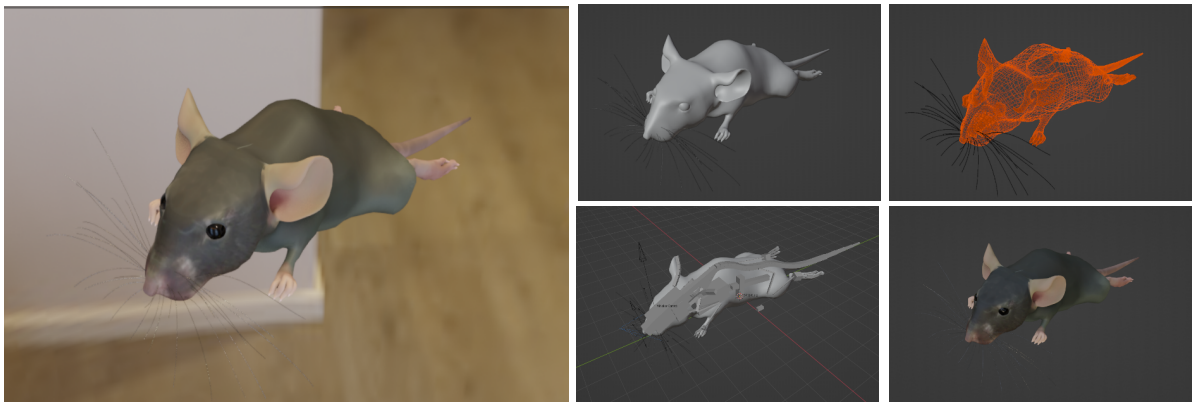


Figure A.7: Visualisations of mouse 3D model. The figure visualises different layers from the 3D mouse model used to generate the animations and, consequently, the prior of 2D poses. The examples are just for illustrative purposes.

Figure A.8: Visualisation of an animation using the 3D mouse model. The video shows a sequence of rendered images using the 3D mouse model. Note that for the main experiments, we did not use any images from this model; just the 2D poses were needed. The figure acts as a link for accessing the video.

Technically, the 3D mouse model consists of multiple objects, each representing a different joint of the mouse's body, such as the snout, tail, and limbs. Blender allows for Python integration, making it easy to extract the positions of such objects in the 3D model for every frame in a given animation. Our primary focus is obtaining the joint positions for each image in the animated sequences to create a prior. The Python script used to extract the joint positions in Blender can be found at the following link: https://github.com/ubcbraincircuits/mCBF/blob/master/mCBF-2d-3d_marker-extraction.py. For illustrative purposes Figure A.8 shows an animation created with the 3D mouse model. Note that we usually disregard the images from the animated sequences and only use the extracted joint positions.

A.3 Implementation details

The structures of networks Φ, Ω, and D are the same as described in Table B.2, Table B.3, and Table B.5 from Appendix B respectively. The network Ψ was implemented in line with [119].

Appendix B

Chapter 4

Overview

In the following sections we provide comparative per-activity quantitative results on Human3.6M (section B.1); and additional qualitative results for Human3.6M [47] (section B.2), MPI-INF-3DHP[49] (section B.3), and HandDB[222] (section B.4) datasets. Moreover, section B.5 includes visualisations of some intermediate representations during and after training the model. Finally, section B.6 provides more details about the implementation and structure of the networks. Our code will be made accessible shortly after the publication of this thesis.

Note that most of the figures in these sections are of high quality. If something is not completely visible on the current figure size, simply zoom in for better visualisation. Some visualisations also serve as links to videos or other visual resources. This information is indicated in the captions of the respective figures.

Project website: https://josesosajs.github.io/imagepose/

Poster (CVPR 2023): https://tinyurl.com/5xwukfby

Paper: https://tinyurl.com/mr3hcush

Supplemental material: https://tinyurl.com/2j33ub85

Pre-print: https://arxiv.org/pdf/2304.02349.pdf

Poster (BMVA Meeting 2022): https://tinyurl.com/cp9z2bss

B.1 Quantitative results on Human3.6M and MPI-INF-3DHP datasets

Method	Assumptions	Dir.	Disc.	Eat	Greet	Phon.	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	Walk	WalkD	WalkT	Avg.(↓)
Chen [219]	Full-3D	89.8	97.6	89.9	107.9	107.3	139.2	93.6	136.1	133.1	240.1	106.6	106.2	87.0	114.0	**90.6**	114.2
Kundu [61]	3D Kin.	**80.2**	81.3	**86.0**	**86.7**	**94.1**	**83.4**	87.5	**84.2**	**101.2**	**110.9**	**86.0**	87.8	**86.9**	**94.3**	90.9	89.4
Ours	Unp. 2D	84.4	**77.8**	89.0	99.2	100.6	101.8	**77.2**	86.5	112.2	144.4	97.3	**80.4**	93.6	103.3	102.5	96.7

Table B.1: Extended quantitative results on the Human3.6M dataset. The P-MPJPE for each activity on the Human3.6M test set (subjects 9 and 11). The performance is compared with two state-of-the-art approaches for which per-activity data is available. Note that by only using unpaired 2D poses, we outperform methods that rely on paired 3D annotations [219]. We perform similarly; and even better for some activities (in bold) than methods relying on 3D kinematic constraints [61].

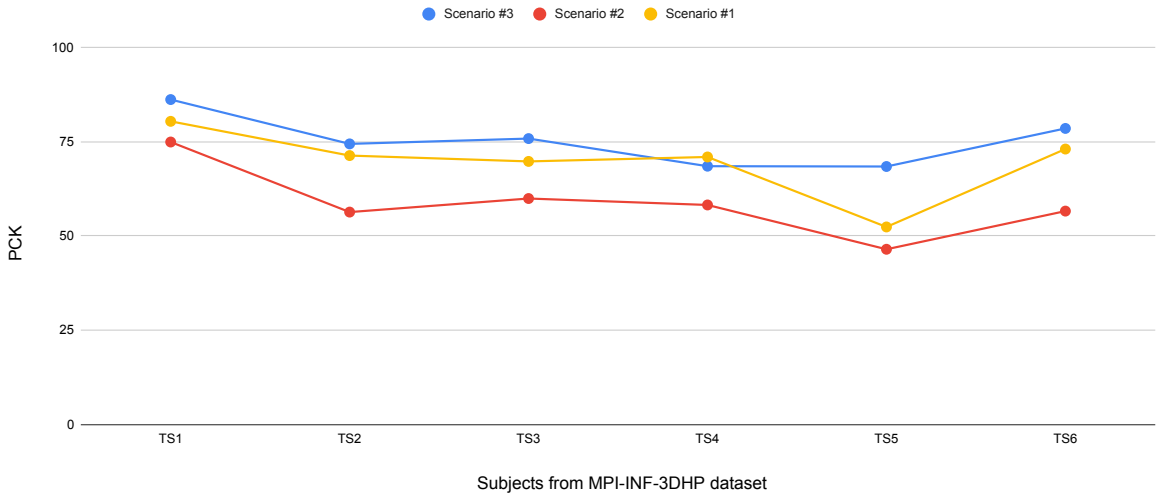


Figure B.1: Distribution of PCK scores per subject in MPI-INF-3DHP dataset. Similar to Human3.6M, the test set of MPI-INF-3DHP dataset consists of subjects identified as (TS_1, ..., TS_6). Apart from the avcerage PCK score provied on the chapter, we breakdown the score to show the individual PCK for each subject in the test set. The lowest score occurs for subject on outdoor settings.

B.2 Qualitative results on Human3.6M

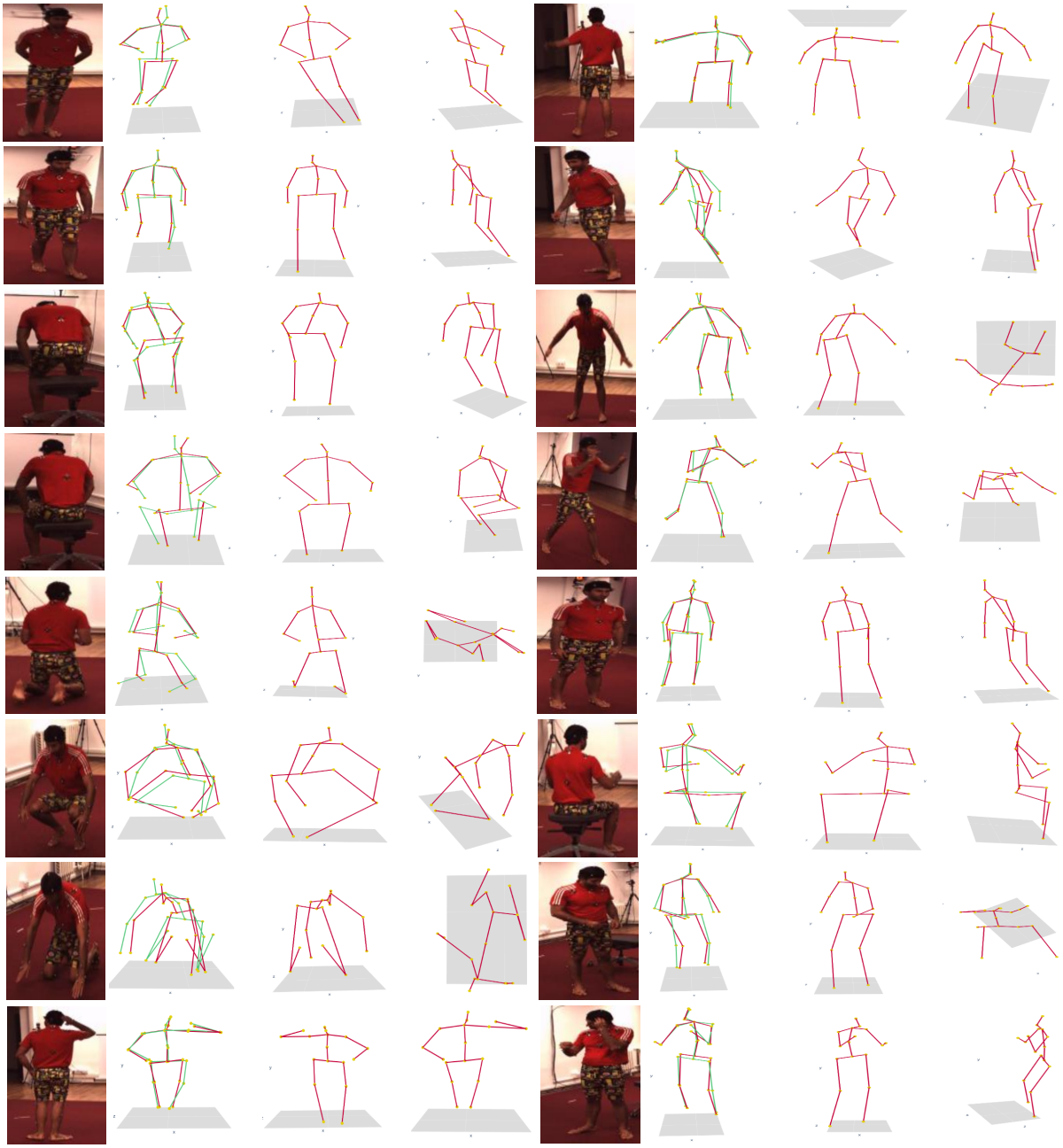


Figure B.2: 3D pose predictions on images corresponding to subject 9 (S9) from Human3.6M dataset. The first and fifth columns show the input image, and the following columns (second and sixth) display the actual 3D pose from the dataset (coloured in green) aligned with the 3D pose predicted by our model (coloured in red). The remaining columns show novel views of the predicted 3D pose.

Qualitative results on Human3.6M (continue)

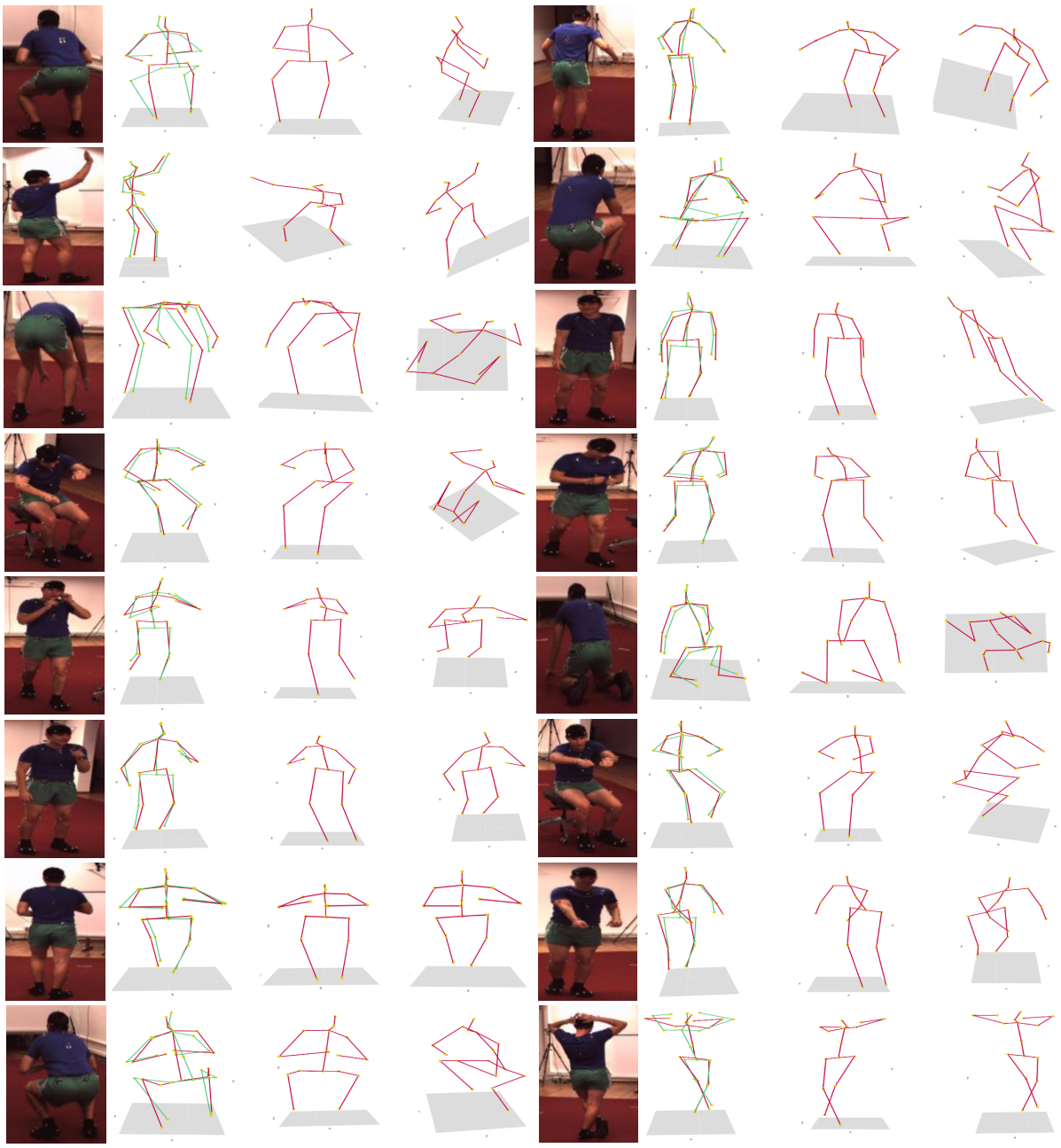


Figure B.3: 3D pose predictions on images corresponding to subject 11 (S11) from Human3.6M dataset. The first and fifth columns show the input image, and the following columns (second and sixth) display the actual 3D pose from the dataset (coloured in green) aligned with the 3D pose predicted by our model (coloured in red). The remaining columns show novel views of the predicted 3D pose.

B.3 Qualitative results on MPI-INF-3DHP

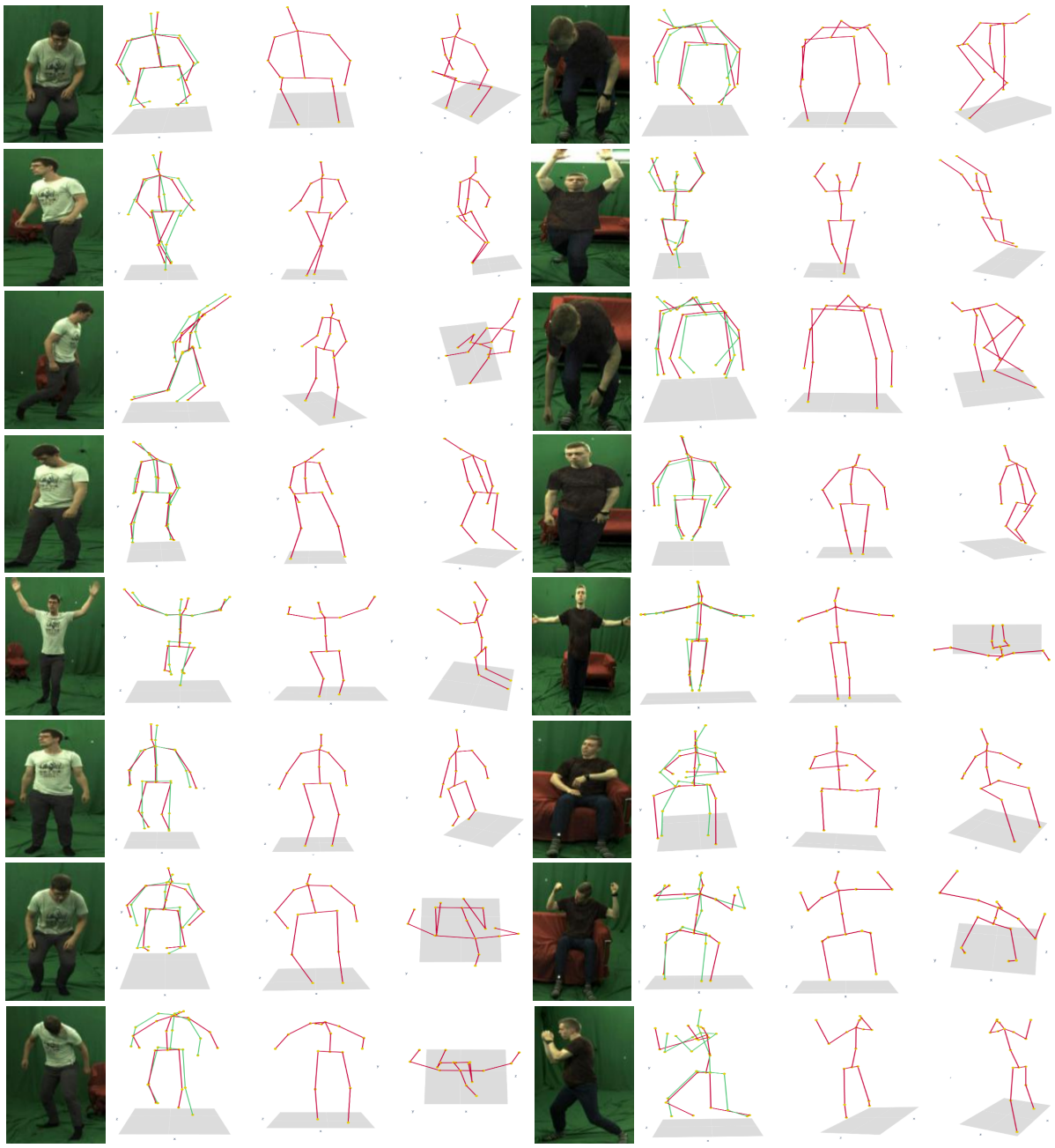


Figure B.4: 3D pose predictions on images corresponding to subjects 1 and 2 from MPI-INF-3DHP dataset. The first and fifth columns show the input image, and the following columns (second and sixth) display the actual 3D pose from the dataset (coloured in green) aligned with the 3D pose predicted by our model (coloured in red). The remaining columns show novel views of the predicted 3D pose.

Qualitative results on MPI-INF-3DHP (continue)

Figure B.5: 3D pose predictions on images corresponding to subjects 3,4,5, and 6 from MPI-INF-3DHP dataset. The first and fifth columns show the input image, and the following columns (second and sixth) display the actual 3D pose from the dataset (coloured in green) aligned with the 3D pose predicted by our model (coloured in red). The remaining columns show novel views of the predicted 3D pose.

B.4 Qualitative results on HandDB

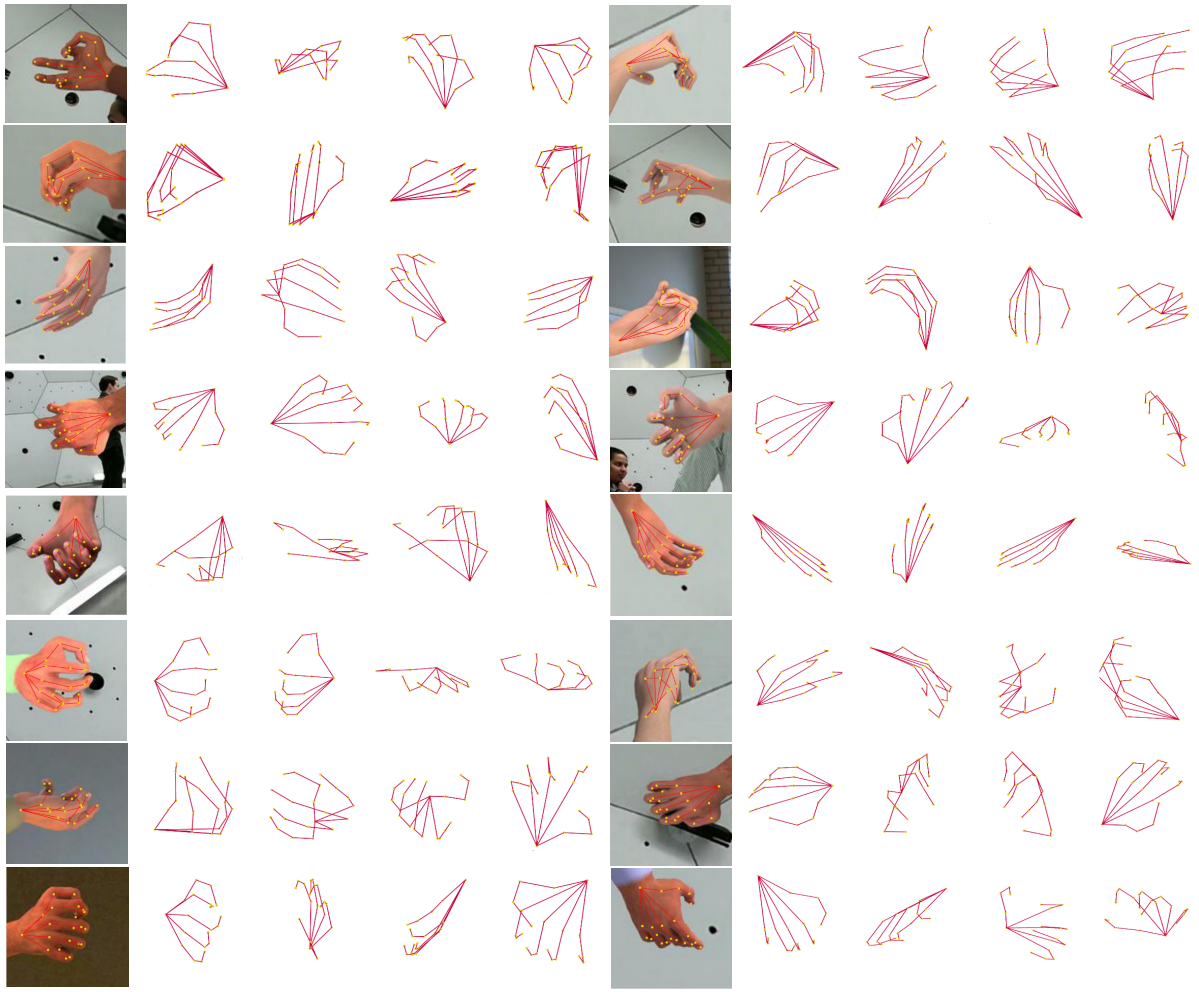


Figure B.6: 3D hand pose predictions on synthetic hand images from HandDB dataset. The first and sixth columns show the input image with its corresponding 2D ground-truth superimposed. The remaining columns show novel views of the predicted 3D hand pose.

B.5 Intermediate representations

Throughout various stages of training our approach, we generate visual representations of the intermediate representations. In particular, we focus on the skeleton images. As can be observed in Figure B.7, the predicted skeleton image improves as the training progresses. It starts showing some blurred and mostly disconnected lines in early iterations. The final iterations display a more aligned skeleton representation of the person's pose depicted on the input. We also show a skeleton image representation of the 2D projection corresponding to the rotated 3D prediction. The final columns illustrate the samples from the unpaired prior of 2D poses.

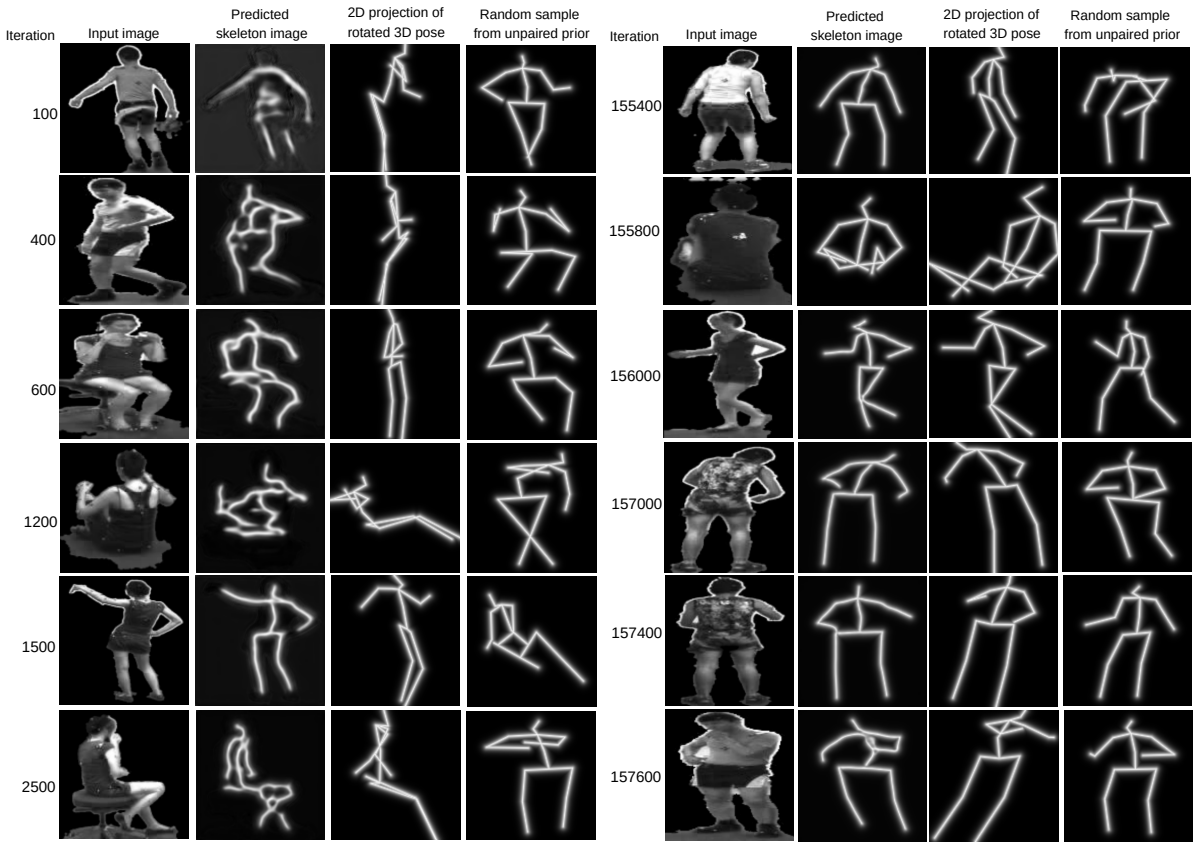


Figure B.7: Intermediate representations during training. We plot the intermediate skeleton image representations from our approach at different stages during training. As can be noticed, the predicted skeleton image is improving with training, showing a more aligned skeleton with the person's pose in the input image. For reference, we also plot the rendered sample from the unpaired prior of 2D poses in the last columns. Due to an issue with the visualisation tool (https://wandb.ai/site), the input image is show on black and white.

We generate and plot the intermediate pose representations as skeleton images using the trained model. Figure B.8 shows the input image with its corresponding skeleton image generated by the trained model. As can be seen, the skeleton mostly aligns with the pose of the person

depicted on the input, which is one of the goals during the training of the model.

Figure B.8: Skeleton images generated with the trained model. The figure shows the input image depicting the subject and its respective skeleton image generated with our trained model.

B.6 Implementation details

Training details: We train the networks Φ, Ω, Λ and D, from scratch according to the loss function (Equation 12) from the main paper. We use the Adam optimiser [203] with learning rate of 2×10^{-4}, and $\beta_1 = 0.5, \beta_2 = 0.999$. Each batch is formed by sampling from the images and randomly sampling from the prior of unpaired 2D poses (which is then transformed to a skeleton image). The batch size is 96. Our model was trained for around 40 hours using one GPU from a NVIDIA DGX-MAX-Q server. The NF is pre-trained in line with [64] as shown on the next section.

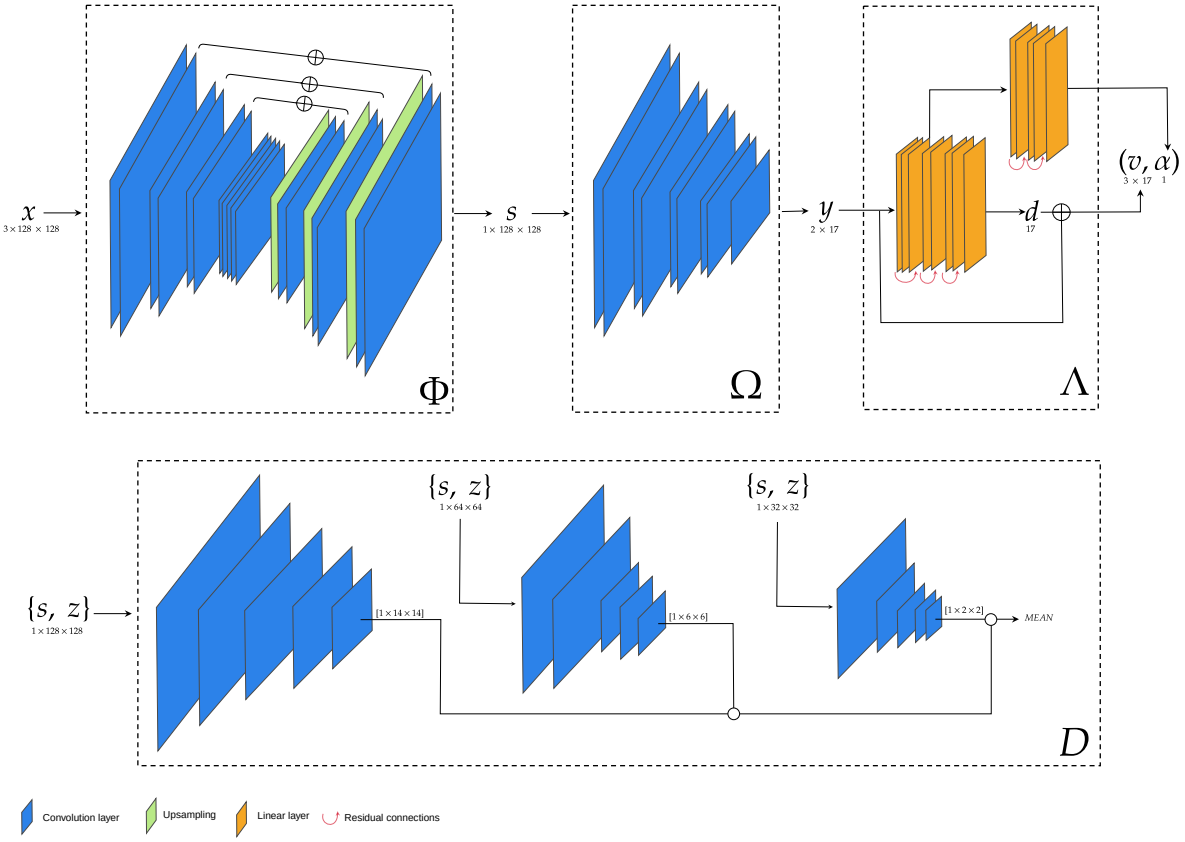


Figure B.9: Pictorial representation of the networks that integrate our model. Blue rectangles represent convolutional layers, and the orange ones the linear layers. Note that to keep the diagrams as simple as possible, we omit some components, such as the size of layers, normalisation layers, and activation functions; we include these elements in Table B.2, Table B.3, Table B.4, and Table B.5 of the next section.

Model components: This section shows the details of the networks used in our model. We include a pictorial representation of all the networks shown in Figure 2 from the main paper. The upper part of Figure B.9 displays the networks needed for the mapping from image x to 3D pose v. The lower part shows the discriminator D needed during training to evaluate the

skeleton images. In particular, Φ and Ω are based on [224, 119], the discriminator D on [121, 119], and the lifting network Λ on [16, 64].

Following [119], with respect to the discriminator D, we use three identical convolutional architectures, inputting different scales of the image: the original image and its downsized versions by $\frac{1}{2}$ and $\frac{1}{4}$, respectively. We take the mean of the patchwise outputs from the three network, as indicated in Figure B.9. The normalising flow network is shown in the following subsection since it requires a more detailed explanation.

Normalising flow

Following [64], we use the network in [236] to represent 4.11. This network consists of consecutive affine coupling blocks like the one shown in Figure B.10. Each coupling block applies a random permutation of the input. In our case, the input $\bar{y}$ is the image in the PCA subspace of the 2D pose $\hat{y}$. After the permutation, it splits the vector into two parts, m_1 and m_2. The first part m_1, is used to predict a scale s and a translation t to deform m_2. In the end, w_1 (or m_1 since it remains unchanged) is concatenated with the deformed m_2 represented as w_2.

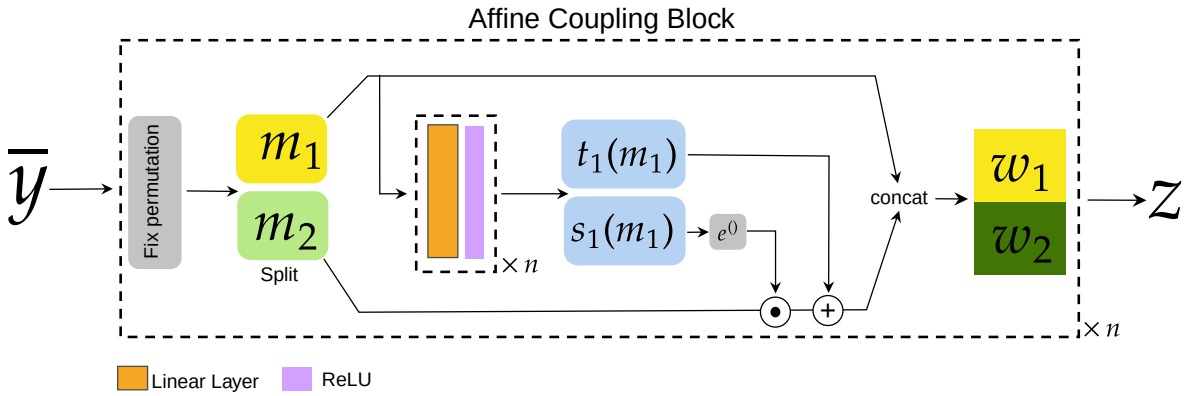


Figure B.10: Affine coupling block. Multiple consecutive coupling blocks integrates the normalising flow. Diagram adapted from the supplemental material of [64].

During the forward pass the scale s and translation t are calculated as a function of m_1, and then used to deform m_2 as follow

$$w_2 = \exp(s(m_1))m_2 + t(m_1) \qquad \& \qquad w_1 = m_1 \tag{B.1}$$

Similarly, the backward part is defined by

$$m_1 = w_1 \qquad \& \qquad m_2 = (w_2 - t(w_1)) \exp(-s(w_1)) \tag{B.2}$$

The determinant of the Jacobian is given by

$$\det\left(\frac{\partial f}{\partial \bar{y}}\right) = \exp\left(\sum_j s(m_1)_j\right). \tag{B.3}$$

Since the Jacobian of f does not need to calculate the Jacobian of the scale s and translation t functions, these could be complex.

Networks structure

Layer	Output Shape	Activation function	Normalisation type
Conv2d	$32 \times 128 \times 128$	ReLU	Batch
Conv2d	$32 \times 128 \times 128$	ReLU	Batch
Conv2d	$64 \times 64 \times 64$	ReLU	Batch
Conv2d	$64 \times 64 \times 64$	ReLU	Batch
Conv2d	$128 \times 32 \times 32$	ReLU	Batch
Conv2d	$128 \times 32 \times 32$	ReLU	Batch
Conv2d	$256 \times 16 \times 16$	ReLU	Batch
Conv2d	$256 \times 16 \times 16$	ReLU	Batch
Conv2d	$256 \times 16 \times 16$	-	-
Conv2d	$256 \times 16 \times 16$	ReLU	Batch
Conv2d	$256 \times 16 \times 16$	ReLU	Batch
Upsampling	$128 \times 32 \times 32$	-	-
Conv2d	$128 \times 32 \times 32$	ReLU	Batch
Conv2d	$128 \times 32 \times 32$	ReLU	Batch
Upsampling	$64 \times 64 \times 64$	-	-
Conv2d	$64 \times 64 \times 64$	ReLU	Batch
Conv2d	$64 \times 64 \times 64$	ReLU	Batch
Upsampling	$32 \times 128 \times 128$	-	-
Conv2d	$32 \times 128 \times 128$	ReLU	Batch
Conv2d	$1 \times 128 \times 128$	-	-
Final output shape: $1 \times 128 \times 128$			

Table B.2: Structure of network Φ.

Layer	Output Shape	Number of parameters	Activation function	Normalisation type
Conv2d	$32 \times 128 \times 128$	1,600	ReLU	Inst.
Conv2d	$32 \times 128 \times 128$	9,248	ReLU	Inst.
Conv2d	$64 \times 64 \times 64$	18,496	ReLU	Inst.
Conv2d	$64 \times 64 \times 64$	36,928	ReLU	Inst.
Conv2d	$128 \times 32 \times 32$	73,856	ReLU	Inst.
Conv2d	$128 \times 32 \times 32$	147,584	ReLU	Inst.
Conv2d	$256 \times 16 \times 16$	295,168	ReLU	Inst.
Conv2d	$256 \times 16 \times 16$	590,080	ReLU	Inst.
Conv2d	$17 \times 16 \times 16$	4,369	None	None
Final output shape: $17 \times 16 \times 16$ **Total params:** 1,177,329				

Table B.3: Structure of network Ω.

Layer	Output Shape	Number of Parameters	Activation function	Normalisation type
Linear	1×1024	35,840	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×17	17,425	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	yReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1	1,025	LReLU	None
Final output shape: $[[1 \times 17], [1 \times 1]]$ **Total params:** 10,550,290				

Table B.4: Structure of network Λ.

Layer	Ouput Shape	Number of Parameters	Activation function	Normalisation type
Conv2d	$64 \times 64 \times 64$	1,088	LReLU	None
Conv2d	$128 \times 32 \times 32$	131,200	LReLU	Inst.
Conv2d	$256 \times 16 \times 16$	524,544	LReLU	Inst.
Conv2d	$512 \times 15 \times 15$	2,097,664	LReLU	Inst.
Conv2d	$1 \times 14 \times 14$	8,193	None	None
Conv2d	$64 \times 32 \times 32$	1,088	LReLU	None
Conv2d	$128 \times 16 \times 16$	131,200	LReLU	Inst.
Conv2d	$256 \times 8 \times 8$	524,544	LReLU	Inst.
Conv2d	$512 \times 7 \times 7$	2,097,664	LReLU	Inst.
Conv2d	$1 \times 6 \times 6$	8,193	None	None
Conv2d	$64 \times 16 \times 16$	1,088	LReLU	None
Conv2d	$128 \times 8 \times 8$	131,200	LReLU	Inst.
Conv2d	$256 \times 4 \times 4$	524,544	LReLU	Inst.
Conv2d	$512 \times 3 \times 3$	2,097,664	LReLU	Inst.
Conv2d	$1 \times 2 \times 2$	8,193	None	None
Final output shape: $[[1 \times 14 \times 14], [1 \times 6 \times 6], [1 \times 2 \times 2]]$ **Total params:** 8,288,067				

Table B.5: Structure of network D.

Appendix C

Chapter 5

Overview

This supplemental material provides extended descriptions of the work done in the corresponding chapter of this thesis. In section C.1, we include more details about the process and resources utilised to collect the data utilised for our experiments. In section C.2 we include visualisations of intermediate representations during and after training the model. Finally, section C.3 includes implementation details.

Note that most of the figures in these sections are of high quality. If something is not completely visible on the current figure size, simply zoom in for better visualisation. Some visualisations also serve as links to videos or other visual resources. This information is indicated in the captions of the respective figures.

Our code will be made accessible shortly after the publication of this thesis (WIP): https://github.com/josesosajs/auto-data-collection.git

Paper (Proceedings): Available soon ...

Pre-print: https://arxiv.org/pdf/2308.03411.pdf

Poster (ICCV 2023): https://tinyurl.com/mpwev5r6

C.1 Dataset details

We utilise a portion of the TigDog dataset [134] to construct our training set. This data is publicly available and can be accessed from here: https://calvin-vision.net/datasets/tigdog/. Note that only the latest version of the dataset contains horse data.

As mentioned in the chapter, we extend the horse data by collecting images from YouTube videos. We manually select a group of public short videos depicting whole-body horses doing different activities (see Table 5.1). We design a pipeline to collect the data automatically illustrate in Figure C.1. It inputs the YouTube video ID and produces images depicting horses and their corresponding segmentation masks. To identify the horses in the videos, we use Detectron2 https://ai.meta.com/tools/detectron2/. Documentation can be accessed here: https://detectron2.readthedocs.io/en/latest/modules/data.html.

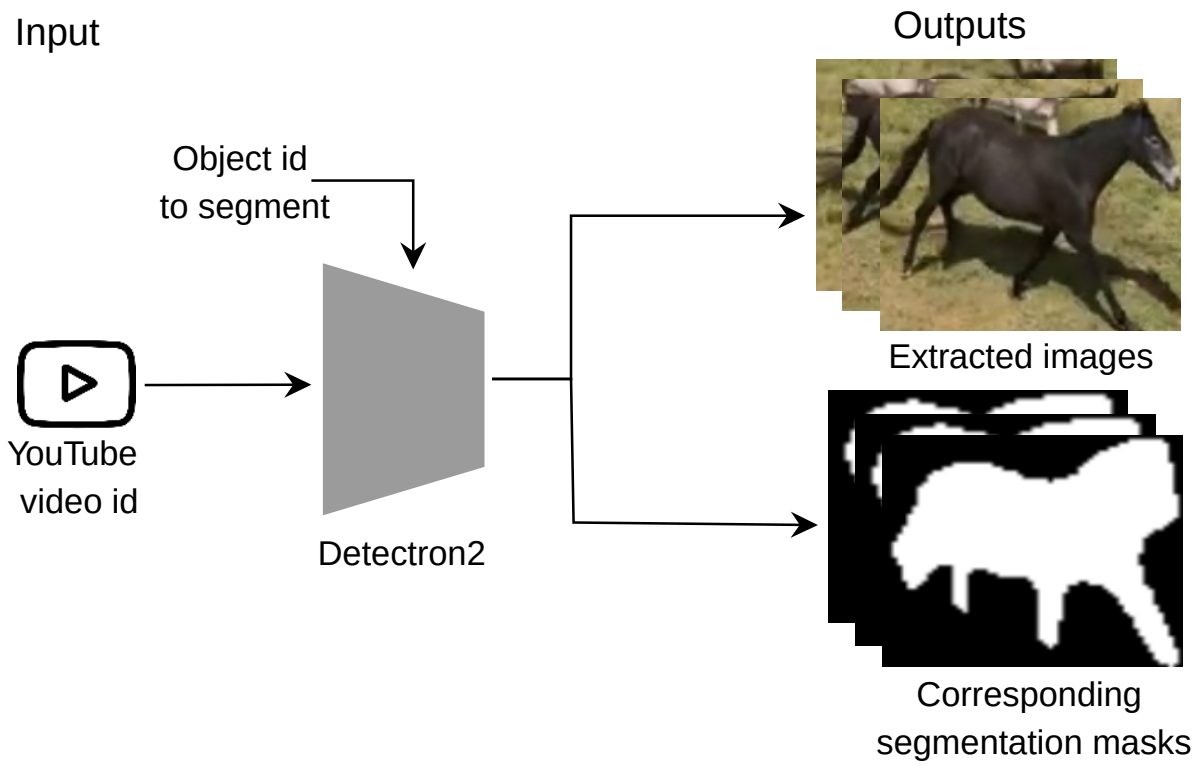


Figure C.1: Diagram summarising the data collection process. Our approach for collecting horse images from videos involves using Detectron2, a pre-trained tool for object segmentation. This helps to identify the images from the videos containing horses and produce the corresponding segmentation masks. Segmenting other animals/objects is possible by changing the object id parameter on the Detectron2 implementation.

We provide our code for implementing the automatic pipeline for collecting the horse images here: https://github.com/josesosajs/auto-data-collection. Due to the need for a GPU

to run Detectron2, we release the Google Colab notebook, allowing for faster and straightforward code use. Connect the notebook with Google Drive for a more uncomplicated experience.

C.2 Intermediate representations

Throughout various stages of training our approach, we generate visual representations of the intermediate representations, as with the human pose estimator. In particular, we focus on the skeleton images. As can be observed in Figure C.2, the predicted skeleton image improves as the training progresses. It starts showing some blurred and mostly disconnected lines in early iterations. The final iterations display a more aligned skeleton representation of the horse's pose depicted on the input. We also show a skeleton image representation of the 2D projection corresponding to the rotated 3D prediction. The final columns illustrate the samples from the prior of synthetic 2D poses.

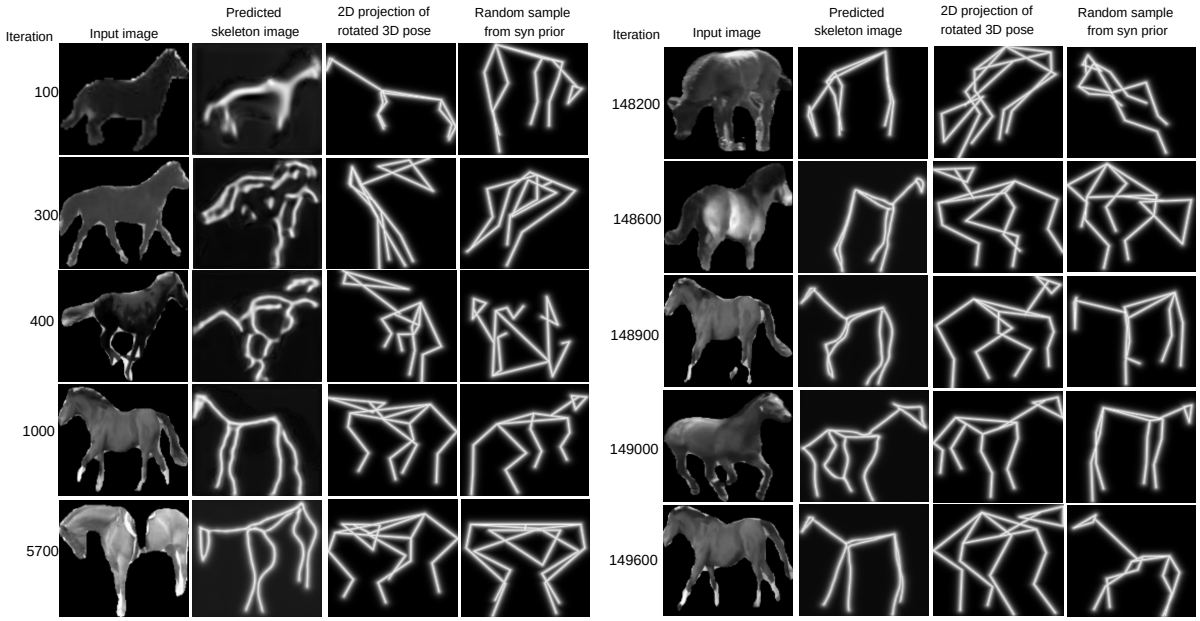


Figure C.2: Intermediate representations during training. We plot the intermediate skeleton image representations from our approach at different stages during training. As can be noticed, the predicted skeleton image is improving with training, showing a more aligned skeleton with the horse's pose in the input image. For reference, we also plot the rendered sample from the prior of synthetic 2D poses in the last columns. Due to an issue with the visualisation tool (`https://wandb.ai/site`), the input image is show on black and white.

We generate and plot the intermediate pose representations as skeleton images using the trained model. Figure C.3 shows the input image with its corresponding skeleton image generated by the trained model. As can be seen, the skeleton mostly aligns with the pose of the horse depicted on the input, which is one of the goals during the training of the model.

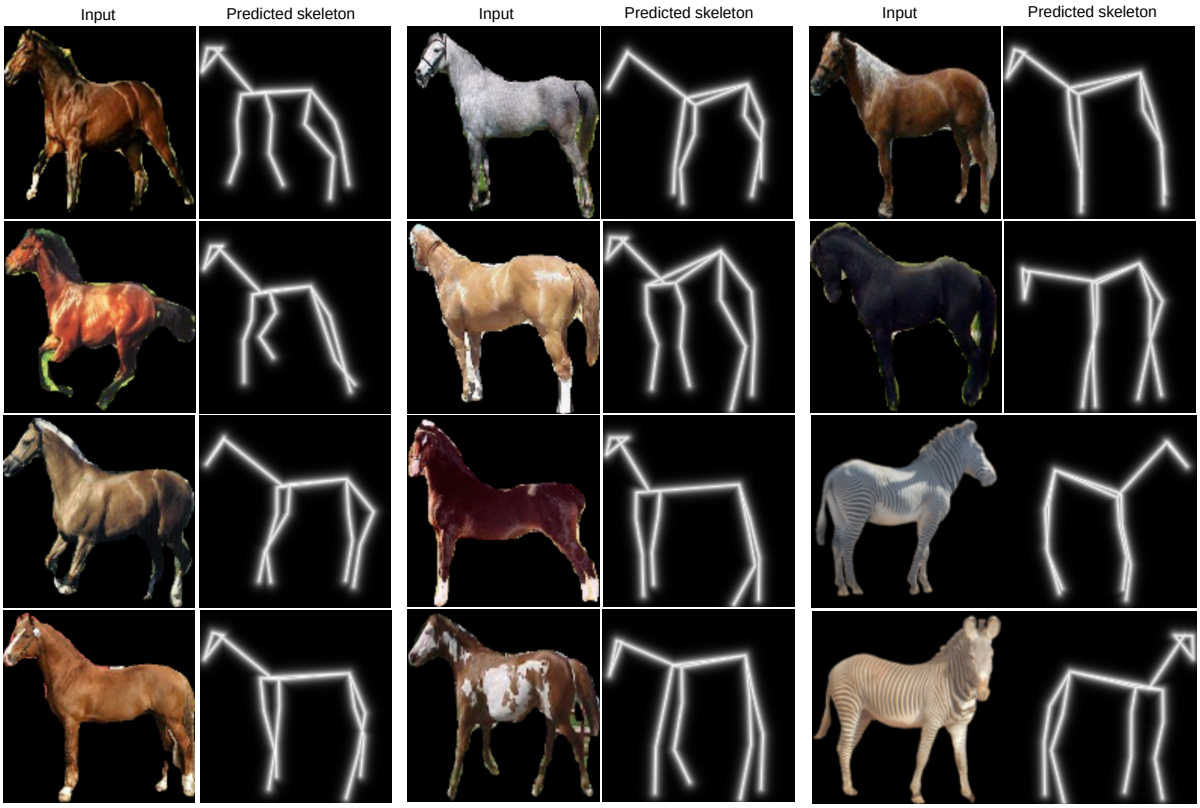


Figure C.3: Skeleton images generated with the trained model. The figure shows the input image depicting the horse and zebras, and its respective skeleton image generated with our trained model.

C.3 Implementation details

The structures of networks Φ, Ω, and D are the same as described in Table B.2, Table B.3, and Table B.5 from Appendix B respectively. The network Λ is a bit different, its structure is illustrate on Table C.1.

Layer	Output Shape	Number of Parameters	Activation function	Normalisation type
Linear	1×1024	35,840	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×1024	1,049,600	LReLU	None
Linear	1×17	17,425	LReLU	None
Final output shape: $[1 \times 17]$, **Total params:** 6,350,865				

Table C.1: Structure of network Λ for horse pose estimation.