A Unified Approach to Pose Estimation in Elephants and Other Quadrupeds using Noisy Labels

Karen Panetta[1], Obafemi Jinadu[1]*, Jamie Heller[1], Srijith Rajeev[1], Kali Pereira[2], Sos Agaian[3]

[1] School of Engineering, Tufts University, Medford, Massachusetts, 02155, United States
[2] Cummings School of Veterinary Medicine, Tufts University, North Grafton, Massachusetts, 01536, United States
[3] Department of Computer Science, College of Staten Island, The City University of New York, New York, 10314, United States
Correspondence and requests for materials should be addressed to O.J. (email: obafemi.jinadu@tufts.edu)

Summary

Pose estimation predicts anatomical landmarks in humans and animals from monocular images or videos. Animal Pose Estimation is crucial for monitoring locomotion, behavior, and activity recognition, playing a key role in wildlife conservation. Single species pose estimation studies capture features unique to the species but generalize sub-optimally, while multi-species studies provide broader generalization by assuming fixed keypoints for all quadrupeds, this oversimplification fails to capture unique anatomical traits in animals such as elephants. To harness the strengths of single-species and multi-species pose estimation, we present QuadPose, a framework that standardizes skeletal structures across datasets and improves generalizability through consistency-dependent pseudo-labelling. Additionally, JumboPose, a manually annotated dataset of 2,078 African elephant images with 33 keypoints tailored to their unique morphology is introduced. Extensive evaluations demonstrate the effectiveness of QuadPose for animal pose estimation. This work establishes a foundation for standardized, cross-species pose estimation, advancing applications in wildlife conservation, and veterinary research.

Introduction

The field of computer vision has tremendously benefited from advances in machine learning. State-of-the-art methods for object detection[1,2,3,4], recognition[5], segmentation[6,7,8,9,10] and pose estimation[11,12,13,14,15,16,17] adopt deep neural network frameworks, such as convolutional neural networks[18,5] and transformer-based architectures[19,20] for feature extraction. Pose estimation involves identifying and localizing anatomical keypoints such as elbows, wrists[11] within an image or video. Pose estimation methods are generally categorized into top-down[21] and bottom-up[22,23] approaches. The top-down approach involves using a detector model[1,2,24,25,26] to isolate regions of interest in the image by obtaining a set of coordinate locations for each detected object and subsequently performs single-person pose estimation. The bottom-up approach does not require a separate detector; instead, it directly predicts keypoints all at once, followed by an association step that groups them into full poses for each individual[27,28,29].

Practical applications that require pose estimation as a critical step include behavior understanding, human-object interaction, and activity recognition. The availability of large-scale datasets, such as COCO[30], OCHuman[31] and MPII[32], has been a driving force behind advancements in human pose estimation (HPE). Despite advances in HPE, progress in animal pose estimation (APE) remains limited. A major challenge is the lack of large-scale, labeled datasets that comprehensively represent diverse animal species. Without standardized datasets and robust models, APE systems struggle to accurately track animal movement, monitor health conditions, and support conservation efforts in real-world settings. To address this, several datasets have been created for specific quadrupeds, with annotations tailored to their anatomical structure. These

include datasets for tigers[33], horses[34], macaque[35], and zebra[36]. Multi-species datasets, such as Animal-Pose[37] and AP-10K[38] with five and fifty-four animal species respectively, aim to capture broader quadruped similarities. These datasets have gained traction in wildlife conservation research. However, annotation styles and skeletal structures across these APE datasets vary. These differences include the number and placement of keypoints, inconsistent naming conventions, and varying anatomical definitions. These discrepancies reduce knowledge transferability in large-scale cross dataset learning. Despite these inconsistences, the shared anatomical similarities among quadrupeds present an opportunity to standardize and consolidate images and labels across most of these datasets into a unified framework, leveraging common features across species. These anatomical similarities can be extended to out-of-domain classes to improve the model's generalizability to unseen animal species.

Another challenge worth addressing is the failure of existing datatsets[37,38,35,33,36,34] to capture critical keypoints in certain quadrupeds, such as elephants, which have unique features not shared by other species, including trunks, large ears, and tusks. This underscores the need for a framework capable of reliably estimating poses for quadruped species with distinct morphologies, such as elephants, while also generalizing effectively to other quadrupeds.

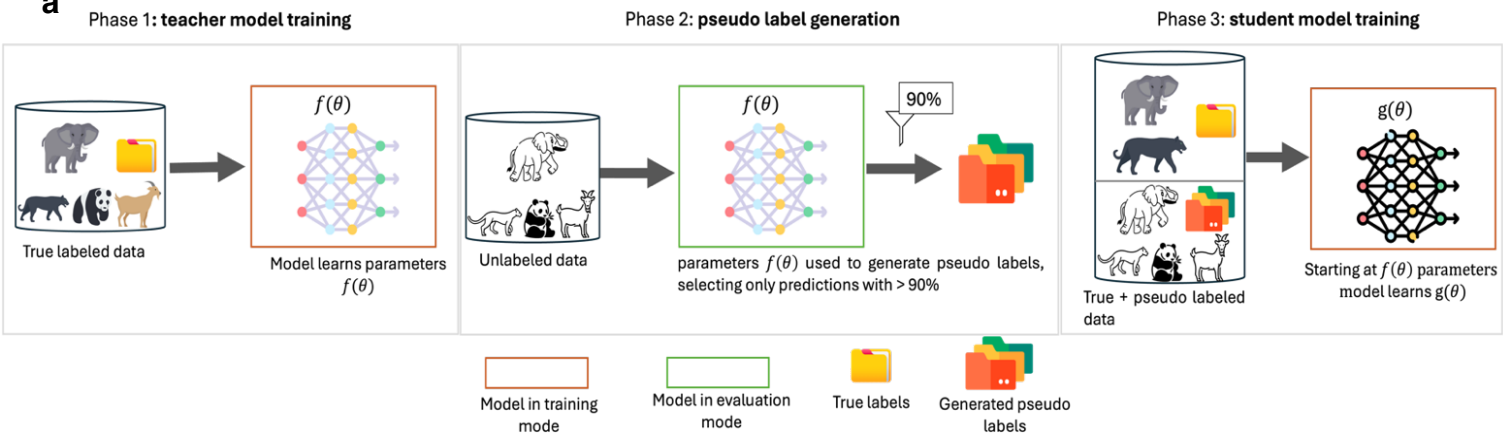
This work, introduces QuadPose, a unified framework for animal pose estimation that enables accurate, species-specific predictions for elephants while improving generalization across diverse quadrupeds. To address the lack of datasets capturing the unique morphology of elephants, we propose JumboPose, a large-scale dataset dedicated to elephant pose estimation, featuring 2,078 manually annotated images with 33 anatomically relevant keypoints.

QuadPose formulates APE as a multi-task learning problem, standardizing pose estimation into two data representations: one tailored specifically for African elephants and another encompassing all other quadrupeds, implemented as a dual-head prediction network. Our framework is developed based on top-down state-of-the-art architectures, including HRNet[18] with polarized self-attention[39], ViTPose[40], and TransPose[41], and incorporates a binary classifier that dynamically routes input data to the appropriate prediction head. Additionally, we introduce a pseudo-labeling strategy that leverages shared anatomical features to enhance generalization to unseen animal species. Extensive evaluations demonstrate that QuadPose achieves state-of-the-art performance, with mAP scores of 81.5, 85.7, and 94.3 on Animal-Pose, AP-10K, and JumboPose, respectively. By standardizing skeletal structures and leveraging multi-task learning, QuadPose not only enhances species-specific accuracy but also improves cross-species generalization. This framework establishes a new benchmark for scalable and robust animal pose estimation, paving the way for broader applications in wildlife conservation, behavioral analysis, and veterinary science. In particular, these capabilities hold significant potential for monitoring free-roaming wildlife populations, enabling automated censusing by age-sex class and facilitating the remote detection of sick or injured individuals. Such advancements are crucial for improving conservation efforts and ensuring timely interventions in challenging field environments.

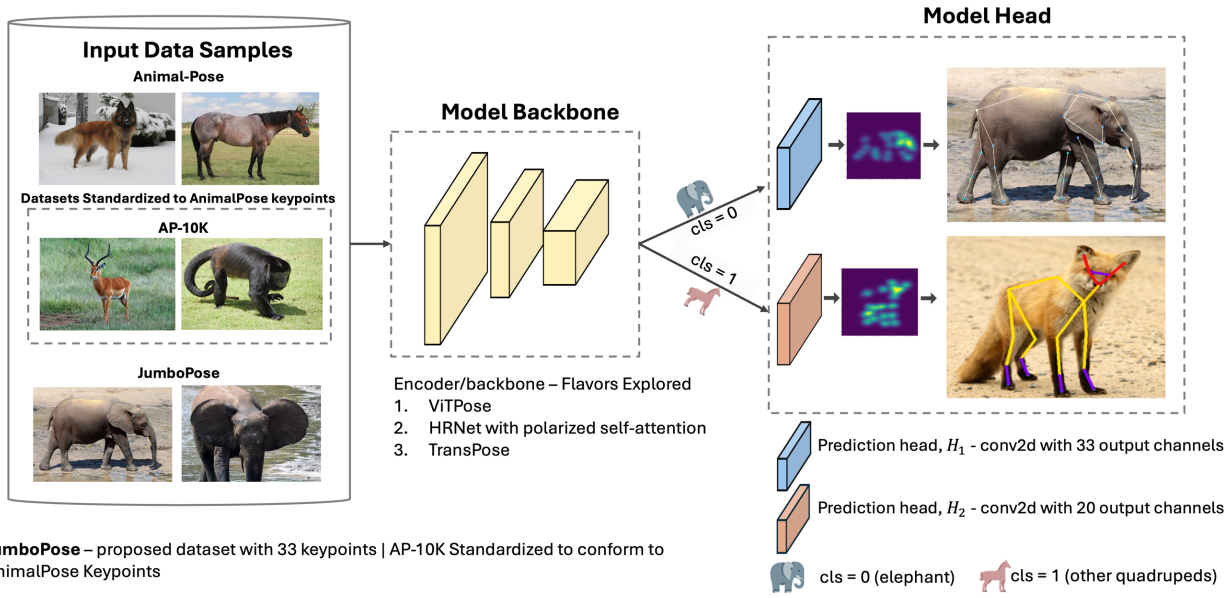
Results

QuadPose employs a three-phase training strategy designed to improve pose estimation across quadrupeds. First, a data standardization step maps input data into two categories: elephants and other quadrupeds. Phase 1 uses a curriculum training approach on manually annotated data. Phase 2 generates high-confidence pseudo-labels based on the model's initial predictions. Finally, Phase 3 applies a typical concurrent training approach, integrating both manually annotated and pseudo-labeled data for refinement (Fig. 1a). Below, we detail the datasets used to train QuadPose.

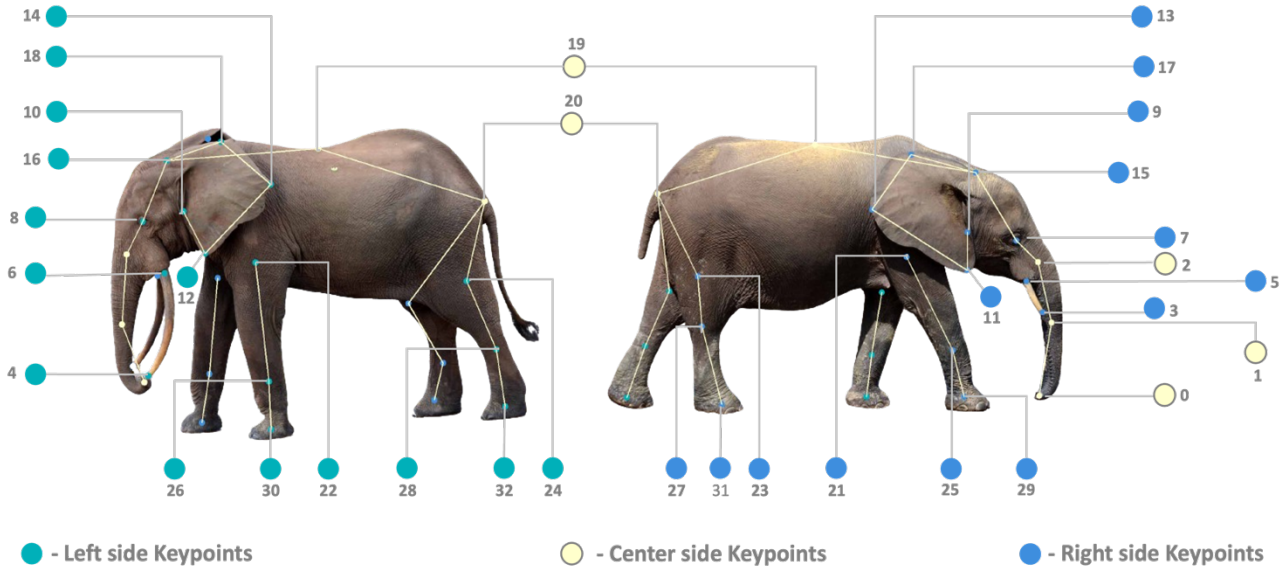
a


Phase 1: teacher model training
Phase 2: pseudo label generation
Phase 3: student model training
True labeled data
$f(\theta)$
Model learns parameters $f(\theta)$
Unlabeled data
$f(\theta)$
90%
parameters $f(\theta)$ used to generate pseudo labels, selecting only predictions with > 90%
True + pseudo labeled data
$g(\theta)$
Starting at $f(\theta)$ parameters model learns $g(\theta)$
Model in training mode
Model in evaluation mode
True labels
Generated pseudo labels

b


Input Data Samples
Animal-Pose
Datasets Standardized to AnimalPose keypoints
AP-10K
JumboPose
Model Backbone
Encoder/backbone – Flavors Explored
1. ViTPose
2. HRNet with polarized self-attention
3. TransPose
cls = 0
cls = 1
Model Head
Prediction head, H_1 - conv2d with 33 output channels
Prediction head, H_2 - conv2d with 20 output channels
cls = 0 (elephant)
cls = 1 (other quadrupeds)
JumboPose – proposed dataset with 33 keypoints | AP-10K Standardized to conform to AnimalPose Keypoints

c


14
18
10
16
8
6
4
12
19
20
26
30
22
28
32
24
27
31
23
21
25
29
13
17
9
15
7
2
5
3
11
1
0
- Left side Keypoints
- Center side Keypoints
- Right side Keypoints

Keypoint	Definition	Keypoint	Definition	Keypoint	Definition	Keypoint	Definition
0	Bottom trunk	9	Right bottom ear	18	Top left tip ear	27	Right back knee
1	Mid trunk	10	Left bottom ear	19	Withers	28	Left back knee
2	Top trunk	11	Right bottom tip ear	20	Tail	29	Right front foot
3	Bottom right tusk	12	Left bottom tip ear	21	Right front elbow	30	Left front foot
4	Bottom left tusk	13	Right side tip ear	22	Left front elbow	31	Right back foot
5	Top right tusk	14	Left side tip ear	23	Right back elbow	32	Left back foot
6	Top left tusk	15	Top right ear	24	Left back elbow		
7	Right eye	16	Top left ear	25	Right front knee		
8	Left eye	17	Top right tip ear	26	Left front knee		

Figure 1 Overview of the QuadPose framework. **a.** Schematic representation of the three-phase training pipeline. The teacher model is first trained using manually labeled data (Phase 1), then the learned parameters are used to generate high confidence pseudo labels from unlabeled data (Phase 2), and finally, a student model is trained using both manually and pseudo labeled data. **b.** Architecture of the QuadPose model, which is based on ViTPose[40], HRNet[39] and, TransPose[41] model backbones. The model head is modified into dual prediction heads, first head is a convolutional layer that outputs 33 joints and second head is a convolutional layer that outputs 20 joints. **c.** Proposed Elephant Dataset, JumboPose, manually labeled dataset of 2,078 African elephants with 33 keypoints along with keys for the labelling scheme. Colored text highlight auxiliary keypoints unique to elephants.

Training Data

The QuadPose framework (Fig. 1b), standardizes data into two formats: one for elephants (JumboPose, Fig. 1c) and another for other quadrupeds. JumboPose, adapted from the ELPephants dataset [42] was originally developed for elephant Re-identification, while the other quadruped dataset integrates annotations from over 50 species across multiple sources. In phase 1, the teacher model is trained using 18,212 images from Animal-Pose[37], AP-10K[38], and JumboPose. Phase 2 expands training samples by generating pseudo-labels from 53,600 unlabeled images across eleven datasets (see Methods). Finally, Phase 3 utilizes over 71,000 images from the previous phases for the student model training. Model weights from the teacher (phase 1) and student (phase 3) are publicly released along with the annotations for JumboPose and can be found at https://github.com/QuadPose.

QuadPose method

The QuadPose is a robust method that handles diverse quadruped pose estimation datasets as two standardized datasets. It leverages top-down pose estimation [40,43, 39,41] base architectures modified with dual prediction heads that conform to the defined unified standards. Top-down methods are chosen for their ability to first detect and classify objects before pose estimation, allowing for a binary classification step that dynamically routes input data to the appropriate prediction head (Fig. 1b). In Phase 1, QuadPose follows a curriculum learning strategy, where training data is introduced progressively to improve model adaptability. By Phase 3, the model undergoes concurrent training, where both manually labeled and pseudo-labeled datasets are trained simultaneously to refine predictions across species.

Benchmarks

To evaluate the effectiveness of the QuadPose framework, we conducted experiments on three animal pose estimation datasets: Animal-Pose, AP-10K, and JumboPose. We compared three training strategies across multiple state-of-the-art architectures modified with dual prediction heads. The training strategies evaluated include 1) the Conventional training strategy, where all datasets are trained simultaneously, serving as a baseline, 2) Progressive training, where data is introduced incrementally using a curriculum learning approach, and 3) Progressive + SSL (semi-

supervised learning), which improves model generalization by using pseudo-labels. Fig. 2 and Table 1 present the mean Average Precision (mAP) scores for each method and model configuration. The global average mAP (Avg. mAP) and relative improvements (Δ Avg. mAP) relative to the conventional baseline method are also reported.

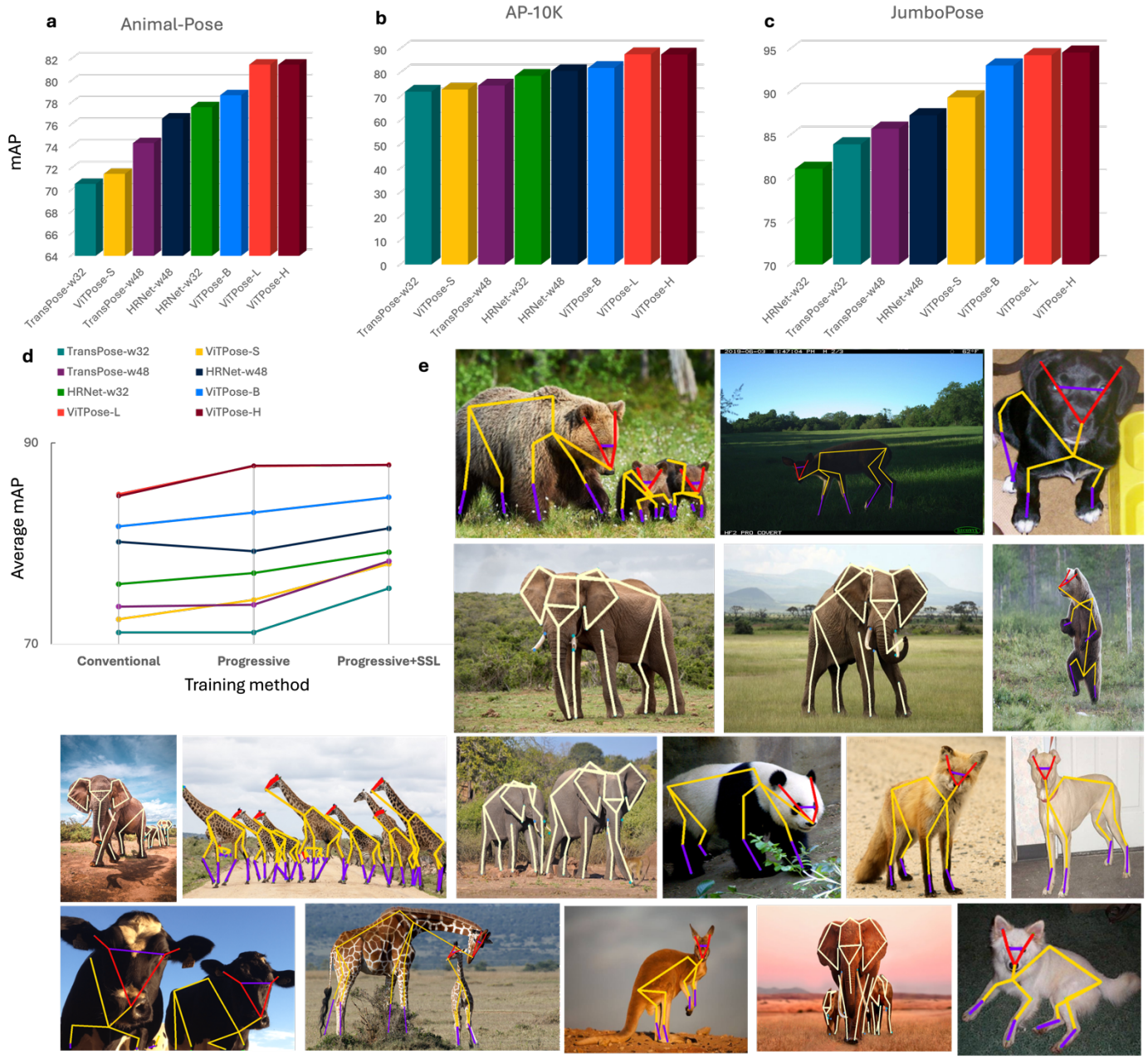


Figure 2. Benchmarking performance of QuadPose across datasets and training strategies. (**a-c)** Mean Average Precision (mAP) scores for QuadPose on Animal-Pose, AP-10K, and JumboPose validation datasets, evaluated across different architectures on the progressive + SSL training approach. The ViTPose family of models achieves the highest mAP across all datasets. **d.** Comparative performance of the three training strategies: Conventional baseline training, Progressive training, and Progressive + semi-supervised learning (SSL) using pseudo-labels **e.** Qualitative results of the multi-task pose estimation problem, showing keypoint predictions for elephants and other quadrupeds, the images used for qualitative analysis were obtained from [38,30,44] and the Cummings School of Veterinary Medicine.

Table 1 results: Performance comparison of QuadPose across different architectures and training strategies

		Dataset (mAP)↑				
Training Method	**Architecture**	**Animal-Pose**	**AP-10K**	**JumboPose**	**Avg. mAP**	**Δ Avg. mAP**
Conventional	HRNet-w48	73.90	74.8	**91.8**	80.17	-
Progressive	HRNet-w48	70.66	77.04	90.04	79.23	-0.94▼
Progressive + SSL	HRNet-w48	**76.57**	**80.71**	87.32	**81.5**	1.33 ▲
Conventional	HRNet-w32	68.49	69.43	**90.0**	75.97	-
Progressive	HRNet-w32	70.62	76.39	84.22	77.07	1.10 ▲
Progressive + SSL	HRNet-w32	**77.60**	**78.70**	81.11	**79.14**	3.17 ▲
Conventional	ViTPose-S	64.10	67.7	85.6	72.47	-
Progressive	ViTPose-S	66.50	68	88.7	74.40	1.93 ▲
Progressive + SSL	ViTPose-S	**71.50**	**73.00**	**89.40**	**77.97**	5.50 ▲
Conventional	ViTPose-B	76.40	78.5	90.2	81.70	-
Progressive	ViTPose-B	76.30	80.3	92.6	83.07	1.37 ▲
Progressive + SSL	ViTPose-B	**78.70**	**82.0**	**93.1**	**84.60**	2.90 ▲
Conventional	ViTPose-L	78.30	83.7	92.7	84.90	-
Progressive	ViTPose-L	80.40	**88.2**	**94.5**	87.70	2.80 ▲
Progressive + SSL	ViTPose-L	**81.50**	87.7	94.3	**87.83**	2.93 ▲
Conventional	ViTPose-H	77.90	83.3	93	84.73	-
Progressive	ViTPose-H	80.30	**88.2**	**94.7**	**87.73**	3.00 ▲
Progressive + SSL	ViTPose-H	**81.00**	87.6	94.6	**87.73**	3.00 ▲
Conventional	TransPose-w48	65.90	63.6	**91.7**	73.73	-
Progressive	TransPose-w48	68.12	65.28	86.53	73.92	0.19 ▲
Progressive + SSL	TransPose-w48	**74.32**	**74.66**	85.76	**78.25**	4.52 ▲
Conventional	TransPose-w32	62.65	60.41	**90.41**	71.16	-
Progressive	TransPose-w32	65.22	64.25	84.05	71.16	0.00
Progressive + SSL	TransPose-w32	**70.58**	**72.11**	83.95	**75.55**	4.39 ▲

The mAP scores (%) for Animal-Pose, AP-10K, and JumboPose datasets using different training strategies: Conventional training (baseline), Progressive training, and Progressive + SSL (semi-supervised learning) with pseudo-labels. The global average mAP across all datasets (Avg. mAP) and the relative improvement (Δ Avg. mAP) compared to the conventional baseline are reported. Results are shown for multiple model architectures, including HRNet, ViTPose, and TransPose. The best-performing strategy for each architecture is highlighted in bold.

Conventional Training

This training strategy provides the baseline for this study. Overall, it can be observed that the larger model variants outperform their smaller counterparts across all datasets considered, with the ViTPose-L and ViTPose-H achieving the highest average mAP of 84.9 and 84.73, respectively. In the HRNet models, the HRNet-w48 achieves an average mAP of 80.17, while the HRNet-w32 attains an average mAP of 75.97. Similarly, within the TransPose family of architectures, TransPose-w32 and TransPose-w48 report average mAP scores of 71.16 and 73.73, respectively. This reflects the influence of model size on performance in this training paradigm. These baseline results establish a reference for assessing the performance gains introduced by the progressive and the progressive + SSL training strategies.

Progressive Training

The progressive training strategy aims to improve model generalization by gradually introducing training data in stages. Compared to conventional training, which processes all datasets simultaneously, progressive training significantly improves mAP scores across Animal-Pose and AP-10K, as shown in Table 1 and Fig. 2. In the ViTPose architectures, there is an average mAP improvement of 2.275, with all three datasets reporting consistent gains. The HRNet and TransPose architectures show similar trends, where the progressive training slightly outperforms or is at par with the conventional training. While an upward trend can be observed holistically, it can be seen from Table 1 that JumboPose performance tends to slightly decline in the HRNet and TransPose architectures as it transitions from conventional to progressive training strategies. This can be attributed to a few factors, including the continual learning approach adopted in training these architectures, where Animal-Pose and AP-10K are introduced first and second, for the task of learning representations to predict poses of general quadrupeds. Lastly, JumboPose is introduced for the task of learning representations to predict poses of elephants (see Methods). With this progressive training configuration, the model gradually catches up to learn elephant-specific poses; additionally, the sheer difference in volume between datasets for other quadrupeds and elephants is also a contributing factor (see methods, Table 2).

Progressive + Semi Supervised Learning (SSL) Training

Here a student model integrates pseudo-labels generated by a progressively trained teacher model, achieves the highest performance gains across all architectures. Notably, some models reach an increase of up to 5.5 in average mAP, demonstrating the effectiveness of this strategy.

For the ViTPose models, the ViTPose-S flavor sees the greatest performance boost from the progressive + SSL, with a 5.5 average mAP improvement over the baseline. Comparing the progressive (which serves as the teacher model) and the progressive + SSL *(*which serves as the student model) strategies, a 3.57 average mAP gain is recorded. For ViTPose-B variant, a 2.90 average mAP increase is reported between the conventional baseline and progressive + SSL approaches, while a 1.56 average mAP improvement is observed between the progressive and progressive + SSL training approaches. For the larger variants, ViTPose-L and ViTPose-H, the gaps between the conventional and progressive + SSL training are 2.93 and 3.00 average mAP, respectively in favor of the progressive + SSL approach. However, analyzing the progressive (teacher model) and progressive + SSL, the performance gains while present begin to plateau, with marginal improvements of 0.18 on ViTPose-L and no improvement on the ViTPose-H. This suggests that while the Progressive + SSL is highly effective for smaller ViTPose architectures, its impact diminishes as model size increases, possibly due to the larger models already capturing rich feature representations during progressive training.

Similarly, the HRNet-based models enjoy a performance jump with the progressive + SSL approach. For HRNet-w48, which achieves an average mAP of 81.50, there is a 1.33 average mAP increase over the baseline. For HRNet-w32 with an average mAP of 79.14, there is a 3.17 average mAP increase over the baseline. The TransPose architectures also, see a significant performance boost with the progressive + SSL approach, whereas in the TrasnPose-w48, with an average mAP of 78.25, there is an average mAP increase of 4.52 when compared to the baseline. For the TransPose-w32 of average mAP 75.55, there is a 4.39 mAP gap over the baseline.

Discussion

This paper introduces QuadPose, a unified standard for pose estimation in mammals and a straightforward yet efficient framework for effectively utilizing noisy labels to improve a model's generalizability to unseen animal breeds and species. By leveraging large amounts of unlabeled

data to generate pseudo-labels, the student model learns better feature representations for localizing joints. Results show that this semi-supervised framework leads to an average performance boost in mAP of about 3.5 across various state-of-the-art pose estimation models. Furthermore, this work conducts a detailed performance evaluation of standard convolutional neural networks and transformer-based networks for animal pose estimation. Results show that ViTPose-based architectures perform best, as they demonstrate superior generalization capabilities and robustness against catastrophic forgetting. Finally, this study presents JumboPose, a large-scale dataset for elephant pose estimation and landmark localization.

Despite its strong performance improvements, QuadPose has certain limitations. One of its primary challenges is handling complex and crowded scenes. Top-down models struggle in highly occluded and densely populated scenarios typically due to detector performance where it might miss objects that are heavily occluded or merge multiple objects of interest into a single bounding box, in such detection failure cases there is no recourse to recovery [37]. Addressing this challenge will require improved multi-instance pose tracking techniques and enhanced keypoint association strategies. Additionally, since QuadPose relies on a pretrained object detector, classification errors of the detector can further impact performance. Misclassification may lead to incorrect routing. For example, if an elephant is wrongly detected as another animal, it will be routed to the general quadruped prediction head instead of the elephant-specific head, and vice versa. This misclassification introduces significant errors in keypoint localization and affects model performance. Future improvements may involve integrating an uncertainty-aware detection system or implementing self-correcting mechanisms to mitigate routing errors. Generalization to Asian elephants is another limitation. The model is trained primarily on African elephants, which have larger ears compared to their Asian counterparts. As a result, the model struggles to accurately predict features of Asian elephants due to their distinct morphological differences. Expanding the JumboPose to include Asian elephants will be crucial for improving the model's adaptability to different elephant species. These limitations are shown in Fig. 3.

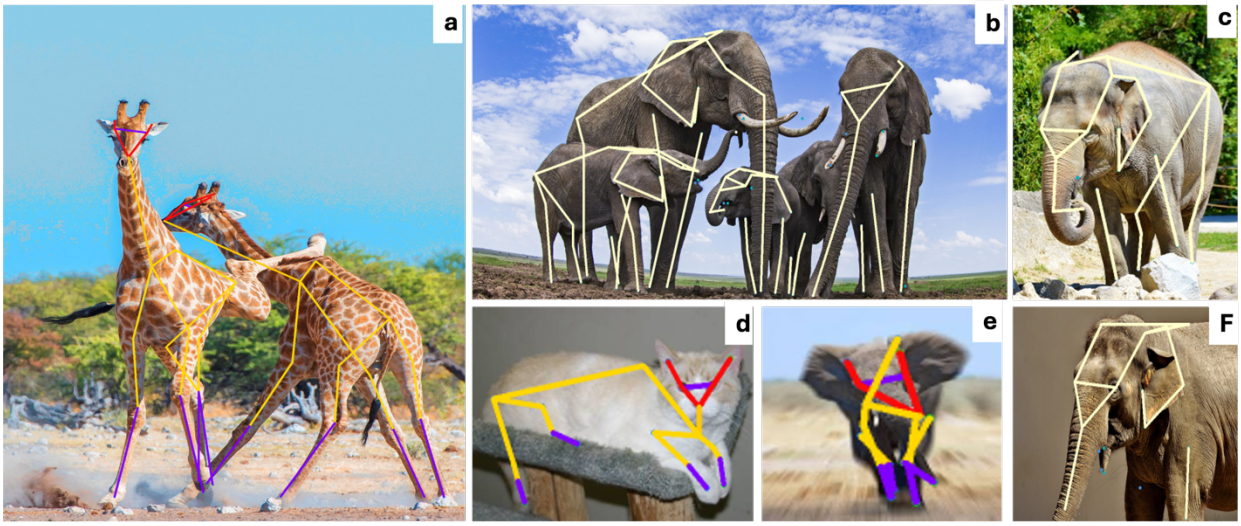


Figure 3. Examples of suboptimal results due to model limitations **a**., **b**., **d.** show the complex, crowded, and occluded poses. **c**. and **f.** illustrate the output of the current model on Asian elephants with smaller ears than their African counterparts **e.** error caused by the detector model misclassifying an elephant, leading to it being incorrectly routed to the general quadruped prediction head.

Future research will focus on mitigating these challenges by reducing the framework's dependence on detector classification performance to minimize misrouting errors. Additionally, expanding the dataset to include Asian elephants will help improve species-specific generalization. Furthermore, developing models more robust to handling extremely crowded scenes will ensure better performance in real-world multi-animal environment where these animals often move in groups. By addressing these limitations, QuadPose can further enhance its applicability in wildlife conservation, behavioral analysis, and veterinary science, making it a more reliable solution for scalable and adaptable animal pose estimation.

Methods

Datasets

Three animal pose estimation datasets were used for phase 1 training and model performance evaluation: Animal-Pose[37], AP-10K[44], and JumboPose. Additionally, eleven supplementary datasets, which do not contain pose annotations, were used for pseudo-label generation. The following sections provide further details on these datasets.

Animal-Pose Dataset

Cao et al.[37] proposed the Animal-pose dataset for developing cross-domain adaptation models for animal pose estimation. The dataset contains 6,117 annotated instances of cats, dogs, sheep, horses, and cows, with 20 keypoints defining their anatomical structures. For full details, refer to Cao et al.[37].

AP-10K Dataset

Yu et al.[45] curated the AP-10K, a large-scale manually annotated dataset for animal pose estimation. It consists of 10,015 images and over 13,000 labeled instances with 23 animal families and 54 species, making it the largest dataset for pose estimation in mammals. The keypoint annotation convention follows a structure similar to human pose estimation, with17 keypoints representing anatomical features of the animals. For full details, refer to Yu et al.[45].

JumboPose Dataset

Elephants, particularly, African elephants possess distinctive features, such as large ears, trunks, and tusks, which differentiate them from most other quadrupeds. These features provide rich additional information that can enhance behavioral understanding. However, current annotation styles in state-of-the-art animal pose datasets do not capture these key anatomical distinctions.
To address this, we introduce JumboPose, a large-scale dataset specifically designed for elephant pose estimation. JumboPose contains over 2,000 labeled images and 6,617 unlabeled images of elephants. It consists of 33 keypoints with 20 keypoints defining the anatomical structure of the elephant similar to other quadrupeds in Animal-Pose[37], and an additional 13 keypoints that capture the nuanced features specific to elephants. The elephant images annotated to create JumboPose was derived from the ElPephants dataset[42] originally designed for elephant re-identification. Fig. 1c illustrates the annotation style used in JumboPose, while Fig. 4 highlights its motivation and structure.

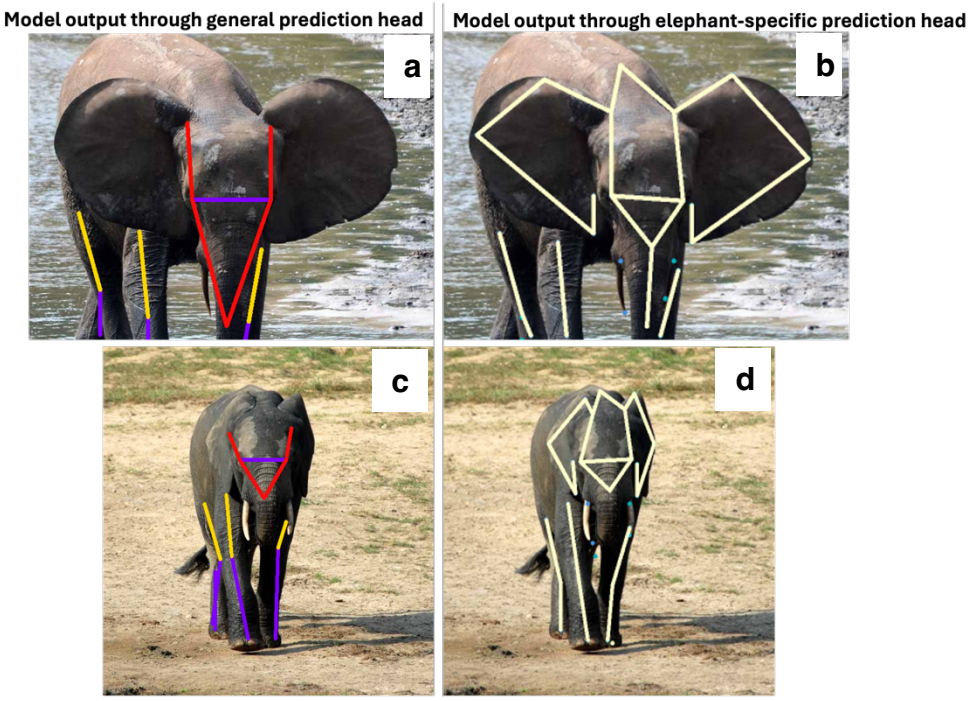


Figure 4. Comparison of model predictions using general quadruped vs. elephant-specific prediction head. **(a., c.)** Model outputs from the general quadruped prediction head, which fails to capture elephant-specific features such as trunks and ears. **(b., d.)** Model outputs from the elephant-specific prediction head, which correctly identifies distinctive anatomical features. These results highlight the importance of JumboPose in improving elephant pose estimation.

Unlabeled Data Curation

The unlabeled data used to generate pseudo labels were sourced from the following datasets; Macaque[35] ,Tiger[33] ,MS-COCO[30], African Wildlife[46], Animal Image Dataset[47], Animals_5 [30], Animals-10[49], Endangered Animals[50], IUCN Animals Dataset[51] , Wild Cats [52], and Asian vs African Elephants[53]. Since MS-COCO is not explicitly an animal dataset, unlabeled elephant images were extracted by running it through an object detection algorithm (YOLOv8[2]).

Table 2: Dataset summarization.

Dataset	Species	Number of keypoints	Number of images
Manually Labeled Data			
Animal-Pose Dataset [37]	5	20	6,117
Ap-10K [45]	54	17	10,015
JumboPose (ours)	1	33	2,078
Unlabeled Data for Pseudo-label Generation			
Tiger [33]	1	-	4,124
Macaque [35]	1	-	13,085
MS COCO[30] (elephants)	-	-	2,202
African Wildlife [46]	4	-	1,504
Animal Image Dataset [47]	3	-	3,000
Animals_5 [48]	10	-	5,233
Animals-10 [49]	6	-	16,148
Endangered Animals [50]	4	-	800
IUCN Animals Dataset [51]	4	-	2,327
Wild Cats [52]	5	-	3,080

Asian vs African Elephants [53]	2	-	2,123
Total Pseudo labels - Elephants			6,617
Total Pseudo-labels - Other Quadrupeds			47,007
Total			71,834

Provides an overview of the datasets utilized for teacher model training (Phase 1) and pseudo-label generation (Phase 3). Three datasets (Animal-Pose, AP-10K, and JumboPose) were used for Phase 1 training, comprising a total of 18,212 manually labeled images. For Phase 3, pseudo-labels were generated from the listed unlabeled datasets, contributing to a total of 71,834 images used in training.
Note: The MS COCO dataset contains 2,202 elephant images, extracted specifically for this study.

Dataset Standardization

The data is standardized into two formats based on quadruped type: elephants and other quadrupeds. The elephant type is straightforward, as JumboPose is the only dataset explicitly designed for elephant pose estimation, to the best of our knowledge. For other quadrupeds, differences in the number of keypoints, labeling style, and keypoint location across datasets necessitate a standardized format for model training. The anatomical structure proposed by Cao et al.[37], is adopted as it accounts for an additional number of useful keypoints like the ears and withers. Keypoints that are not annotated in certain datasets are assigned a "missing" tag and left unannotated to maintain consistency across datasets. Fig. 5 shows how the AP-10K annotation style is remapped accordingly.

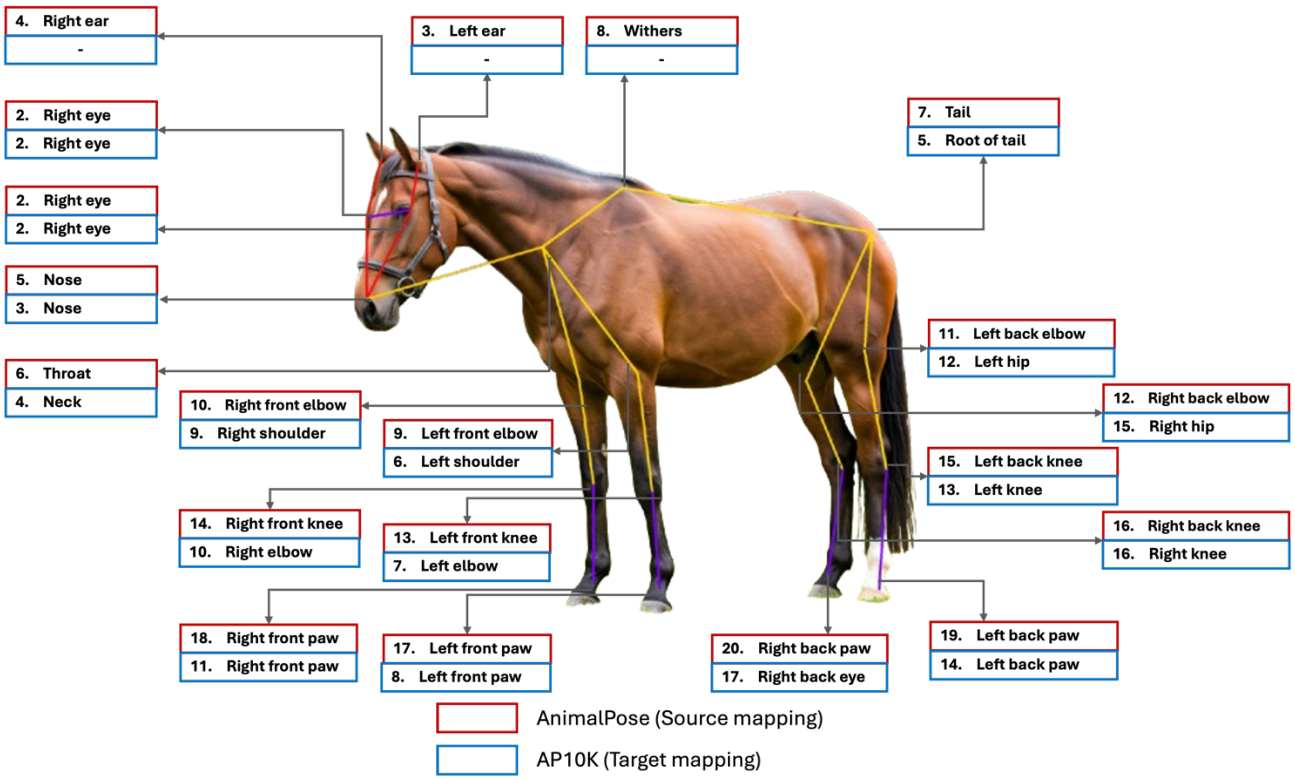


Figure 5. This figure illustrates how AP-10K keypoints are remapped and standardized to align with the Animal-Pose anatomical structure.

QuadPose Training

The proposed model is tasked with two objectives: 1) elephant pose estimation, routed to prediction head, H_1 and 2) other quadruped pose estimation, routed to prediction head, H_2. The general architectural overview is provided in Fig. 1b. The dual-head architecture gradually learns and retains good representations for both tasks. As shown in Fig. 1a, there are three phases in the training pipeline. Curriculum learning, which is a method used to gradually increase the complexity of data samples in the training process[54] is adopted in training, specifically for task 2. A simple yet efficient curriculum learning strategy is adopted, where data samples are introduced in order of difficulty, progressing from easy examples (examples with well-defined poses with most joints visible) to more challenging ones (examples of animals with occluded or missing joints). This strategy is adopted in training the teacher (f_θ) network.

For the first phase, training is done on three datasets: JumboPose (33 keypoints) for task 1, and Animal pose (20 keypoints) and AP-10k (17 keypoints) for task 2. A curriculum based on the number of annotated keypoints is adopted. This helps determine which datasets are prioritized per task at multiple intervals during training, ensuring the model progressively learns from data distributions that provide the most information before adapting to less informative ones, traversing from the known to the less known. This structured progression improves cross-dataset learning and enhances the model's generalization capabilities across different species. The training strategies empirically derived to give the best performance by model architecture are:

a) ViTPose-based models:
    - Initial training begins with JumboPose and Animal-Pose as it simultaneously provides standard baselines for both tasks.
    - Finally, AP-10K dataset is progressively introduced allowing for better refinement of task 2 representations

b) HRNet and TransPose-based models
    - Initial training starts with Animal-Pose as it provides the standard baseline for task 2.
    - AP-10k is introduced.
    - Finally, JumboPose is introduced in a continual learning manner to learn task 1 representations while retaining task 2's representations.

In the second phase, pseudo-labels are generated from a large set of unlabeled data of unseen animal species using the learned weights of the teacher network f_θ. Each image sample is classified as either 'easy' or 'hard' based on the pose complexity, which is determined by the number of high-confidence keypoints detected in each instance. The underlying hypothesis is that an animal's pose is better defined when its complete anatomical structure is visible. By setting a high confidence threshold, unreliable labels are filtered out, ensuring only accurate labels contribute to training. Due to resource constraints and to prevent the models from over-committing to low-confidence and potentially misleading labels, only 'easy' samples are utilized in this study.

In the third phase, the ground-truth annotated datasets, and the pseudo-labeled data are used to train the student network, $g_\emptyset$. The teacher network distills learned knowledge into the student, improving the model's ability to generalize across diverse species. This knowledge distillation process helps the student network capture robust feature representations and enhances overall pose estimation.

Model Specific Training Configurations

The following shows the specific learning settings employed for each architecture. All models were trained on a single NVIDIA V100 GPU and implemented using the PyTorch deep learning framework[55].

ViTPose model training. The learning rate was reduced at epochs 18 and 150 to improve training stability and performance. Initially, the learning rate reduction was scheduled only at epoch 150, coinciding with the introduction of the AP-10K dataset. However, empirical analysis revealed that an earlier reduction at epoch 18 further stabilizes training and enhances convergence. As shown in Fig. 6, early reduction in learning rate improves training stability. Table 3 summarizes the training settings used for the ViTPose family of models.

Table 3: ViTPose Training Settings

Starting LR	0.0005
LR Scheduler	Multi-step
LR Step Factor	0.1
LR Steps (epochs)	[18,150, 400, 450]
Batch size	32
Optimizer	Adam[56]
Data Arrangement for progressive training Phase 1	
Epochs 0 -149	Animal-Pose + JumboPose
Epochs 150 - 349	Animal-Pose + AP-10K + JumboPose
Data Arrangement for progressive training Phase 3	
Epochs 350 - 500	Animal-Pose + AP-10K + Pseudo_other_quadrupeds + JumboPose + Pseudo_elephants

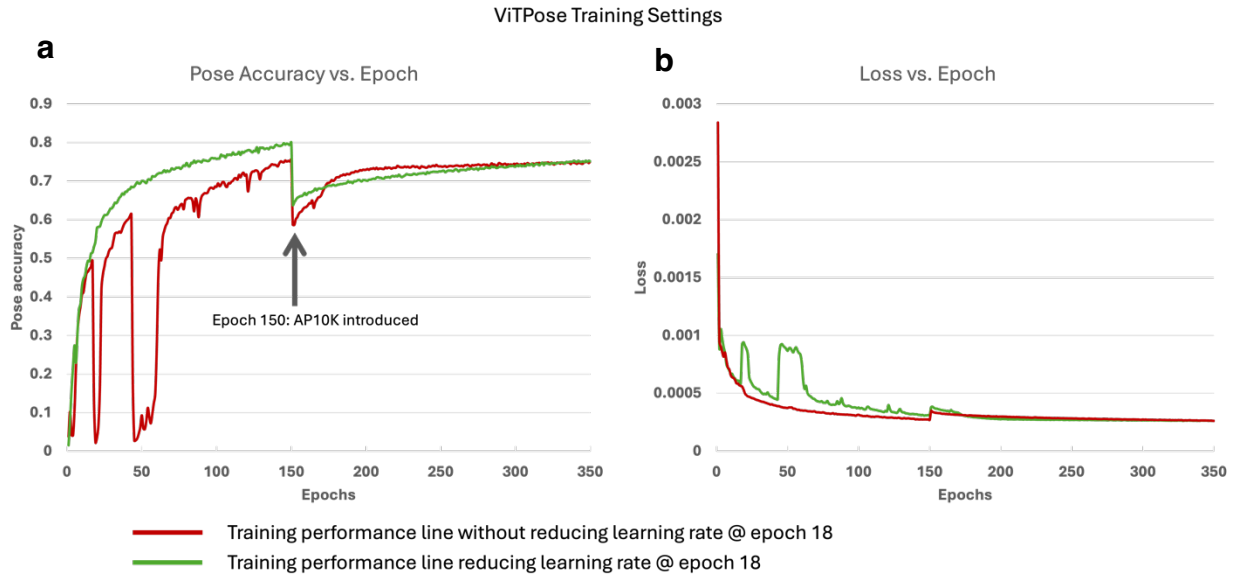


Figure **6**. ViTPose-B phase 1 training performance **a.** The pose accuracy per epoch **b.** The Loss per epoch: the red line - when the learning rate is not reduced at the 18th epoch, leading to an unstable training that eventually converges, the green line - when the learning rate is reduced at the 18th epoch leading to a more stable training with better performance. Note, the dip at the 150th epoch occurs as a result of introducing AP-10K dataset in the progressive training.

HRNet and TransPose model training. These models were found to be more sensitive to training with the proposed dual-prediction head modification compared to the ViTPose-based models. Hence, the need for a carefully designed continual learning approach. A common issue encountered in continual learning is catastrophic forgetting, where models lose previously learned information when adapting to new tasks[57]. This occurs because the model continuously updates

its parameters, often overwriting past knowledge from an initial task in favor of a more recent task. An ideal multitask model mitigates this by learning an optimal parameter space that effectively generalizes across tasks while preserving previously acquired knowledge**.** To mitigate catastrophic forgetting in HRNet and TransPose-based models, we adopted the training settings outlined in Table 4. Specifically, we reduced the learning rate by a smaller multiplication factor of 0.05, which helps in stabilizing updates and retaining existing knowledge. By implementing a more gradual learning rate reduction, the model's updates become less aggressive, thereby minimizing the risk of overwriting previously learned information.

Table 4: HRNet & TransPose Training Settings

Starting LR	0.001
LR Scheduler	Multi-step
LR Step Factor	0.05
LR Steps (epochs)	120
Batch size (HRNet I TransPose)	32 I 24
Optimizer	Adam[56]
Data Arrangement for progressive training Phase 1	
Epochs 0 -74	Animal-Pose
Epochs 75 - 119	Animal-Pose + AP-10K
Epochs 120 - 349	Animal-Pose + AP-10K + JumboPose
Data Arrangement for progressive training Phase 3	
Epochs 350 - 500	Animal-Pose + AP-10K + Pseudo_other_quadrupeds + JumboPose + Pseudo_elephants

Cost function

The mean squared error (MSE) is used to evaluate the Euclidian distance between the predicted and ground-truth keypoints. The objective function, L_{joint} for training the network using a set of labeled and pseudo-labeled datasets, is given as

$$L_{joint} = L_{sup} + cL_{pseudo} \tag{1}$$

Where L_{sup} and L_{pseudo} are the mean squared errors (MSE) on the manually labeled data (supervised) and pseudo-labeled examples, respectively. The binary flag c determines if pseudo-labels are used:

$$c = \begin{cases} 1, if\ using\ pseudo - labels \quad (student\ training, phase\ 3) \\ 0, otherwise \quad (teacher\ training, phase\ 1) \end{cases} \tag{2}$$

Each loss term, L_{sup} or L_{pseudo} is the sum of MSE losses for task 1 (elephant pose estimation) and task 2 (other quadruped pose estimation):

$$L = L_{MSE}(y_i^1, \hat{y}_i^1) + L_{MSE}(y_i^2, \hat{y}_i^2) \tag{3}$$

Where y_i^1 and y_i^2 are the ground-truth labels for tasks 1 and 2, respectively while $\hat{y}_i^1$ and, $\hat{y}_i^2$ correspond to model predictions for tasks 1 and 2.

$$L_{pseudo} = \sum_{j=0}^{N} w_j^{\emptyset} L(I_j^T, m(I_j^T|\emptyset) \tag{4}$$

L_{pseudo} is similar to equation (8) in the work proposed by Cao et al. [37] for self-paced selection of pseudo-labels and $m(I_j^T|\emptyset)$ is the output by the model of current weights $\emptyset$ on an input image I_j^T. $w_j^{\emptyset}$ denotes whether the pose prediction on I_j^T is a hard or easy example.

$$w_j^{\emptyset} = \begin{cases} 1.0, if\ C(m(I_j^T|\emptyset)) > \mu \\ 0, otherwise \end{cases} \tag{5}$$

Where $C(m(I_j^T|\emptyset))$ denotes the output confidence score on I_j^T by the current model and μ is the threshold to filter unreliable outputs. μ is set to 0.9 for our task to select only high confidence pseudo-labels.

Algorithms

Algorithm 1 outlines the training process using the dual prediction head architecture. Where a data sample could either be elephant (k =1*)* or other quadrupeds (k = 2*)*. Both sample types pass through the same encoder backbone, which extracts features and learn representations common to all quadrupeds. For the prediction heads, if k = 1(elephant), the extracted features are routed to the prediction head, H_1 to generate 33 heatmaps corresponding to the keypoints. If k = 2 (other quadrupeds), the extracted features are routed to prediction head, H_2 to produce 20 heatmaps corresponding to the predicted key points.

Algorithm 1 Dual-Head Model Training

Input: Training data, $D = (x_i^k,\ y_i^k): k \in \{1, 2\}$
$k = 1 \triangleleft$ elephant, $k = 2 \triangleleft$ other quadrupeds
$x_i^k, y_i^k \triangleleft ith$ input image and ground truth respectively
Encoder backbone, Enc
Prediction Heads, $H_1\ and\ H_2$

1: **Initialize** Encoder backbone Enc and Prediction Heads $H_2\ and\ H_2$
2: **repeat**
3: $(x_i^k,\ y_i^k) \sim D \triangleleft$ sample batch from dataset
$\emptyset = Enc(x_i^k) \triangleleft data\ goes\ into\ encoder\ backbone$
4: $if\ k == 1$:
5: $\hat{y}_i^1 = H_1(\emptyset) \triangleleft$ Route $\emptyset$ to H_1
6: $elif\ k == 2$:
7: $\hat{y}_i^2 = H_2(\emptyset) \triangleleft$ Route $\emptyset$ to H_2
8: $L = loss(y_i^1, \hat{y}_i^1) + loss(y_i^2, \hat{y}_i^2)$
9: Take gradient step to update $Enc, H_1\ and\ H_2$
10: **until** training converges

Algorithm 2 describes the inference process of the dual prediction head architecture. This strongly leverages the top-down nature of the pose estimation algorithms considered. First, an image is processed by a detector, which outputs the object class along with the bounding box coordinates.

The detected class is then binarized, assigning '1' to elephants and '2' to other quadrupeds. The bounding box coordinates define a cropped image region, which is passed through the encoder backbone for feature extraction. The extracted features are then routed identically to Algorithm 1.

Algorithm 2 Dual-Head Model Inferencing

Input: Image $= x$
Detector d
Trained model Encoder backbone Enc and Prediction Heads H_1 and H_2

Load trained Encoder backbone, Enc and Prediction Heads, $H_{2,}$ and H_2 weights
$x^k = d(x) \triangleleft$ $detected\ object\ from\ image\ with\ predicted\ class, k$
$\emptyset = Enc(x^k) \triangleleft$ $data\ goes\ into\ encoder\ backbone$
$if\ k\ \ ==\ 1$: $predicted\ class\ is\ '1'$ (elephant class)
$\hat{y}_i^1 = H_1(\emptyset) \triangleleft$ Route $\emptyset$ to H_1
$elif\ k\ \ ==\ 2$: $predicted\ class\ is\ '2'$ (other quadruped class)
$\hat{y}_j^2 = H_2(\emptyset) \triangleleft$ Route $\emptyset$ to H_2

Model Architectures

QuadPose employs top-down models, including ViTPose[40], HRNet with polarized self-attention for improved representation capacity[39,18] and TransPose[41], each modified with dual prediction heads to enable multi-task pose estimation for elephants and other quadrupeds. This method follows a detection-first approach, which isolates objects of interest using bounding boxes while simultaneously leveraging class information as a signal to route the detected objects to the appropriate prediction head. Specifically, the classification is binarized into two categories: (1) Class 1- if an elephant is detected, it is routed to prediction head H_1 and (2) Class 2 – if a general quadruped is detected, it is routed to prediction head H_2. This adaptive routing mechanism ensures that each animal is processed by the most relevant prediction head. A pretrained faster R-CNN[1] model is used for object detection during inference.

Evaluation Metric

The Object Keypoint Similarity (OKS)[58] calculates the distance between predicted keypoints, and ground-truth points normalized by the object's scale[59]. OKS values serve as thresholds for computing mean average precision (mAP), which ranges from 0 to 1, with higher values indicating more accurate keypoint localization.

$$OKS = \frac{\sum_i exp(-d_i^2\ /2s^2k_i^2)\delta(v_i\ >\ 0)}{\sum_i \delta(v_i\ >\ 0)} \tag{6}$$

Where,

- d_i is the Euclidean distance between the ground-truth and predicted keypoint
- s is the square root of the object segment area
- k is the per-keypoint constant that controls fall off
- v_i is the visibility flag that can be 0, 1, 2 for not labeled, labeled but not visible and visible and labeled respectively
- $\delta(v_i\ >\ 0)$ ensures only labeled keypoints contribute

Acknowledgements

The authors thank Allen Rutberg for his valuable insights and Victor Oludare for his insightful contributions and code assistance. Funding was partly provided by the Humane Society of the United States and the Cummings School of Veterinary Medicine at Tufts University, the Tufts Elephant Conservation Alliance (TECA), and The Bailey Wildlife Foundation.

Author Contributions

Conceptualization, K. Panetta, O. J., and S. R.; Methodology, O. J., S. R., and K. Panetta; Software, K. Panetta, O. J., S. R., J. H.; Validation, K. P., O. J., S. R., J. H., S. A.; Formal Analysis, K. Panetta, O. J., S. A.; Investigation, O. J., S. R., J. H.; K. Panetta.; Resources, K. Panetta.; Data Curation, O. J.,; Writing— Original Draft, O. J., S. R., K. Panetta.; Writing— Review and Editing, K. Panetta, O. J., S. R., J. H., S. A., K. Pereira; Visualization, O. J., K. Pereira; Supervision, K. Panetta, S.A.; Project Administration, K. Panetta, S.A.; Funding Acquisition, K. Panetta. All authors have read and agreed to the published version of the manuscript.

Competing Interests

The authors declare no competing interests.

Data Availability

JumboPose is made publicly available with download instructions at https://github.com/QuadPose.

Code Availability

QuadPose source code is available at https://github.com/QuadPose. All other requests should be made to the corresponding author.

References

1. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 1137–1149 (2015).
2. Varghese, R. & M., S. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)* 1–6 (IEEE, Chennai, India, 2024). doi:10.1109/ADICS58448.2024.10533619.
3. Carion, N. *et al.* End-to-End Object Detection with Transformers. Preprint at https://doi.org/10.48550/ARXIV.2005.12872 (2020).
4. Liu, W. *et al.* SSD: Single Shot MultiBox Detector. (2015) doi:10.48550/ARXIV.1512.02325.
5. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* **2016-December**, 770–778 (2015).
6. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.* **9351**, 234–241 (2015).
7. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. Preprint at https://doi.org/10.48550/ARXIV.1703.06870 (2017).
8. Oludare, V., Kezebou, L., Jinadu, O., Panetta, K. & Agaian, S. Attention-based two-stream high-resolution networks for building damage assessment from satellite imagery. in *Multimodal Image Exploitation and Learning 2022* (eds. Agaian, S. S., Asari, V. K., DelMarco, S. P. & Jassim, S. A.) vol. 12100 121000L (SPIE, 2022).
9. Kirillov, A. *et al.* Segment Anything. Preprint at https://doi.org/10.48550/ARXIV.2304.02643 (2023).
10. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. Preprint at https://doi.org/10.48550/ARXIV.1606.00915 (2016).

11. Cao, Z., Hidalgo, G., Simon, T., Wei, S. E. & Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 172–186 (2018).
12. Mathis, A. *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **21**, 1281–1289 (2018).
13. Ye, S. *et al.* SuperAnimal pretrained pose estimation models for behavioral analysis. *Nat. Commun.* **15**, 5165 (2024).
14. Pereira, T. D. *et al.* Fast animal pose estimation using deep neural networks. *Nat. Methods* **16**, 117–125 (2019).
15. Lauer, J. *et al.* Multi-animal pose estimation, identification and tracking with DeepLabCut. *Nat. Methods* **19**, 496–504 (2022).
16. Dong, C. & Du, G. An enhanced real-time human pose estimation method based on modified YOLOv8 framework. *Sci. Rep.* **14**, 8012 (2024).
17. Yang, J., Zeng, A., Zhang, R. & Zhang, L. X-Pose: Detecting Any Keypoints. Preprint at https://doi.org/10.48550/ARXIV.2310.08530 (2023).
18. Wang, J. *et al.* Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 3349–3364 (2021).
19. Vaswani, A. *et al.* Attention Is All You Need. Preprint at https://doi.org/10.48550/ARXIV.1706.03762 (2017).
20. Dosovitskiy, A. *et al.* An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Preprint at https://doi.org/10.48550/ARXIV.2010.11929 (2020).
21. Ning, G., Liu, P., Fan, X. & Zhang, C. A Top-down Approach to Articulated Human Pose Estimation and Tracking. *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.* **11130 LNCS**, 227–234 (2019).
22. Geng, Z., Sun, K., Zhang, Z. & Wang, J. Bottom-Up Human Pose Estimation Via Disentangled Keypoint Regression.
23. Cao, Z., Simon, T., Wei, S. E. & Sheikh, Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognit. CVPR 2017* **2017-January**, 1302–1310 (2016).
24. Iqbal, U. & Gall, J. Multi-Person Pose Estimation with Local Joint-to-Person Associations. Preprint at https://doi.org/10.48550/ARXIV.1608.08526 (2016).
25. Gkioxari, G., Hariharan, B., Girshick, R. & Malik, J. Using k-Poselets for Detecting People and Localizing Their Keypoints. in *2014 IEEE Conference on Computer Vision and Pattern Recognition* 3582–3589 (IEEE, Columbus, OH, USA, 2014). doi:10.1109/CVPR.2014.458.
26. Papandreou, G. *et al.* Towards Accurate Multi-person Pose Estimation in the Wild. Preprint at https://doi.org/10.48550/ARXIV.1701.01779 (2017).
27. Li, M., Zhou, Z., Li, J. & Liu, X. Bottom-up Pose Estimation of Multiple Person with Bounding Box Constraint.
28. Pishchulin, L. *et al.* DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. Preprint at https://doi.org/10.48550/ARXIV.1511.06645 (2015).
29. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M. & Schiele, B. DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model. Preprint at https://doi.org/10.48550/ARXIV.1605.03170 (2016).
30. Lin, T.-Y. *et al.* Microsoft COCO: Common Objects in Context. in *Computer Vision – ECCV 2014* (eds. Fleet, D., Pajdla, T., Schiele, B. & Tuytelaars, T.) vol. 8693 740–755 (Springer International Publishing, Cham, 2014).
31. Zhang, S.-H. *et al.* Pose2Seg: Detection Free Human Instance Segmentation. in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 889–898 (IEEE, Long Beach, CA, USA, 2019). doi:10.1109/CVPR.2019.00098.
32. Andriluka, M., Pishchulin, L., Gehler, P. & Schiele, B. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. in *2014 IEEE Conference on Computer Vision and Pattern Recognition* 3686–3693 (IEEE, Columbus, OH, USA, 2014). doi:10.1109/CVPR.2014.471.
33. Li, S., Li, J., Tang, H., Qian, R. & Lin, W. ATRW: A Benchmark for Amur Tiger Re-identification in the Wild. *MM 2020 - Proc. 28th ACM Int. Conf. Multimed.* **20**, 2590–2598 (2019).
34. Mathis, A. *et al.* Pretraining boosts out-of-domain robustness for pose estimation. in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)* 1858–1867 (IEEE, Waikoloa, HI, USA, 2021). doi:10.1109/WACV48630.2021.00190.

35. Labuguen, R. *et al.* MacaquePose: A Novel "In the Wild" Macaque Monkey Pose Dataset for Markerless Motion Capture. *Front. Behav. Neurosci.* **14**, 268 (2021).
36. Graving, J. M. *et al.* DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife* **8**, e47994 (2019).
37. Cao, J. *et al.* Cross-Domain Adaptation for Animal Pose Estimation. *Proc. IEEE Int. Conf. Comput. Vis.* **2019-October**, 9497–9506 (2019).
38. Yu, H. *et al.* AP-10K: A Benchmark for Animal Pose Estimation in the Wild. Preprint at https://doi.org/10.48550/ARXIV.2108.12617 (2021).
39. Liu, H., Liu, F., Fan, X. & Huang, D. Polarized self-attention: Towards high-quality pixel-wise mapping. *Neurocomputing* **506**, 158–167 (2022).
40. Xu, Y., Zhang, J., Zhang, Q. & Tao, D. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. Preprint at https://doi.org/10.48550/arXiv.2204.12484 (2022).
41. Yang, S., Quan, Z., Nie, M. & Yang, W. TransPose: Keypoint Localization via Transformer. Preprint at https://doi.org/10.48550/ARXIV.2012.14214 (2020).
42. Korschens, M. & Denzler, J. ELPephants: A Fine-Grained Dataset for Elephant Re-Identification. in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* 263–270 (IEEE, Seoul, Korea (South), 2019). doi:10.1109/ICCVW.2019.00035.
43. Xu, Y., Zhang, J., Zhang, Q. & Tao, D. ViTPose++: Vision Transformer for Generic Body Pose Estimation. Preprint at https://doi.org/10.48550/ARXIV.2212.04246 (2022).
44. Price, J. (July 25, 2024). The kangaroo: All you need to know about Australia's most iconic animal - and its famous hop [Image]. *DiscoverWildlife*. https://www.discoverwildlife.com/animal-facts/mammals/kangaroo-facts.
45. Yu, H. *et al.* AP-10K: A Benchmark for Animal Pose Estimation in the Wild. (2021).
46. Ferreira, B. (2020). African Wildlife, version 1. Retrieved January 3, 2025 from https://www.kaggle.com/datasets/biancaferreira/african-wildlife.
47. Banerjee, S. (2022). Animal Image Dataset (90 Different Animals), version 5. Retrieved January 3, 2025 from https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals
48. Trivedi, Y. V. (2021). animals_5, version 1. Retrieved January 3, 2025 from https://www.kaggle.com/datasets/ytrivedi1/animals-5.
49. Alessio, C. (2020). Animals-10, version 1. Retrieved January 3, 2025 from https://www.kaggle.com/datasets/alessiocorrado99/animals10.
50. Jamil, S. (2021). Endangered Animals, version 1. Retrieved January 3, 2025 from https://www.kaggle.com/datasets/sonain/endangered-animals.
51. Antoreepjana. (2021). IUCN Animals Dataset, version 1. Retrieved January 3, 2025 from https://www.kaggle.com/datasets/antoreepjana/iucn-animals-dataset.
52. Sahovic, E. (2021). wild_cats, version 1. Retrieved January 3, 2025 from https://www.kaggle.com/datasets/enisahovi/cats-projekat-4.
53. Vivek. (2022). Asian vs African Elephants, version 1. Retrieved January 3, 2025 from https://www.kaggle.com/datasets/vivmankar/asian-vs-african-elephant-image-classification/data.
54. Bengio, Y., Louradour, J., Collobert, R. & Weston, J. Curriculum learning. *ACM Int. Conf. Proceeding Ser.* **382**, (2009).
55. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. Preprint at https://doi.org/10.48550/ARXIV.1912.01703 (2019).
56. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. Preprint at https://doi.org/10.48550/ARXIV.1412.6980 (2014).
57. Kirkpatrick, J. *et al.* Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci.* **114**, 3521–3526 (2017).
58. Ronchi, M. R. & Perona, P. Benchmarking and Error Diagnosis in Multi-Instance Pose Estimation. Preprint at https://doi.org/10.48550/ARXIV.1707.05388 (2017).
59. COCO - Common Objects in Context. Available at https://cocodataset.org/#keypoints-eval. (Accessed: 20-Dec-2021).