Université de Neuchâtel
Faculté des sciences économiques

ACTION RECOGNITION OF GREAT APE BEHAVIORS AND COMMUNICATIVE GESTURES USING DEEP LEARNING

PhD thesis submitted to the Faculty of Economics and Business

Information Management Institute

University of Neuchâtel

For the PhD degree in Computer Science

by

Michael FUCHS

Approved by the dissertation committee:

Prof. Kilian STOFFEL, University of Neuchâtel, thesis co-director

Dr. MER Paul COTOFREI, University of Neuchâtel, thesis co-director

Prof. Adrian HOLZER, University of Neuchâtel

Prof. Klaus ZUBERBÜHLER, University of Neuchâtel

Prof. Rolf INGOLD, University of Fribourg

Prof. Hatem GHORBEL, HE-Arc Engineering

Thesis defended on January 30, 2025.

unine
Université de Neuchâtel
Faculté des sciences économiques

IMPRIMATUR POUR LA THÈSE

Action recognition of great ape behaviors and communicative gestures using deep learning

Michael FUCHS

UNIVERSITÉ DE NEUCHÂTEL
FACULTÉ DES SCIENCES ÉCONOMIQUES

La Faculté des sciences économiques,
sur le rapport des membres du jury

Prof. Kilian STOFFEL, Université de Neuchâtel, co-directeur de thèse
Dr. MER Paul COTOFREI, Université de Neuchâtel, co-directeur de thèse
Prof. Adrian HOLZER, Université de Neuchâtel
Prof. Klaus ZUBERBÜHLER, Université de Neuchâtel
Prof. Rolf INGOLD, Université de Fribourg
Prof. Hatem GHORBEL, HE-Arc Ingénierie (HES-SO), St-Imier

autorise l'impression de la présente thèse.

Neuchâtel, le 11 février 2025

Le doyen
Peter Fiechter

Faculté des sciences économiques
Avenue du 1er-Mars 26
CH-2000 Neuchâtel
Email: secretariat.seco@unine.ch
Tel: +41 32 718 15 00
www.unine.ch/seco

To *Sonya*,
my supportive wife and exceptional sparring partner,

and

To *Shilo* and *Maya*,
our beautiful children,
who inspire me to learn something new every day.

Acknowledgements

This thesis would not have been possible without the help and support of many individuals, to whom I would like to express my deepest gratitude.

First and foremost, I would like to thank my supervisors, Prof. Kilian Stoffel and Dr. MER Paul Cotofrei, for their invaluable guidance and support throughout this endeavor. I am also grateful to Prof. Klaus Zuberbühler and Dr. Emilie Genty for our fruitful collaboration and for sharing their passion for great apes.

I would like to thank all my colleagues at the Information Management Institute of the University of Neuchâtel, including Prof. Adrian Holzer, Prof. Giuseppe Melfi, Prof. Catalin Starica, Prof. Eric Simon, Dr. Vladimir Macko, Dr. Kristoffer Bergram, Dr. Martina Raggi, Dr. Pierluigi Giosi, Dr. Alessio De Santo, Dr. Aditya Purohit, Dr. Selena Baset, Dr. Iulian Ciorascu, Eugenia Cotofrei, Eliane Maalouf, Natalia Bartlomiejczyk, Manon Berney, Romain Claret, Michael Palma Mendes, and Abdessalam Ouaazki.

I am also thankful to other members of the NCCR Evolving Language, including Prof. Daphné Bavelier, Prof. Richard Hahnloser, Dr. Nianlong Gu, Dr. Guanghao You, Dr. Remo Nitschke, and Dr. Erik Ringen, as well as the staff at Basel Zoo, including Adrian Baumeyer, David Lehnert, and all those involved in the care of the chimpanzees.

Finally, I thank my family and friends for their unwavering support and encouragement throughout these years. Your belief in me has been a constant source of strength and motivation.

Abstract

The study of great ape behavior and communication is essential for understanding the evolutionary foundations of human language and social interaction. However, traditional methods relying on manual video annotation are time-consuming, labor-intensive, and limited in scalability. Recent advancements in computer vision and deep learning offer transformative potential to automate the recognition of great ape behaviors and gestures, yet their application to this domain remains limited.

This dissertation introduces novel approaches to address these challenges by leveraging deep learning techniques and datasets. It presents *ASBAR* (*Animal Skeleton-Based Action Recognition*), a framework that combines pose estimation and action recognition into a unified pipeline, achieving competitive accuracy in classifying natural great ape behaviors in the wild while significantly reducing computational and storage requirements.

Additionally, it introduces *ChimpBehave*, a dataset of zoo-housed chimpanzee videos annotated for behavior recognition, enabling the study of domain adaptation and cross-dataset generalization. Results from benchmarking video- and skeleton-based models reveal the robustness of skeleton-based methods in handling visual variability across datasets.

Further, the dissertation develops *FineChimp*, a fine-grained dataset specifically designed for recognizing great ape gestures. With expert annotations across 38 gesture classes and multiview recordings, *FineChimp* serves as a benchmark for gesture recognition, demonstrating the efficacy of state-of-the-art deep learning models in decoding the nuances of great ape communication.

By integrating innovative computer vision techniques with detailed behavioral data, this work automates and enhances the study of great ape behavior and communication, offering scalable tools for primatology research. These contributions have implications for conservation, behavioral science, and the broader understanding of animal behaviors and communication systems.

Keywords: Computer vision, deep learning, great apes, chimpanzees, animal behavior, gestural communication, action recognition, pose estimation, annotated datasets, artificial intelligence, gesture recognition, automated behavioral analysis, video analysis.

Résumé

L'étude des comportements et de la communication des grands singes est essentielle à la compréhension des fondements évolutionnaires du langage humain et de ses interactions sociales. Cependant, les méthodes traditionnelles, qui reposent sur l'annotation manuelle de données vidéo, sont laborieuses, chronophages et peu efficientes. À l'inverse, les récentes avancées en vision par ordinateur et en apprentissage profond offrent un potentiel nouveau pour automatiser la reconnaissance des comportements et des gestes des grands singes. Cela dit, leurs applications à ce domaine restent, pour l'heure, limitées.

Cette thèse propose des approches novatrices pour répondre à ces défis en tirant parti des techniques d'apprentissage profond et des jeux de données associés. Elle présente *ASBAR* (dont l'acronyme français serait *RAABS*, pour *Reconnaissance d'Actions Animales Basée sur les Squelettes*), un cadre qui combine l'estimation de pose à la reconnaissance d'actions à travers une approche unifiée, atteignant des résultats compétitifs dans la classification des comportements des grands singes en milieu naturel, tout en réduisant drastiquement les besoins computationnels et de stockage.

Elle introduit également *ChimpBehave*, un jeu de données vidéo annoté pour la reconnaissance des comportements de chimpanzés en captivité, qui permet l'étude de l'adaptation au domaine et de la généralisation entre jeux de données. L'évaluation de modèles basés soit sur la vidéo, soit sur les squelettes révèle la robustesse de ces derniers face à la variabilité visuelle entre jeux de données.

En outre, cette thèse propose *FineChimp*, un jeu de données d'actions fines conçu spécifiquement pour la reconnaissance des gestes des grands singes. Avec ses 38 classes de gestes annotées par des experts et ses enregistrements provenant de multiples points de vue, *FineChimp* permet l'étalonnage des modèles de reconnaissance de gestes et démontre l'efficacité des modèles d'apprentissage profond de pointe pour décoder les nuances de la communication des grands singes.

En intégrant des techniques innovantes de vision par ordinateur à des données comportementales détaillées, ce travail automatise et enrichit l'étude des comportements et de la communication des grands singes, en apportant des outils évolutifs à la recherche en primatologie. Ces contributions ont des implications pour la conservation animale, les sciences

comportementales et, de manière générale, la compréhension des comportements et des systèmes de communication animaliers.

Mots-clés: Vision par ordinateur, apprentissage profond, grands singes, chimpanzés, comportement animal, communication gestuelle, reconnaissance d'actions, estimation de pose, jeux de données annotés, intelligence artificielle, reconnaissance de gestes, étude comportementale automatisée, analyse vidéo.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation and Scope

The widespread adoption of deep learning technologies has increasingly enabled Artificial Intelligence (AI) to assist humans, revolutionizing interactions between humans and machines. The study of animal behavior has similarly benefited from this technological shift, with scientists progressively relying on computer vision techniques to automatically detect and identify individuals and quantify their behaviors [149, 27, 223, 98]. Quantifying animal behaviors is highly relevant in biology [193], and the automation of such methods has significant implications for fields like computational ethology [4], conservation ecology [201, 62], and neuroscience [102, 154]. This trend in animal behavior research is expected to accelerate, as advances in Vision Foundation Models demonstrate that general-purpose architectures can perform comparably to domain-specific methods in animal behavior recognition with minimal adaptations [188].

Despite these advancements, deploying AI for animal behavior recognition in real-world scenarios remains challenging. For instance, existing deep learning models often fail to differentiate between fundamental locomotive behaviors, such as walking and running, in great apes [17]. Such distinctions, however, are critical for applications like monitoring individual well-being. In contrast, models trained on human-centric datasets can recognize up to 600 actions with over 90% accuracy [206] and are already deployed in diverse applications, from autonomous vehicles [234] to automated surveillance systems [144].

One significant barrier to progress in animal behavior recognition is the scarcity of publicly available data. For instance, MammalNet [27], the largest public dataset for animal behavior analysis, includes *only* 539 hours of video annotated with 12 animal behaviors, distributed across fewer than 20,000 video clips and 173 animal categories. In contrast, the human-centric Kinetics700 dataset, released four years earlier [22], contains approximately

1,800 hours of video annotated with 700 human activities across 650,000 clips. This stark disparity arises from both economic and logistical factors.

Human-centric datasets benefit from the abundance of user-generated video content on social media platforms, often pre-labeled with tags or descriptions. Annotating human videos is inexpensive and does not require specialized expertise, while large datasets can also be created in controlled environments with actors performing predefined actions [175]. Conversely, collecting animal video data, particularly in primatology, presents unique challenges. Fieldwork often requires months in remote areas to record individuals in their natural habitats, and videos are typically gathered for specific research objectives (e.g., studying ape sleeping patterns), limiting their volume and variety. Annotation demands meticulous work by a small pool of domain experts, and while research findings are frequently shared publicly, the original videos and annotations are often withheld, creating a critical gap for data-hungry deep learning algorithms.

Despite these challenges, AI-assisted primatology is rapidly evolving. In 2023 alone, at least 16 studies employed deep learning methods for computer vision tasks, such as species identification [222, 27, 158, 36, 224], individual detection and tracking [222, 129, 124, 220, 119, 162], face recognition [173, 107], pose estimation [222, 124, 213, 119, 7, 36, 224], and behavior recognition [124, 113, 119, 27, 15]. Remarkably, over 40% of these studies also published relatively large public datasets containing annotated images or videos of primates [222, 129, 124, 119, 27, 36, 224].

In contrast, the narrower field of automated great ape behavior recognition remains relatively underexplored, despite its high relevance. Notably, all non-human great ape species are classified as either endangered or critically endangered [87], and automated systems for population monitoring could significantly aid conservation efforts [201]. Moreover, as our closest evolutionary relatives, great apes provide unique insights into the origins and evolution of human language through the study of their communicative systems and gestural signals [30, 195, 160].

Despite its significance, deep learning for great ape behavior understanding is still in its infancy, with fewer than 10 contributions to date [167, 17, 16, 5, 149, 15, 27, 124, 53] (see Sect. 2.4.2). Practical implications for real-life deployment remain largely unproven, and the recognition of great ape communicative gestures unexplored. Furthermore, only three public video datasets are available for benchmarking great ape behaviors: *PanAf500* [17], *PanAf20K* [17], and *ChimpACT* [124]. While valuable individually, these datasets present significant limitations collectively: (i) each focuses on a distinct machine learning task (e.g., multi-class behavior recognition, multi-label behavior recognition, or spatio-temporal behavior detection); (ii) they use independent annotation schemes, which are not always aligned

(see Tabel T4 in Appendix C); and (iii) cumulatively, they cover only 40 behavior classes, representing just 10% of the the number of terms documented in chimpanzee ethograms [150].

These limitations highlight the need for more comprehensive methodologies and datasets to advance great ape behavior recognition, bridging the gap between potential applications and practical deployment.

Great ape behavior recognition relies on action recognition techniques, a machine learning task that predicts the action class of a video clip [2]. This is typically approached using either *video-based* or *skeleton-based* methods [204]. Video-based models use entire video sequences as input, leveraging all pixel information from frames, capturing subtle visual patterns (e.g., blinking eyes or finger movements), and incorporating scene context [109]. However, the high dimensionality of video data demands significant computational resources and large amounts of training data. Skeleton-based models, in contrast, rely on low-dimensional representations of skeletal structures, focusing on motion patterns rather than appearance [47]. These models are computationally efficient and robust in cross-subject scenarios but may lack granularity and rely on accurate pose estimation. Figure 1.1 shows an example of how images can be transformed into skeletal representations.

To date, research in great ape behavior classification has *only focused on video-based approaches*, despite the potential advantages of skeleton-based methods for capturing motion-specific behaviors. Furthermore, these studies *do not investigate the out-of-distribution generalization capacity of models*, i.e., how well models perform on test data whose distribution differs from that of the training set [80]. This limitation is particularly critical for video-based approaches, which are known to be more sensitive to distribution shifts [26, 217]. Out-of-distribution generalization is an essential consideration in animal behavior research, especially in primatology, where individuals are often recorded in diverse environments (e.g., in nature or zoos) and species exhibit shared behaviors but with varying visual appearances (e.g., a climbing chimpanzee versus a climbing orangutan). Lastly, while previous deep learning studies have explored great ape behaviors, they *have yet to address the automated recognition of great ape gestures*, a key component of their communication systems.

The aforementioned considerations highlight critical gaps in the field that need to be addressed. This dissertation contributes to computer vision by designing, developing, and implementing deep learning models, methods, and tools for understanding great ape behavior and communication. Specifically, we focus on supervised action recognition techniques to investigate the automated classification of great ape behaviors and communicative gestures. Alongside our findings, we publicly release models, code, and annotated video datasets to facilitate future research.

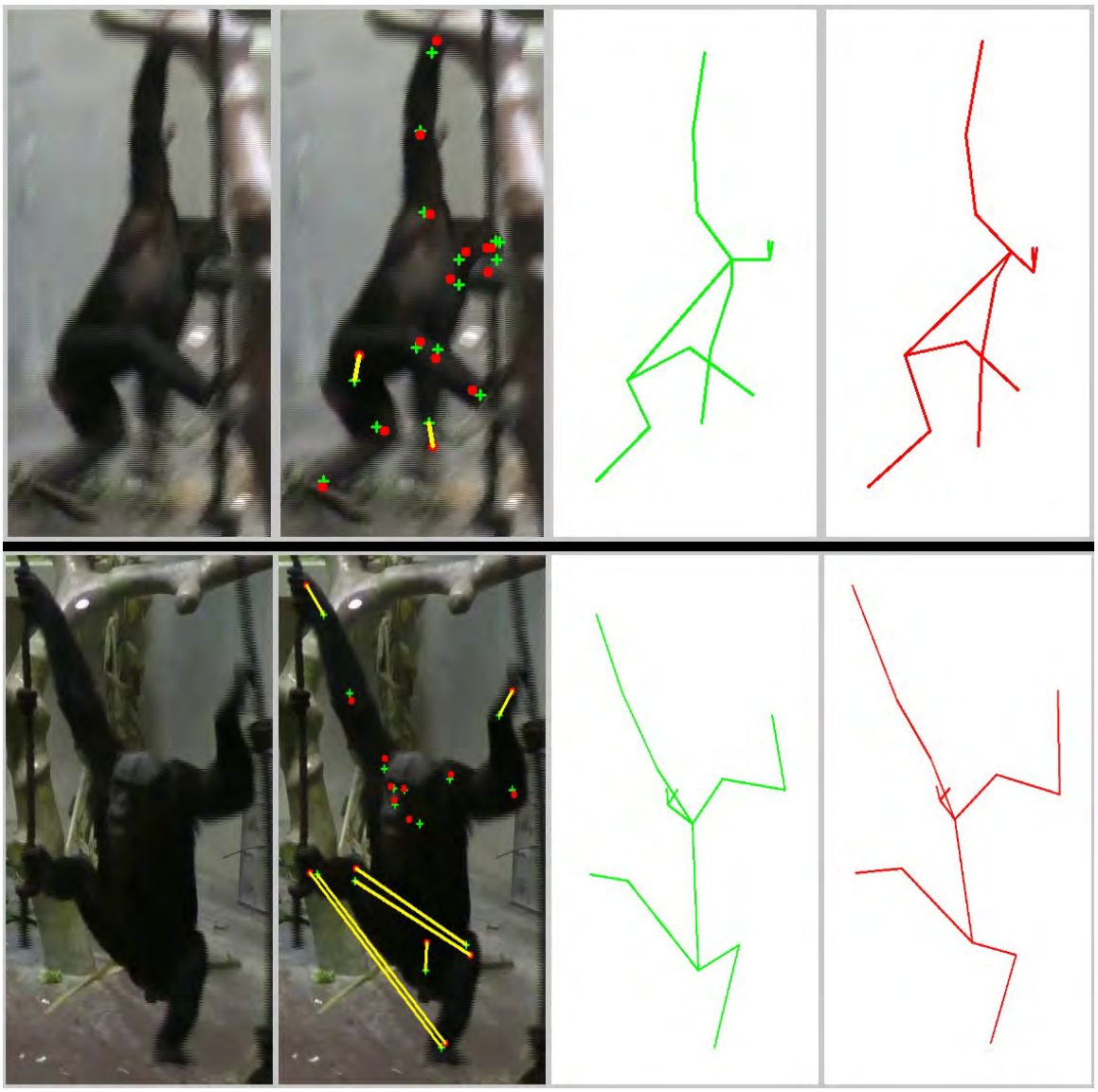

Fig. 1.1 **Comparison of chimpanzee images and their skeletal representations**. From left to right: (1) the original image, (2) ground truth keypoint coordinates (green crosses) alongside the pose estimation model's predictions (red dots), with prediction errors exceeding the length of the nose highlighted by yellow segments, (3) the ground truth skeleton (green), and (4) the predicted skeleton (red). A video-based action recognition approach processes the high-dimensional images shown on the far left, while a skeleton-based approach uses the low-dimensional representations on the far right. Note in the bottom row that the pose estimation model inverted the left/right knees and ankles in its prediction. However, this inversion does not significantly alter the skeletal representation.

1.2 Problem Statement

The previous considerations reveal several gaps in the literature that remain unaddressed. The overarching problem statement of this dissertation can thus be summarized as follows: *How can deep learning action recognition methods be applied to animal behavior understanding, particularly in recognizing great ape behaviors and communicative gestures?*

This general problem statement is broken down into three specific research questions:

RQ1 *How can skeleton-based action recognition be successfully applied to the study of animal behaviors, particularly for recognizing great ape behaviors?*

RQ2 *How do video-based and skeleton-based great ape behavior recognition methods compare in their ability to generalize to out-of-distribution data?*

RQ3 *How can action recognition methods be effectively applied to the recognition of fine-grained great ape communicative gestures?*

1.2.1 Skeleton-Based Behavior Recognition

Skeleton-based action recognition offers several advantages, including lightweight architectures, strong generalization capabilities across individuals, and the potential to generalize across species. However, such methods require the prior identification of an individual's pose through *pose estimation*.

Unsurprisingly, pose estimation—the task of inferring body part coordinates from images [21]—has become a central focus in animal behavior research. In recent years, there has been a surge in open-source frameworks (e.g., DeepLabCut [133, 134, 108], SLEAP [153, 155], or AniPose [94]) and datasets dedicated to animal pose estimation (e.g., [149, 20, 226, 224, 36, 106, 132]. While these advancements have driven progress, pose estimation is only a means to an end: quantifying behavior. Despite this progress, skeleton-based action recognition methods have not yet been effectively applied to the study of animal behaviors, let alone the specific domain of great ape behaviors.

Several key aspects must be considered when tackling this topic:

- *Pose Estimation Accuracy*: How can large, unspecific pose estimation datasets be leveraged to achieve the accuracy required for subsequent behavior classification?
- *Transferability*: Among the various skeleton-based action recognition methods developed for humans, can any be adapted for large-scale application to great apes?

- *Framework Accessibility*: Can a general framework be designed to enable other researchers to use the same model and data pipeline for studying other species or behaviors?

These considerations lead to the first research question:

RQ1: *How can skeleton-based action recognition be successfully applied to the study of animal behaviors, particularly for recognizing great ape behaviors?*

1.2.2 Out-of-Distribution Generalization

Skeleton-based methods for human action recognition have demonstrated several advantages over video-based approaches. In particular, video-based models often suffer from low out-of-distribution generalization capacity. This means that while such models may perform well in the specific visual contexts on which they are trained, they frequently fail to generalize to visually different environments during inference.

In contrast, skeleton-based methods rely only on an abstracted representation of the scene—namely, the estimated pose data—making them presumably more robust to variations in scene composition, individual appearances, lighting conditions, and other factors. This robustness is especially pertinent to animal behavior research, where video data is scarce and diverse. For instance, videos may be sourced from vastly different recording environments (e.g., African forests versus man-made zoos) and depict species that, while visually distinct, share significant morphological similarities (e.g., gorillas and orangutans).

Understanding how action recognition models generalize to out-of-distribution data is critical for studying many animal species, particularly great apes, and has not yet been investigated in the literature.

This motivates our second research question:

RQ2: *How do video-based and skeleton-based great ape behavior recognition methods comparatively generalize on out-of-distribution data?*

1.2.3 Recognition of Great Ape Gestures

Great ape communication is unique among animal communication systems, particularly in its use of gestures (i.e., gestural signals), which serve as expressions of *intention* during social interactions. Studying the gestural communication of non-human great apes is therefore essential for understanding the origins and evolution of human language.

While prior work has demonstrated the use of deep learning methods for classifying general great ape behaviors [17, 124], no existing contributions have focused on recognizing

great ape gestural signals. Given the fine-grained nature of these gestures and their importance in understanding communicative intent, addressing this gap is a significant step forward in great ape behavior research.

Thus, the third research question of this dissertation is:

RQ3: *How can action recognition methods be successfully applied to the recognition of fine-grained great ape communicative gestures?*

1.3 List of Contributions

Each of the aforementioned research questions is addressed in the main three chapters of this dissertation, with each chapter presenting several key contributions. These contributions have been made possible by leveraging open-source resources, such as deep learning models and frameworks, publicly available datasets, annotation tools, and pedagogical material. As a gesture of reciprocity, all relevant materials for these contributions—including datasets, annotations, trained models and weights, and source code—will be made openly accessible to the research community.

1.3.1 The ASBAR Framework

The primary contribution of Chapter 3 is the development of the ASBAR framework, which stands for *Animal Skeleton-Based Action Recognition*. This framework fully integrates two key machine learning tasks—pose estimation and action recognition—into a unified system with complete data pipelines and model workflows.

1. *Pose Estimation Module*: Built upon DeepLabCut, one of the most widely adopted pose estimation toolboxes in animal research.

2. *Behavior Recognition Module*: Built upon MMAction2, an open-source platform for video understanding, enabling action classification.

While we test ASBAR on a particularly complex task — namely, the classification of great ape behaviors in the wild — the framework itself is species- and behavior-agnostic. This design allows it to accommodate the needs of researchers working on various animal behaviors across different species.

Unlike most human-centric skeleton-based approaches that predominantly rely on Graph Convolutional Networks (GCNs), our model pipeline incorporates PoseConv3D [39], a CNN-based architecture. PoseConv3D has demonstrated notable robustness to noisy pose

estimations and challenging action classifications, making it particularly well-suited for real-world animal behavior recognition where data quality can be inconsistent.

To promote accessibility and usability, we designed a Graphical User Interface (GUI) that encapsulates all of ASBAR's core functionalities. This user-friendly interface is particularly appropriate for researchers with limited programming expertise, facilitating broader adoption of the framework. The ASBAR framework is publicly accessible at:

https://github.com/MitchFuchs/asbar

1.3.2 General Primate Pose Estimator

Training pose estimation models using large, diverse datasets can present significant challenges. Leveraging ASBAR's pose estimation module, we rigorously train and evaluate nine pose estimation models on the OpenMonkeyChallenge dataset [224], which contains over 100,000 images of 26 different primate species annotated with 17-keypoint poses.

- *Within-Domain and Out-of-Domain Evaluation*: We assess model performance both within-domain (on OpenMonkeyChallenge) and out-of-domain (on a subset of video frames from the PanAf500 dataset [167]) to identify the model with the best generalization capacity.
- *Detailed Metrics*: We provide an in-depth performance analysis, reporting metrics both at the species and individual body-part levels.

Additionally, we release a set of annotations consisting of nearly 5,500 keypoint ground-truth annotations, labeled by expert primatologists on the PanAf500 dataset. This data will allow the research community to build upon our work and improve pose estimation results.

1.3.3 Great Ape Skeleton-based Behavior Recognition

Using the best-performing model identified above, we demonstrate a methodology for extracting skeletal poses of great apes from the PanAf500 dataset. This dataset includes over 180,000 video frames (equivalent to approximately 2 hours of footage) of chimpanzees and gorillas filmed in their natural habitats, and annotations for nine common great ape behaviors in 500 video segments, resulting in approximately 6,700 examples for behavior recognition.

Using ASBAR's action recognition module, we classify behaviors solely from extracted skeleton data, achieving performance comparable to previously reported video-based methods. Remarkably, this result is achieved without fine-tuning the pose estimation model, while reducing the dataset volume by a factor of 20 through skeleton extraction.

This contribution underscores the potential of skeleton-based methods as lightweight, data-efficient alternatives to traditional video-based approaches for animal behavior recognition. This advantage is particularly relevant for researchers operating in resource-constrained environments, where access to power, storage, or computational resources may be limited. Notably, this work represents the first application in the literature of skeleton-based action recognition to great apes, an important milestone in the field.

1.3.4 The ChimpBehave Dataset

In Chapter 4, we address the need for expanded publicly accessible datasets to investigate model generalization capacity for great ape behavior recognition. To this end, we introduce a novel dataset, ChimpBehave, which consists of:

- Approximately 215,000 high-resolution video frames (equivalent to around 2 hours and 20 minutes) of chimpanzees recorded at the Basel Zoo.
- Over 1,300 video segments (comprising approximately 10,000 examples), each labeled by an expert primatologist with one of eight common great ape behaviors.

Among the eight behaviors, seven were specifically chosen to overlap with those in the PanAf500 dataset. While the annotated behaviors align, the visual and recording conditions of ChimpBehave and PanAf500 are intentionally distinct. This design establishes a foundation for studying model generalization capacity and enables future researchers to explore domain adaptation techniques in primatology.

To date, ChimpBehave is the largest publicly available video dataset specifically annotated with chimpanzee behaviors. The dataset is publicly accessible at:

https://github.com/MitchFuchs/ChimpBehave

1.3.5 ChimpBehave Baselines

To establish a reference point for future research, we provide two initial baselines for video-based and skeleton-based methods on the ChimpBehave dataset. These baselines are designed to facilitate direct comparison and serve as a benchmark for future developments.

While the performances of the video-based and skeleton-based methods are statistically tied, the skeleton-based method consistently achieves higher mean scores and lower variance across all considered evaluation metrics, suggesting potentially greater reliability. Notably, this is achieved without any fine-tuning phase on the ChimpBehave dataset of the pose estimation model, further underscoring its robustness and efficacy.

1.3.6 Cross-dataset Generalization

We further evaluate and compare the performance of video-based and skeleton-based approaches in two different scenarios, respectively:

- *Within-dataset*: Models are trained and tested on the same dataset.
- *Cross-dataset*: Models are trained on one dataset (e.g., ChimpBehave) and tested on another (e.g., PanAf500).

Our results demonstrate that, while video-based and skeleton-based methods perform comparably well in within-dataset settings, skeleton-based approaches consistently outperform video-based methods in cross-dataset scenarios. These findings highlight the superior generalization capacity of skeleton-based methods in diverse visual contexts, making them particularly valuable for primatologists studying various primate species across visually heterogeneous environments.

1.3.7 The FineChimp Dataset

In Chapter 5, we shift our focus from general great ape behaviors to fine-grained great ape actions, specifically their communicative gestures. Since no publicly available dataset exists for great ape gestures, we introduce FineChimp, a second novel dataset as part of this dissertation's contributions.

The FineChimp dataset comprises over 100,000 video frames (approximately 1 hour of footage), and around 2,000 annotated examples - each labeled by an expert primatologist specialized in great ape gestural signals - with one of the 38 classes of chimpanzee gestures. This dataset is unique in several regards: (i) it is the first machine learning dataset specifically targeting animal gestural communication, (ii) it offers the most fine-grained classification of actions among existing great ape datasets, and (iii) it includes multi-view recordings of chimpanzees, with up to five synchronized viewpoints. FineChimp is publicly available for further research and development, at

https://github.com/MitchFuchs/FineChimp

1.3.8 Automated Recognition of Great Ape Gesture

To evaluate the FineChimp dataset, we benchmark several video-based action recognition models and explore various optimization strategies, including model pretraining, data augmentation techniques, hyperparameter tuning, and optical flow incorporation for motion

enhancement. Our best-performing model - a novel large dual-stream video transformer - achieves a Top-1 Accuracy that is 6 times higher and a Mean Class Accuracy that is 20 times higher, compared with the baseline (near random) classifier.

These results demonstrate the feasibility and effectiveness of automated recognition of fine-grained great ape communicative gestures, laying the groundwork for a promising research direction: leveraging computer vision to study animal gestural communication.

1.3.9 Multi-View Camera System

As part of the data collection process for FineChimp, we designed and deployed a custom multi-view camera system inside the chimpanzee enclosure at the Basel Zoo, enabling automated recordings from multiple viewpoints. The system's features include automatic primate detection for efficient recording triggers, over-the-network messaging to synchronously start and end recordings across viewpoints, high-quality video recording capabilities (up to 2K@50fps), low-power consumption (less than 15W per viewpoint), and cost-effective design using materials priced at approximately 300 CHF per viewpoint.

These properties make the system particularly appealing to animal behavior researchers operating under challenging conditions, such as limited power, budgets, or remote locations. All custom-designed software for the camera installation has been made open-source, ensuring accessibility and reproducibility for future research initiatives.

1.4 Thesis Structure

This dissertation is organized into six chapters:

- Chapter 1 introduces the dissertation, emphasizing the importance of automatically recognizing great ape behaviors and communication systems using deep learning. It provides an overview of AI-assisted primatology, identifies gaps in the current literature, and presents the main research questions and contributions of the dissertation.

- Chapter 2 reviews the theoretical foundations and advancements relevant to this work, covering deep learning methodologies such as convolutional neural networks and transformers in computer vision. It explores tasks like pose estimation and action recognition, including video- and skeleton-based approaches, and concludes with a survey of computer vision applications for non-human primates, focusing on great ape behavior recognition.

- Chapter 3 introduces *ASBAR* (*Animal Skeleton-Based Action Recognition*), a framework that integrates pose estimation and behavior recognition into a unified pipeline. Leveraging large public datasets, this chapter demonstrates the framework's effectiveness in classifying natural behaviors of great apes in the wild, achieving competitive accuracy while reducing data size and computational requirements.
- Chapter 4 presents *ChimpBehave*, a novel dataset of zoo-housed chimpanzee videos annotated for behavior recognition. This chapter investigates out-of-distribution generalization in cross-dataset settings, benchmarks video- and skeleton-based models, and highlights the superior performance of skeleton-based approaches under significant visual variability.
- Chapter 5 focuses on the automated recognition of great ape gestures, detailing the development and benchmarking of *FineChimp*, a dataset featuring multiview recordings and expert-level annotations across 38 gesture classes. This chapter demonstrates the use of state-of-the-art deep learning models for fine-grained action recognition.
- Chapter 6 concludes the dissertation by summarizing its contributions and discussing potential directions for future research.

Supplementary material, including images, tables, as well as theoretical and technical support, is provided in a series of appendices.

1.5 List of Publications

This dissertation has led to the following past or upcoming publications:

Fuchs, M., Genty, E., Zuberbühler, K., and Cotofrei, P. (2024c). ASBAR: an Animal Skeleton-Based Action Recognition framework. Recognizing great ape behaviors in the wild using pose estimation with domain adaptation. *eLife*, 13:RP97962

Fuchs, M., Genty, E., Bangerter, A., Zuberbühler, K., and Cotofrei, P. (2024a). From Forest to Zoo: Domain Adaptation in Animal Behavior Recognition for Great Apes with ChimpBehave. 4th Workshop on CV4Animals: Computer Vision for Animal Behavior Tracking and Modeling, In conjunction with CVPR 2024

Fuchs, M., Genty, E., Bangerter, A., Zuberbühler, K., Odobez, J.-M., and Cotofrei, P. (2024b). From Forest to Zoo: Great Ape Behavior Recognition with ChimpBehave. Currently under review with minor revisions for the International Journal of Computers Vision

Fuchs, M., Genty, E., Zuberbühler, K., and Cotofrei, P. (2025). Automated Recognition of Great Ape Gestures. Manuscript in preparation

As well as this online resource:

Genty, E. and Fuchs, M. (2023). *GApS: A Coding Scheme for Great Apes Signals in ELAN*. https://greatapesgestures.github.io

Chapter 2

Related Work

2.1 Deep Neural Networks

Most humans can easily identify whether a chimpanzee is walking, sitting, or climbing in a video. They can understand a scene by identifying distinctive elements (e.g., *chimpanzee, substrate, trees, background*), recognizing motion (*chimpanzee moving*), analyzing relationships (*chimpanzee moving on substrate*), and logically inferring an action based on the combination of these elements (*the chimpanzee is walking*). For humans, this is a trivial exercise, even if they have never seen that particular video before. For computers, however, the same task is extraordinarily complex. After all, computers only *understand* data as sequences of zeros and ones.

One way to enable a computer to differentiate between a walking or sitting chimpanzee is to use traditional (non-deep) machine learning approaches on hand-designed features. For instance, one could train a model, such as a Support Vector Machine [31], to map inputs (the hand-crafted features) to outputs (the behaviors to recognize). For example, features might include whether the chimpanzee's legs are moving, the angle formed between its spine and the ground, and which body parts are in contact with the substrate. Using these features, the model could predict that the chimpanzee is walking if its legs are moving and its spine forms an acute angle with the ground, or sitting if its bottom is in contact with the substrate. Increasing the number and complexity of these features—essentially the model's *representation*—would enhance the model's ability to recognize actions accurately.

However, such approaches have significant limitations. First, they require models to learn from extracted features rather than directly from videos, necessitating extensive human input (models do not inherently recognize "legs" or their movement). Second, as tasks become more complex—such as differentiating between approximately 40 subtle communicative

gestures—the process of selecting, defining, and annotating features becomes impractical and inefficient.

To address these challenges, deep learning has emerged as a powerful paradigm, introducing models and methods capable of learning complex visual patterns directly from images and videos with minimal human intervention. Since the advent of AlexNet [103], deep learning methods have outperformed traditional approaches based on hand-crafted features in computer vision tasks such as image classification (e.g., recognizing objects in an image). Similarly, in *action recognition* task—identifying which action (or behavior) is displayed in a video—deep learning methods have become the dominant approach.

In this new paradigm, models learn through experience by extracting meaningful patterns directly from raw data. Unlike traditional approaches, deep learning models do not simply map predefined features to outputs but also learn the representations themselves. These models build complex representations by combining simpler ones in a hierarchical, sequential manner. For instance, early layers in a model's architecture may learn to recognize edges, colors, and textures. These are then used to detect more complex elements, such as chimpanzee legs or arms, and finally, to determine whether the legs are moving to predict that the chimpanzee is walking.

Moreover, deep learning models consider millions of additional features beyond leg motion to predict actions in a video clip. They may account for variations in leg movement and visual perspectives (e.g., viewing a leg from the side, above, or partially occluded). Remarkably, this approach requires no explicit human programming of abstract concepts like "leg" or "movement." Instead, the model learns these (potential similar) representations during training to optimize the relationship between its inputs and outputs.

This dissertation focuses primarily on two key deep learning tasks: *pose estimation* (Sect. 2.2) and *action recognition* (Sect. 2.3). Both tasks are addressed within a *supervised learning* framework. Broadly, machine learning models can be categorized as supervised or unsupervised, depending on whether they have access to labeled output during training (other areas of learning, such as reinforcement learning, are beyond the scope of this work).

In this case, all learning occurs in a supervised manner, meaning the output data is fully annotated, enabling models to assess the accuracy of their predictions during training. For action recognition, this involves providing algorithms with labeled classes of actions represented in videos. For pose estimation, models are trained on images annotated with the (x, y) ground-truth coordinates of keypoints.

Understanding how deep neural networks learn appropriate representations from visual data to accurately predict animal poses or recognize behaviors requires a combination of mathematical, statistical, and programming concepts. The following sections provide an

overview of these concepts. We begin with a discussion of logistic regression (Sect. 2.1.1), then explore more sophisticated models, including Multilayer Perceptrons (Sect. 2.1.2), Convolutional Neural Networks (Sect. 2.1.3), Graph Convolutional Networks (Sect. 2.1.4), and Transformers (Sect. 2.1.5).

2.1.1 Logistic Regression - An Example

Deep learning belongs to the broader field of machine learning. In this section, we present fundamental theoretical concepts that explain how models can learn with supervision [9]. We illustrate these concepts using the example of a binary classification task, where the model is taught to predict whether an image contains a sitting ape or not.

Consider a dataset $\chi \in \mathbb{R}^{n_x \times m}$ with m examples (images), each composed of n_x features (pixels). Each example is labeled with either a 0 or 1, where $y = 1$ indicates that the image contains a sitting ape. During training, the model's goal is to learn associations between examples and their corresponding ground-truth labels, ideally generalizing this knowledge to previously unseen images. In other words, given an image $\boldsymbol{x} \in \mathbb{R}^{n_x}$, the model must estimate a probability distribution, so that its prediction is $\hat{y} = p(y = 1|\boldsymbol{x})$. To achieve this, we start with a linear regression model:

$$\hat{y} = \boldsymbol{w}^\intercal \boldsymbol{x} + b \tag{2.1}$$

where $\boldsymbol{w} \in \mathbb{R}^{n_x}$ are weights assigned to each pixel, and $b \in \mathbb{R}$ is a bias parameter. Together, $\boldsymbol{w}$ and b are the model parameters to be learned. Learning involves defining a *cost* (or *loss*) function to optimize. For linear regression, the Mean Squared Error (MSE) is commonly used:

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{m}||\hat{\boldsymbol{y}} - \boldsymbol{y}||_2^2 \tag{2.2}$$

This cost function calculates the average squared distances between the ground-truth labels and predictions. Optimizing the MSE means to find parameters $\boldsymbol{w}$ and b that result in the smallest error. The minimum occurs where the gradient of the cost function with respect to $\boldsymbol{w}$ is zero:

$$\nabla_{\boldsymbol{w}}\text{MSE} = \boldsymbol{0}. \tag{2.3}$$

In practice, this linear regression model would most likely fail at categorizing images in a classification task, the main reason being that it is purely linear and therefore cannot achieve

to find a viable solution for non-linear problems such as those of computer vision. A more sophisticated model in machine learning is the logistic regression, having the form:

$$\hat{y} = \sigma(\boldsymbol{w}^\intercal \boldsymbol{x} + b) \tag{2.4}$$

where

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.5}$$

is the sigmoid function mapping the input to the range $(0,1)$, representing probabilities. The cost function J, averaging the binary cross-entropy across all examples, is defined as:

$$J(\boldsymbol{w},b) = \frac{1}{m}\sum_{i=1}^{m}\mathscr{L}(\hat{y}^{(i)},y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(\hat{y}^{(i)}) + (1-y^{(i)})\log(1-\hat{y}^{(i)})] \tag{2.6}$$

where $\boldsymbol{w}$ and b are the learnable weights and parameters of the model, m the number of examples, $\hat{y}^{(i)}$ and $y^{(i)}$ respectively the prediction and ground-truth label for the i-th example.

The optimal values of $\boldsymbol{w}$ and b that minimize the cost function can be found using the *gradient descent* approach, an iterative algorithm that minimizes a function $f : \mathbb{R}^n \to \mathbb{R}$ by computing its gradient (the vector defined by partial derivatives) and taking small steps in the opposite direction of the gradient.

In practice, instead of computing gradients over the entire dataset (which is computationally expensive), models use *stochastic gradient descent* (SGD) approach, updating parameters iteratively using small, randomly selected mini-batches of training examples.

While logistic regression can be applied to simple binary tasks like detecting sitting apes, its capacity for real-world computer vision applications is limited. However, understanding logistic regression provides the foundation for extending these principles to deep learning models, which employ layers of linear transformations, non-linear activations, and much greater model depth to solve complex problems in computer vision.

2.1.2 Multilayer Perceptrons

Multilayer perceptrons (MLPs) are among the most fundamental types of neural networks and are ubiquitous in modern computer vision models. The primary goal of an MLP is to approximate a target function f^*. For instance, in an image classification task, the function $y = f^*(\boldsymbol{x})$ maps an input $\boldsymbol{x}$ (e.g., an image) to an output y (e.g., a class label). Here, the input $\boldsymbol{x} \in \mathbb{R}^n$ is a vector where each component x_i represents a feature of the input, such as pixel values of the image. In the context of image data, $\mathbb{R}^n$ is often expressed as $\mathbb{R}^{w\times h\times c}$, where w, h, and c denote the image width, height, and number of channels, respectively.

An MLP can be expressed as $\boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{\theta})$, where f is the approximation function of the MLP, and $\boldsymbol{\theta}$ represents the set of learnable parameters in the network (i.e., the weights and biases of the network).

MLPs, as their name suggests, consist of multiple layers of perceptrons (or artificial neurons) arranged sequentially. This design is loosely inspired by biological neurons in the brain. The first layer of an MLP is called the *input layer*, the final layer is the *output layer*, and the layers in between are referred to as *hidden layers*. MLPs are *feedforward* networks, meaning the input signal flows in one direction—from the input layer, through the hidden layers, to the output layer—without any backward connections.

MLPs are also *fully connected* networks, meaning every perceptron in layer l is connected to all perceptrons in the preceding layer $l-1$ and the following layer $l+1$, with each connection weighted by a scalar.

2.1.2.1 Perceptrons

A perceptron is a simple computational unit that maps an input vector to a scalar output. Each perceptron performs two basic operations:

1. **Linear Transformation:** The perceptron computes the dot product of its input vector $\boldsymbol{x}$ with the corresponding weights $\boldsymbol{w}$ and adds a bias term b:

$$z = \boldsymbol{w}^{\intercal}\boldsymbol{x} + b. \tag{2.7}$$

2. **Non-linear Activation:** The resulting scalar z is passed through a non-linear activation function σ, which introduces non-linearity and enables the network to learn complex patterns. Typical activation functions include the sigmoid, tanh, and ReLU functions [1].

For a layer of perceptrons, the forward pass computes the output vector $\boldsymbol{x}^{(l+1)}$ for the next layer:

$$\boldsymbol{x}^{(l+1)} = \sigma(\boldsymbol{x}^{(l)} \cdot \boldsymbol{w}^{(l)} + \boldsymbol{b}^{(l)}), \tag{2.8}$$

where $\boldsymbol{x}^{(l)}$ represents the input to layer l, $\boldsymbol{w}^{(l)}$ and $\boldsymbol{b}^{(l)}$ are the weights and biases of the layer, and σ is the activation function.

2.1.2.2 Architectural Considerations

The design of an MLP depends on the complexity of the task and the dataset. Determining the optimal number of hidden layers and the number of perceptrons in each layer remains an active area of research.

In regression tasks, the output layer typically consists of a single perceptron, whose output corresponds to the predicted value of the dependent variable. For classification tasks, the output layer usually contains as many perceptrons as there are classes. In this case, the output values are often passed through a final softmax function to produce a probability distribution over the K classes:

$$\text{softmax}(\boldsymbol{x})_i = \frac{exp(x_i)}{\sum_{k=1}^{K} exp(x_k)} \tag{2.9}$$

2.1.2.3 Training MLPs

The process by which an MLP learns follows the same principles outlined in Section 2.1.1. During training, mini-batches of examples are passed through the network in a forward pass, producing predictions and computing the average cost (loss) over the mini-batch.

The gradients of the cost function with respect to the model parameters are then calculated in a backward pass using the chain rule. This process, known as *backpropagation*, computes how each parameter contributes to the error and updates the weights and biases accordingly using stochastic gradient descent.

Formally, the parameters are updated as follows:

$$\boldsymbol{w} := \boldsymbol{w} - \alpha \frac{\partial J}{\partial \boldsymbol{w}}, \quad \boldsymbol{b} := \boldsymbol{b} - \alpha \frac{\partial J}{\partial \boldsymbol{b}}, \tag{2.10}$$

where α is the learning rate, J is the cost function, and $\frac{\partial J}{\partial \boldsymbol{w}}$ and $\frac{\partial J}{\partial \boldsymbol{b}}$ are the gradients with respect to $\boldsymbol{w}$ and $\boldsymbol{b}$.

2.1.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been one of the most dominant architectures in computer vision research in the last decade. However, the core principle of using convolutions on images dates back much further, first described in the late 1980s in [110]. This seminal work demonstrated that 2D convolutions could effectively automate the recognition of handwritten digits. At the time, the method was computationally expensive, and datasets were relatively small.

More than 30 years later, CNNs reshaped the field with the introduction of AlexNet in 2012 [103]. At a time when state-of-the-art methods for image classification relied primarily on traditional machine learning or MLPs, AlexNet achieved a groundbreaking performance in a challenge to classify objects across 1000 categories. It outperformed the second-best model

by more than 8 percentage points, creating a shockwave in the computer vision community. The original paper has been cited over 136,000 times to date.

This revolution was driven by CNNs' ability to drastically reduce the number of connections and parameters compared to MLPs of similar depth, coupled with their efficient training on GPUs, leveraging high parallelization performance.

2.1.3.1 Building Blocks of CNNs

CNNs primarily consist of two fundamental building blocks: convolutional layers and pooling layers. These are based on simple mathematical operations:

1. **Convolutional Layers:** In 2D CNNs, a 2D input signal $I \in \mathbb{R}^{h \times w}$ (e.g., an image) with a grid topology is transformed by sliding a 2D kernel $K \in \mathbb{R}^{m \times n}$ over it. For each overlapping region between the input and the kernel, the element-wise dot product is computed and stored as an element of the output, called the *feature map* $S \in \mathbb{R}^{h_s \times w_s}$. Mathematically, this can be expressed as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n) K(i-m, j-n) \tag{2.11}$$

   In practice, I and K are often square matrices, where $h = w$ and $n = m$. The input I can be an image (in the input layer) or a feature map (in hidden layers). The dimensions of S depend on the input size I and several hyperparameters: kernel size n, stride s, and zero-padding p. The output dimensions are calculated as:

$$h_s = w_s = \left\lfloor \frac{h + 2p - n}{s} + 1 \right\rfloor \tag{2.12}$$

   After generating the feature map, a bias is added, and the result is passed through a non-linear activation function, such as ReLU.

   Kernels can enhance various visual features, such as edges or textures. However, unlike traditional image processing, CNNs learn these kernel parameters during training via backpropagation, optimizing them for the specific task. Intuitively, the network discovers its own image processing techniques that best enable it to solve the task at hand. The number of kernels in a convolutional layer is a hyperparameter that determines the depth of the output feature map. For instance, processing a grayscale 224×224 image with 5 kernels that preserve resolution results in a feature map of dimensions $224 \times 224 \times 5$.

2. **Pooling Layers:** Pooling layers, typically applied after the non-linear activation of convolutional layers, reduce the dimensions of the feature map. The most common pooling method is max pooling, which retains only the maximum value in a given region of pixels. Pooling has two main benefits: (i) *Dimension reduction* - reducing the size of the feature map decreases the computational cost of subsequent layers; (ii) *Translation invariance* - the representation becomes less sensitive to small translations in the input, enhancing robustness.

2.1.3.2 Structure of CNNs

Modern CNN architectures for image classification often consist of multiple convolutional and pooling layers, followed by a few fully connected layers. The number of kernels typically increases with each successive convolutional layer, creating deeper feature maps that capture increasingly abstract representations of the input. The final representation is passed through a softmax function to produce a probability distribution over the target classes.

2.1.3.3 Motivation of CNNs

The efficacy of CNNs is rooted in three key principles:

1. *Sparse Connections:* CNNs use significantly fewer parameters than MLPs without compromising performance. For example, consider a medium-resolution RGB image of 1000×1000 pixels (1 megapixel, or 3 million features with 3 color channels). A fully connected MLP with 1000 perceptrons in the first layer would require 3 billion parameters in this layer alone. In contrast, using 64 kernels of size 7×7 in a CNN would result in fewer than 10,000 trainable parameters in the first layer [75]. This drastic reduction makes CNNs far more feasible for real-world applications.

2. *Parameter Sharing:* Unlike MLPs, where each input feature has a unique weight, CNNs share kernel weights across the input through the sliding window operation. This significantly reduces memory requirements and improves statistical efficiency.

3. *Equivariance:* CNNs preserve the spatial structure of the input. For example, if a chimpanzee in an image shifts position, the feature map representation of the chimpanzee shifts equivalently. This property is particularly advantageous for detecting patterns that appear across different regions of the image, such as edges, faces, or body parts, using the same set of kernel parameters.

2.1.3.4 Extension to Videos

CNNs have also been extended to process three-dimensional inputs, such as videos. The pioneering work in [88] introduced 3D convolutions, enabling kernels to slide not only across spatial dimensions (width and height) but also across time. This adaptation captures the *spatio-temporal* representation of video volumes.

Where 2D kernels focus on spatial relationships within a single frame, 3D kernels operate on pixel regions across consecutive frames, effectively capturing motion and temporal patterns. This approach has inspired numerous advancements in action recognition [23, 199, 43] (see Sect. 2.3.2 for more details).

2.1.4 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) adapt the principles of traditional image-based CNNs [103] for graph-structured data. In their seminal work, [101] applied GCNs to graph data in a semi-supervised manner, targeting the classification of unlabeled nodes in partially labeled graphs.

A graph $G(V,E)$ is defined by its set of N nodes $v_i \in V$ and undirected edges $(v_i, v_j) \in E$. The task of classifying missing node labels is achieved by minimizing the following loss function:

$$\mathscr{L} = \mathscr{L}_0 + \lambda \mathscr{L}_{reg} \tag{2.13}$$

where $\mathscr{L}_0$ is the supervised loss, responsible for label prediction on the labeled nodes, and $\mathscr{L}reg$ is a graph Laplacian regularization term, defined as:

$$\mathscr{L}_{reg} = \sum_{i,j} A_{ij} \| f(\boldsymbol{X}_i) - f(\boldsymbol{X}_j) \|^2 = f(\boldsymbol{X})^T \Delta f(\boldsymbol{X}) \tag{2.14}$$

The regularization term $\mathscr{L}_{reg}$ enforces the assumption that connected nodes in the graph are likely to share similar labels. For each node i, its feature vector $\boldsymbol{X}_i$ is passed through a differentiable function f, such as a neural network with learnable parameters $\boldsymbol{w}$, to predict its label $f(\boldsymbol{X}_i)$. The squared distances between the predicted labels of connected nodes are then summed, weighted by the adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$. The entries of $\boldsymbol{A}$ define the connections between nodes, which can be binary or weighted.

In vectorized form, this computation applies $f(\cdot)$ to the matrix $\boldsymbol{X}$ (the concatenation of all node features). The Laplacian matrix Δ, handling the regularization, is defined as:

$$\Delta = \boldsymbol{D} - \boldsymbol{A}, \tag{2.15}$$

where $D_{ii} = \sum_j A_{ij}$ is the degree matrix, whose diagonal entries count the neighbors of each node. The hyperparameter λ balances the importance of the supervised loss $\mathscr{L}_0$ and the regularization term $\mathscr{L}reg$.

2.1.4.1 Applications of GCNs in Action Recognition

While originally applied to node classification in citation networks, GCNs have demonstrated their versatility and can be generalized to any data represented as graphs, including human skeletons. This flexibility makes them particularly relevant to skeleton-based action recognition (discussed in Sect. 2.3.3), where joints and their connections can naturally be modeled as graph structures.

2.1.5 Transformers

In their seminal 2017 paper "Attention Is All You Need" [202], the authors introduced the Transformer, a novel architecture that has since become foundational across diverse fields of deep learning. As of today, this paper has been cited nearly 145,000 times. Transformer models underlie many transformative AI technologies, including Large Language Models (LLMs) like the GPT family [18], which are revolutionizing human-AI interaction.

Initially developed for machine translation in Natural Language Processing (NLP), Transformers have since been adapted for computer vision applications involving images [38] and videos [196]. Today, Transformer-based models consistently outperform traditional CNN-based methods in computer vision tasks such as image classification [38] and action recognition [206].

2.1.5.1 In Natural Language Processing

At the core of the Transformer are several key data transformations. To illustrate, we consider GPT-3 [18], a prominent autoregressive LLM. GPT-3 predicts the next most likely word in a sequence of text. For example, given the input "*The capital city to study primatology in Switzerland is*," a well-trained model might output "*Neuchâtel*."

Tokenization and embeddings: Textual input consists of sentences of varying lengths. The first step involves breaking sentences into smaller units, called *tokens* (e.g., words, subwords, or punctuation marks). In GPT-3, the vocabulary consists of approximately 50,000 tokens. Each token is then *embedded* into a high-dimensional vector space through a linear transformation. This embedding process retains semantic similarity, mapping tokens with similar meanings closer together in the embedding space.

In GPT-3, the embedding space has a dimensionality of 12,288. To preserve positional information, a positional encoding vector is added to each token embedding. These positional encodings, often derived from sine and cosine functions of different frequencies [202], inform the network about the token's position in the sequence. The resulting embeddings serve as inputs to the model's encoder.

Encoder: The encoder processes the sequence of token embeddings, transforming them into contextualized representations. Each token exchanges information with others, updating its meaning based on the sentence's context. For instance, the token "capital" in "*The capital city to study primatology in Switzerland is*" is associated with "most important place," but its meaning would differ in other contexts, such as "*The initial capital was invested over 10 years*."

To achieve this, the encoder comprises several *attention blocks*, each containing multiple *attention heads*. These attention mechanisms determine which tokens are most relevant to each other. For each token embedding $\boldsymbol{e}_i \in \mathbb{R}^{12,288}$, three vectors are computed: (i) a query vector $\boldsymbol{q}_i = Q\boldsymbol{e}_i$, where $Q \in \mathbb{R}^{128\times 12,288}$; (ii) a key vector $\boldsymbol{k}_i = K\boldsymbol{e}_i$, where $K \in \mathbb{R}^{128\times 12,288}$; (iii) a value vector $\boldsymbol{v}_i = V\boldsymbol{e}_i$, where $V \in \mathbb{R}^{12,288\times 12,288}$. For a sequence of N tokens, pairwise dot products between query and key vectors produce an attention matrix $\mathbb{R}^{N\times N}$. Each element represents the relevance between tokens. The resulting scores are normalized using a softmax function (Eq. 2.9), and these normalized weights are used to compute weighted sums of the value vectors.

The attention mechanism can be compactly expressed as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.16}$$

where d_k is the dimensionality of the query and key vectors. Scaling by $\sqrt{d_k}$ stabilizes gradients during training.

A single attention block consists of multiple attention heads operating in parallel. For example, GPT-3 uses 96 heads per block, each with its own Q, K, and V matrices. The encoder itself stacks 96 such blocks, interleaved with MLP layers (Sect. 2.1.2) and residual connections [75].

Model output: The model predicts the next token by passing the final embedding through a linear transformation (unembedding) and a softmax function. This produces a probability distribution over the vocabulary, identifying the most likely next token.

Self-Supervised Pre-training: GPT-3 is trained on approximately 500 billion tokens using a self-supervised approach. During training, the model predicts the next word in a sequence (e.g., "*The capital city to study primatology in Switzerland is*" $\rightarrow$ "*Neuchâtel*"). To maxi-

mize training data usage, sentences are repeatedly truncated and used as input, predicting progressively earlier tokens.

Another approach, used by models like BERT [97], involves masking random tokens within a sequence and training the model to predict these masked tokens. BERT also employs a class token ([CLS]) to represent the sequence as a whole, facilitating tasks like sentiment analysis or question answering.

2.1.5.2 In Computer Vision

VisionTransformer: Among numerous efforts to adapt Transformers for image classification in computer vision, one of the most successful implementation remains Vision Transformer (ViT) [38], which follows the original Transformer architecture with "the fewest possible modifications." The authors introduced only a few key adjustments, primarily focusing on how images are embedded and how the encoder produces a classification output.

To address the embedding of images, ViT divides each image into a sequence of patches, typically 16×16 pixels in size. Each patch is flattened into a vector of 256 entries and embedded into a high-dimensional space through a linear transformation. To enable class prediction, a classification embedding is prepended to the sequence at position 0. This classification token's hidden state, after the encoder's forward pass, serves as input to a final classification MLP. This modification is inspired by the class token introduced in BERT, as described in the previous section.

As with many large Transformer architectures, the key to ViT's success lies in pre-training on large datasets, followed by fine-tuning on task-specific datasets. Training on ImageNet alone (1.3 million images in 1,000 classes [35]) is insufficient for ViT to outperform CNN-based models like ResNets. This limitation is likely due to Transformers' lack of inductive biases such as translation equivariance, which is inherent to CNNs. However, when pre-trained on larger datasets, such as JFT-300M (303 million images across 18,000 classes [187]), and fine-tuned on smaller datasets, ViT consistently outperforms ResNets on several image classification benchmarks.

Masked Autoencoders (MAE): Similar to the application of masked autoencoders in NLP (e.g., BERT), Vision Transformers can benefit from self-supervised learning using masked autoencoding. Masked Autoencoders (MAE) [73] enable more efficient training and improved performance on smaller datasets like ImageNet. MAE employs an encoder-decoder architecture where a large portion of image patches (e.g., 75%) is randomly masked during pre-training.

Only the unmasked patches are processed by the encoder, significantly reducing memory and computation requirements. The decoder then reconstructs the original image by

processing a sequence of tokens composed of encoded visible patches and placeholders for masked patches, with positional embeddings added to maintain spatial context. The decoder outputs reconstructed pixel values for the masked patches, and the reconstruction error is measured using a Mean Squared Error (MSE) loss. Importantly, the decoder is only used during pre-training, as image recognition tasks rely solely on the encoder.

Extension to Video Understanding: The success of MAEs in image-related tasks quickly extended to video understanding. Similar to their application to 2D image patches, Vision Transformers can be pre-trained in a self-supervised manner by masking 3D patches in video data [45, 196]. Masking ratios as high as 90% leverage the spatiotemporal redundancy inherent in videos, leading to significant reductions in training time.

The most recent advancement, VideoMAEv2 [206], achieves state-of-the-art performance by scaling both the model and dataset sizes. The architecture incorporates ViT-g as the backbone, an enhanced version of ViT with billion-level parameters [230]. Effective pre-training is achieved using a dual-masking strategy, masking video patches in both the encoder and the decoder. This pre-training process leverages a dataset of 1.35 million videos in a self-supervised manner, further pushing the boundaries of video understanding.

2.2 Pose Estimation

2.2.1 Task Definition

Body pose estimation refers to the task of predicting the coordinates of anatomical keypoints, such as joints and bones, from visual data [21]. For 2D images, these keypoints are typically inferred as coordinates in the image plane. Figure 2.1 illustrates how standard networks predict keypoint positions, such as limbs. Some frameworks extend this capability to predict depth information, enabling 3D keypoint localization [12, 137]. In practice, more robust 3D annotations can be obtained using markers or depth-sensor cameras, which are commonly applied in 3D pose estimation tasks.

The set of keypoints is often defined using an anatomically simplified structure that best represents the skeletal configuration of the individual. However, the choice of annotated landmarks can vary depending on the dataset used. For example, in human pose estimation, the NTU-RGB-D 120 dataset [120], widely used for 3D body pose annotations, includes 25 keypoints. In contrast, the framework introduced in [21] annotates a more compact set of 18 keypoints.

Pose estimation is a critical prerequisite for skeleton-based action recognition. Typically, each RGB frame of a video clip undergoes preprocessing to extract a set of (x, y) keypoint

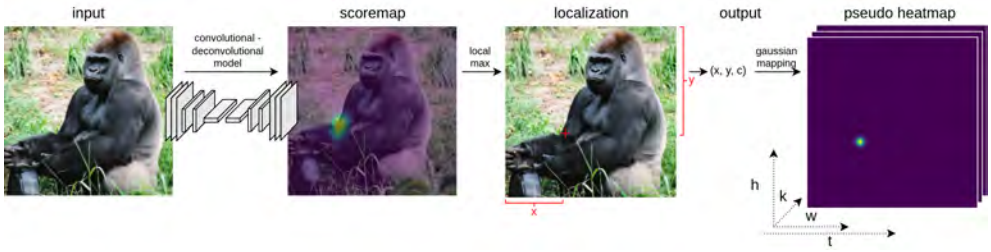

Fig. 2.1 **From RGB image to pseudo-heatmaps.** The transformation of an RGB image into a pseudo-heatmap. The input image is processed by a Conv-Deconv architecture to generate a probabilistic scoremap for each keypoint location (e.g., the right elbow). By identifying the local maxima in the scoremap, the keypoint coordinates and confidence values can be extracted. A Gaussian transformation is then applied to produce pseudo-heatmaps, which serve as input to the subsequent behavior recognition model.

coordinates, along with a corresponding network confidence score c (Figure 2.1). This transformation is generally performed using supervised CNNs trained specifically for keypoint detection.

The field of pose estimation has seen significant advancements in computer vision over the past decade, driven by extensive research efforts. Following the categorization introduced in [235], pose estimation methods can be broadly classified as single or muli-individual approaches.

2.2.2 Single-Individual Pose Estimation

Single-individual pose estimation assumes that only the coordinates of one individual need to be predicted in an image. If multiple individuals are present, their corresponding bounding box coordinates must be known beforehand. Approaches to this task generally fall into two categories: regression-based and heatmap-based methods, distinguished by their respective data pipelines.

2.2.2.1 Regression-Based Methods

The seminal work by Toshev et al. [197] introduced pose estimation using regression with CNNs. In this approach, the network architecture is similar to AlexNet [103], but instead of predicting a classification label, the model outputs a pose vector: $\hat{y} \in \mathbb{R}^{2\times k}$ where k is the number of keypoints. This vector contains two values per keypoint corresponding to the x and y-coordinates. The network parameters are learned, for example, by minimizing the L_2

loss (or Mean Squared Error, see Eq. 2.2) between the predicted pose and its ground-truth counterpart.

2.2.2.2 Heatmap-Based Methods

In contrast, heatmap-based methods do not directly regress keypoint coordinates. Instead, the model predicts k heatmaps (one per keypoint), where each heatmap has the same spatial dimensions as the input image [211]. Each pixel in a heatmap represents the probability of that pixel containing the corresponding keypoint.

Ground-truth heatmaps are generated using a Gaussian transformation, where the keypoint location is represented as a Gaussian peak centered at the corresponding coordinates The model parameters are then learned by minimizing the discrepancy (e.g., using L_2 loss) between the predicted and ground-truth heatmaps.

The primary advantage of heatmap-based methods is their ability to preserve spatial information throughout the pipeline. However, this comes at the cost of increased computational requirements and a larger number of trainable parameters.

Model Architecture: To produce heatmaps, models typically pass the input signal through a series of convolutional and pooling layers, which progressively downsample the spatial dimensions while increasing feature depth. Subsequently, the signal is upscaled to its original resolution using deconvolutional layers (the reverse operation of convolution) in a symmetric fashion [205]. This process transforms the representation from a high-to-low-to-high resolution, where maintaining spatial accuracy at lower resolutions is crucial.

HRNet: To address this challenge, HRNet (High-Resolution Network) [189] proposes maintaining a high-resolution representation throughout the forward pass. HRNet achieves this by combining multiple parallel subnetworks at varying resolutions, ensuring that spatial details are preserved without significant loss of accuracy. HRNet remains a state-of-the-art architecture for pose estimation, with subsequent improvements addressing specific limitations. For instance, Lite-HRNet [225] optimized the network to reduce computational costs and HigherHRNet [28] was designed to improve performance on individuals of varying scales and sizes. A variant of the HRNet model is employed in Chapter 4 of this dissertation.

2.2.3 Multi-Individual Pose Estimation

The task of single-individual pose estimation can be extended to multi-individual pose estimation, where the goal is to identify keypoints for multiple individuals present in an image. This introduces additional challenges, as the model must determine which detected keypoints belong to which individuals without access to predefined bounding boxes. Existing

methods can generally be categorized as either *top-down* or *bottom-up*, depending on the structure of their data pipeline.

2.2.3.1 Top-Down Approaches

In the top-down paradigm, models first detect the location of individuals, typically by predicting their bounding box coordinates. Subsequently, single-individual pose estimation techniques are applied to each detected individual independently to infer their keypoint locations.

2.2.3.2 Bottom-Up Approaches

In contrast, bottom-up approaches begin by detecting all keypoint candidates within an image. A subsequent step, often referred to as assembly, determines which keypoints belong to the same individual. This paradigm has inspired notable advancements, including DeepCut [159] and DeeperCut [86], which eventually led to the development of DeepLabCut [133, 147, 108], a widely adopted framework for multi-animal pose estimation.

DeeperCut employs a ResNet [75] as a feature extractor to generate deep visual representations of an input image. This is followed by a series of deconvolutional layers that progressively upsample the signal to recover the original image resolution. Inspired by semantic segmentation techniques [208], the network outputs keypoint probabilistic scoremaps—pixel-wise probabilities indicating the location of specific keypoints (see Fig. 2.1).

The process can be summarized as follows:

- For each keypoint, the local maxima of the scoremap are identified using the *argmax* transformation, yielding the predicted coordinates (x, y) and confidence c.
- During training, target scoremaps are generated by assigning a probability of 1 to all pixels within a specified distance of the ground-truth (x, y) coordinates and 0 elsewhere.
- The model parameters are optimized by minimizing the cross-entropy loss between the predicted scoremap and the target scoremap using stochastic gradient descent.

DeepLabCut (DLC) extends the principles of DeeperCut to multi-animal, markerless pose estimation. This open-source framework is specifically designed for detecting the body parts of various animal species, providing tools for pose estimation, tracking, and identification. DLC is one of the first toolboxes to adapt human pose estimation advances for animal research.

For keypoint detection, DLC allows the selection of various CNN backbones, including ResNet [75] and EfficientNet [192], with multiple depth options to balance accuracy and

computational efficiency. DLC's widespread adoption can be attributed to its high detection accuracy, even for complex animal poses, the availability of a large collection of pre-trained models for various animal species, and for having an active and engaged user community, which has contributed to its continuous improvement and versatility.

DLC has become a standard tool for researchers across diverse fields, including neuroscience, ecology, and animal behavior studies (e.g., [214, 70, 231]). The ASBAR framework proposed in Chapter 3 integrates DLC in its pose estimation module, leveraging its robust and well-established architecture for animal behavior research.

2.3 Action Recognition

2.3.1 Definition

According to Aggarwal's nomenclature [2], human activities can be categorized into four hierarchical levels:

- *Gestures*: Elementary movements (e. g., "raising a hand" or "tilting the head")
- *Actions*: Sequences of chronologically organized gestures (e. g., "kicking" or "eating")
- *Interactions*: Activities involving at least two individuals and/or objects (e.g., "shaking hands" or "grabbing a bottle")
- *Group activities*: Activities performed by conceptually defined groups of multiple individuals and/or objects (e. g., "two teams playing volleyball" or "boats sailing.")

This nomenclature can be extended to non-human beings, by adapting the corresponding classes of activities.

The goal of action recognition is to classify an action from an unseen video into its correct activity category. Formally, it is a classification task in which a model learns a function: $f : \mathbb{R}^n \rightarrow \{1, ..., k\}$ where the input is a video clip of dimensions $n = w \times h \times t \times c$ (width, height, time, and channels), and the output corresponds to one of k predefined classes (e.g., 1= sitting, 2=walking).

2.3.2 Video-Based Action Recognition

Action recognition approaches can be categorized in numerous ways (e.g., sequential, syntactic, description-based). This section focuses on models that rely solely on space-time volumes as input.

A video consists of a sequence of 2D images (frames) arranged chronologically. Each frame contains data projected from a 3D real-world scene onto the camera's image plane, stored as a discretized grid of pixels of dimensions $w \times h$ (ignoring RGB channels). By concatenating t frames, a video clip can be represented as a 3D space-time volume with dimensions $w \times h \times t$.

Action recognition, therefore, requires capturing and understanding both *spatial features* within individual frames, and *temporal associations* across successive frames. While the challenge of spatial feature extraction has been addressed successfully in tasks such as image classification and object detection—thanks to deep CNNs [103] — integrating temporal information remains an active area of research.

2.3.2.1 Trends in Video-Based Action Recognition

According to Zhu et al. [237], deep learning approaches for action recognition on videos have evolved along three main trends:

1. *Two-Stream Networks*: Introduced in [181], the Two-Stream Network architecture incorporates a dual-path design. One CNN processes the spatial information from video frames, while a second CNN simultaneously processes the optical flow stream to capture motion. The predictions from both streams are fused to produce the final action classification.

2. *3D Convolutional Networks*: This trend extends the principle of 2D convolution to the temporal dimension using 3D convolutional kernels [88]. These kernels simultaneously capture spatial and temporal information across the video. Early limitations included insufficient computational resources and the lack of large-scale datasets required to train such deep architectures effectively [23, 199].

3. *Efficiency and Large-Scale Training*: further developments have emphasized computational efficiency, training on larger datasets, and deployment in real-world applications [43, 118, 236]. These efforts aim to improve model scalability, robustness, and generalization to practical scenarios.

2.3.2.2 The Rise of Video Transformers

In recent years, the focus has shifted from CNN-based architectures to video transformers for action recognition. Video transformers have demonstrated superior performance on benchmark datasets, outperforming CNN-based models in many cases [206]. This trend

highlights the growing importance of attention mechanisms and large-scale pretraining strategies in modeling spatiotemporal patterns within videos.

2.3.3 Skeleton-based Action Recognition

Skeleton-based action recognition involves identifying actions based on a sequence of skeletal joint data, represented as coordinate lists. These coordinates can either be captured by sensors (e.g., depth cameras or motion capture systems) or extracted via markerless pose estimation models.

The primary advantage of using pre-identified body coordinates for action recognition is the significant reduction in the input space dimensionality. For instance, a traditional Full HD image of dimensions 1920×1080 pixels contains over 12 million input features. In contrast, skeleton-based models operate on just a few dozen keypoints, dramatically reducing model complexity, number of trainable parameters and overall computational requirements. Additionally, joint trajectories are inherently robust to changes in illumination, background clutter, or variations in the scene, which makes skeleton-based methods particularly appealing for real-world applications [219], especially in primatology.

In contrast, the primary limitation of skeleton-based approaches lies in the need to pre-identify the pose of individuals. This additional processing step introduces dependency on pose estimation accuracy, which can be challenging, especially for non-human subjects. For human-centric action recognition, this challenge is largely mitigated by the availability of open-source pose estimation models, such as OpenPose [21], which effectively reduce the prior workload, or Large-scale datasets with accurate joint annotations obtained from depth sensors, such as NTU-RGB-D 120 [120]. However, achieving similar performance for non-human subjects, such as great apes, requires further research and methodological adaptations.

Skeleton-based action recognition approaches can generally be categorized either as GCN-based or CNN-based methods. GCN-based methods represent skeletal data as graphs, where body joints serve as vertices and bones (connections between joints) act as edges. Action recognition is performed by learning effective graph representations. An overview of the spatio-temporal GCN model is provided in section 2.3.3.1, while others GCN-based approaches, widely used in the literature, may be found in Appendix A.

Conversely, CNN-based methods are an alternative research direction, which applies CNNs directly on images of detected keypoints, treating the keypoints as spatial features. This method offers a straightforward and computationally efficient pipeline while leveraging the power of CNN architectures. This approach, particularly the PoseConv3D model [39] (see Sect. 2.3.3.2), is instrumental in our experiments described in Chapters 3 and 4.

2.3.3.1 Spatial Temporal GCN

A major advancement in human action recognition was led by the adaptation of Graph Convolutions Networks (GCNs) to the graph-structured data of the human skeleton.

GCNs originally introduced by [101] (see section 2.1.4 for details) were first adapted in [219] to the human skeleton to infer an action category with performance exceeding those of prior techniques. Their work focuses on the interpretation of body parts movements, seen as small local groups of joints. A spatial temporal graph $G = (V, E)$ is constructed, in which the nodes are human skeleton joints and the edges represent, on one hand, their natural relationships in each frame and, on the other hand, their temporal correspondence through all frames. The graph therefore consists of a set of joints $V = \{v_{ti} | \, t = 1, \ldots, T, i = 1, \ldots, N\}$ corresponding to N keypoints over T frames, and a set of edges E, made from two subsets $E_S = \{v_{ti}v_{tj} | (i, j) \in H\}$ with H being the set of anatomical body joints and $E_F = \{v_{ti}v_{(t+1)i}\}$ their temporal connection between frames.

Each node's features vector is made of its coordinates and the estimation confidence of the i-th joint on frame t. Through multiple neural layers, the input feature vector is transformed in higher-level feature maps, on which a softmax classifier can predict the action as the model's final output. Forward propagation is similar to [101] and the model's parameters are trained by backpropagation using stochastic gradient descent.

For each pixel x_i in a 2D image input, traditional image CNNs rely on the definition of its surrounding pixels and the layer-specific weights for their features' linear projection. Similarly, a GCN also needs to identify nodes' neighborhood and layer-specific weights. Thus, to extend this technique to a skeleton-based approach, the node's spatio-temporal neighborhood can be seen as the subset of all nodes $\{v_{qj}\}$, whose spatial distance from the focal node v_{ti} at time t, i. e. $d(v_{tj}, v_{ti})$, is smaller or equal to a parameter K and whose temporal distance, i. e. $|q-t|$, is smaller or equal to $\lfloor \Gamma/2 \rfloor$ (where K and Γ are predefined parameters). The absolute value of the temporal distance enables nodes from previous frames to be considered as part of the neighborhood $(q < t)$.

To compute the layer-wise linear projection of the features, nodes' label-specific weights are introduced. Each node v_{qj} of the neighborhood $B(v_{ti})$ is labeled according to the following mapping:

$$l_{ST}(v_{qj}) = l_{ti}(v_{tj}) + (q - t + \lfloor \Gamma/2 \rfloor) \times K$$

where the first term, $l_{ti} : B(v_{ti}) \rightarrow \{0, \ldots, K-1\}$, is the spatial, frame-dependent mapping, and the second term, $(q - t + \lfloor \Gamma/2 \rfloor) \times K$ is the temporal mapping. The overall label mapping allows the weights to be trained in a label-specific way; intuitively, changes in human joints

that are spatially and temporally close must have different predictive weights than ones that are further apart.

Training occurs in a similar fashion to that of [101], while the layer output features is the addition of the linear projections of each label. The model achieves higher performance than previous works on Kinetics [96] and NTU-RGB+D datasets [175].

2.3.3.2 CNN-based methods

More recent model architectures, such as PoseConv3D in [39], have demonstrated superior performance when applying 3D-CNNs to pose estimated data rather than GCNs. Particularly in the context of animal behavior recognition, this approach is more suitable, as it significantly outperforms previous GCN-based architectures in differentiating between subtly different actions (such as in the case of FineGym [176]) and is more robust with noisy pose data. Furthermore, PoseConv3D can deal with multi-individual settings without additional computation expense (where GCN techniques see their number of trainable parameters and FLOPs increase linearly with each additional individual), generalizes better in cross-dataset setting, and can easily integrate dual-modality of pose and RGB data.

In comparison with GCN approaches, this type of architecture uses pose data to create 3D heatmap volumes instead of graphs. From a set of pose coordinates (x_k, y_k, c_k) corresponding to the (x, y) coordinates and c confidence of the k-th keypoint in a frame of size $H \times W$, a heatmap J can be generated by applying the following Gaussian transformation:

$$J_{kij} = e^{-\frac{(i-x_k)^2+(j-y_k)^2}{2\sigma^2}} \cdot c_k \tag{2.17}$$

where i, j refer to the pixel frame coordinates and σ is the variance of the Gaussian map. For each frame, a total of K heatmaps are produced. After transforming all T frames from the sample (i.e., video clips), all generated heatmaps are stacked in a 3D volume of size $K \times T \times H \times W$. This data can then be used to train an adapted video-based action recognition 3D-CNN model such as [199, 23, 43] in a supervised manner using stochastic gradient descent (see Fig. 2.2).

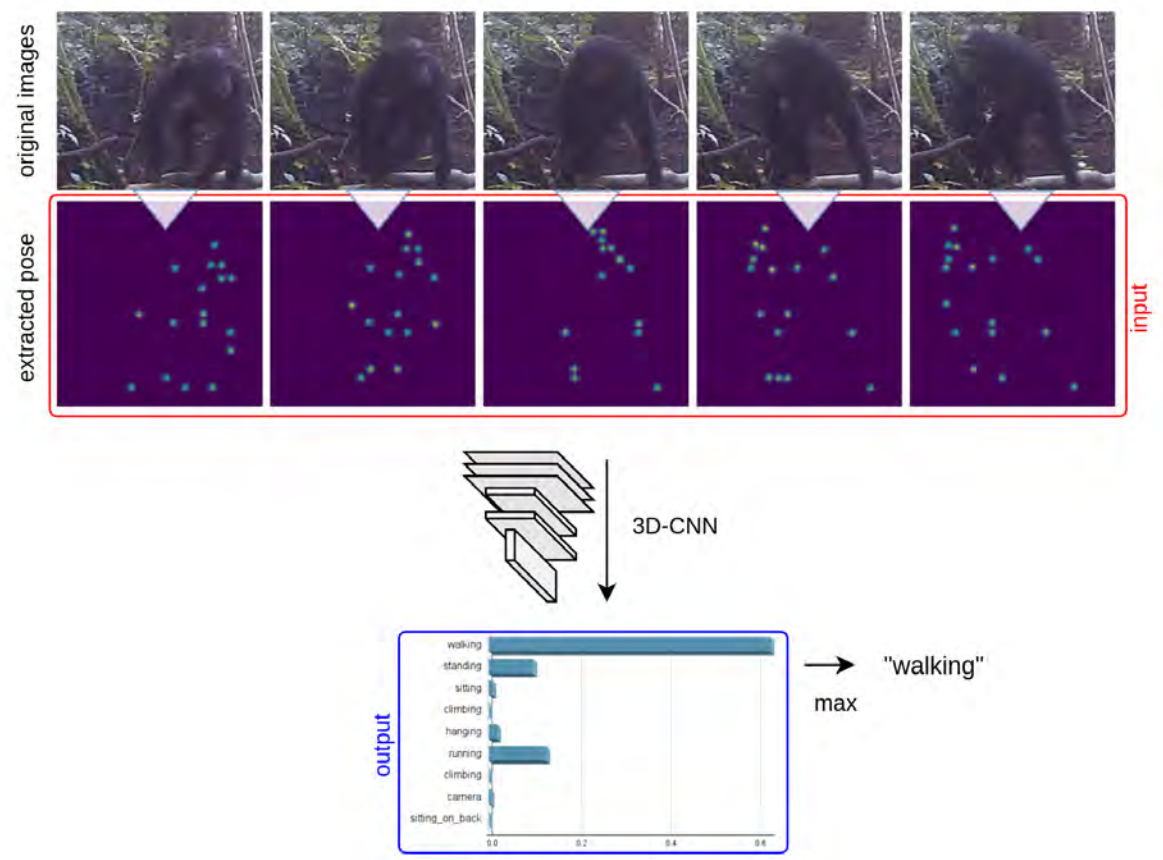

Fig. 2.2 **From extracted poses to behavior classification.** From a set of consecutive RGB frames, the animal pose is extracted, transformed into pseudo-heatmaps, and stacked as input of the behavior recognition model. A 3D-CNN is trained to classify the represented action into the correct behavior category (e.g., here 'walking')

2.4 Computer Vision Applications for Non-Human Primates

2.4.1 Overview of Scientific Contributions

Table 2.1 provides an overview of various publications investigating computer vision applications for non-human primates. While not exhaustive, the table specifically highlights contributions that apply deep learning methods to great ape data. For each paper, we indicated several information, as whether great apes are included in the data and, if applicable, the specific great ape species, the computer vision tasks addressed, and whether the data used is publicly accessible. The same set of contributions is visualized in Fig. 2.3 for further clarity.

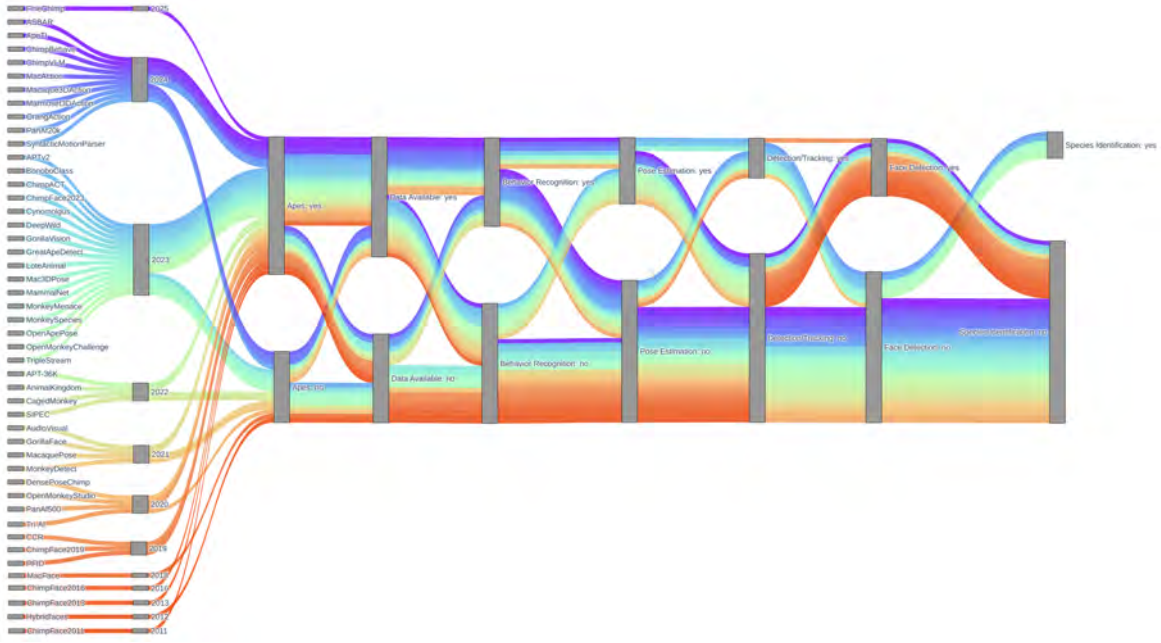

Fig. 2.3 **Overview of scientific contributions in computer vision for non-human primates.** This diagram visualizes a summary of deep learning contributions in computer vision applied to non-human primates from 2011 to 2025. Each flow represents a study or dataset, categorized by year of publication, inclusion of great apes, data availability and computer vision tasks. Best seen zoomed in.

While a detailed description of every task is beyond the scope of this dissertation, the key computer vision tasks listed in Table 2.1 are defined as follows:

- *Face Recognition (FR)*: A classification task that associates an individual's face with its identity. This process can be framed as an image classification problem. It may include a prior face detection step, in which the spatial location of the individual's head is identified using object detection methods [172].
- *Species Identification (SI)*: A classification task where the model predicts the species to which an individual belongs [158].
- *Detection/Tracking (D/T)*:
  - *Detection*: The task of locating one or multiple individuals in an image, typically addressed using object detection techniques.
  - *Tracking*: The task of associating detections over time to identify which detection corresponds to the same individual across a sequence of frames [124].

- *Pose Estimation (PE)*: As previously described in Section 2.2, pose estimation involves predicting the coordinates of anatomical keypoints (e.g., joints or body parts) for an individual [36].
- *Behavior Recognition (BR)*: As detailed in Section 2.3, behavior recognition involves classifying actions or activities from a sequence of video frames [17].

Table 2.1 **List of scientific contributions in computer vision for non-human primate.** This non-exhaustive list orders most prominent works by year of publication and symbolic name.
C=Chimpanzees / B=Bonobos / G=Gorillas / O=Orangutans / BR=Behavior Recognition / FR=Face Recognition / PE=Pose Estimation, D/T=Detection/Tracking / SI=Species Identification

Symbolic Name	Year	Great Apes	Species	Tasks	Data Available
FineChimp (ch. 5)	2025	yes	C	BR	yes
ApeTI [130]	2024	yes	C	FR	yes
ASBAR (ch. 3)	2024	yes	C, G	BR, PE	yes
ChimpBehave (ch. 4)	2024	yes	C	BR	yes
ChimpVLM [16]	2024	yes	C, G	BR	
MacAction [131]	2024			BR, PE	yes
Macaque3DAction [135]	2024			BR	yes
Marmoset3DAction [93]	2024			BR, PE	yes
OrangAction [53]	2024	yes	O	BR	
PanAf20K [17]	2024	yes	C, G	BR, D/T, SI	yes
SyntacticMotionParser [139]	2024			BR	
APTv2 [222]	2023	yes	C, G, O	PE, D/T, SI	yes
BonoboClass [129]	2023	yes	B	D/T	yes
ChimpACT [124]	2023	yes	C	BR, PE, D/T	yes
ChimpFace2023 [173]	2023	yes	C	FR	yes

Continued on next page

Name	Year	Great Apes	Species	Tasks	Data Available
Cynomolgus [113]	2023			BR	
DeepWild [213]	2023	yes	C	PE	yes
GorillaVision [107]	2023	yes	G	FR	
GreatApeDetect [220]	2023	yes	C, G	D/T	
LoteAnimal [119]	2023			BR, PE, D/T	yes
Mac3DPose [7]	2023			PE	
MammalNet [27]	2023	yes	C, G, O	BR, SI	yes
MonkeyMenace [162]	2023			D/T	
MonkeySpecies [158]	2023			SI	
OpenApePose [36]	2023	yes	C, B, G, O	PE, SI	yes
OpenMonkeyChallenge [224]	2023	yes	C, B, G, O	PE, SI	yes
TripleStream [15]	2023	yes	C, G	BR	
AnimalKingdom [149]	2022	yes	C, G, O	BR, PE	yes
APT-36K [223]	2022	yes	C, G, O	PE	yes
CagedMonkey [190]	2022			PE	yes
SIPEC [127]	2022			BR	
AudioVisual [5]	2021	yes	C	BR	
GorillaFace [14]	2021	yes	G	FR	yes
MacaquePose [106]	2021			PE	yes
MonkeyDetect [104]	2021			D/T	
DensePoseChimp [168]	2020	yes	C		yes
OpenMonkeyStudio [8]	2020			BR, PE	yes
PanAf500 [167]	2020	yes	C, G	BR	yes
Tri-AI [66]	2020	yes	C, O	FR	yes
CCR [6]	2019	yes	C	D/T, FR	
ChimpFace2019 [172]	2019	yes	C	FR	

Continued on next page

Name	Year	Great Apes	Species	Tasks	Data Available
PFID [180]	2019	yes	C	FR	
MacFace [215]	2018			FR	
ChimpFace2016 [48]	2016	yes	C	FR	yes
ChimpFace2013 [121]	2013	yes	C	FR	
Hybridfaces [122]	2012			FR	
ChimpFace2011 [42]	2011	yes	C	FR	

2.4.2 Great Ape Behavior Recognition

In Table 2.1, several contributions focus on the recognition of great ape behaviors. This section highlights key milestones and methodologies from these studies.

One of the earliest studies in great ape behavior recognition is presented in [5], which explores the multimodal combination of visual and audible signals to infer whether wild chimpanzees are performing one of two actions: buttress drumming or nut cracking. The study introduces a pipeline composed of several 2D-CNNs (e.g., for audio spectrogram classification and individual detection), culminating in the use of 3D-CNNs for final spatio-temporal classification. This pioneering contribution laid the groundwork for deep learning applications in primate behavior recognition.

The work presented in [167] marks a significant step forward, as it provides the first classification of great ape behavior on a large-scale video dataset, i. e. *PanAf500*, which will be later made public in [17]. To classify behaviors, the authors employ C2D [181], an early dual-stream model that fuses RGB and optical flow data before classification.

Building on this foundation, [15] expands the dual-stream model architecture to incorporate a third stream using dense pose information from [168]. In this approach, video clips are preprocessed into optical flow and dense pose representations. The features are extracted using ResNet [200]. The system is trained using metric learning, leveraging reciprocal triplets with cross-entropy loss. At inference time, a k-NN classifier is used to predict action classes based on the learned feature embeddings.

In [17], two datasets are published: *PanAf500* and *PanAf20K*. Both datasets feature footage collected in the same natural settings, yet they differ significantly in scale and scope. PanAf20K is substantially larger, containing nearly 20,000 video clips annotated across 18 behaviors, while PanAf500 is more compact, comprising 500 videos with 9 classes of more

refined behavioral actions. The authors provide a benchmark for both datasets, evaluating the performance of several deep learning models. They include three CNN-based architectures: I3D [23], 3D ResNet-50 [69], and X3D [43], as well as two Transformer-based models: MViTV2 [117] and TimeSformer [11].

More recently, ChimpVLM [16] demonstrated performance improvements on both PanAf500 and PanAf20K by introducing a multimodal transformer architecture. In this approach, BERT is used to embed textual behavior descriptions (ethogram information) into the model pipeline, enhancing the model's action classification performance by combining video and text modalities. Note that, to date, this work has not yet been peer-reviewed.

In parallel, *ChimpACT* [124] presents a chimpanzee-specific video dataset annotated for three machine learning tasks: Detection/Tracking, Pose Estimation, and Action Recognition. For action recognition, ChimpACT includes approximately 160,000 frames annotated with multi-label behaviors across 23 classes. The authors provide a benchmark of five CNN-based models for spatio-temporal action detection.

OrangAction [53] represents one of the first attempts to specifically recognize orangutan behaviors. While novel, this study relies on CreateML, a black-box service offered by Apple. The data used contains only two individuals, raising concerns about potential bias and overfitting. Moreover, the system fails to predict any action classes for 87% of the test data, limiting its practical impact.

MammalNet [27] and *AnimalKingdom* [149] are two large-scale video datasets that include footage of great apes labeled for behavior recognition. Notably, MammalNet comprises 539 hours of video across 173 species, and AnimalKingdom includes 50 hours of video covering 850 species. However, great apes represent only a small fraction of their content, and their practical applications in primatology have yet to be demonstrated.

Chapter 3

Skeleton-Based Action Recognition for Great Ape Behaviors

The present chapter is based on the following published work:

Fuchs, M., Genty, E., Zuberbühler, K., and Cotofrei, P. (2024c). ASBAR: an Animal Skeleton-Based Action Recognition framework. Recognizing great ape behaviors in the wild using pose estimation with domain adaptation. *eLife*, 13:RP97962

Abstract

The study and classification of animal behaviors have traditionally relied on direct human observation or video analysis, processes that are labor-intensive, time-consuming, and prone to human bias. Advances in machine learning for computer vision, particularly in pose estimation and action recognition, offer transformative potential to enhance the understanding of animal behaviors. However, the integration of these technologies for behavior recognition remains underexplored, particularly in natural settings.

We introduce *ASBAR* (*Animal Skeleton-Based Action Recognition*), a novel framework that integrates pose estimation and behavior recognition into a cohesive pipeline. To demonstrate its utility, we tackled the challenging task of classifying natural behaviors of great apes in the wild.

Our approach leverages the OpenMonkeyChallenge dataset, one of the largest open-source primate pose datasets, to train a robust pose estimation model using DeepLabCut. Subsequently, we extracted skeletal motion data from the PanAf500 dataset, a collection of in-the-wild videos of gorillas and chimpanzees annotated with nine behavior categories. Using PoseConv3D from MMAction2, we trained a skeleton-based action recognition model, achieving a Top-1 accuracy of 75.3%. This performance is comparable to previous video-based methods while reducing input data size by approximately 20-fold, offering significant advantages in computational efficiency and storage.

To support further research, we provide an open-source, terminal-based GUI for training and evaluation, along with a dataset of 5,440 annotated keypoints for replication and extension to other species and behaviors.

All models, code, and data are publicly available at: https://github.com/MitchFuchs/asbar.

3.1 Introduction

Direct observation and manual annotation of animal behaviors are labor-intensive, time-consuming, and prone to human error [203]. These methods also face significant limitations, such as information loss in low-visibility settings or during complex, fast-paced social

interactions involving multiple individuals. Video recording and post-hoc annotation have thus become the preferred methods for studying animal behavior. They enable detailed identification and interpretation of behaviors, while also facilitating reliability testing and replication of coding. However, the manual annotation of videos remains a significant bottleneck, underscoring the need for automated systems that can streamline animal behavior analysis. Machine learning tools have the potential to identify relevant video sections containing social interactions and automatically classify behaviors, significantly expanding the scope and robustness of observational studies and enhancing our understanding of animal behaviors [4].

Recent advancements in machine learning and computer vision offer innovative avenues for building such systems. In particular, action recognition models can learn deep representations of video features and classify these features into behavior categories. Within deep learning, two primary approaches to action recognition have emerged: video-based methods and skeleton-based methods.

On one hand, video-based action recognition involves analyzing RGB video data to identify spatio-temporal patterns that characterize actions. This approach often relies on Convolutional Neural Networks (CNNs) [103] adapted to the temporal domain. Notable models include Two-Stream CNNs [181], C3D [199], I3D [23], (2+1)D ResNet [200], and SlowFast [43]. These methods have been extended to classify animal behaviors [183, 116, 167, 46, 13, 127] and multimodal audio-visual data [5].

On the other hand, skeleton-based action recognition predicts behavior classes based on the skeletal structure and motion of the body [47, 39]. This approach relies on an additional preprocessing step, *pose estimation*, which detects body parts, such as joints and bones, and extracts their coordinates from video frames [21]. While skeleton-based methods require the added step of pose estimation, they offer several advantages for computational ethology [4, 203, 72]:

(i) *Cross-subject behavior recognition*: These models focus on skeletal motion rather than external appearance, allowing them to generalize across individuals within the same species (e.g., [120] for humans, [8, 186] for non-human animals).

(ii) *Robustness to visual setting changes*: Video-based models are sensitive to lighting conditions, background variations, and other subtle changes in input data [26, 191, 126]. Comparatively, skeleton-based methods are less affected by these variations [68].

(iii) *Reduced computational complexity*: Extracting pose coordinates reduces the dimensionality of video data, lowering computational costs and power consumption [47]. This is particularly beneficial for field researchers with limited resources.

(iv) *Geometric quantification*: Pose estimation provides a pre-computed geometric representation of body motion and behavior [203, 154].

A major challenge for skeleton-based methods in animal behavior analysis lies in obtaining accurate pose-estimated data. While human pose estimation benefits from extensive open-source datasets and state-of-the-art detectors, achieving similar performance for animals remains challenging. Fortunately, there has been a surge in annotated animal pose datasets (e.g., Animal Kingdom [149], Animal Pose [20], AP-10K [226], OpenMonkeyChallenge [224], OpenApePose [36], MacaquePose [106], Horse-30 [132]) and open-source pose estimation frameworks (e.g., DeepLabCut [133, 134, 108], SLEAP [153, 155], AniPose [94]). Despite these advancements, the use of pose estimation for behavior recognition remains underexplored, particularly in natural settings. Challenges include the lack of datasets with both keypoint coordinates and behavior annotations, and the tendency to treat pose estimation and behavior recognition as separate tasks rather than parts of an integrated approach.

To address these challenges, we introduce ASBAR, an innovative framework for animal skeleton-based action recognition. Our contributions include:

- An integrated pipeline, which combines a DeepLabCut-based pose estimation module with a behavior recognition module from MMAction2 [141]. The pipeline is encapsulated in a terminal-based GUI, allowing researchers to train and evaluate models without programming knowledge, even in remote or cloud-based environments.
- A robust primate keypoint detector, by leveraging the OpenMonkeyChallenge dataset [224], which spans 26 primate species. Additionally, we provide detailed performance metrics for species and individual body parts and release 5,440 high-quality keypoint annotations for great apes in their natural habitats.
- A methodology for wild behavior analysis using the PanAf500 dataset [167], a two-hour collection of camera trap videos annotated with nine locomotive behaviors. We demonstrate that our skeleton-based pipeline achieves performance comparable to existing video-based methods.

3.2 The ASBAR Framework

ASBAR is an integrated data and model pipeline (marked in red in Fig 3.1) designed to address two sequential machine learning tasks: *pose estimation* and *action recognition*.

The first module, responsible for animal pose estimation (marked in green in Fig 3.1), incorporates key features of DeepLabCut (DLC), a widely used framework for multi-animal,

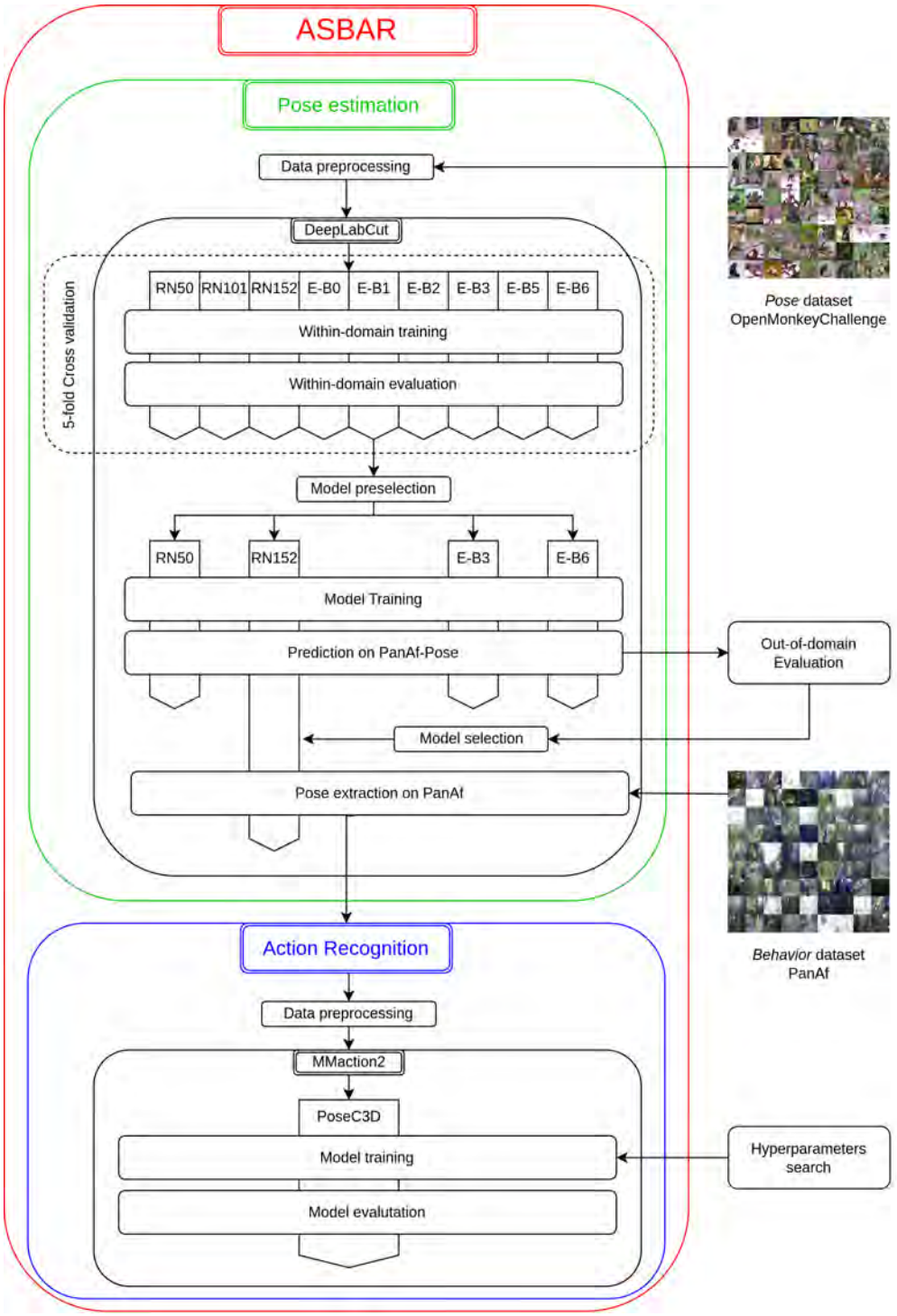

Fig. 3.1 **The ASBAR Framework.** The ASBAR framework's data and model pipeline (red) comprises two modules: a *pose estimation* module (green) based on DeepLabCut and an *action recognition* module (blue) integrating models from MMAction2.

markerless pose estimation [133, 134, 108]. This module includes functionality for project creation, dataset preparation, model training and evaluation, configuration editing, and video analysis.

The second module, focused on behavior recognition (marked in blue in Fig 3.1), integrates APIs from MMAction2 [141], a comprehensive platform for action recognition and video understanding. Specifically, this module employs PoseConv3D [39], a convolutional neural network (CNN) model tailored for skeleton-based action recognition.

3.2.1 Pose and Behavior Datasets

ASBAR operates on two distinct datasets: a *pose* dataset and a *behavior* dataset. The pose dataset contains images annotated with 2D keypoint coordinates and is used to train the pose estimator model. The behavior dataset comprises video clips annotated with specific behaviors.

Ideally, both datasets originate from the same visual distribution—for example, pose images being a subset of video frames from the behavior dataset. However, in practice, annotating a dataset with both pose and behavior labels is time-consuming and costly. A pragmatic approach involves combining pose and behavior datasets from different visual distributions. For instance, Internet images annotated with keypoints can complement video data labeled with behaviors recorded in the wild. In such cases, the pose dataset is considered *within-domain*, while the behavior dataset is referred to as *out-of-domain*.

3.2.2 Pose Estimation Module

The goal of the pose estimation module is to extract pose information from the behavior dataset using a trained pose estimator. This module encompasses four key functionalities: *data preprocessing*, *model benchmarking*, *model selection*, and *pose extraction*.

Data Preprocessing: The module provides four preprocessing steps:

- *Data formatting*: Ensures the pose dataset meets DeepLabCut's structural requirements.
- *Data selection*: Allows users to customize the dataset by selecting specific species, annotated keypoints, or excluding invisible keypoints. For example, users can limit the dataset to chimpanzees and bonobos, focusing on three visible keypoints (e.g., eyes and nose).
- *Dataset splitting*: Supports options for no cross-validation, 5-fold cross-validation, or 10-fold cross-validation to enable statistical validation of the model's performance.

- *Configuration setup*: Enables customization of DeepLabCut's configuration files, including training hyperparameters.

Model Benchmarking: Given the importance of high pose prediction performance for behavior recognition accuracy, the framework facilitates benchmarking various pose estimation models. Users can evaluate models with different backbones (e.g., ResNet or EfficientNet) and depths (e.g., ResNet50, ResNet101, EfficientNet-B0).
Model Selection: When the pose and behavior datasets share the same visual distribution, benchmarking results suffice for model selection. However, for out-of-domain behavior datasets (Sect.3.2.1), additional evaluation is necessary, as within-domain performance does not guarantee robustness to visual domain shifts. Models with EfficientNet backbones, for example, have demonstrated superior generalization to out-of-distribution scenarios compared to ResNet models [132]. Evaluating out-of-domain performance involves comparing model predictions with manually labeled video frames from the behavior dataset. More details are provided in Sect.3.3.3.
Pose Extraction: The selected model extracts pose information from the behavior dataset. Users can specify a particular model snapshot or allow the framework to choose the snapshot with the lowest test set error.

3.2.3 Action Recognition Module

The action recognition module classifies behaviors in the behavior dataset using pose data extracted from the first module. This module includes three functionalities: *data preprocessing*, *model training*, and *model evaluation*.
Data Preprocessing: To enable behavior recognition, the module implements four preprocessing steps:

- *Prediction filtering*: Retains only the highest-confidence keypoint predictions for each frame. For each keypoint, the most confident coordinates within the labeled bounding box are kept, while others are discarded.

- *Data sampling*: Extracts sequences of consecutive frames that meet specific time and behavior label constraints (see Sect. 3.3.5 for details).

- *Data formatting*: Converts skeleton data into a structure compatible with PoseConv3D.

- *Configuration setup*: Allows customization of PoseConv3D's configuration, including hyperparameter settings.

Model Training: Users can train several variations of PoseConv3D available in the MMAction2 toolbox [39, 141]. These variations include different 3D-CNN backbones (e.g., SlowOnly [44], C3D [199], X3D [43]) with an I3D classification head [23]. Training can be distributed across multiple GPUs to reduce computation time.
Model Evaluation: The module produces probabilistic classifications, returning a ranked list of behavior candidates with associated confidence probabilities. The behavior with the highest confidence is used to calculate Top-1 Accuracy, the percentage of correctly predicted samples. Other metrics, such as Top-*k* Accuracy (percentage of ground-truth behaviors within the top-*k* predictions) and Mean Class Accuracy (average Top-1 accuracy across behavior classes), are also supported.

3.3 Materials and Methods

3.3.1 Datasets and Data Annotation

For the classification of great ape behaviors in their natural habitat, we utilized two primary datasets: OpenMonkeyChallenge and PanAf500. Additionally, we manually labeled a subset of keypoint coordinates from PanAf500, referred to as PanAf500-Pose.
OpenMonkeyChallenge: OpenMonkeyChallenge (OMC) [224] is a benchmark dataset containing 111,529 images of 26 primate species, designed for non-human primate pose estimation challenges. The dataset includes images sourced from the web, three U.S. National Primate Research Centers, and multiview cameras at the Minnesota Zoo. Each image is annotated with species, bounding box coordinates, and 2D pose information for 17 keypoints, including the nose, eyes, head, neck, shoulders, elbows, wrists, hips, tail, knees, and ankles. For occluded keypoints, annotators were instructed to provide the most likely location and specify visibility.

The dataset is divided into training (60%), validation (20%), and testing (20%) subsets. While the testing annotations are withheld for competition purposes, we combined the training and validation sets to create a comprehensive *pose* dataset containing 89,223 images. Examples of these images are shown in Fig 3.2 (left).
PanAf500: The Pan African Programme "The Cultured Chimpanzee" [136] aims to enhance understanding of the ecological and evolutionary factors influencing chimpanzee behavioral diversity. This program has amassed thousands of hours of footage from camera traps deployed in Central African forests. The PanAf500 dataset consists of 500 15-second videos (180,000 frames at 24 FPS) of chimpanzees and gorillas, annotated with bounding box coordinates for ape detection [221, 220] and behaviors for action recognition [167]

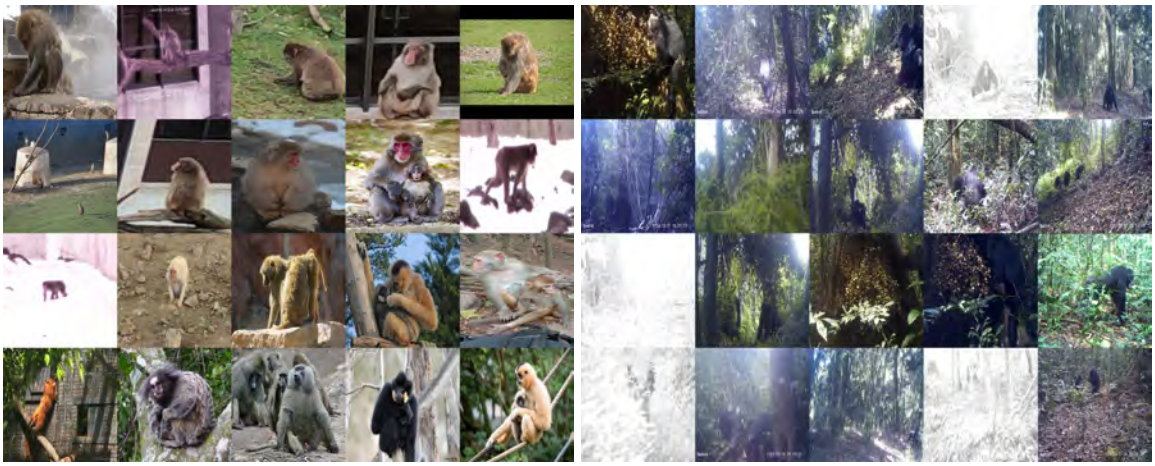

Fig. 3.2 **Examples from the *pose* and *behavior* datasets.** (*Left*) Sample images from the *OpenMonkeyChallenge* dataset, one of the largest collections of primate images annotated with 2D poses. This dataset contains over 100,000 images from 26 primate species. (*Right*) Sample video frames from the *PanAf500* dataset, comprising 500 videos of gorillas and chimpanzees recorded in African forests using camera traps. The dataset includes annotations for bounding boxes and behaviors. Visual challenges include small individual sizes due to camera distance, abundant vegetation, nocturnal imaging, and varying backgrounds.

The dataset includes nine annotated behaviors: 'walking,' 'standing,' 'sitting,' 'running,' 'hanging,' 'climbing up,' 'climbing down,' 'sitting on back,' and 'camera interaction.' The class distribution exhibits a long-tail pattern [34], with three *head* classes ('walking,' 'standing,' and 'sitting') each containing over 1,000 samples. In contrast, *tail* classes such as 'running,' 'climbing up,' 'climbing down,' 'sitting on back,' and 'camera interaction' have fewer than 100 samples each. Examples from this dataset are displayed in Fig 3.2 (right).
PanAf500-Pose:To supplement the PanAf500 dataset, we manually annotated 5,440 keypoints across 320 images, using the same keypoints as in OMC. The annotation process involved three steps:

(i) *Image selection*: We first shortlisted 4,000 images using predictions from ResNet152 and EfficientNet-B6 models based on high overall prediction confidence (Section 3.3.3). From this shortlist, 320 frames were manually selected to represent diverse scenes, lighting conditions, postures, sizes, and species, while minimizing consecutive frames;

(ii) *Mini-clip generation*: For each selected frame, we generated a 34-frame mini-clip (24 frames before and 10 frames after) to capture motion and aid in labeling occluded keypoints;

(iii) *Keypoint annotation*: We employed a semi-automated annotation process, initially leveraging predictions from the ResNet152 model. These predictions were refined using DeepLabCut's Napari plugin [182]. In the first phase, a non-trained annotator (MF) adjusted the predictions. The annotations were then finalized by a great ape behavior and signaling expert (EG) [57], ensuring high-quality labels.

3.3.2 Evaluation Metrics

3.3.2.1 Evaluation Metrics for Pose Estimation

Mean Average Euclidean Error (MAE): MAE is the primary evaluation metric in DeepLabCut and measures the average Euclidean distance between the ground-truth labels ($\hat{y} \in \mathbb{R}^2$) and the model predictions ($y \in \mathbb{R}^2$):

$$\text{MAE} = \frac{1}{J}\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{J}||\hat{y}_{jk} - y_{jk}|| \tag{3.1}$$

Here, J is the number of images (e.g., 89,223 in OMC) and K is the number of keypoints (e.g., 17 in OMC). Refer to Fig. S2 in Appendix B for a visual comparison of predictions and MAE examples.

Percentage of Correct Keypoint - nasal dorsum (PCK): PCK measures the percentage of keypoints that fall within a specified distance of the ground-truth. PCK is computed as:

$$\text{PCK} = \frac{1}{J}\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{J}\delta(||\hat{y}_{jk} - y_{jk}|| < \varepsilon) \tag{3.2}$$

Here, $\delta(\cdot)$ is an indicator function that outputs 1 when the condition is met and 0 otherwise. The threshold distance ε is equal to the nasal dorsum length, defined as the distance between the midpoint of the eyes and the tip of the nose calculated for each frame as:

$$\varepsilon = \left|\left|\hat{y}_{\text{nose}} - \frac{1}{2}(||\hat{y}_{\text{left eye}} - \hat{y}_{\text{right eye}}||)\right|\right| \tag{3.3}$$

See Fig. S1 in Appendix B for an example of nasal dorsum calculation.

Normalized Error Rate (NMER): NMER quantifies the mean normalized error by dividing the raw pixel distance between the predicted and ground-truth keypoints by the square root of the bounding box area [132]:

$$\text{NMER} = \frac{1}{J}\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{J}\frac{||\hat{y}_{jk} - y_{jk}||}{\sqrt{w_j h_j}} \tag{3.4}$$

Here, w and h denote the width and height of the bounding box, respectively.

Both PCK and NMER are size- and distance-normalized metrics, unlike MAE, making them more robust for evaluating diverse scenarios.

3.3.2.2 Evaluation Metrics for Action Recognition

Similar to [167], we use the three following action recognition metrics, whose mathematical formulas can be found in Appendix E.

Top-1 Accuracy: Top-1 Accuracy measures the percentage of samples where the model's highest-confidence prediction matches the ground-truth label.

Top-3 Accuracy: Top-3 Accuracy measures the percentage of samples where the ground-truth label appears within the top three predictions of the model.

Mean Class Accuracy (MCA): MCA calculates the average accuracy across all classes, giving equal weight to each class irrespective of sample size. See Fig. 3.8 for details.

3.3.3 Methods for Pose Estimation

Data Preprocessing For our experiment, we utilized all annotated data from the OpenMonkeyChallenge (OMC) dataset, encompassing 26 species and 17 keypoints, including invisible ones. Due to OMC's large size, a 5-fold cross-validation approach was chosen for model benchmarking.

Within-domain Models Benchmarking: We evaluated the within-domain performance of nine pose estimation models, including three ResNet architectures (RN-50, RN-101, RN-152) and six EfficientNet variants (B0, B1, B2, B3, B5, B6) [75, 192]. All models were pretrained on ImageNet [165] and then trained on OMC for 40,000 iterations, a duration estimated as sufficient for loss convergence in preliminary tests using the largest network (EfficientNet-B6).

Default training hyperparameters and augmentation settings were used except for the batch size, which was set to 16 (the maximum fitting into the GPU memory for EfficientNet-B6 on an NVIDIA A100 40GB). The learning rate schedule followed the defaults: 1.0e-04 until iteration 7,500, 5.0e-05 until iteration 12,000, and 1.0e-05 for the remainder. We performed 5-fold cross-validation, splitting the dataset into 80% training and 20% testing subsets, ensuring that all 89,223 images were included in the test set once. All models were trained remotely on the Google Cloud platform with NVIDIA A100 (40GB) or V100 (16GB) GPUs, using ASBAR's GUI (see Fig. S4 in Appendix B for examples of elements).

Model snapshots were saved every 5,000 iterations and evaluated on the test set. Each model's eight snapshots were evaluated across all five folds ($8 \times 5 = 40$ evaluations per model). To handle the computational load, we customized DLC's evaluation pipeline to

batch data processing and limit evaluations to the test set. While this modification was not integrated into the released framework, the default DLC evaluation method remains available in ASBAR's GUI for reproducibility.

Model Shortlisting: Since the ultimate goal was to predict keypoints on the out-of-domain *behavior* dataset, we shortlisted four models based on their robustness and generalization potential: (i) ResNet-152: Best-performing within-domain model; (ii) ResNet-50: Widely used and popular among researchers; (iii) EfficientNet-B6: Demonstrated strong generalization to out-of-domain data in prior studies [132]; (iv) EfficientNet-B3: Despite lower within-domain accuracy, it balances strong out-of-domain generalization [132] with low computational cost (1.8G FLOPs versus 4.1G for ResNet-50 and 11G for ResNet-152).

The shortlisted models were retrained on the full OMC dataset (89,223 images) with no test set. Training was extended to 100,000 iterations to accommodate the larger training dataset. All other hyperparameters were unchanged. Model snapshots were saved every 5,000 iterations, producing 20 snapshots per model for evaluation.

3.3.4 Methods for Pose Extraction

After evaluating pose estimation performance (see Sect. 3.4.1 for results), ResNet-152 was selected for pose extraction. This model was applied to all frames in the *behavior* dataset to predict keypoint candidates.

Given the visual differences between the *pose* and *behavior* datasets, the pose estimation model's confidence threshold was lowered to 10^{-6} to maximize keypoint candidate generation and minimize cases of "no prediction." Skeletal poses were extracted by filtering these candidates to retain only the 17 keypoints with the highest confidence scores within the annotated bounding boxes.

3.3.5 Methods for Behavior Recognition

Data Preprocessing: The methodology proposed by [167] was followed for data sampling. A minimum threshold of 72 consecutive frames (equivalent to 3 seconds) exhibiting the same behavior was set to ensure the inclusion of prolonged and meaningful behavioral patterns. Selected video clips were divided into samples of 20 consecutive frames, with no gaps or overlaps between samples.

The dataset was randomly split at the video level into training, validation, and testing subsets, using a 70-15-15 distribution. The pose extraction output was formatted and stored as triplets of $(x, y, \text{confidence})$ coordinates for each keypoint.

Model Training: A PoseConv3D model [39] with a ResNet3dSlowOnly backbone and an I3D classification head was selected for behavior recognition. This architecture was chosen for its strong performance on NTU60-XSub [120], a benchmark dataset for human action recognition, as reported by [39].

To adhere strictly to a skeleton-based approach, the model was trained exclusively on pose-estimated data, without incorporating the multimodal RGB+Pose capability. Only joint keypoints (excluding limbs) were used, with a sigma value of 0.6 (examples illustrated in Fig.2.2). Probabilistic confidence scores (c_k in Equation 2.17) were not considered, given the intentionally low confidence threshold during pose extraction.

The model weights were initialized from pretraining on the FineGym dataset [176]. Training was conducted for 50 epochs using two NVIDIA RTX 2080 Ti GPUs (2×11GB). A class-balanced focal loss [34] was employed to address the imbalanced class distribution ($\beta = 0.992$, $\gamma = 2$). Other hyperparameter choices included: batch size of 32; initial learning rate of 0.005 (with momentum of 0.9 and cosine annealing); weight decay of 0.01 and a dropout ratio of 0.8 (i.e. strong regularization to avoid overfitting). Other hyperparameters and augmentation settings followed those used in [39].

3.4 Results

To showcase the ASBAR framework's capability in animal behavior recognition from pose estimation, we selected a particularly challenging task: classifying great ape natural behaviors in the wild.

This section details the experimental results. First, we present the evaluation of pose estimation models, including both within-domain and out-of-domain results, leading to the selection of an optimal model for pose extraction (Section 3.4.1). Additional insights into model performance at keypoint and species levels are provided in Section 3.4.2. Finally, skeleton-based behavior classification results are reported and compared to existing studies (Section 3.4.3).

3.4.1 Results of Pose Estimation

Within-domain evaluation: We compared the performance of all nine models after 40,000 iterations by constructing 95% confidence intervals for MAE using a t-distribution ($\alpha = 0.025$, $\nu = 4$), given the small sample size from cross-validation. The results (as seen in Figure 3.3) show that: (i) ResNet-152 achieved the best performance (14.05 ± 0.199), statistically outperforming other models; (ii) ResNet-101 ranked second (14.334 ± 0.080);

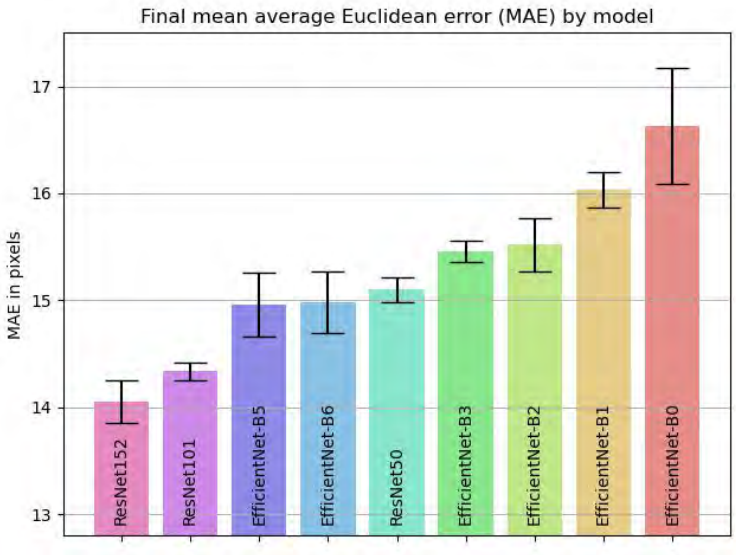

Fig. 3.3 **Final within-domain model performance.** Mean and 95% confidence intervals of the MAE (in pixels) after 40,000 iterations (end of training). Disjoint confidence intervals indicate statistically significant differences. ResNet-152 demonstrates significantly better performance compared to all other models in this task.

(iii) EfficientNet-B5 (14.958 ± 0.299), EfficientNet-B6 (14.981 ± 0.288), and ResNet-50 (15.098 ± 0.12) had overlapping confidence intervals, making their performances statistically indistinguishable; (iv) EfficientNet-B3 (15.455 ± 0.097) and EfficientNet-B2 (15.519 ± 0.25) performed slightly worse; (v) EfficientNet-B1 (16.031 ± 0.167) and EfficientNet-B0 (16.631 ± 0.546) exhibited the highest error rates.

In addition, the performance of each model during training is visualized through deviation charts of their snapshot variants, showing the mean and standard deviation of MAE and PCK in Figure 3.4.

Out-of-domain Evaluation: Each saved model snapshot was tested on the PanAf500-Pose ground-truth annotations. To reduce the influence of noisy predictions, the minimum confidence threshold for pose prediction was maintained at the default value of 0.1.

Our results indicate that ResNet-152 generalizes best to out-of-domain data (Fig 3.5), achieving the highest overall PCK-nasal dorsum of 54.17% across all keypoints ($n = 5,440$) and the lowest normalized error rate (NMER) of 10.19%.

Pose extraction: Based on ResNet-152's performance at 60,000 iterations—achieving a detection rate of 53.9% (very close to the highest observed value) and a minimal NMER of 10.19%—this snapshot was selected as the final keypoint detector for pose extraction.

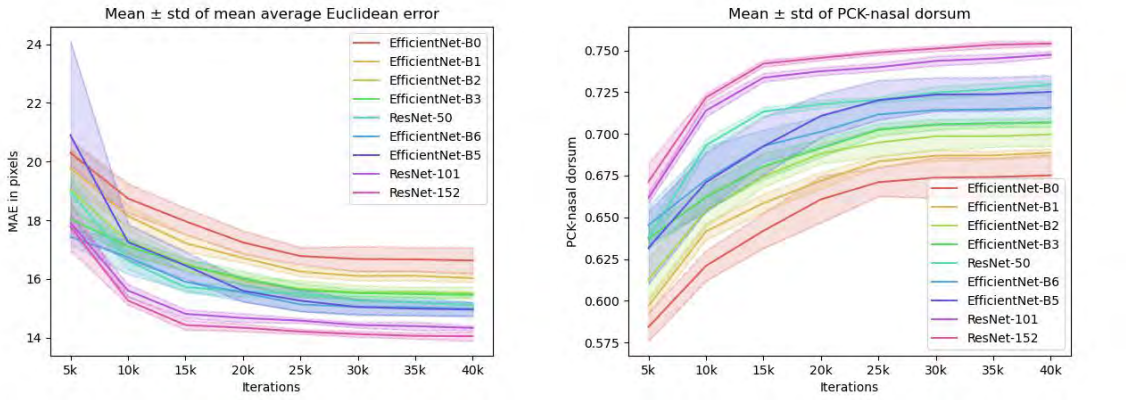

Fig. 3.4 **Model's relative performance throughout 'within-domain' training.** The mean ± std of the Mean Average Euclidean Error (MAE) in pixels (*left*, lower is better) and percentage of correct keypoint (PCK nasal dorsum) (*right*, higher is better) for all nine model variations. Evaluation results of 5-fold cross-validation on test set data, at every 5,000 iterations.

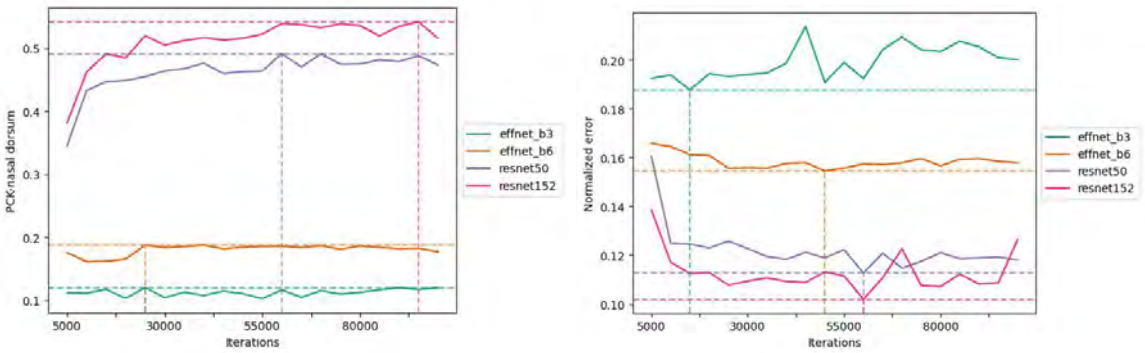

Fig. 3.5 **Out-of-Domain performance on PanAf500-Pose.** Models are evaluated using two metrics that account for the animal's relative size and distance: PCK nasal dorsum (*left*, higher is better) and normalized error rate (*right*, lower is better). ResNet-152 demonstrates superior performance in predicting great ape poses in their natural habitat. Vertical and horizontal dashed lines indicate the maximum and minimum values, along with the corresponding number of iterations. ResNet-152 at 60,000 iterations is selected for pose extraction.

3.4.2 Alternative Performance Evaluation

To gain deeper insights into the final pose estimation model's performance, we evaluated it across keypoints and species using both OMC and PanAf500-Pose datasets. A confidence threshold of 0.1 was applied throughout.

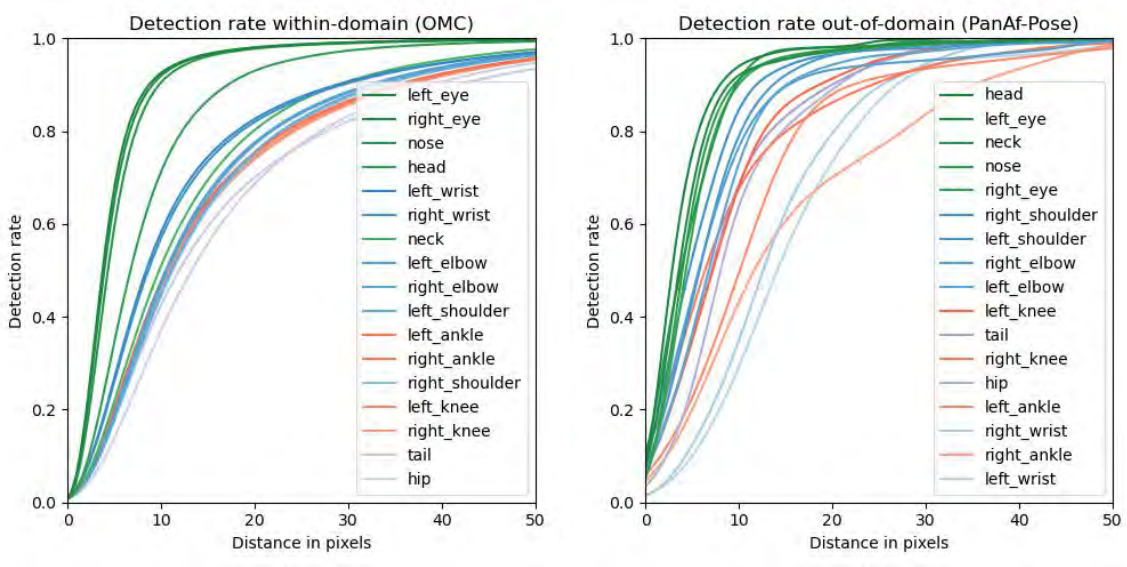

Fig. 3.6 **Keypoint detection rate on within-domain vs. out-of-domain test data.** The keypoint detection rate, defined as the percentage of keypoints detected within a given pixel distance, is shown for OMC (*left*) and PanAf500-Pose (*right*). For example, within a distance of 10 pixels or less, the nose is detected in approximately 95% of the 89,223 images in OMC. In contrast, the tail is detected within the same distance in only about 38% of cases.

3.4.2.1 Keypoint Detection Rate

Our analysis revealed that not all keypoints are equally detectable for non-human primates. Detection rates, computed as the cumulative distribution of predicted distances in pixels (Fig 3.6), highlight the following trends at test time for OMC ($n = 89,223 \times 17 = 1,516,791$):

- Facial features (e.g., nose, eyes) are the easiest to detect.
- Keypoints on the head are more accurately predicted than those below the neck.
- Upper body limbs (e.g., wrists, elbows, shoulders) are detected more reliably than lower body limbs (e.g., ankles, knees).
- Limb extremities (e.g., wrists, ankles) are predicted more accurately than proximal keypoints (e.g., elbows, knees).
- Hip and tail positions are the most challenging to predict accurately.

These trends can be attributed to (i) the distinct visual features of facial keypoints, (ii) the prominence of heads and limb extremities, and (iii) the occlusion and ambiguity of lower body parts and tails.

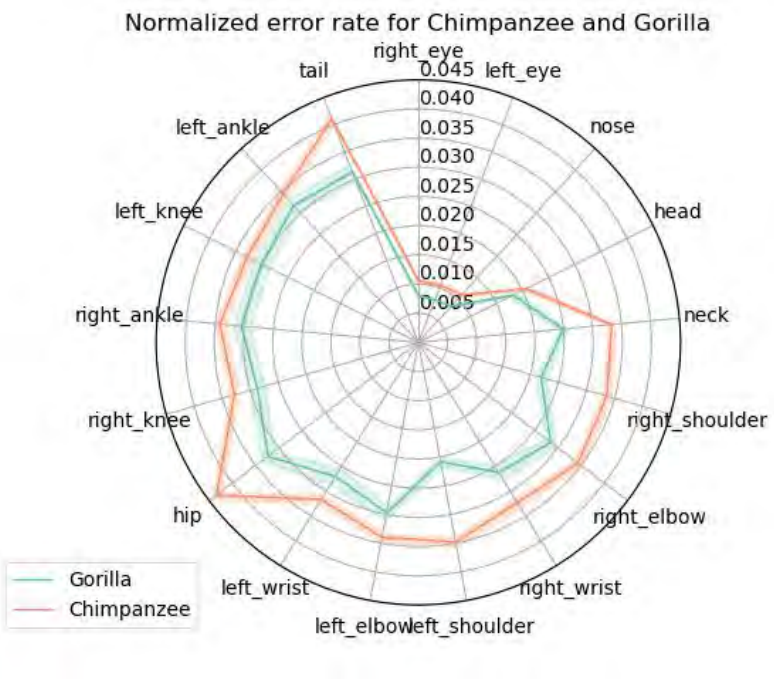

Fig. 3.7 **Normalized error rate for chimpanzees and gorillas in OMC.** Mean and 95% confidence intervals for the normalized error rate (NMER). Disjoint confidence intervals indicate statistical significance. The model demonstrates lower error rates for all gorilla keypoints, suggesting higher prediction accuracy for this species.

Comparing results from PanAf500-Pose ($n = 320 \times 17 = 5,440$) shows a similar S-shaped distribution, indicating the model's robustness to domain shifts. However, lower detection rates for specific keypoints may result from the precise annotations in PanAf500-Pose compared to OMC, where annotations are occasionally inconsistent (e.g., labeling fingers instead of wrists or toes instead of ankles).

3.4.2.2 Per-Species Accuracy

To evaluate species-specific performance, we analyzed chimpanzees ($n = 6,190$) and gorillas ($n = 1,777$) in OMC using the normalized error rate (NMER) and 95% confidence intervals (Fig 3.7).

Results indicate a statistically significant dependence on species, with gorillas consistently showing lower error rates than chimpanzees across all keypoints. This suggests that the model detects keypoints more accurately for gorillas. Additional species-level analysis is provided in Fig S3 in Appendix B.

3.4.3 Results of Behavior Recognition

The results of behavior classification, summarized in Table 3.1, demonstrate the successful application of our skeleton-based action recognition pipeline for animals. In the context of automating the recognition of great ape behaviors in the wild—a highly relevant yet challenging task—our approach achieves accuracy comparable to other video-based techniques, such as those reported in [167].

Table 3.1 **Performance comparison with previous studies.** Comparison of Top-1 Accuracy, Top-3 Accuracy, and Mean Class Accuracy (MCA) between ASBAR and previous video-based methods. ASBAR achieves comparable performance to video-based approaches across all metrics.

	Approach	**Top1 Accuracy**	**Top3 Accuracy**	**MCA**
Two-Stream CNNs [167]	video-based	73.5%	94.1%	42.3%
ASBAR	skeleton-based	**75.3%**	**95.4%**	**47.0%**

To the best of our knowledge, this is the first use of a skeleton-based method for classifying great ape behaviors. Notably, the entire behavior dataset after pose extraction (i.e., the input features for the behavior classifier) requires less than 60 MB of storage in text format—approximately 20 times smaller than the storage requirements of the same dataset using a video-based approach. For ethologists working in the field, where computational, storage, and transfer resources are often limited, this represents a significant improvement without sacrificing performance in behavior recognition.

The normalized confusion matrix of the final behavior recognition model is shown in Fig.3.8. The model tends to overfit on *head* behavior classes, which have more samples in the dataset (see Sect.3.2.1). For instance, the model frequently overpredicts 'walking,' the second most represented class, at the expense of *tail* classes. The false positive rates (i.e., misclassification rates) for 'walking' on 'sitting on back,' 'climbing up,' 'climbing down,' 'running,' and 'camera interaction' are 0.74, 0.42, 0.50, 0.89, and 0.40, respectively.

Interestingly, the other two *head* classes—'standing' and 'sitting'—show near-zero false positive rates for the same *tail* classes. This discrepancy may be explained by the static nature of 'standing' and 'sitting,' which involve stationary poses, compared to the dynamic movements in 'walking' and most *tail* classes, where pose estimation accuracy may be lower.

Additionally, the true positive rates for 'sitting on back' and 'running' (i.e., their per-class accuracy) are extremely low, at 0.11 each. Both are predominantly misclassified as 'walking.' This likely stems from the similarity in skeletal poses across these behaviors, making it challenging for the model to differentiate between them using only skeleton data, particularly given the limited sample sizes for these classes.

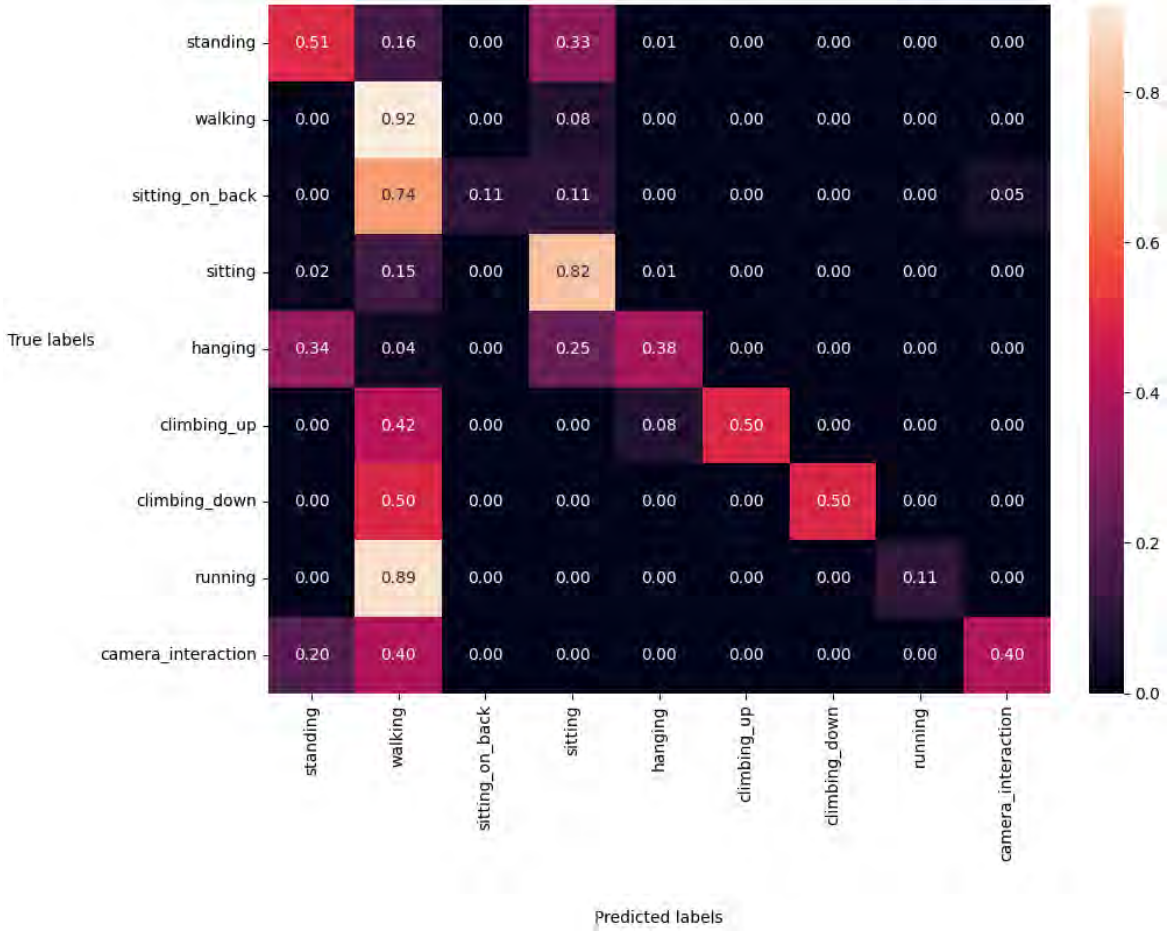

Fig. 3.8 **Normalized confusion matrix of behavior recognition.** For each true behavior label (rows), the percentage of predictions across all predicted behaviors (columns) is shown. For instance, 51% of samples labeled as 'standing' were correctly classified, while 16% were misclassified as 'walking' and 33% as 'sitting.' The diagonal cells represent the per-class accuracy, and their average corresponds to the Mean Class Accuracy (MCA) metric. A perfect classification model would yield a normalized confusion matrix with values of 1 on the diagonal and 0 elsewhere.

3.5 Discussion

Despite the growing availability of open-source resources, such as large-scale animal pose datasets and machine learning toolboxes for pose estimation and human skeleton-based action recognition, their integration for animal behavior recognition—particularly in natural settings—remains largely unexplored. With ASBAR, a framework combining animal pose estimation and skeleton-based action recognition, we provide a comprehensive data and model pipeline, methodology, and GUI to assist researchers in automatically classifying

animal behaviors via pose estimation. We hope these resources will become valuable tools for advancing the understanding of animal behavior within the research community.

To illustrate ASBAR's capabilities, we applied it to the challenging task of classifying great ape behaviors in their natural habitat. Our skeleton-based approach achieved accuracy comparable to previous video-based studies for Top-K and Mean Class Accuracies. Additionally, by reducing the input size of the action recognition model by a factor of approximately 20 compared to video-based methods, our approach requires significantly less computational power, storage space, and data transfer resources. These qualities make ASBAR particularly suitable for field researchers working in resource-constrained environments.

Our framework and results are built on the foundation of shared and open-source materials, including tools like DeepLabCut [133], MMAction2 [141], and datasets such as OpenMonkeyChallenge [224] and PanAf500 [167]. This underscores the importance of making resources publicly available, especially in primatology, where data scarcity often impedes progress in AI-assisted methodologies. We strongly encourage researchers with large annotated video datasets to make them publicly accessible to foster interdisciplinary collaboration and further advancements in animal behavior research.

3.5.1 Challenges and Future Directions

While our results are promising, there are areas for improvement in both pose estimation and action recognition tasks.

Pose Estimation: Out-of-domain PCK metrics for pose estimation hovered just above 0.5, indicating that nearly half of the predicted keypoints were outside the acceptable range of the ground-truth coordinates. Accurate pose estimation is critical for downstream behavior classification. Future work could address this by fine-tuning the pose estimation model on the *behavior* dataset before pose extraction. Additionally, training on more specific datasets, such as OpenApePose [36], could improve performance. Techniques to reduce the domain gap between *pose* and *behavior* datasets [207] or leveraging pseudo-labels for semi-supervised learning [20, 111, 142] could also enhance generalization.

Interestingly, EfficientNet architectures performed worse than ResNet-152 in both within-domain and out-of-domain evaluations, contrary to prior results in animal pose estimation [132]. This discrepancy may stem from suboptimal hyperparameter tuning (e.g., fixed learning rate schedules instead of cosine decay) for EfficientNet models. Future studies should optimize hyperparameters individually for each architecture to fully explore their potential.

Behavior Recognition: While our skeleton-based pipeline achieved comparable results to previous studies, the overall accuracy remains relatively low, which could limit its practical

deployment in the field. Comparisons to human-centric studies, where abundant datasets for both pose estimation and action recognition have led to higher performance [39], highlight the need for additional public datasets [149, 27, 20] to drive progress in AI-assisted animal behavior research.

From an algorithmic perspective, using keypoint detection as pose scoremaps rather than compressing them into (x, y, c) triplets could improve performance, particularly when pose predictions are less accurate [39]. Incorporating RGB-Pose dual-modality could further enhance classification accuracy, especially for behaviors with similar skeletal motion, such as 'walking,' 'running,' and 'sitting on back.'

3.6 Conclusion

This study demonstrates the practical utility and relevance of skeleton-based action recognition approaches in animal behavior research. We hope the tools, methodologies, and insights presented here will inspire further applications of skeleton-based techniques to study a broader range of behaviors and animal species. Future advancements in pose estimation, action recognition, and dataset availability will undoubtedly enhance the impact of such approaches in ethology and beyond.

3.7 Acknowledgments

We extend our sincere gratitude to the team behind the Pan African Programme: 'The Cultured Chimpanzee', along with their partners, for granting us permission to use their data for this study. For access to the videos from the dataset, please reach out directly with the copyright holder Pan African Programme at http://panafrican.eva.mpg.de. In particular, we would like to thank H. Kuehl, C. Boesch, M. Arandjelovic, and P. Dieguez. Further acknowledgments go to: K. Corogenes, E. Normand, V. Vergnes, A. Meier, J. Lapuente, D. Dowd, S. Jones, V. Leinert, E. Wessling, H. Eshuis, K. Langergraber, S. Angedakin, S. Marrocoli, K. Dierks, T. C. Hicks, J- Hart, K. Lee, M. Murai and the team at Chimp&See.

The work that allowed for the collection of the PanAf500 dataset was made possible due to the generous support from the Max Planck Society, Max Planck Society Innovation Fund, and Heinz L. Krekeler. By extension, we also wish to thank: Foundation Ministre de la Recherche Scientifique, and Ministre des Eaux et Forêts in Cote d'Ivoire; Institut Congolais pour la Conservation de la Nature and Ministre de la Recherche Scientifique in DR Congo; Forestry Development Authority in Liberia; Direction des Eaux, Forêts Chasses et de la

Conservation des Sols in Senegal; and Uganda National Council for Science and Technology, Uganda Wildlife Authority, and National Forestry Authority in Uganda.

In addition, we would like to thank the team at NCCR Evolving Language and in particular Guanghao You, for allowing us to use their computational platform.

Chapter 4

Out-Of-Distribution Generalization

The present chapter is based on the following peer-reviewed works:

Fuchs, M., Genty, E., Bangerter, A., Zuberbühler, K., and Cotofrei, P. (2024a). From Forest to Zoo: Domain Adaptation in Animal Behavior Recognition for Great Apes with ChimpBehave. 4th Workshop on CV4Animals: Computer Vision for Animal Behavior Tracking and Modeling, In conjunction with CVPR 2024

and

Fuchs, M., Genty, E., Bangerter, A., Zuberbühler, K., Odobez, J.-M., and Cotofrei, P. (2024b). From Forest to Zoo: Great Ape Behavior Recognition with ChimpBehave. Currently under review with minor revisions for the International Journal of Computers Vision

Abstract

This chapter addresses the significant challenge of recognizing behaviors in non-human primates, specifically focusing on chimpanzees. Automated behavior recognition is crucial for both conservation efforts and the advancement of behavioral research. However, it is often hindered by the labor-intensive process of manual video annotation. Despite the availability of large-scale animal behavior datasets, effectively applying machine learning models across varied environmental settings remains a critical challenge due to the variability in data collection contexts and the specificity of annotations.

In this chapter, we introduce *ChimpBehave*, a novel dataset featuring over 2 hours and 20 minutes of video (approximately 215,000 video frames) of zoo-housed chimpanzees, meticulously annotated with bounding boxes and behavior labels for action recognition. ChimpBehave uniquely aligns its behavior classes with existing datasets, enabling the study of domain adaptation and cross-dataset generalization between different visual settings. Furthermore, we benchmark our dataset using both video-based and skeleton-based CNN action recognition models, providing baselines for within- and cross-dataset evaluations. Our results demonstrate that in cross-dataset experiments with substantial visual changes, the skeleton-based approach performs statistically significantly better than the video-based method. The dataset, models, and code can be accessed at: https://github.com/MitchFuchs/ChimpBehave

4.1 Introduction

The development of machine learning tools to recognize animal behaviors from videos plays a critical role in ecology and ethology. Automated systems for recognizing chimpanzee behaviors could offer a broad spectrum of applications, ranging from enhancing conservation efforts to providing valuable insights into the behavior of great apes. Moreover, non-invasive technologies developed for their well-being can significantly benefit chimpanzees — an endangered species — in both wild and captive settings. For instance, these systems could monitor population dynamics in natural habitats or provide timely signals of behavioral abnormalities in unwell individuals to caretakers in zoos.

As one of humans' closest living relatives, chimpanzees have been the subject of extensive scientific research in fields such as ecology, comparative cognition, neuroscience, and evolutionary biology. This research often relies on videos, whose manual annotation is time-consuming and labor-intensive. The advancement of algorithms for animal behavior classification can, therefore, significantly benefit researchers by expediting the labeling process and/or reducing its overall cost.

However, such algorithms require large amounts of data for training before they can be effectively deployed in the observational fields of behavioral studies. To address this, large-scale animal datasets have recently been created to adapt human-centered action recognition models for animal behavior classification (see, e.g., Animal Kingdom [149] and MammalNet [27]). Although comprehensive, these datasets lack the fine-grained annotations needed to capture the complex behaviors of great apes.

To bridge this gap, more specialized datasets, such as ChimpACT [124] and PanAf [17], have been developed to target species-specific behaviors across various environments - ranging from zoo settings to wild forests - and under diverse recording conditions (see Section 4.2.1 for details). While invaluable, their practical relevance for researchers may be constrained by the distinctiveness of their visual, environmental, and recording settings. For instance, behavior recognition models trained on zoo-specific data may overfit to that context, limiting their ability to generalize to other zoo environments, let alone to data from natural forest habitats. As a result, researchers collecting new data in their own settings may find such models unsuitable unless substantial effort is invested in annotating and fine-tuning them for their specific needs.

A critical factor in animal behavior recognition, therefore, is the model's capacity to generalize across diverse environments. A key limitation of both ChimpACT and PanAf is that their sets of annotated behaviors are unique and non-overlapping, making them challenging to use in generalization studies. To address this limitation, we introduce *ChimpBehave*, a novel dataset whose class labels are aligned with PanAf, yet feature starkly different visual and recording conditions. ChimpBehave contains footage of chimpanzees filmed in a zoo environment with handheld cameras focused on individual animals, whereas PanAf consists of recordings from stationary outdoor camera traps in natural forests, capturing both chimpanzees and gorillas. This design makes ChimpBehave an ideal resource for studying cross-dataset generalization in great ape behavior recognition.

Challenges in cross-domain generalization have also been widely explored in human studies. A promising approach to reducing the dependency on domain-specific behavioral data and annotations in humans has been the adoption of *skeleton-based* methods. Unlike *video-based* approaches, which rely on full RGB videos, skeleton-based methods predict

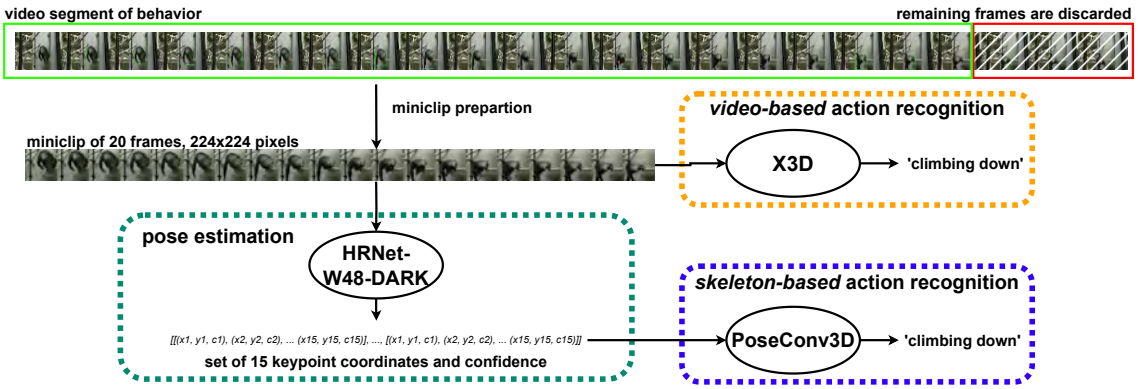

Fig. 4.1 **Representation of our setting**. Miniclips of 20 consecutive frames are extracted from all video segments and fed into X3D, a video-based action recognition model for behavior classification. In parallel, the pose is extracted with HRNet, and then fed into PoseConv3D for skeleton-based action recognition.

behavior by analyzing sequences of extracted skeletal poses. Recent studies have shown that this approach can effectively recognize even fine-grained, complex human actions, such as athletes' movements in gymnastics competitions [39, 176].

As highlighted in various reviews on human data [47, 26, 217], skeleton-based representations tend to be more robust to variations in visual settings, including changes in illumination and background. This robustness is particularly advantageous in cross-domain generalization, where models are applied to new environments with differing visual characteristics. Additionally, the reduction in data from high-dimensional video frames to lightweight 2D joint coordinates offers significant benefits, especially in resource-constrained environments, such as low-power field settings [52].

These promising prospects have drawn attention from the animal research community, with a growing focus on *pose estimation* - the task of predicting an individual's posture via joint coordinates. However, translating skeletal poses into practical animal behavior classification has been only marginally explored, particularly in non-laboratory environments.

Two major challenges arise when considering a skeleton-based approach rather than a video-based one. First, relying solely on skeletal poses while discarding the remaining pixel information could result in lower behavior classification accuracy. Second, estimating accurately the pose of animals presents unique challenges. For example, detecting joint coordinates in animals like chimpanzees is significantly more difficult than in humans due to factors such as the scarcity of publicly annotated data, the variety of poses, frequent occlusions during social interactions, and the low visual contrast of their appearance.

Despite these valid concerns, we argue, based on our experimental results, that a skeleton-based method may not only perform comparably to a video-based method within a single

dataset but can also be statistically significantly more robust in cross-dataset settings. Specifically, when trained on data from one visual environment and tested on a completely different one, skeleton-based approaches can outperform video-based models, even without additional pose estimation labeling costs.

In summary we make the following contributions:

- We introduce *ChimpBehave*, a dataset for great ape behavior recognition, featuring over 2 hours and 20 minutes of video (approximately 215,000 video frames) annotated with fine-grained behaviors and bounding boxes. Its label classes are specifically aligned with existing datasets to facilitate cross-dataset generalization experiments.

- We establish a *performance baseline* on ChimpBehave using CNN-based models for both video-based and skeleton-based action recognition, with X3D [43] and PoseConv3D [39], respectively. We use a rigorous evaluation protocol, including a stratified 5-fold cross-validation procedure to validate our results.

- We examine both *within-dataset* and *cross-dataset generalization* using other existing datasets for video-based and skeleton-based methods. Our results demonstrate that while both methods deliver comparable performance when trained and tested within a single dataset, the skeleton-based approach consistently outperforms the video-based method in cross-dataset scenarios.

The chapter is organized as follows: In Section 4.2, we review related work, introducing various non-human primate datasets used in computer vision applications and providing an overview of key action recognition studies focusing on non-human primates. In Section 4.3, we present the proposed ChimpBehave dataset, detailing its description, data collection process, and annotation methodology. Section 4.4 outlines our methods and experiments, including descriptions of the datasets used, the evaluation metrics, and the methodology behind both our video-based and skeleton-based approaches. This section also covers the models, experimental setup, and protocol details.

Our experimental results are presented in Section 4.5, structured into three subsections: results on the ChimpBehave dataset, within-dataset evaluations, and cross-dataset model performance. Finally, we conclude with a discussion of our findings, some limitations, and suggestions for future research in Section 4.6.

4.2 Related Work

4.2.1 Non-human Primate Datasets

The growing number of animal datasets designed for computer vision tasks has prominently included non-human primates (NHP), reflecting their importance across various ethological and ecological studies. These datasets span a wide array, from those encompassing multiple animal orders [149, 27, 222, 119] to those focusing specifically on primates [224], apes [124, 17, 36], monkeys [190], and particularly macaques [8, 106, 131].

These datasets showcase notable diversity in annotations and tasks, including species identification [27, 222, 224, 17, 36], animal detection and tracking [222, 124, 17, 119], pose estimation [149, 222, 224, 124, 36, 190, 8, 106, 131, 119], and behavior recognition [149, 27, 124, 17, 8, 131, 119].

Among these datasets, only two are dedicated specifically to great ape behavior recognition: ChimpACT [124] and PanAf [17]. These datasets adopt distinct approaches, highlighting the challenges of applying computer vision across diverse contexts. ChimpACT captures the daily life of a young chimpanzee in a zoo environment, characterized by man-made backgrounds, dynamic handheld camera movements, and a longitudinal focus on a single individual. In contrast, PanAf documents the behaviors of chimpanzees and gorillas in their natural habitats, using static cameras deployed in African forests to capture a wide variety of ape populations. These differing settings underline the challenges faced by computer vision models in adapting to varying visual environments. Table 4.1 provides a comparative overview of the main features of these datasets, while Table T4 in Appendix C details their annotated behavioral classes.

PanAf is further divided into two subsets: PanAf20K and PanAf500. PanAf20K comprises over 7 million frames labeled with broad ecological behaviors through crowdsourcing, designed for multi-label behavior recognition at the video level. In contrast, PanAf500 is a smaller, fine-grained dataset focused on multi-class behavior recognition, with individual bounding boxes annotated with their corresponding behaviors. In this chapter, we focus on PanAf500 for its precise annotation scheme, referring to it as PanAf for simplicity.

While ChimpACT and PanAf are valuable datasets, they exhibit limitations when considered for combined use in primatology research. ChimpACT targets spatio-temporal behavior detection with annotations emphasizing social interactions and broad locomotive behaviors (e.g., 'moving', 'resting', 'climbing'). PanAf, however, focuses on fine-grained individual locomotive actions, such as 'walking,' 'running,' 'sitting,' and 'climbing down.' These differences in design and annotation complicate cross-dataset analyses, making direct model comparisons challenging.

ChimpBehave bridges this gap by combining a visual and filming setup reminiscent of ChimpACT, with recordings of individuals in man-made environments featuring camera motion and zooming, while aligning its behavioral annotation scheme with PanAf's detailed locomotive action categories. This design makes ChimpBehave an ideal resource for primatology, enabling straightforward out-of-distribution generalization experiments and facilitating model comparisons across diverse visual environments.

Table 4.1 **Main feature comparison of ChimpBehave, PanAf and ChimpACT for Behavior Recognition (BR).**

	ChimpBehave	**PanAf**		**ChimpACT**
		P500	*P20K*	
Species	Chimpanzees	Chimpanzees & Gorillas		Chimpanzees
Environment	Zoo / Man-made	Forest / Natural		Zoo / Man-made
Location	Basel (CH)	Tropical Africa		Leipzig (D)
Recording method	Focal sampling	Camera trapping		Focal sampling
Cameras	Moving	Fix		Moving
Resolution	1920x1080@25	720x404@24		720x578@25 / 1280x720@25
Annotated frames for BR	213,000	180,000	7,000,000	160,000
Locomotive Behaviors	8	7	3	4
Total Behaviors	8	9	18	23
Annotations by primatologist	✓	✗	✗	✗
Task	Multi-Class BR	Multi-Class BR	Multi-Label BR	Multi-Label BD

4.2.2 Behavior Recognition for Non-Human Primates

With advancements in deep learning, including Convolutional Neural Networks (CNNs) and Transformers, several studies have applied these techniques to the automated recognition of

NHP behaviors. To date, these efforts have primarily focused on macaques [127, 113, 8], monkeys [119], and apes [124, 17, 167, 5, 52, 15, 16].

Key advancements in this domain leverage action recognition (AR) techniques, which are essential for classifying behaviors from video sequences. Below, we provide an overview of recent research focusing on great apes. These techniques are generally categorized into three main approaches: video-based, skeleton-based, and multimodal.

Video-Based AR: Video-based action recognition analyzes visual features directly from raw video data, capturing movements and interactions within the pixel data of each frame. Several studies have employed such methods specifically for NHP behavior recognition. In [167], experiments on the PanAf dataset were conducted using C2D [181], an early dual-stream model architecture that fuses RGB and optical flow data before classification. Subsequently, [15] expanded the number of data streams by incorporating a third path to include dense pose, as presented in [168]. The PanAf dataset was later benchmarked in [167] using various CNN-based and Transformer-based models.

Additionally, several models for spatiotemporal action detection - detecting actions in both space and time within videos - have been benchmarked on videos annotated with chimpanzee behaviors in ChimpACT [124].

Skeleton-Based AR: Skeleton-based approaches, in contrast, focus on tracking the movement of key body points or joints, constructing a skeletal representation of the subject to recognize specific actions or behaviors. This approach was first applied to great ape behavior classification in ASBAR [52], a framework that combines DeepLabCut pose estimation modules [133, 108] with PoseConv3D [39] for action recognition.

Multimodal AR: The recent rise of Vision-Language Models (VLMs) [233], which combine video frames and text as input, was first adapted for great ape behavior classification in [16]. In addition, [5] incorporated multimodal signals by including audio cues to enhance chimpanzee behavior classification.

4.3 The ChimpBehave Dataset

In this section, we present the details of the dataset we have built and its main features.

Dataset Description: ChimpBehave consists of 1,362 annotated video segments of chimpanzees, each labeled for multi-class action recognition, specifying a unique behavioral class along with corresponding bounding boxes. These segments were derived from a collection of 50 longer videos recorded in 2016 at the Basel Zoo indoor enclosure (see examples in Fig. 4.2 and Fig. S5 in Appendix C).

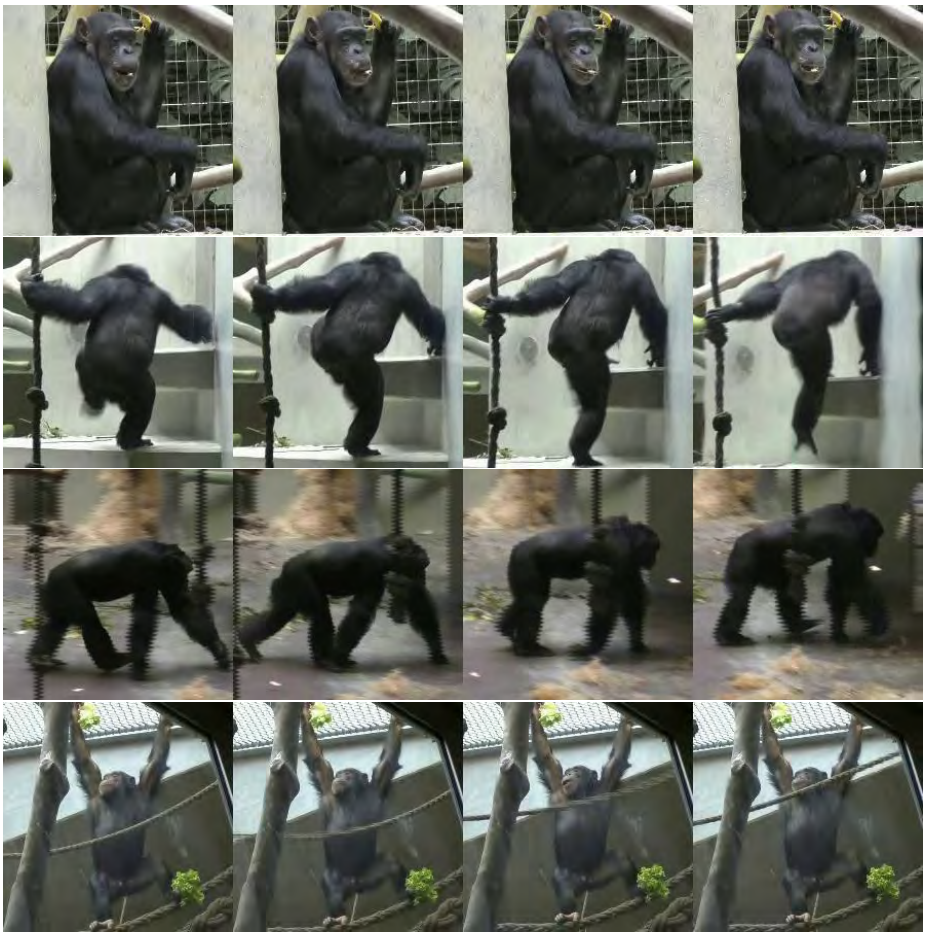

Fig. 4.2 **Walking, hanging, sitting, or climbing up?** Identifying which chimpanzee behavior is depicted in these images is trivial for most humans. For algorithms, however, this is not always the case, especially when exposed to videos from previously unseen environments.

The Basel Zoo houses a group of 15 chimpanzees (*Pan troglodytes verus*), including 12 females (6 adults, 1 subadult, 3 juveniles, and 2 infants) and 3 males (1 adult, 1 juvenile, and 1 infant). The group resides in a facility consisting of six indoor lodges (totaling 233.3 m^2) and two outdoor enclosures (totaling 477 m^2). The lodges are equipped with climbing structures, ropes, puzzle boxes, and other enrichment features.

Each original video was captured using *focal sampling*, where one of nine chimpanzees was tracked while also capturing its surroundings and other conspecifics. The filming conditions were naturalistic, including camera motion, zooming, and occasional shaking. All videos were recorded with a handheld camera at a resolution of 1920x1080 pixels and 25 fps.

The annotated video segments were selected for their clear depiction of behaviors, with an emphasis on including less frequent locomotive behaviors whenever possible. In total, the dataset comprises approximately 215,000 annotated video frames.

Bounding Box Annotations: To label all video frames with bounding boxes, we employed a three-stage process, described below:

1. A single annotator (MF) manually labeled over 16,000 video frames from 206 video segments. Annotations were done on every tenth frame, with an interpolation approach used for the remaining frames. All annotations were performed in Label Studio (https://labelstud.io/) across a variety of scenes. For each chimpanzee, a minimum of 500 frames were annotated, and for each of the 50 original focal videos, at least two segments or 130 frames were included.

2. These labels were then used to fine-tune QDTrack [152] on the MMAction2 platform [141], a state-of-the-art Multiple Object Tracking (MOT) model pretrained on the ChimpACT dataset [124]. This model was selected due to its demonstrated effectiveness for this task, as highlighted in [124]. Fine-tuning was conducted for 6 epochs using 168 video segments, leaving 38 segments for evaluation. The final model achieved the following scores on common MOT metrics at test time: Recall 0.576, Precision 0.706, HOTA 0.499, mAP 0.557.

3. We then used this model to predict tracking bounding boxes for all video segments, manually refining the tracks to correct any ID swaps and linearly interpolating predictions where necessary (see Fig. 4.3 for more details). Each track was individually reviewed, and only valid sequences were retained in the final dataset. The distribution of bounding box sizes is visualized in Fig. 4.4. The code for MOT fine-tuning and data conversion between MMAction2 and Label Studio is available in our code repository. Fig. S16 provides further details on our iterative fine-tuning process.

Keypoint Annotations: The coordinates of 1,500 keypoints were manually annotated by MF. This includes 15 keypoints in 100 individual frames, with two frames selected from each of the 50 focal videos.

Behavior Annotations: An expert primatologist (EG) meticulously labeled the dataset's behavior annotations for all video segments using ELAN (https://archive.mpi.nl/tla/elan). The annotator focused on eight mutually exclusive behavioral classes that represent some of the most common primate behaviors, namely 'sitting', 'standing', 'walking', 'running', 'hanging', 'swinging', 'climbing down', and 'climbing up' (see full ethogram in Table T1 in Appendix C). These classes were selected because they represent a diverse range of locomotion and posture-related behaviors commonly exhibited by primates. They capture both stationary (e.g., 'sitting,' 'standing') and dynamic (e.g., 'walking,' 'running,' 'swinging') actions, providing a comprehensive dataset for action recognition.

Fig. 4.3 **Tracking example after correcting IDs and interpolating missing frames (red bounding box)**. When the individual passed behind the pole, the network lost its track and assigned a new ID. All predicted tracks were manually reviewed, and proper IDs reassigned programmatically where necessary. Missing frames were reconstructed using linear interpolation. Best seen zoomed in. Code is available in the project repository.

From an ecological perspective, these behaviors are integral to understanding primate activity patterns, energy expenditure, and environmental interactions. For instance, locomotor behaviors like 'climbing up' and 'swinging' are directly tied to arboreal navigation, which is critical for species living in forested habitats. Similarly, stationary behaviors such as 'sitting' and 'standing' are often associated with feeding or vigilance, offering insights into primates' social and foraging strategies.

In addition, seven of these classes were intentionally selected to match those annotated in [17], in order to facilitate cross-dataset analysis. Similar to [17], we observe a long-tail class distribution in our dataset, as shown in Fig. 4.5. Accordingly, we refer to 'sitting', 'standing', and 'walking' as *head* classes, while the remaining five are considered *tail* classes.

4.4 Method and Experiments

Our objective is to study and compare within-dataset and cross-dataset generalization performance of standard video-based and skeleton-based action recognition models. To achieve this, we first describe the datasets and data preparation process, followed by the evaluation

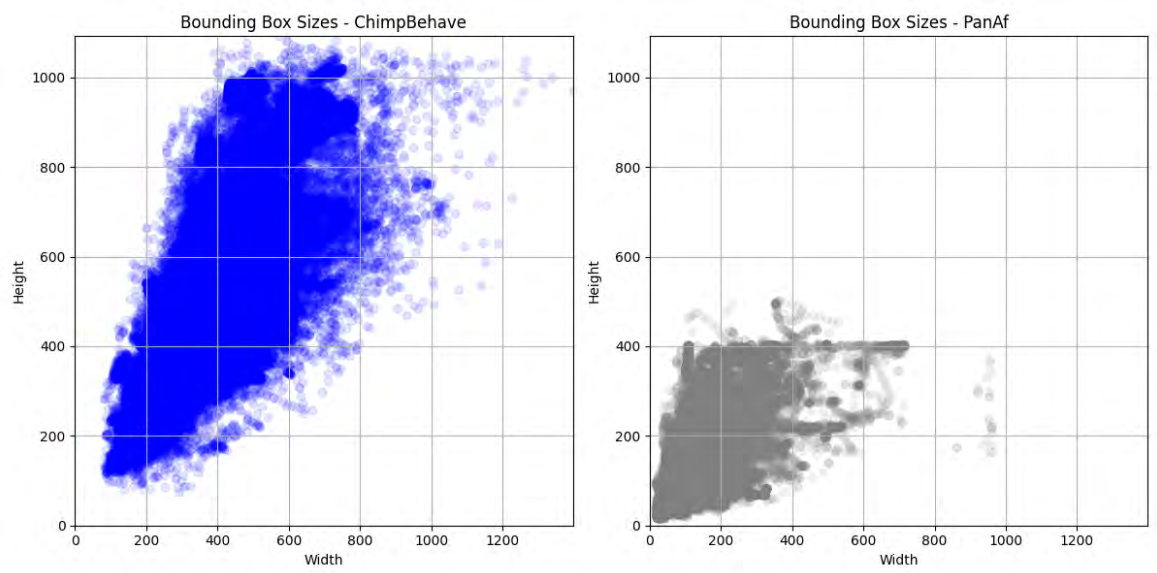

Fig. 4.4 **Bounding box sizes in the ChimpBehave (*left in blue*) and PanAf (*right in grey*) datasets**. Each dot represent the size of one bounding box annotated with behavior.

metrics used in our experiments. We then describe the video-based behavior recognition model and its implementation details, followed by the skeleton-based recognition model, including keypoint estimation. See Fig. 4.1 for on overview of our data pipeline. Finally, we outline the experimental protocol.

4.4.1 Datasets and Data Preparation

In addition to ChimpBehave (described in Sec. 4.3), we relied on three additional datasets introduced below.

4.4.1.1 OpenApePose

The OpenApePose dataset [36] comprises 71,868 images annotated with the 2D poses of various ape species, namely chimpanzees (approximately 25%), gorillas (18%), orangutans (18%), bonobos (16%), gibbons (13%), and siamangs (10%). The images capture individuals in diverse settings, including zoos, sanctuaries, and field sites. Each image contains pose annotations for one individual, including 17 keypoints: nose, eyes, head, neck, shoulders, elbows, wrists, sacrum (center between hips), knees, and ankles. We used this dataset to train our pose estimation model (Sec. 4.4.4.1).

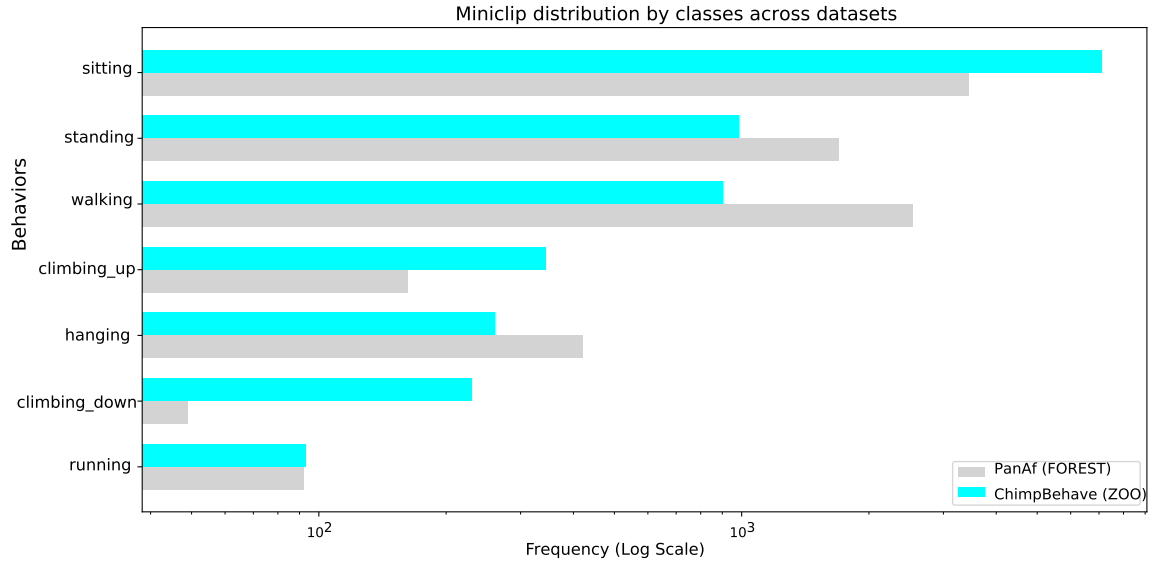

Fig. 4.5 **Behavior frequency distribution in the ChimpBehave and PanAf datasets**. Frequencies are plotted on a logarithmic scale to highlight the long-tailed characteristic of the data

4.4.1.2 ChimpACT

ChimpACT [124] is a dataset that includes videos annotated for animal tracking, pose estimation, and action detection. It comprises 16,028 images annotated for multi-animal 2D pose estimation, with a total of 56,324 annotated poses from a group of chimpanzees housed at the Leipzig Zoo, Germany. The environment is man-made, and the labeled images are derived from 163 longer video segments (see Section 4.2.1 for more details). It includes 16 labeled keypoints: eyes, upper and lower lips, neck, shoulders, elbows, wrists, root of hips, knees, and ankles. This dataset was also used to train our pose estimation model (Sec. 4.4.4.1), and we used its tracking data to pretrain our bounding box tracker (Sec. 4.3).

4.4.1.3 PanAf

The Pan African Programme 'The Cultured Chimpanzee' [136] aims to enhance the understanding of evolutionary and ecological factors that influence chimpanzee behavioral diversity. As part of this effort, numerous hours of footage were collected using camera traps placed in the forests of Central Africa. From this collection, 500 videos, each 15 seconds long (totaling 180,000 frames at 24 fps, resolution 720x404), were annotated with bounding boxes for ape detection and action labels for behavior recognition [17, 167] (see Fig. S6 in Appendix C for image examples). The nine annotated behaviors include 'walking', 'standing', 'sitting', 'running', 'hanging', 'climbing up', 'climbing down', 'sitting on back',

and 'camera interaction'. This dataset was used in our experiments for action recognition (Sec. 3.4). Note that this dataset is referred to as PanAf500 in [17], distinct from PanAf20k, which includes coarser-grained actions and is not employed in our experiments.

4.4.1.4 Data Preparation

Behavioral Class Selection: In our experiments on ChimpBehave and the results presented in Sec. 4.5.1, we included all eight behavioral classes in our training and testing data. However, when examining within-dataset and cross-dataset generalization in Sec. 4.5.2.1 and Sec. 4.5.2.2, we only included the seven classes that overlap with PanAf, namely 'walking', 'standing', 'sitting', 'running', 'hanging', 'climbing up', and 'climbing down'.
Miniclip Preparation: For our experiments, we converted both ChimpBehave and PanAf videos into *miniclips*, i.e., sequences of 20 consecutive video frames associated with one unique behavioral class and without frame overlap. Any remaining frames beyond the last multiple of 20 were discarded. The coordinates of the individual's bounding boxes within each miniclip were used to calculate the global minimum and maximum coordinates of the region to crop. Each miniclip was then resized to 224x224 pixels. This process yielded 10,043 unique miniclips from ChimpBehave and 8,404 from PanAf.
Keypoint Extraction: In our skeleton-based approach, each miniclip mentioned above was converted into a series of keypoint coordinates and confidence scores using the Dark-HRNet-W48 model, as described in Sec. 4.4.4.1.

4.4.2 Evaluation Metrics

Pose Estimation: To evaluate the performance of our pose estimation model on images from ChimpBehave and PanAf, we relied on the Normalized Mean Error Rate (NMER) and Percentage of Correct Keypoint - Nasal Dorsum (PCK-ND), two metrics suggested in [52], where the set of 2D pose annotations from PanAf were made available. (see Sect. 3.3.2 for formulas).

- *Normalized Mean Error Rate (NMER)* measures the average error distance per image and per keypoint between a predicted keypoint and its ground truth coordinates, normalized by a factor proportional to the dimensions of the individual's bounding box.

- *Percentage of Correct Keypoint - Nasal Dorsum (PCK-ND)* gives the percentage of keypoints predicted within a certain radius of the ground truth coordinates. In this case, the radius is defined by the length of the individual's nasal dorsum (nose bridge).

Behavior Recognition: At the behavioral class level, we compute the following metrics (see Appendix E for formulas):

- *Precision*: The proportion of correctly predicted miniclips of a behavioral class among all miniclips predicted as that class.
- *Recall*: The proportion of actual miniclips of a behavioral class that are correctly recognized by the model.
- *F1 Score*: The harmonic mean of Precision and Recall, balancing both metrics, which often involve a trade-off.
- *False Positive Rate (FPR)*: The proportion of miniclips that do not belong to a class but are incorrectly predicted as belonging to it.
- *False Negative Rate (FNR)*: The proportion of actual miniclips of a class that are missed by the model.

At the dataset level, we evaluate each model using the following action recognition metrics: Top-1 Accuracy, F1 Score (weighted), Mean Class Accuracy (MCA), and Mean Average Precision (mAP).

- *Top-1 Accuracy* measures the proportion of correctly classified miniclips, making it sensitive to imbalanced class distributions.
- *F1 Score (weighted)* is useful for imbalanced datasets, as it combines precision and recall, providing a single score while accounting for the distribution of classes.
- *Mean Class Accuracy (MCA)* evaluates the average accuracy across all classes, using the same weight per class.
- *Mean Average Precision (mAP)* provides a comprehensive measure by averaging precision across different recall levels for each class, capturing the model's overall ability to identify relevant miniclips. This metric is not sensitive to imbalanced class distributions.

4.4.3 Video-Based Behavior Recognition

Behavior Recognition Model: X3D [43], a standard CNN-based model for action recognition, was chosen as our comparative baseline. This model was selected due to its strong performance on the PanAf benchmark, as presented in [17]. X3D's architecture incrementally

builds upon a small 2D image classification model, expanding along several network dimensions, including space, time, depth, and width. This progressive expansion is designed to optimize the trade-off between model complexity and performance, enabling highly efficient models without sacrificing accuracy.

Implementation Details: Each X3D model was trained from scratch for 50 epochs using default hyperparameters, with a batch size of 8. Training was performed using *SGD* optimization, with an initial learning rate of 0.01, momentum of 0.9, and weight decay of 0.0001.

4.4.4 Skeleton-Based Behavior Recognition

The quality of the skeleton-based action recognition model depends heavily on the capacity of the pose estimation model to reliably detect and localize keypoints. In the following section, we first present the design and training of the keypoint detector module, followed by the behavior recognition model.

4.4.4.1 Pose Estimation

Data Preparation: We merged images and pose annotations from the OpenApePose and ChimpACT datasets, resulting in a total of 87,896 images annotated with 128,192 poses, comprising chimpanzees (approximately 58%), gorillas (10%), orangutans (10%), bonobos (9%), gibbons (7%), and siamangs (6%). Only keypoints present in both datasets were included: 'left eye', 'right eye', 'neck', 'left shoulder', 'right shoulder', 'left elbow', 'right elbow', 'left wrist', 'right wrist', 'hip/sacrum', 'left knee', 'right knee', 'left ankle', and 'right ankle'. Additionally, we unified the 'nose' coordinates from OpenApePose and the 'upper lip' coordinates from ChimpACT into a single keypoint labeled 'nose or upper lip'. This merged dataset was split into train/validation/test partitions using a 68/16/16 split.

Model Training: We trained an HRNet-W48 model [189], enhanced with the DARK method [232], due to its superior performance on ChimpACT, as demonstrated in [124]. The weights were initialized from a model pretrained on COCO-WholeBody V1.0 [89], and fine-tuned for 120 epochs. Additionally, we enforced that each keypoint should be considered *visible*, regardless of whether the original annotation suggested otherwise, to encourage the network to predict joint coordinates even when occluded.

Evaluation on ChimpBehave and PanAf: To assess the model's prediction accuracy on ChimpBehave, we annotated 1,500 ground truth keypoint coordinates and used them to compute the Normalized Mean Error Rate (NMER) and PCK-Nasal Dorsum (PCK-ND). We applied the same metrics to images from the PanAf dataset, using keypoint coordinates from

Table 4.2 **Evaluation metrics of our pose estimation network on an image subset of ChimpBehave and PanAf.**

	# images	# keypoints	NMER	PCK-ND
ChimpBehave	100	1500	9.48%	47.0%
PanAf (in [52])	320	4800	9.08%	59.19%

[52]. Results are shown in Table 4.2, whereas qualitative prediction examples are shown in Fig. 4.6 and in Figs. S7 to S10.

Several observations can be made based on these results: 1) Despite training on a relatively large dataset, the model achieves only moderate performance on both datasets, failing to correctly detect even half of the keypoints on ChimpBehave (PCK-ND: 47%); 2) Similar to [52], not all keypoints are equally easy to detect. For instance, facial features are detected more accurately than limb extremities (see keypoint-wise metrics in Fig. 4.7); 3) Despite ChimpBehave's higher image resolution compared to PanAf, the metrics indicate better pose estimation performance on PanAf. A possible explanation is that PanAf's pose annotations in [52] were primarily made on video frames depicting less dynamic behaviors, such as 'walking,' 'sitting,' and 'standing.' In contrast, our set of annotations in ChimpBehave deliberately included more challenging poses from fast-paced behaviors, such as 'climbing down' and 'running.'

4.4.4.2 Behavior Recognition Model

The PoseConv3D model [39], designed for action recognition, was employed in this study. This model was chosen based on its high accuracy in recognizing complex human actions, as demonstrated in [39], and its proven efficacy for great ape behavior recognition, as shown in [52]. Notably, PoseConv3D uses only CNNs for action recognition, whereas most other skeleton-based models rely on Graph Convolutional Networks (GCNs). GCNs are generally less robust in handling noisy pose estimates [227, 47]. In PoseConv3D's pipeline, keypoint coordinates and confidence scores are first transformed into 3D heatmap volumes, which are then processed by a 3D-CNN classifier for action recognition (see [39] or [52] for more model details).

Implementation Details: Each PoseConv3D model was trained from scratch for 25 epochs using default hyperparameters, with a batch size of 32, optimized using *SGD* (initial learning rate of 0.2, momentum of 0.9, and weight decay of 0.0003). We used only *keypoint* data (including confidence scores), excluding *limb* data, and did not employ *RGB+Pose multimodality* to clearly differentiate between methods.

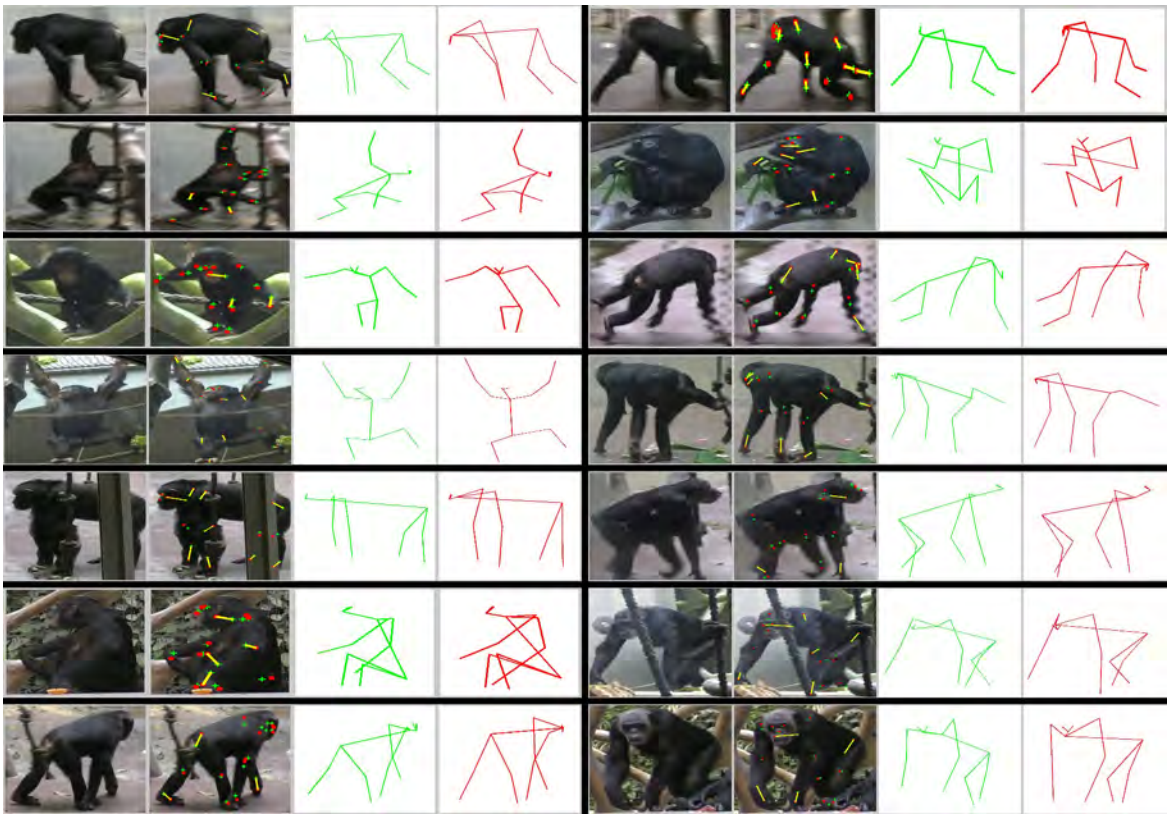

Fig. 4.6 **Pose estimation examples on ChimpBehave**. From left to right: 1) original image cropped around the individual bounding box. 2) keypoint ground truth (green crosses) and HRNet prediction (red dots), the prediction error is highlighted when greater than the nasal dorsum length (yellow segments). 3) ground truth skeleton (in green). 4) predicted skeleton (in red). Here we present some examples where we consider the pose prediction to be relatively good, ie. where both skeletons share relatively similar appearances. See Figs. S7 to S10 for more examples.

4.4.5 Experimental Protocol

Within-Dataset Cross-Validation Procedure: To validate all within-dataset experimental results, we followed a standard stratified 5-fold cross-validation procedure. In this approach, each of the five models was trained on 4 folds (representing 80% of the dataset) and validated on the remaining 20%. This ensures that all dataset miniclips are used exactly once for validation. The same folds were used for both video-based and skeleton-based methods. To maintain a similar class distribution across folds, miniclips were sampled proportionally for each class. Miniclip selection for each fold was done in the order of their appearance in the database, sorted by video name and then by frame numbering. This approach groups miniclips at the video level as much as possible, ensuring better generalization across videos, similar to the train/validation/test partitioning used in [17].

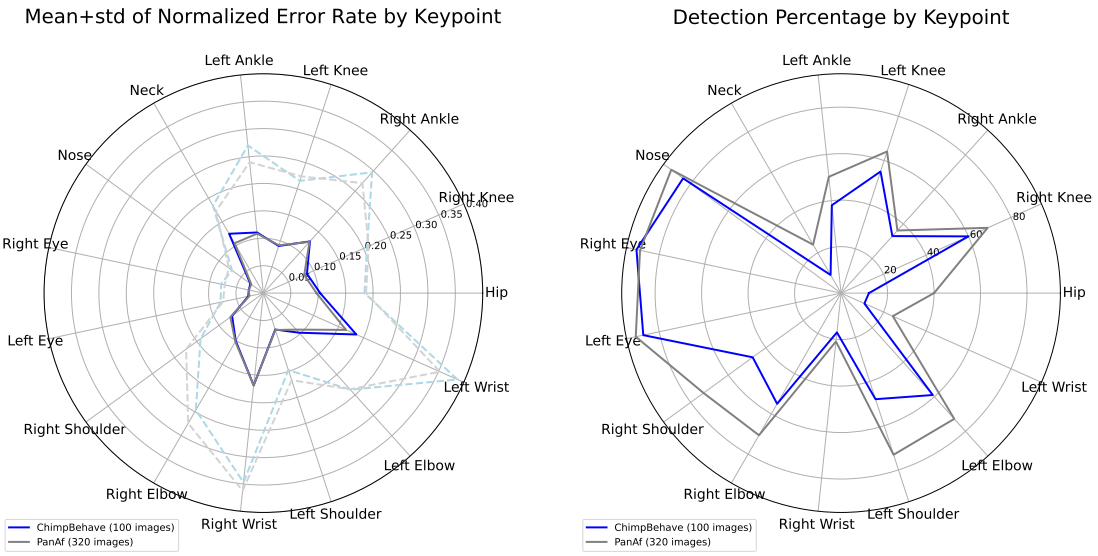

Fig. 4.7 **Pose estimation metrics by keypoint for both datasets**. (*Left*) Normalized error rate: mean (solid) and mean+std (dotted). (*Right*) Detection percentage within nasal dorsum distance.

Cross-Dataset Cross-Validation Procedure: In the cross-dataset evaluation, we used all models trained on each group of 4 folds of one *source* dataset (i.e., 80% of the miniclips from one dataset) and tested them on all the miniclips of the second *target* dataset. This procedure allows us to assess the generalization capability of the models across datasets.
Confidence Intervals: To assess the statistical significance of the results, we calculated 95% confidence intervals based on the Student distribution with degrees of freedom $\nu = 4$, for the average of each metric calculated using the five evaluations from the 5-fold cross-validation procedure.
Experimental Details: All training and evaluation for the models were conducted using the MMaction2 platform [141]. The experiments were run on the HPC cluster at the University of Neuchâtel, utilizing 4x NVIDIA RTX 2080 Ti GPUs (each with 11GB of memory). For model selection, we chose the final epoch based on its Top-1 accuracy on the validation set.

4.5 Results

In this section, we present the results obtained on ChimpBehave for both video-based and skeleton-based methods (Sec. 4.5.1). In Section 4.5.2, we incorporate data from the PanAf dataset to examine and compare within-dataset and cross-dataset generalization for both methods.

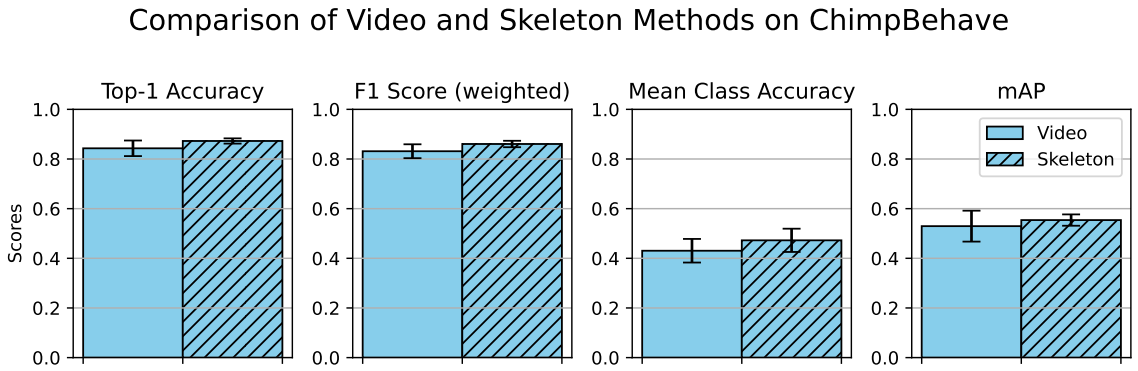

Fig. 4.8 **Evaluation metrics of ChimpBehave (8 classes).**

4.5.1 Behavior Recognition on ChimpBehave

Overall Observations: We present our classification results for all eight behavioral classes in ChimpBehave for both video-based and skeleton-based methods in Fig. 4.8 (see Fig. S13 in Appendix C for UMAP visualization). Overall, we observe relatively high Top-1 Accuracy and weighted F1 scores for both methods, while significantly lower results are observed for MCA and mAP. Unlike Top-1 Accuracy and weighted F1, MCA and mAP give equal importance to each behavioral class in the dataset. The lower scores for these metrics suggest that, while both methods can generally predict behaviors well, some behaviors are more challenging to recognize than others.

Several factors may contribute to this discrepancy. One major factor is the unbalanced distribution of miniclips across behavioral classes. As highlighted in Section 4.3, ChimpBehave exhibits a typical long-tailed class distribution, with head classes ('sitting', 'standing', and 'walking') being overrepresented compared to tail classes ('hanging', 'climbing up', 'climbing down', 'running', 'swinging').

Class-Level Observations: To gain better insights into the models' ability to recognize different behaviors, we report class-level metrics in Table 4.3 and visualize the corresponding confusion matrices in Fig. S14.

We observe that Precision, Recall, and F1 Score per class tend to decrease as the number of miniclips per class decreases. None of these metrics exceed a mean of 0.6 for any of the tail classes, showcasing the difficulty in recognizing these behaviors. Conversely, the high FPR for 'sitting' (Video-Based (VB): 0.145 ± 0.024, Skeleton-Based (SB): 0.154 ± 0.028) highlights the models' tendency to overfit this class, likely due to its high frequency in the dataset. This overfitting may occur particularly when model training and selection are based on the Top-1 Accuracy score, as in our setting.

Table 4.3 **Class-level metrics (Mean ± Std) for ChimpBehave (8 classes) across 5-fold cross-validation.**

	Precision	Recall	F1 Score	FPR	FNR
Video-based					
Sitting	0.941 ± 0.008	0.962 ± 0.031	0.951 ± 0.013	0.145 ± 0.024	0.038 ± 0.031
Standing	0.799 ± 0.083	0.712 ± 0.112	0.744 ± 0.062	0.021 ± 0.013	0.288 ± 0.112
Walking	0.661 ± 0.071	0.717 ± 0.116	0.679 ± 0.050	0.039 ± 0.016	0.283 ± 0.116
Hanging	0.497 ± 0.220	0.433 ± 0.145	0.455 ± 0.167	0.014 ± 0.008	0.567 ± 0.145
Climbing Up	0.305 ± 0.048	0.507 ± 0.084	0.377 ± 0.044	0.042 ± 0.010	0.493 ± 0.084
Climbing Down	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.001	1.000 ± 0.000
Running	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000
Swinging	0.166 ± 0.144	0.113 ± 0.130	0.123 ± 0.134	0.005 ± 0.004	0.887 ± 0.130
Skeleton-based					
Sitting	0.939 ± 0.010	0.986 ± 0.008	0.962 ± 0.005	0.154 ± 0.028	0.015 ± 0.008
Standing	0.885 ± 0.050	0.765 ± 0.051	0.819 ± 0.031	0.011 ± 0.006	0.235 ± 0.051
Walking	0.713 ± 0.031	0.779 ± 0.047	0.744 ± 0.032	0.031 ± 0.004	0.222 ± 0.047
Hanging	0.597 ± 0.064	0.444 ± 0.153	0.498 ± 0.120	0.008 ± 0.002	0.557 ± 0.153
Climbing Up	0.468 ± 0.073	0.364 ± 0.100	0.393 ± 0.051	0.016 ± 0.007	0.636 ± 0.100
Climbing Down	0.336 ± 0.066	0.265 ± 0.161	0.259 ± 0.122	0.014 ± 0.010	0.735 ± 0.161
Running	0.229 ± 0.146	0.064 ± 0.039	0.098 ± 0.060	0.002 ± 0.001	0.936 ± 0.039
Swinging	0.136 ± 0.128	0.114 ± 0.138	0.119 ± 0.131	0.006 ± 0.004	0.886 ± 0.138

Other factors may contribute to the difficulty in recognizing certain classes of behaviors. For instance, the duration of behaviors can vary significantly; individuals may sit for extended periods, whereas running is typically brief, making short-duration behaviors harder to recognize. Additionally, the use of handheld video recordings can introduce variability in video quality, further complicating the recognition of fast-moving behaviors. Lastly, some behaviors, such as sitting or walking, are exhibited more uniformly across scenes, while others, like climbing down, may vary visually depending on the environment (e.g., descending a rope versus a platform).

Method Comparison: When comparing video-based and skeleton-based performance, we do not observe any statistically significant difference between them, as the 95% confidence intervals of all four metrics overlap. However, it is worth noting that the mean performance of the skeleton-based method is higher across each metric, with smaller variance, suggesting potentially more reliable performance under the experimental conditions.

A notable observation is the video-based model's tendency to misclassify 'climbing down' and 'swinging' as 'climbing up' (FPR 0.042 ± 0.01), a pattern less pronounced in the skeleton-based model (FPR 0.016 ± 0.007). Additionally, the video-based method exhibits extremely poor performance, with a lack of correct classifications for the 'climbing down' and 'running' classes. As shown in the confusion matrices in Fig. S14, the network predominantly predicts 'walking' instead of 'running' and 'climbing up' instead of 'climbing down.' While the exact causes of these misclassifications are difficult to determine, it is likely that the visual similarity between these actions leads the network to learn overlapping spatial representations, making it unable to effectively distinguish their temporal features. Notably, 'running' and 'climbing down' are among the most fast-paced behavioral classes in the dataset. Despite the overall low performance, the skeleton-based method achieves comparatively better results for these two classes.

4.5.2 Behavior Recognition: Within-Dataset and Cross-Dataset

In this section, we investigate how both video-based and skeleton-based methods perform within a single dataset and in a cross-dataset setting. For this analysis, we restrict the set of behavioral classes to the seven common classes present in both datasets, namely 'sitting', 'standing', 'walking', 'hanging', 'climbing up', 'climbing down', and 'running'. Our results are presented in Fig. 4.9. To emphasize the unique visual features of each dataset and simplify discussion, we refer to the ChimpBehave dataset as ZOO and the PanAf dataset as FOREST in this section.

4.5.2.1 Within-Dataset Results

Overall Observations: Similar to the findings in Sec. 4.5.1, we observe a discrepancy between the generally high Top-1 Accuracy and F1 Score and the relatively lower MCA and mAP. The former two metrics emphasize individual miniclips, while the latter two are averaged by the number of classes, giving each class equal weight. As previously discussed, this discrepancy can be attributed to unbalanced class distributions and certain behaviors being more difficult to differentiate. For further details, class-level metrics and confusion matrices are presented in Table T2 and Fig. S15.

Dataset Comparison: When comparing performance between ZOO and FOREST, both methods achieve statistically significantly better results in ZOO for Top-1 Accuracy and F1 Score. Three major visual factors may contribute to this result: (1) higher contrast between individuals and their background in ZOO, as seen in Fig. S12, (2) the overall higher image resolution in ZOO (1920x1080) compared to FOREST (720x404), and (3) the smaller

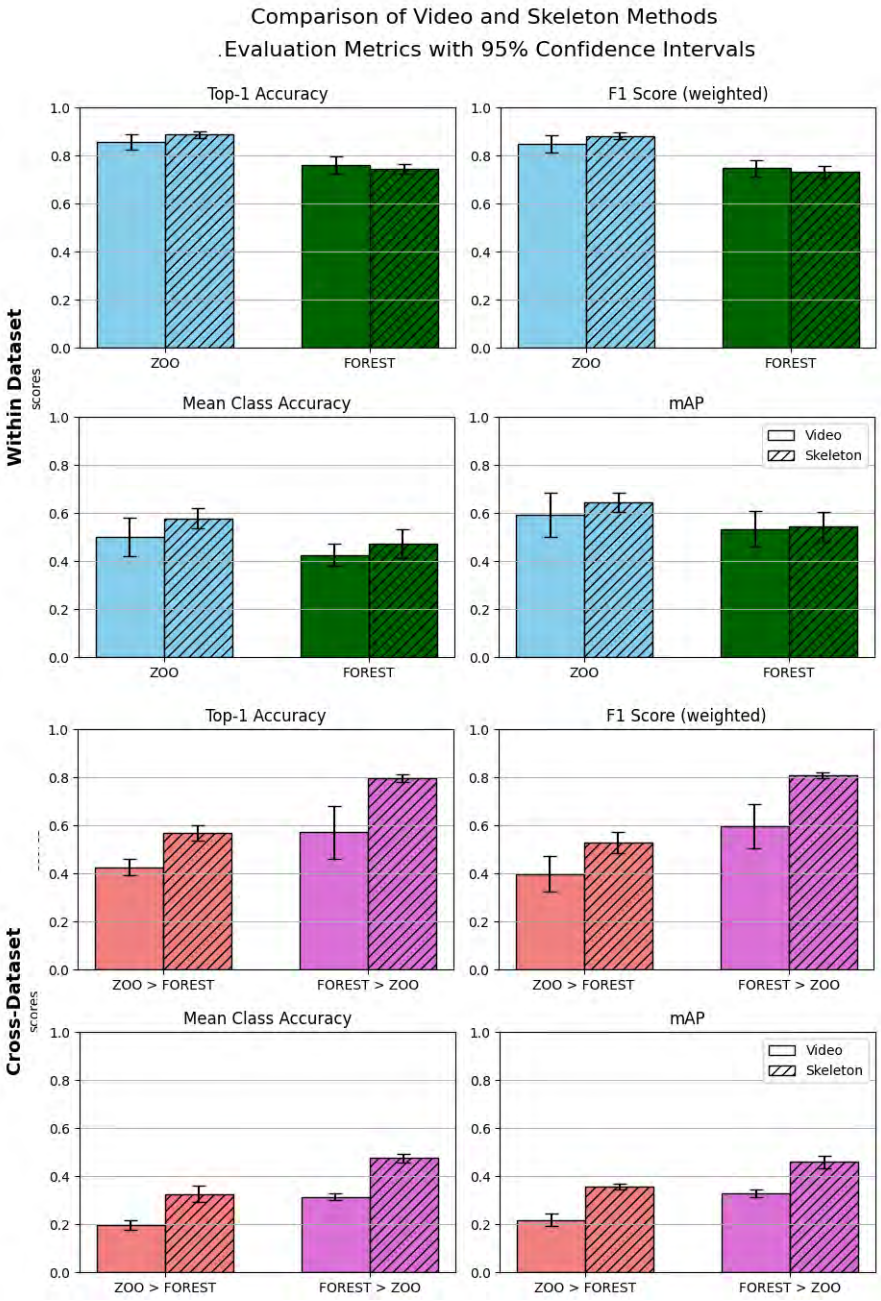

Fig. 4.9 **Evaluation metrics for ZOO (ChimpBehave, 7 classes) and FOREST (PanAf, 7 classes)**, within-dataset (*top row*) and cross-dataset (*bottom row*).

relative size of individuals in FOREST, as the camera is fixed, whereas focal sampling in ZOO follows each individual.

However, a different pattern emerges when examining MCA and mAP. For these two metrics, there is no significant difference between ZOO and FOREST for the video-based method, but the skeleton-based method achieves statistically higher scores in ZOO. This

suggests that while behaviors may not be easier to differentiate in ZOO or FOREST using full video data, they may be easier to distinguish when relying on skeletal pose information.
Method Comparison: When comparing video-based and skeleton-based methods, we do not observe any statistically significant difference in scores across any metrics. However, the skeleton-based method shows higher mean scores across all metrics in ZOO and higher MCA and mAP in FOREST. In contrast, the video-based method shows higher mean scores in FOREST for Top-1 Accuracy and F1 Score. While no statistical evidence supports the superiority of either method, it is worth noting that the pose estimation algorithm used by the skeleton-based method was neither trained nor fine-tuned on data from ZOO or FOREST. Fine-tuning on these datasets may potentially improve the classifier's performance.

4.5.2.2 Evaluating Generalization: Cross-Dataset Results

Overall Observations: In general, most models in the cross-dataset setting (i.e., trained on one dataset and tested on the other) exhibit much lower scores compared to within-dataset performance.
From ZOO to FOREST: This trend is particularly noticeable for video-based models trained on ZOO and tested on FOREST, where all metrics are reduced to about 50% of their within-dataset measurements. Similarly, the skeleton-based method shows a reduction of approximately 35%-40% across all metrics in this setting. This indicates that both methods suffer greatly from visual domain shifts between the datasets. However, while no statistical difference was observed between the methods in the within-dataset setting for ZOO, the skeleton-based method performs statistically significantly better across all metrics in the ZOO→FOREST cross-dataset setting.

The comparatively higher robustness of the skeleton-based approach aligns with previous observations in human-centric action recognition, where skeleton-based methods were found to be less sensitive to visual changes, such as variations in appearance, background and illumination [68]. This robustness is particularly relevant here, as the FOREST dataset includes nighttime infrared recordings and captures the behaviors of two great ape species, chimpanzees and gorillas.
From FOREST to ZOO: A similar downward trend is observed for the video-based model between the within-dataset and cross-dataset settings for FOREST→ZOO, although the drop is smaller (around 20%-33%). Interestingly, the skeleton-based method shows an increase in Top-1 Accuracy and F1 Score in the cross-dataset setting compared to its within-dataset scores, while MCA and mAP remain statistically tied. In the cross-dataset setting, the skeleton-based method outperforms the video-based method across all metrics.

Head Classes: As shown in Table T3 and Fig. S15, in the ZOO→FOREST setting, both video-based and skeleton-based methods tend to predict many instances as 'sitting', the predominant class in the training dataset. This is reflected by the extremely high average FPR for 'sitting' (VB: 0.516, SB: 0.534). Conversely, many test examples of the other two head classes, 'standing' and 'walking', are missed, as shown by their high average FNR (VB: 0.819, SB: 0.713 for 'standing'; VB: 0.710, SB: 0.648 for 'walking'). The individual average F1 scores for these head classes are: 'sitting' (VB: 0.587, SB: 0.703), 'standing' (VB: 0.224, SB: 0.390), and 'walking' (VB: 0.362, SB: 0.494).

In contrast, in the FOREST→ZOO setting, head classes are recognized much better overall, as evidenced by their average F1 scores: 'sitting' (VB: 0.720, SB: 0.924), 'standing' (VB: 0.324, SB: 0.685), and 'walking' (VB: 0.532, SB: 0.616). This may explain why the performance decline is more pronounced in ZOO→FOREST than in FOREST→ZOO.

Tail Classes: Examining class-level metrics for tail classes reveals a clear distinction between the methods. For the video-based method, none of the tail classes achieve more than 5% in average F1 Score in the cross-dataset setting. Specifically, 'hanging', 'climbing up', 'climbing down', and 'running' obtain average F1 scores of 0.032, 0.050, 0.015, and 0.009 in ZOO→FOREST, and 0.017, 0.037, 0.003, and 0.008 in FOREST→ZOO, respectively. In contrast, the skeleton-based method consistently achieves average F1 scores above 10% in both cross-dataset settings, with scores reaching up to 28.9% for 'climbing up' in FOREST→ZOO. For instance, the average F1 scores for 'hanging', 'climbing up', 'climbing down', and 'running' are 0.126, 0.201, 0.139, and 0.100 in ZOO→FOREST, and 0.266, 0.289, 0.191, and 0.151 in FOREST→ZOO.

4.6 Discussion

In this section, we discuss the results of our experiments, highlight some limitations - especially in the context of real-life applications - and make suggestions for future research.

The ChimpBehave dataset, introduced in Sec. 4.3, is currently the largest publicly available great ape-specific dataset for multi-class behavior recognition in terms of annotated video frames. Unlike previous datasets, its annotations are curated by expert primatologists, ensuring high-quality and contextually accurate labeling. Additionally, the dataset features significantly higher video resolution compared to existing resources. ChimpBehave also provides a novel platform for researchers to investigate methods in domain adaptation and out-of-distribution generalization, which could lead to the development of animal behavior recognition models that are more robust to visual and contextual variability.

However, this dataset has several limitations, the most significant being its unbalanced class distribution. While certain behaviors, like sitting and walking, naturally occur more frequently than others, such as hanging or running, their dominance in the data distribution may affect overall model performance, resulting in a tendency to overfit to the more prevalent classes. In the case of ChimpBehave, data augmentation techniques could help mitigate this class frequency disparity, such as oversampling underrepresented classes or creating miniclips with overlapping video frames. Another approach could involve implementing weighted loss functions that incentivize correct classification of less frequent behaviors. From a practical perspective, in real-life scenarios - such as an automated system monitoring animal health or well-being - one would want the system to reliably detect behaviors like running. However, in our experiments, models often confuse running with walking, which would be unsuitable for such applications. Future research should place special emphasis on behaviors that, while less frequently displayed, convey high ecological importance. Compared to other datasets that emphasize annotations of social interactions among conspecifics in great apes [124], our dataset focuses exclusively on locomotive behaviors of single individuals, which may limit its applicability to certain aspects of primatology.

Our methodology, as detailed in Sec. 4.4, employed two convolutional-based action recognition models, X3D and PoseConv3D, to evaluate the dataset. While we acknowledge the significant advancements in recent model architectures, particularly transformers, it is likely that newer state-of-the-art models could achieve superior performance. For instance, as demonstrated in [16], transformer architectures have outperformed CNN-based models on datasets like PanAf. However, we deliberately did not include transformers in our experiments to ensure a *fair* comparison between video-based and skeleton-based approaches, specifically within a context relevant to primatology.

In human-centric video action recognition, the remarkable performance of video-transformers in recent years can be largely attributed to their ability to leverage pretraining on massive datasets, often exceeding one million video samples, followed by fine-tuning on smaller downstream tasks [206]. Conversely, skeleton-based approaches are typically benchmarked on much smaller datasets, where ground-truth poses are often derived from sensor-based systems [175]. When accurate body poses are not part of the dataset, the performance of skeleton-based models generally lags behind video-based models. For instance, the best Top-1 Accuracy achieved by skeleton-based approaches on Kinetics-400 [96] does not exceed 40%, as reported in [218].

Given the ethical considerations inherent in primatology, particularly when working with great apes in natural settings, the availability of *large-scale* datasets annotated with both behaviors and highly accurate pose information remains highly improbable. Thus, our

focus was to evaluate methods that align with realistic, ethical, and contextually relevant conditions for studying primate behavior. Future work could explore the application of vision transformers, particularly those pretrained on large-scale human-centric datasets, to assess their potential in primate-centric tasks and their generalizability in scenarios with limited annotated data (see e. g. [188]).

While the results in Sec. 4.5 offer comparisons between video-based and skeleton-based behavior recognition models, they do not consider the potential of multimodal models that combine video and skeletal data. Indeed, PoseConv3D, the skeleton-based model used in our experiments, natively supports such a design choice, and this approach has been shown to achieve higher performance on multiple human action datasets [39]. Although we chose to keep both modalities separate for research purposes, we believe that combining them could yield higher performance. Furthermore, to the best of our knowledge, PoseConv3D is currently the only algorithm that has demonstrated success in skeleton-based behavior recognition of great apes [52]. Expanding the evaluation to include additional skeleton-based models could provide a more comprehensive assessment of this task and represents a promising direction for future work.

When examining model generalization in cross-dataset settings (Sec. 4.5.2.2), our results reflect performance on out-of-distribution data without any fine-tuning on the target test data. In practice, researchers often have access to at least some labeled data from the target domain. Future experiments could explore scenarios where fine-tuning is applied to test whether the generalization capacity observed in our experiments remains robust. Similarly, as noted in Sec. 4.5.2.1, the pose estimation model used in our skeleton-based approach was not fine-tuned on either the ChimpBehave or PanAf datasets. In practice, fine-tuning on these datasets is recommended, as the quality of estimated poses can significantly impact the final behavior classifier's performance [39]. Additionally, PanAf includes footage of both gorillas and chimpanzees, while ChimpBehave focuses exclusively on chimpanzees. Anatomical and behavioral differences between these species could pose challenges for algorithms trained solely on chimpanzee data. Future work could investigate within-PanAf performance differences between gorilla and chimpanzee footage, providing insights into species-specific model behavior and addressing the extent to which training exclusively on chimpanzees impacts cross-species generalization.

4.7 Conclusion

The field of animal behavior studies, like many other scientific domains, may be entering a new era where human effort is increasingly complemented by artificial intelligence. Tech-

nologies such as those presented in this chapter could become valuable tools for researchers by, for example, pre-annotating regions of interest in video clips or discarding irrelevant sequences in lengthy footage. While frequently observed behaviors such as walking, sitting, and standing can often be accurately recognized by automated systems, less common behaviors involving more complex body movements - such as climbing up or down - are still prone to misclassification. For many practical applications, this level of accuracy may not yet be sufficient for field deployment. However, studies on human actions have shown the potential of machine learning models to accurately recognize highly sophisticated activities when more data is available. As more annotated behavioral data of non-human primates becomes accessible, we believe that automated systems will play an increasingly important role in efforts to halt biodiversity loss and protect endangered species more effectively.

4.8 Ethical Statement

We received ethical agreement for this study from the Commission d'Ethique de la Recherche of the University of Neuchâtel (agreement number: 01-FS-2017) and the Kantonales Veterinäramt BS at Basel Zoo.

4.9 Acknowledgement

We extend our gratitude to the Basel Zoo, its staff, and its director, Adrian Baumeyer, for granting us the opportunity to conduct our data collection within their facilities.
Furthermore, we wish to express our appreciation to NCCR Evolving Language, Swiss National Science Foundation Agreement #51NF40_180888 and Grant No. CR31I3_166331 awarded to A.B. and K.Z. for their financial support in data collection and annotation.
Special thanks are owed to the members of the SIG *Ape Gestures*, including Daphné Bavelier, Richard Hahnloser, Nianlong Gu, and Remo Nitschke, for their contributions.

Chapter 5

Automated Recognition of Great Ape Gestures

The present chapter is based on a working draft paper, being prepared for submission:

Fuchs, M., Genty, E., Zuberbühler, K., and Cotofrei, P. (2025). Automated Recognition of Great Ape Gestures. Manuscript in preparation

Abstract

Great ape gestures provide critical insights into the origins and evolution of human language, showcasing intentionality and flexibility comparable to human communication systems. While these gestures span diverse contexts and modalities, their study has largely relied on labor-intensive manual video coding, limiting scalability and hindering broader research efforts. Recent advances in deep learning have facilitated automated recognition of animal behaviors, but the fine-grained recognition of great ape gestures remains an unexplored challenge due to the complexity of subtle, fast-paced movements.

To address this gap, we introduce *FineChimp*, the first dataset and benchmark specifically designed for the automated recognition of great ape gestures. Featuring approximately 2,000 multiview video samples annotated with 38 distinct gesture classes, *FineChimp* offers a unique resource for fine-grained multi-class action recognition. We present comprehensive benchmarks, evaluating state-of-the-art deep learning models, including a dual-stream VideoMAEv2 video transformer that achieves a mean Top-1 Accuracy of $65.1 \pm 2.2\%$ and a Mean Class Accuracy of $63.3 \pm 5.9\%$, significantly surpassing a near-random baseline.

5.1 Introduction

The study of great ape communication has garnered significant attention due to its relevance in understanding the origins and evolution of human language [30, 195, 160]. Unlike many animal communication systems, great ape signals are expressions of *intentions*, as observed in their vocalizations [171, 63, 148], and particularly in their use of gestural signals [194, 55, 19]. Notably, great ape gestures play a crucial role in their daily lives, helping them coordinate social interactions by influencing others' behavior or mental states [33] in ways comparable to human language [77, 76, 78]. These gestures span various contexts, such as initiating grooming, requesting food, or inviting play.

Like humans, great apes use gestures in flexible and goal-directed ways, sometimes conveying referential meanings, such as beckoning to invite others to approach [83, 58]. They even purposefully use gestures to interact with humans [24, 58]. The gestural repertoire

Fig. 5.1 **'Grabbing,' 'poking,' or 'grabbing and pulling'?** Chimpanzees use tactile gestures to communicate, much like humans. Each row of images corresponds to a distinct gesture, illustrating the visual complexity of differentiating them. From top to bottom, the depicted gestures are 'poke other,' 'grab,' 'grab,' and 'grab pull.'

of great apes is diverse, encompassing dozens of gesture types used across visual, audible, and tactile modalities [163, 55, 82], with significant overlap across species and even similarities to human infants [61, 60]. Studying great ape communication, especially identifying language-like features in their signaling systems, thus can provide valuable insights into the evolutionary foundations of human language.

Despite the wealth of knowledge gained through meticulous observation and manual video coding of great ape gestures [57], this approach has significant limitations. Video coding is highly labor-intensive, requires specialized expertise, and therefore restricts the scalability of research efforts. Automating this process represents a critical challenge with the potential to significantly advance our understanding of ape communication.

In recent years, deep learning-based technologies in computer vision have made substantial progress in understanding animal behaviors. These methods have enabled the automatic detection and recognition of various great ape locomotive and social behaviors in videos

[5, 167, 15, 52, 124, 17, 16, 50]. However, unlike broader behavioral categories, the automated recognition of great ape gestures presents unique challenges and has yet to be explored in the literature. Gestures often require a fine-grained understanding of subtle, fast-paced movements, with nuances that can elude even novice human observers (see Fig. 5.1).

To address this gap, we make the following contributions:

- We introduce *FineChimp*, a novel dataset and benchmark for the automated recognition of great ape gestures. The dataset includes approximately 2,000 samples (1 hour or 100,000 video frames) of multiview recordings with expert-level annotations across 38 non-overlapping gesture classes, offering a unique and challenging resource for fine-grained multi-class action recognition.
- We present the first benchmark for automated gesture recognition in great apes, exploring optimal model architectures, pretraining scenarios, hyperparameter tuning, data augmentation strategies, and the integration of RGB with optical flow. Our best-performing model, a dual-stream VideoMAEv2 video transformer, achieves a mean $\pm$ std Top-1 Accuracy of $65.1 \pm 2.2\%$ and a Mean Class Accuracy of $63.3 \pm 5.9\%$, significantly outperforming a near-random classifier.
- We open-source the design and software for a cost-effective multiview camera system, enabling automated and synchronized recordings with features like primate detection, high-quality video capture, and low power consumption.

All data, models, and code are available at: https://github.com/MitchFuchs/FineChimp

5.2 Related Work

5.2.1 Deep Learning for Animal Communication

Animal Behaviors: Deep learning has garnered increasing interest among animal behavior researchers in recent years, as techniques initially developed for human subjects have been successfully adapted to non-human animal species (e.g., [188, 201, 134, 157]). Specifically, various methods have proven useful when applied to chimpanzees for tasks such as face recognition [172], individual detection, tracking, and re-identification [124], pose estimation [36, 52, 213], and dense pose estimation [168]. Moreover, there has been a surge of research in animal action recognition aimed at automatically classifying chimpanzee behaviors from videos [17, 15, 124], skeleton-based data [52, 50], audiovisual signals [5], and videos paired with text [16].

Animal Communication: In the more specific context of automated studies of animal communication, deep learning has gained significant traction, particularly in the analysis of bioacoustic signals [184, 143]. For example, fundamental tasks such as vocalization detection and classification now benefit from multi-species datasets designed for model benchmarking [67], large transformer models capable of detecting vocalizations even with limited data for new species [65], and systems that are increasingly versatile across different animal species [10]. Additionally, deep learning has been applied to the analysis of vocalizations from birds [59], marine mammals [179, 71], meerkats [170], dogs [210], elephants [54], and other animals. Primate vocalizations have also received notable attention, with studies on lemurs [164], marmosets [169, 216], and chimpanzees [10]. Other significant work using *traditional* (i.e., non-deep) machine learning techniques includes research on marmosets [156], orangutans [41], and chimpanzees [37].

Animal Gestures: Despite these advances, none of the aforementioned literature addressed the classification of visual signals associated with animal gestural communication. Some studies employ statistical tools to categorize visual patterns of ape gestural communication [138] from a bottom-up perspective [64]. However, this approach relies on labor-intensive annotations of a selection of hand-crafted features rather than learning directly from video data. Furthermore, the video collections used in such work have not been made publicly available. To the best of our knowledge, FineChimp is the only publicly available video dataset specifically designed for the automatic recognition of gestures characteristic of animal communication.

5.2.2 RGB+Optical Flow for Action Recognition

Computer vision has traditionally relied on 2D-CNNs to extract meaningful *spatial* information from 2D images for tasks such as object detection and image classification. By extension, a video can be seen as a sequence of successive static images, i.e., 2D images distributed along a temporal axis. Consequently, it is not surprising that 2D-CNN architectures were among the first to be adapted for extracting *temporal* information in video-based action recognition. Early approaches applied 2D-CNNs to individual video frames, averaging the action class predictions across frames, or by stacking multiple frames as input to the model [95]. However, these methods were limited in their ability to effectively learn temporal patterns, which led researchers to explore the combined use of RGB and optical flow.

Optical flow represents *motion* in a video, capturing the temporal information of spatial changes between successive frames. It is computed as a displacement vector field, indicating the motion of each pixel between consecutive frames. This vector field comprises n two-dimensional vectors (where n is the number of pixels in an image), each with horizontal and

vertical components that can be represented as two single-channel images. These images can then serve as input for computer vision algorithms to enhance action recognition performance.

Several foundational works in CNN-based human action recognition have highlighted the utility of dual-stream architectures, where models learn from both RGB and optical flow features in parallel. For instance, the two-stream 2D-CNN architecture introduced by [181] inspired a substantial body of research, establishing the use of optical flow alongside RGB information as a common practice. Another groundbreaking development was the introduction of I3D [23], which extended a similar dual-stream architecture with the power of 3D-CNNs.

The interest in optical flow has since remained strong in the computer vision community. Researchers have worked on making its use more computationally efficient, such as by employing knowledge distillation [81] to reduce complexity at test time [32, 185], or by leveraging attention mechanisms on optical flow-augmented data [114]. In fine-grained human action classification, motion features derived from optical flow have been shown to be increasingly important for distinguishing subtle differences in actions. In contrast, appearance features extracted from RGB data are more effective for recognizing broader, coarser activities [176].

In primatology, optical flow has also proven effective for behavior recognition. Studies such as [167, 15] have demonstrated that using optical flow improves the classification accuracy of great ape behaviors observed in their natural habitats. Similarly, [113] showcases its utility in recognizing behaviors of cynomolgus monkeys in laboratory settings.

Following this line of research, our final model adapts VideoMAEv2 [206] into a dual-stream RGB+Optical Flow architecture.

5.2.3 Related Datasets

Video Datasets of Great Apes: The advent of deep learning-based technology for human-centric video understanding has sparked growing interest among researchers in recognizing great ape behaviors using computer vision. In recent years, at least four public video datasets of great ape actions have been proposed: PanAf500 [167], PanAf20K [17], ChimpACT [124], and ChimpBehave [50]. With the exception of PanAf20K, the other three datasets were used in our experiments and are described in Sect. 5.4.1.

FineChimp distinguishes itself from its predecessors through its focus on visual animal communication and fine-grained classification task. Its behavior video clips are notably short, averaging just 1.7 seconds (with a median of 1.1 seconds), allowing for the precise capture of distinct actions compared to the much longer clips in previous datasets, such as ChimpBehave (with an average length of 6.2 seconds). FineChimp also offers a significantly

broader scope, featuring 38 action classes—far surpassing the 8, 9, 18, and 23 classes included in ChimpBehave, PanAf500, PanAf20K, and ChimpACT, respectively. Moreover, FineChimp uniquely emphasizes actions tied to visual animal communication, unlike datasets such as ChimpACT, where social interactions account for less than 35% of the annotations.
Animal Multi-Views: Using multiview data streams for human activity understanding has a long history, culminating in the development of large-scale datasets for model benchmarking, such as NTU RGB+D 120 [120]. This dataset features 8 million video frames related to 120 human actions, captured from up to 155 different viewpoints using various sensors (e.g., RGB+D, 3D joints, and infrared signals). However, such extensive data collection with highly accurate measurements is rarely feasible for non-human animal datasets.

Nonetheless, research on animal behavior has increasingly leveraged multiview video setups, primarily for laboratory rodents (e.g., [85, 128]) but also for other species, including pigs [3], horses [112], cheetahs [91], pigeons [145], songbirds [166], and even underwater species using stereo recordings [40]. Similarly, studies incorporating multiview data for non-human primates have focused on Japanese macaques [135, 146], rhesus macaques [8, 131, 127], crab-eating macaques [113], and marmosets [92, 228, 29]. As far as we are aware, FineChimp is the first public available dataset to include multiview videos of great apes.

5.3 The FineChimp Dataset

In this section, we present the details of the FineChimp dataset, highlighting its main features, including dataset statistics, challenges for machine learning applications, and a description of the filming setup, data collection, and annotation process.

5.3.1 Dataset Description

Overview: FineChimp consists of 1,996 video clips, each labeled with one of 38 gesture classes typical of chimpanzee gestural communication. All recordings were captured using a multi-view camera setup installed in the indoor chimpanzee enclosure at Basel Zoo, Switzerland. The dataset comprises 57 minutes of footage, with clips lasting an average of 1.7 ± 2.9 seconds and a median duration of 1.1 seconds. Captured at 30 frames per second (FPS), the dataset contains over 100,000 individual frames, with each clip including an average of 51 ± 87 frames and a median of 32 frames. The shortest clip contains 8 frames (corresponding to a duration of 0.26 seconds), whereas the longest clip contains 1690 frames

(i.e. 56.3 seconds). Figure 5.2 illustrates the distribution of the number of clips per number of frames.

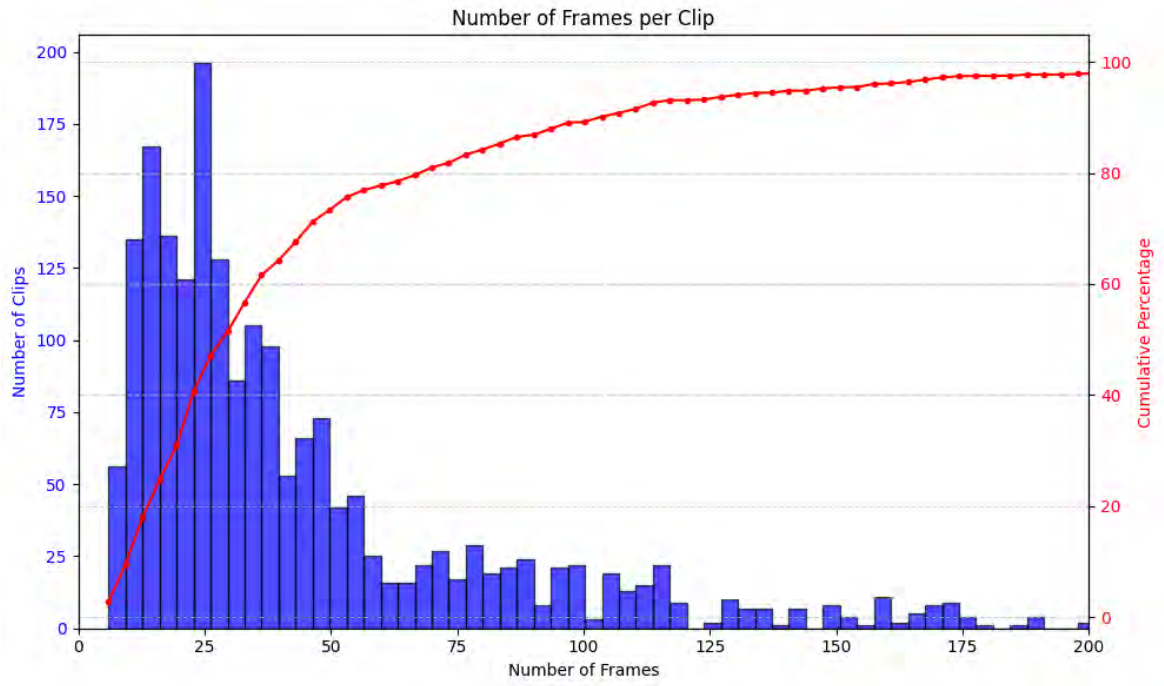

Fig. 5.2 **Distribution of the number of video clips by the number of frames in the FineChimp dataset**. Approximately 50% of the clips include less than 32 frames (corresponding to a duration of 1.1 seconds), highlighting the challenge of this classification task.

Classes of Gestures: Each video clip in FineChimp is assigned to one of 38 gesture classes, as detailed in Table T4 in Appendix D. These classes were selected based on the corresponding annotations, ensuring that each class contains a minimum of five video clips with at least eight frames per clip.

Class Distribution: The number of items per class in FineChimp follows a typical long-tail distribution, as shown in Fig. 5.3. The two most frequent classes, 'grab' and 'touch,' account for 23.1% and 12.6% of the data, respectively (see examples in Fig. 5.4). Conversely, the 20 least frequent classes each represent less than 1% of the dataset.

Multiview Data: Each gesture may appear in up to five video clips, as the scene is simultaneously recorded from five distinct viewpoints (see Section 5.3.2 for details).

Image Quality: Video clips in FineChimp were originally recorded at a resolution of 640x480 pixels at 30 FPS in H.264 format and later re-encoded to AVI. However, the clips included in the dataset are typically smaller than 640x480, as they were cropped to focus on the region of interest, i.e., the gesture and its protagonist(s) (see Section 5.3.4 for further details).

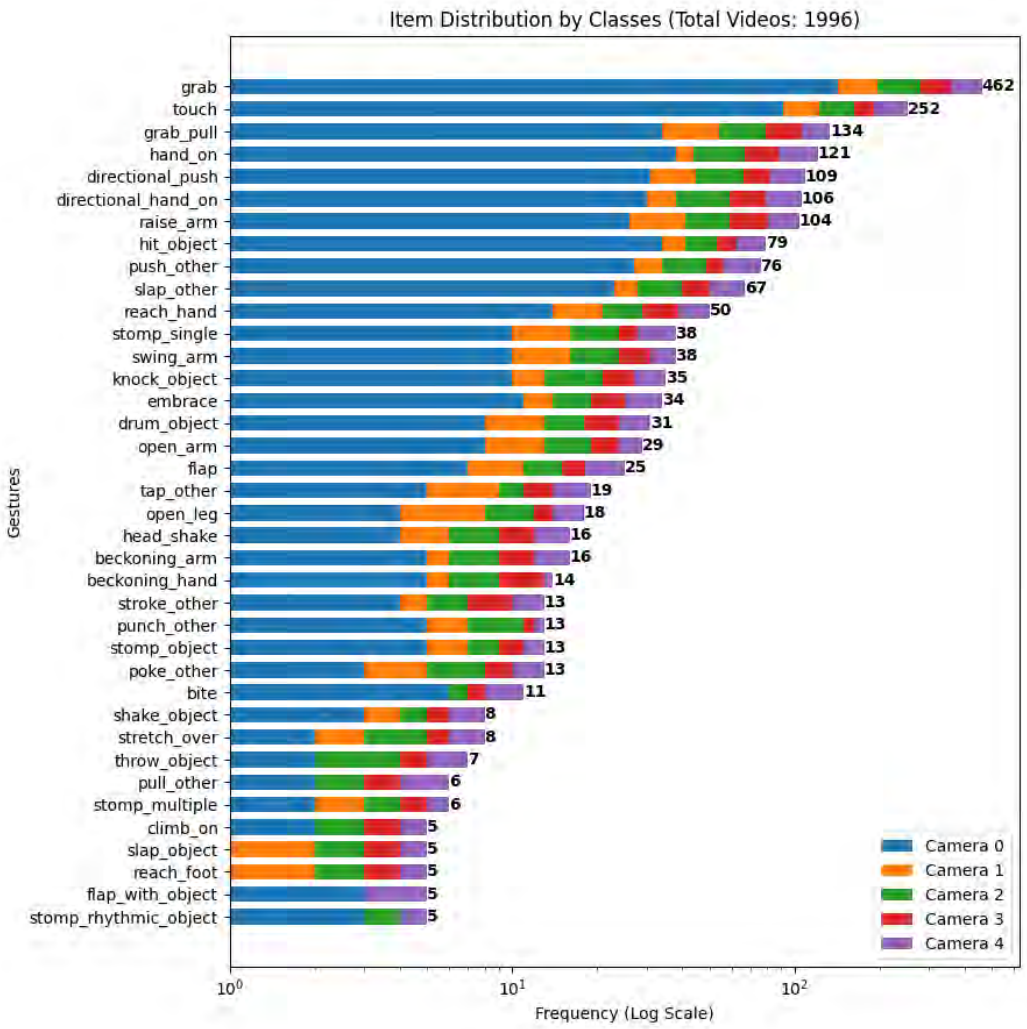

Fig. 5.3 **Long-tail class distribution in the FineChimp dataset**. The dataset comprises 38 classes of fine-grained actions, representing typical great ape communicative gestures. Collected across five viewpoints, it includes nearly 2,000 video clips.

5.3.2 Zoo Installation

Chimpanzee Group and Housing Conditions: The information on the chimpanzee group structure and about the facility residence at Basel Zoo were detailed in Section 4.3.

Camera Installation: Six fixed cameras were installed in the indoor enclosure, all directed toward the same area to capture the same scene from multiple angles. One camera provides a bird's-eye view and serves as the *detector*, while the other five record from side angles. Fig. 5.5 shows a representation of the scene, whereas Fig. 5.6 shows an example of a gesture class captured from three different viewpoints.

Fig. 5.4 **Examples of 'touch' (top) and 'grab' (bottom)**. These two distinct gestures often differ only in the subtle closing movement of the fingers on the receiver's body. These two classes are the most frequent in the dataset, collectively accounting for over 35% of all annotations.

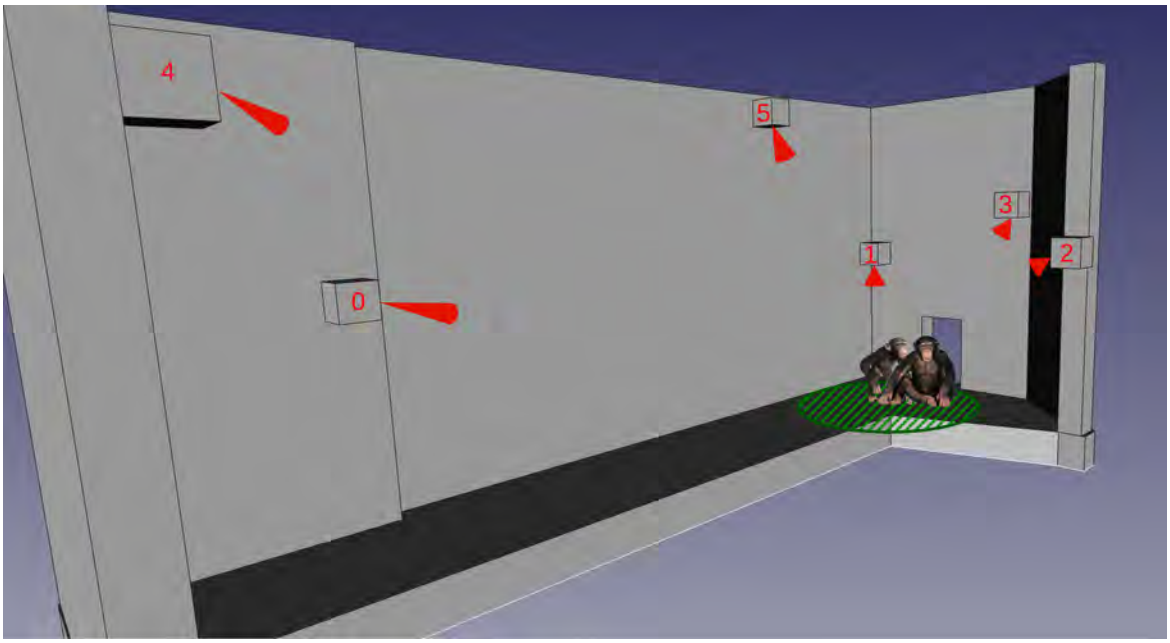

Fig. 5.5 **3D representation of the camera setup**. All cameras (red cones) are directed toward a central scene (green disk) covering approximately 4m². Camera 0 is used for manual gesture annotations, while Cameras 1 to 4 augment the number of viewpoints. Camera 5 functions solely as a *detector*, signaling the presence or absence of chimpanzees in the scene to the other cameras.

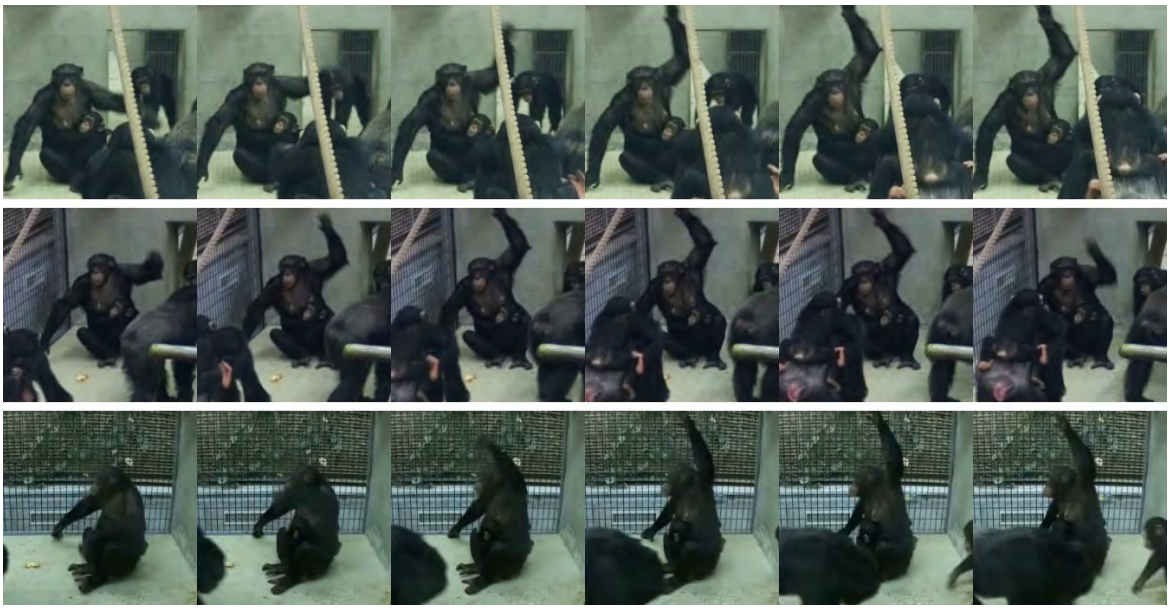

Fig. 5.6 **Example of an 'raise arm' gesture captured from three different viewpoints**: from top to bottom, Cameras 0, 4, and 2.

Hardware: Each camera setup includes a Raspberry Pi 4B (8GB RAM) and a HQ camera with a 12.3 MP Sony IMX477 sensor. Four cameras use 6mm lenses (3MP resolution, F1.2 aperture, 63° field angle), while two use 16mm telephoto lenses (10MP resolution, F1.4–16 aperture, 44.6°–21.8° field angle). All Raspberry Pis are connected via Ethernet to a shared NAS for storage.
Software: The system uses a custom software pipeline for automated recording. The *detector* continuously runs YOLOv8 [90], trained on the OpenMonkeyChallenge dataset [224]. When a chimpanzee is detected, a *start* signal is sent via UDP to the other cameras to begin recording. When no individuals are detected, a *stop* signal is sent to end recording. The software is available on the project repository.

5.3.3 Data Collection

Raw Data Collection: From Nov. 14, 2023, to Jan. 11, 2024, the system recorded approximately 6,770 raw video clips per viewpoint, resulting in a total of 33,850 multiview clips. For each camera, the total recording duration was approximately 21 hours (equivalent to 2.25 million frames), with an average clip length of 11.1 seconds.
Clips of Interest: To reduce manual screening time, we employed computer vision techniques to preselect clips of interest from Camera 0. Specifically, we used the QDTrack multiple object tracking model [152], which was initially pretrained on the ChimpACT

dataset [124] and subsequently fine-tuned on approximately 9,000 video frames of our own data, labeled with chimpanzee tracks. Applying this model to all clips from Camera 0 allowed us to preselect 2630 clips, based on the criterion that at least two individuals were tracked for a minimum of 30 consecutive frames.

5.3.4 Data Annotation

Gesture Annotations: A primatologist (EG) with expertise in great ape gestures identified and labeled 750 communicative gestures across 460 recordings. Following the coding scheme introduced in [57], the recordings were annotated in ELAN (https://archive.mpi.nl/tla/elan) and subsequently trimmed to include only the segments where the gestures were observed.
Multiview Augmentation and Cropping: The same temporal segments were trimmed from the four additional viewpoints (i. e. cameras 1 to 4). A single reviewer (MF) manually inspected all segments and included them in the dataset only if the gesture was visible. Segments were cropped to focus on the gesture and interacting individuals, minimizing background and irrelevant elements (see the examples from Fig. 5.7).

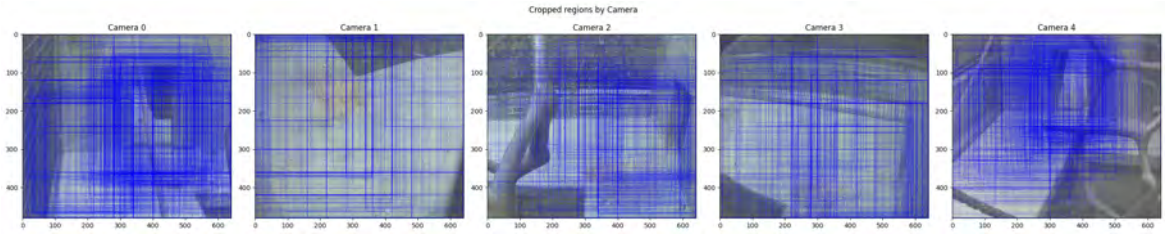

Fig. 5.7 **Examples of cropped regions by camera**. Each blue rectangle illustrates the extracted cropped region for video clips in the dataset, focusing on the area of interest while excluding irrelevant background elements.

5.3.5 Challenges for Action Recognition

Identifying and annotating great ape communicative gestures is a highly specialized task in primatology, often requiring months or even years of expertise. Automating this process poses significant challenges for computer vision algorithms but also provides an opportunity to advance technology to match human-level performance in scientific domains.

Some challenges specific to *FineChimp* dataset include:

- *Short and Fast-Paced Clips*: Some video clips are very short, often containing fewer than 10 frames, with rapid movements that are difficult to capture.

- *Subtle Class Differences*: Differentiating certain gestures can be visually subtle or even time-dependent. For example, 'grab' and 'touch' are distinguished only by finger actions (Fig. 5.4), while 'touch' and 'hand on' involve the same action but differ in the duration of contact (prolonged beyond two seconds for the latter).
- *Long-Tailed Class Distribution*: The dataset's long-tail distribution (Fig. 5.3) poses challenges for balancing accuracy across classes.
- *Typical Animal Dataset Limitations*: Issues such as partial occlusions, indistinct appearances among individuals, and auxiliary motions can complicate automated classification.

5.4 Method and Experiments

This section details the methodology employed to implement the task of automated recognition of great ape gestures. Several strategies are considered for evaluation, related to the weight initialization, model architecture, frame sampling, data augmentation, and cross-view generalization.

5.4.1 Datasets

In addition to FineChimp, five additional datasets were involved in task implementation, as detailed below:

OpenMonkeyChallenge: This dataset comprises 111,529 images of 26 non-human primate species, designed for 2D pose estimation [224]. The images originate from diverse sources, including the internet, three National Primate Research Centers in the USA, and a multiview camera setup within the Japanese macaques' enclosure at the Minnesota Zoo. All images have a resolution of at least 500x500 pixels. The bounding box annotations from this dataset were used to train the chimpanzee *detector*, as described in Section 5.3.2.

ChimpACT: ChimpACT contains 163 videos totaling 160,500 frames, annotated for chimpanzee detection, tracking, pose estimation, and action recognition [124]. The videos were recorded at resolutions of either 720x578 or 1280x720 pixels at 25 FPS, with a total duration of approximately 1 hour and 47 minutes. Captured at the Leipzig Zoo in Germany, the recordings used *focal sampling*, where videographers focused on a single individual. A unique aspect of this dataset is its longitudinal documentation of a young male chimpanzee's developmental trajectory over several years. The tracking annotations in ChimpACT were used to pretrain the QDTrack multi-object tracking model, as described in Section 5.3.3.

PanAf500: This dataset consists of 500 video clips, each 15 seconds long, amounting to a total duration of 2 hours and 5 minutes corresponding to 180,000 frames at 24 FPS with a resolution of 720x404 pixels [17]. The footage was collected via fixed cameras in the forests of Central Africa, capturing nine behaviors of chimpanzees and gorillas in their natural habitat. PanAf500 was instrumental in pretraining an action recognition model (see Section 5.4.3.1).
ChimpBehave: ChimpBehave includes approximately 1,300 video clips annotated for chimpanzee behavior recognition, amounting to 2 hours and 20 minutes (215,000 frames) at 25 FPS with a resolution of 1920x1080 pixels [50]. Recorded at the Basel Zoo, the footage focuses on the same chimpanzee group mentioned in Section 5.3.2. Unlike FineChimp, ChimpBehave was filmed using *focal sampling* with a hand-held camera from the visitors' area. The dataset covers eight common chimpanzee behaviors, seven of which overlap with those in PanAf500, enabling cross-dataset generalization studies. This dataset was also used for model pretraining, as detailed in Sec. 5.4.3.1.
Kinetics-400: Kinetics-400 is a large-scale action recognition dataset consisting of 400 human action classes, each containing at least 400 video clips [96]. The footage, sourced from YouTube, comprises approximately 306,000 video clips, each 10 seconds long, totaling around 850 hours of content. This dataset was utilized to pretrain the optical flow-based models detailed in Section 5.4.6.

5.4.2 Evaluation Metrics

We selected the following metrics to evaluate the performance of our models (see Appendix E for formulas).
Gesture Recognition Metrics: At the gesture class level, we calculated:

- *Precision*: The proportion of correctly predicted clips of a class among all clips predicted as that class.
- *Recall*: The proportion of clips of a class correctly predicted by the model.
- *F1 Score*: The harmonic mean of Precision and Recall, balancing both metrics to account for their trade-off.
- *False Positive Rate (FPR)*: The proportion of clips that do not belong to a class but incorrectly predicted as belonging to it.
- *False Negative Rate (FNR)*: The proportion of clips of a class incorrectly predicted by the model.

Dataset-Level Metrics: At the dataset level, we evaluated each model using the following metrics:

- *Top-1 Accuracy (Top-1 Acc)*: The proportion of clips correctly classified into their respective classes (metric sensitive to class imbalance).
- *Top-3 Accuracy (Top-3 Acc)*: The proportion of clips where the correct class is among the top three predictions.
- *Mean Class Accuracy (MCA)*: The average classification accuracy across all classes, assigning equal weight to each class regardless of its frequency in the dataset.

5.4.3 Model Pretraining, Architecture, and Frame Sampling

In the following subsections, we first evaluate the impact of various weight initialization strategies, then benchmark several action recognition model architectures, and finally compare frame sampling strategies of different lengths.

5.4.3.1 Model Pretraining

Weight initialization plays a critical role in the performance of action recognition models [181, 23]. Pretrained weights, especially from large-scale datasets, can improve results significantly when training data is limited and accelerate model convergence [74]. To assess the impact of pretraining, we identified and compared five weight initialization scenarios: training from scratch (no pretraining), three variants of weights pretrained on non-human primate (NHP) behavior recognition datasets, and weights pretrained on the Kinetics-400 human action recognition dataset. For a first evaluation, in each scenario the same model (X3D) was trained and evaluated for Top-1 Acc, Top-3 Acc and MCA metrics. All experiments used identical hyperparameters, as detailed in Table 5.1.

Given the emergence of video datasets for great ape behavior recognition [50, 17, 124], we hypothesized that pretraining on these datasets, which contain coarser behavioral annotations, could improve FineChimp's fine-grained action classification. Specifically, we evaluated:

- *No Pretraining*: Training the model from scratch, without any weight initialization. While this approach is less common, it can sometimes achieve competitive results [74].
- *Pretraining on ChimpBehave:* Initializing weights from the best checkpoint of a model trained on the ChimpBehave dataset [50]. This dataset focuses on behaviors of the same chimpanzee group as FineChimp, recorded in the same indoor zoo environment.

The model was trained for 23 epochs on eight behaviors ('walking,' 'standing,' 'sitting,' 'running,' 'climbing up,' 'climbing down,' 'hanging,' and 'swinging'), achieving Top-1 Accuracy: 81.68, Top-3 Accuracy: 95.02, and MCA: 37.8 on ChimpBehave.

- *Pretraining on PanAf:* Initializing weights from the best checkpoint of a model trained on the PanAf500 dataset, which documents behaviors of wild chimpanzees and gorillas in African forests [50]. The model was trained for 38 epochs on seven behaviors ('walking,' 'standing,' 'sitting,' 'running,' 'climbing up,' 'climbing down,' and 'hanging'), achieving Top-1 Accuracy: 73.71, Top-3 Accuracy: 96.07, and MCA: 39.23 on PanAf.

- *Pretraining on ChimpBehave+PanAf:* Combining ChimpBehave and PanAf datasets to increase diversity and size. The datasets were merged along their seven common behavior classes, and the same train/test split and hyperparameters from [50] were used. The model achieved Top-1 Accuracy: 81.45, Top-3 Accuracy: 97.02, and MCA: 51.54 on the merged dataset.

- *Pretraining on Kinetics-400:* Initializing weights from a model trained on Kinetics-400 [96], a large-scale human action dataset. The model achieved a Top-1 Accuracy of 76.0 and a Top-5 Accuracy of 92.3 on Kinetics-400 [43].

5.4.3.2 Model Architecture

We benchmarked several action recognition models, including three CNN-based architectures and two transformer-based architectures.

- *X3D:* X3D [43] is a CNN-based architecture specifically designed for efficient action recognition. It achieves state-of-the-art performance with a compact model size and computational efficiency.

- *SlowFast R50 and R101:* SlowFast [44] employs a dual-path architecture that processes video data at different temporal resolutions, effectively capturing both slow and fast motions. We benchmarked two variants, R50 and R101, which differ in their backbone complexity.

- *TimeSformer:* TimeSformer [11] is a transformer-based architecture that applies attention mechanisms across both spatial and temporal dimensions, achieving high accuracy on various action recognition benchmarks.

Table 5.1 **Model benchmarking hyperparameters.** Values marked with an asterisk (*) are considered default parameters in the following sections and will only be mentioned when they are set at a different value.

Model Architecture	**X3D***	**SlowFast**	**SlowFast**	**TimeSformer**	**VideoMAEv2**
Architecture Details	M*	ResNet50	ResNet101	Space only	Base
Batch Size	8*			4	8
Initial Learning Rate	0.01*			0.005	
LR Scheduler	MultiStepLR*				
Optimizer	SGD*				
Momentum	0.9*				
Weight Decay	0.0001*				
Number of Epochs	50*				
Frame Sampling	Uniform*				
Number of frames	8 frames*				16 frames
Data Augmentation	No*				
Pretrained Weights	Kinetics400*				
Modality	RGB*				

- *VideoMAEv2:* VideoMAEv2 [206] builds on the Video Masked Autoencoder (VideoMAE) framework, introducing enhancements that improve representation learning for video data, especially in transformer-based architectures.

The list of the hyperparameters considered by these models and their default values are provided in Table 5.1.

5.4.3.3 Frame Sampling

Despite the significant variation in clip lengths within the dataset (see Fig. 5.2), action recognition models require input of predefined dimensions, typically including batch size, number of RGB channels, frame width, frame height, and the number of video frames. In this study, we fixed the first four dimensions to 8, 3, 224, and 224, respectively, while experimenting with different frame counts to determine the optimal choice: 4, 8, 16, 24, or 32 frames.

It is important to note the practical trade-off between the number of frames and computational cost. As for *which* frames to select, two common strategies are *fixed-stride sampling* and *uniform sampling*. Fixed-stride sampling selects frames evenly spaced throughout each clip, whereas uniform sampling selects frames randomly within t subsets of frames uniformly distributed across the clip (where t is the number of frames).

In our experiments, we adopted uniform sampling for the following reasons: (i) Previous research in fine-grained human action recognition shows that uniform sampling outperforms fixed-stride sampling for clips of varying lengths [39]. (ii) Uniform sampling allows the model to observe the action in its entirety, avoiding potential biases from fixed-stride sampling, which may emphasize the beginning of an action. (iii) The randomness of uniform sampling ensures that the same clip can present different frame subsets during each training epoch, introducing variability and reducing overfitting.

5.4.4 Data Augmentation

State-of-the-art action recognition models are often trained on large-scale datasets with hundreds of classes, each containing hundreds of training clips. For instance, the latest Kinetics-700 dataset includes 700 classes, each with 450 to 1,000 clips, totaling over 545,000 videos [22]. In contrast, FineChimp contains approximately 1,600 training clips, which poses risks of overfitting and reduced generalization.

Additionally, the dataset's class distribution is unbalanced. For example, 'grab' and 'touch' each have over 250 examples, whereas 10 classes have fewer than 10 clips. This imbalance can lead models to overfit to frequent classes and perform poorly on rarer ones [50].

To address these issues, we employed data augmentation techniques to expand the training set. These methods are categorized into three types: geometrical, non-geometrical, and temporal augmentation. Each model's training pipeline also includes standard augmentations such as centered cropping and flipping, which are common practices not detailed here.

5.4.4.1 Geometrical Data Augmentation

Geometrical augmentations involve transformations that alter the spatial characteristics of an image, such as rotations, translations, and affine transformations [105]. We implemented the following techniques:

- *Rotation:* Each training clip is augmented with random rotations of up to $\pm 10^{\circ}$.
- *Affine Transformations:* Each training clip undergoes random rotations ($\pm 10^{\circ}$) combined with affine transformations, such as scaling within the range [0.8, 1.2].

5.4.4.2 Non-Geometrical Data Augmentation

Non-geometrical augmentations modify the visual properties of images, such as color space, noise injection, or cropping [105]. We tested three scenarios:

- *+2 Crops:* Each training clip is augmented with up to two additional random crops. For 'touch' and 'grab,' only one crop is added to balance their high frequency in the dataset.
- *+5 Crops:* Each training clip is augmented with up to five additional random crops.
- *Noise and Color Jittering:* Each training clip is augmented with one version that includes random noise and color jittering.

5.4.4.3 Temporal Data Augmentation

For video data, temporal augmentation generates new training examples by subsampling video frames in various patterns [25]. We created additional 16-frame clips from videos with at least 32 frames, ensuring minimal redundancy. Note that this method does not preserve the original class distribution since classes with more clips exceeding the 32-frame threshold will produce more augmented samples. Three subsampling strategies were tested:

- *Random Sampling:* Generates one 16-frame clip by randomly selecting frames without replacement.
- *Sliding Windows:* Generates one 16-frame clip by selecting consecutive frames starting at a random position.
- *Stratified Sampling:* Generates one 16-frame clip by sampling frames distributed across the beginning, middle, and end of the clip.

5.4.5 Cross-View Generalization

Video-based action recognition models are often view-dependent, with accuracy deteriorating when the model encounters videos captured from novel angles. For instance, a gesture such as raising an arm may appear visually distinct depending on whether the individual is facing the camera or turned away. Additionally, variations in lighting or background can further degrade model performance [50].

Efforts in human action recognition have aimed to address these challenges by developing models that learn view-invariant action representations [174]. Such research has also inspired the creation of large-scale, multimodal, multi-view datasets, including RGB, RGB-D, 3D skeletons, and infrared modalities, as seen in NTU RGB+D 120 [120].

We established a baseline for cross-view generalization using the protocol described in [175]. For each camera, all items from that camera were used for testing, while items from

the remaining cameras were used for training. This process was repeated for all five cameras in FineChimp. Note that the number of test examples varies across viewpoints, as certain gestures may not be visible from all angles and that some viewpoints do not include all gesture classes.

5.4.6 Multimodality: Optical Flow

Building on the demonstrated value of incorporating optical flow alongside RGB data, both in fine-grained human action recognition and primatology (see Sect. 5.2.2), we extended our experiments by introducing a second VideoMAEv2 network as a parallel *temporal* data stream. The main implementing steps are detailed in the following.

Data Preprocessing: Each video clip in FineChimp (without data augmentation) was converted to optical flow using the TV-L1 algorithm [229] implemented in OpenMMLab's DenseFlow [209]. The resulting optical flow was resized to 224x224 pixels, ensuring consistency with the RGB input dimensions.

Model Pretraining: Proper weight initialization is critical for VideoMAEv2, as training from scratch is not feasible given the limited data in FineChimp. To address this, we leveraged the optical flow of the Kinetics-400 dataset as a large-scale pretraining resource. First, Kinetics-400 was converted to optical flow using the same method described above. We initialized the VideoMAEv2 model with weights pretrained on *RGB* Kinetics-400. Since the optical flow input has only two channels (compared to three for RGB), the initial weights of the first layer (patch embeddings) were adjusted by averaging the RGB channel weights. Second, the network was then fine-tuned on Kinetics-400 optical flow for 50 epochs. On the Kinetics-400 validation set, the model achieved a Top-1 accuracy of 30.0%, Top-5 accuracy of 55.5% and MCA of 30.0%.

Model Fine-Tuning on FineChimp: Several tests were conducted to determine the optimal fine-tuning hyperparameters. The final optical flow VideoMAEv2 model was fine-tuned for 100 epochs on 2 GPUs, each with a batch size of 8. The initial learning rate was set to 0.01, with a momentum of 0.9 and a weight decay of 0.001, using cosine annealing for learning rate scheduling. Additionally, a dropout ratio of 0.5 was applied to the model head for regularization. All other parameters were left at their default values.

Late Fusion: To integrate predictions from the RGB and optical flow streams, we employed late fusion. Weighted averages of the class prediction distributions from both streams were calculated using a weighting factor λ, where $\lambda \in [0,1]$ represents the weight given to the RGB stream, and $(1-\lambda)$ corresponds to the optical flow stream. Results for different values of λ are discussed in Sect. 5.5.4.

5.4.7 Experimental Protocol

Cross-Validation: We employed a standard stratified 5-fold cross-validation protocol, where the dataset is divided into five randomly distributed folds, each comprising 20% of the data. The initial distribution of gesture classes is maintained across all folds to ensure consistency. For each experiment, training and testing are conducted five times, such that in each iteration one fold is used for testing and the remaining merged four folds are used for training. This ensures that every clip in the dataset is used exactly once for testing.

Confidence Intervals: All metrics are reported with 95% confidence intervals, calculated using the Student's t-distribution with degrees of freedom $\nu = 4$. Consequently, if for two strategies, the confidence intervals for the same evaluation metric do not overlap, the strategy having the best evaluation value is considered statistically significantly more efficient than the second strategy.

Optimization Strategy: From a theoretical viewpoint, identifying the best set of parameters' value which optimizes the performance of a particular model can be done by applying a "grid" approach, on the space of all possible combinations of parameters' values. We identified several factors which may influence the performance of an action recognition model, as pretraining (5 strategies), architecture (5 choices), number of sampled frames (5 values), or data augmentation (8 transformations). An optimization strategy based on a grid search would imply, only for these factors, the evaluation of a number of $5 \times 5 \times 5 \times 8 = 1000$ experiments (supposing that all others hyperparameters are keep constant), requiring a huge amount of resources (particularly computational power). Therefore, for practical considerations, we decided to apply a "greedy" optimization strategy, in which the optimal value of a particular factor is determined through a series of experiments independently of other factors.

Experimental Hardware Details: All training and evaluation procedures were conducted using the MMAction2 platform [141]. Most model training was performed on the HPC cluster at the University of Neuchâtel, using 4 NVIDIA RTX 2080 Ti GPUs (each with 11GB of memory). An exception was made for the VideoMAEv2 model, which was trained at the University of Zurich on 2 NVIDIA A6000 GPUs (each with 49GB of memory). The final model for each experiment was selected based on its Top-1 accuracy on the validation set.

5.5 Results

This section discuss and analyze the performances obtained during multiple series of experiments, where each series was designed, according to the "greedy" strategy, to identify the best value of a particular factor, as detailed in Sect. 5.4.7.

5.5.1 Performances of Model Pretraining, Architecture, and Frame Sampling

In the following, we focus on the analysis of the effects of various scenarios concerning the model pretraining, model architecture, and frame sampling.

5.5.1.1 Model Pretraining

Training from Scratch: Given the complexity of distinguishing subtle ape gestures, the amount of data in FineChimp is insufficient to train X3D from scratch effectively. Specifically, the mean Top1-accuracy of 24.8 barely exceeds the proportion of the most frequent class in the dataset ('grab' with 23.1%). Furthermore, the average MCA is 3.2, which approaches the lowest possible value of 2.6 (i.e., $100/38$), equivalent to the performance of a random classifier.

Pretraining on Great Ape Behavior Datasets: The three pretraining scenarios using great ape behavior datasets, as described in Sect. 5.4.1, show moderate improvements compared to training from scratch. For Top1-accuracy, pretraining on either PanAf (CI $[25.5, 27.2]$) or ChimpBehave (CI $[25.3, 27.5]$) offers only modest, but not statistically significant benefits compared to no pretraining (CI $[23.0, 26.6]$). However, combining both datasets (PanAf500 + ChimpBehave) significantly improves Top1-accuracy, with a confidence interval $[26.8, 28.8]$.

For Top3-accuracy, the improvements are more pronounced, especially in scenarios involving ChimpBehave. Pretraining on PanAf500 + ChimpBehave (CI $[49.8, 58.6]$) and ChimpBehave alone (CI $[49.3, 53.6]$) both reach statistical significance compared to no pretraining (CI $[43.8, 48.1]$).

Similarly, MCA confidence intervals demonstrate significant improvement for all three pretraining scenarios with great ape behavior datasets (CI $[4.2, 7.2]$, $[4.3, 7.6]$, and $[6.2, 9.3]$) compared to no pretraining (CI $[2.7, 3.8]$). These results suggest that (i) pretraining on great ape behavior data positively impacts gesture recognition, and (ii) increasing the size of the pretraining dataset improves the final model's performance.

Pretraining on Kinetics 400: Studies in human action recognition have demonstrated that pretraining on Kinetics 400, a large-scale dataset containing 400 human actions and $300,000$ video clips, consistently boosts model performance across architectures and datasets [23]. In our experiments, X3D pretrained on Kinetics 400 significantly outperforms all other pretraining scenarios, underscoring the value of large-scale human activity datasets in recognizing ape gestures.

The results of these experiments on FineChimp dataset are summarized in Table 5.2.

Table 5.2 **Performance evaluation for pretraining scenarios.** Pretraining on Kinetics-400 (K400), a large-scale human action dataset, significantly outperforms both pretraining on great ape behavior datasets (PA = PanAf500, CB = ChimpBehave) and training from scratch.

Pretraining	Top-1 accuracy	Top-3 accuracy	MCA
none	24.8 ± 0.7 [23.0, 26.6]	45.9 ± 0.8 [43.8, 48.1]	3.2 ± 0.2 [2.7, 3.8]
PA	26.4 ± 0.3 [25.5, 27.2]	49.4 ± 1.1 [46.3, 52.6]	5.7 ± 0.5 [4.2, 7.2]
CB	26.4 ± 0.4 [25.3, 27.5]	51.5 ± 0.8 [49.3, 53.6]	6.0 ± 0.6 [4.3, 7.6]
PA+CB	27.8 ± 0.4 [26.8, 28.8]	54.2 ± 1.6 [49.8, 58.6]	7.8 ± 0.5 [6.2, 9.3]
K400	**51.2** ± 1.0 [48.4, 54.0]	**78.2** ± 0.7 [76.1, 80.3]	**40.9** ± 2.3 [34.5, 47.3]

5.5.1.2 Model Benchmarking

Among the three CNN-based action recognition models evaluated in our experiments (X3D, SlowFast R50, and SlowFast R101), and pretrained on Kinetics-400, we observe similar performances, with no statistically significant differences across all three metrics.

However, contrasting results emerge when comparing these CNN-based models to transformer-based architectures. Contrary to expectations that newer architectures would outperform older ones, TimeSformer underperforms significantly compared to its CNN-based counterparts. This result aligns with prior observations that the variant tested (space-only TimeSformer) does not outperform either X3D (M) or SlowFast (R50/R101) on the Kinetics 400 benchmark. Additionally, TimeSformer has 86.11M trainable parameters, far exceeding those of X3D (3.8M) and SlowFast (R50: 34.6M, R101: 62.9M).

In contrast, VideoMAEv2, the second transformer-based architecture in our benchmarks, significantly outperforms all other models across all three metrics. This statistically significant improvement establishes VideoMAEv2 as our final classifier.

The results of model benchmarking are summarized in Table 5.3.

Table 5.3 **Performance evaluation for model benchmarking.** VideoMAEv2, a recent state-of-the-art video transformer, significantly outperforms other model architectures.

Model	Top1-accuracy	Top3-accuracy	Mean Class Accuracy
X3D	51.2 ± 1.0 [48.4, 54.0]	78.2 ± 0.7 [76.1, 80.3]	40.9 ± 2.3 [34.5, 47.3]
SlowFast (Resnet 50)	50.4 ± 0.7 [48.5, 52.2]	75.4 ± 1.2 [72.0, 78.8]	41.0 ± 1.9 [35.7, 46.2]
SlowFast (Resnet 101)	50.9 ± 1.0 [48.3, 53.6]	74.3 ± 1.0 [71.5, 77.1]	40.3 ± 1.7 [35.6, 45.0]
TimeSformer	36.9 ± 1.1 [33.8, 40.0]	65.8 ± 1.3 [62.1, 69.5]	25.5 ± 1.7 [20.6, 30.3]
VideoMAEv2	**61.4** ± 0.7 [59.4, 63.4]	**84.3** ± 0.7 [82.3, 86.4]	**57.4** ± 0.8 (55.1, 59.7]

Table 5.4 **Performance evaluation for frame sampling.** Using 16 frames as input achieves the best performance, indicating it is the optimal input length.

Method	Top-1 accuracy	Top-3 accuracy	MCA
4 frames	45.1 ± 0.4 [44.0, 46.1]	72.9 ± 0.6 [71.3, 74.5]	31.8 ± 1.2 [28.6, 35.0]
8 frames	51.2 ± 1.0 [48.4, 54.0]	78.2 ± 0.7 [76.1, 80.3]	40.9 ± 2.3 [34.5, 47.3]
16 frames	**53.7** ± 0.5 [52.2, 55.1]	**80.7** ± 0.8 [78.5, 82.8]	**46.1** ± 1.0 [43.2, 48.9]
24 frames	52.8 ± 1.2 [49.4, 56.2]	78.7 ± 0.8 [76.3, 81.0]	46.0 ± 2.8 [38.3, 53.8]
32 frames	51.4 ± 1.2 [48.0, 54.7]	79.0 ± 0.9 [76.6, 81.4]	44.0 ± 2.1 [38.2, 49.8]

5.5.1.3 Frame Sampling

When analyzing the impact of the number of frames on evaluation metrics, we observe that the mean of each metric increases as the number of frames rises from 4 to 16, but then decreases as the input dimension continues to grow from 16 to 32. Specifically, the model performs statistically better when using 16 frames compared to 4, suggesting that longer inputs can be beneficial for this complex task, which contains significant temporal information. However, using even longer video clips (e.g., 24 or 32 frames) tends to result in slightly lower evaluation scores (but not statistically significant).

We attribute this trend to two potential factors: (i) Shorter video clips (e.g., those with 10 or 20 frames) may require frame duplication to meet the model's input dimensions, which could introduce redundancy. (ii) The model may lose generalization capability by overfitting to lengthy patterns in the training data.

The results of frame sampling experiments are presented in Table 5.4.

5.5.2 Performances of Data Augmentation

As outlined in Sect. 5.4.4, we identified eight different data augmentation techniques categorized as geometrical, non-geometrical, and temporal. The results of these experiments (all based on a pretrained X3D model) are presented in Table 5.5, alongside a baseline without augmentation for comparison.

From a general perspective, none of these techniques, when applied individually, produced a statistically significant boost in any of the three evaluation metrics compared to the baseline (without data augmentation). However, several notable observations can still be made: (i) Both cropping methods (*+2 crops* and *+5 crops*) resulted in mean scores equal to or higher than the baseline across all three metrics. However, *+5 crops* did not yield significant benefits over *+2 crops*, especially considering the increased training time required. (ii) All

Table 5.5 **Performance evaluation for data augmentation.** Techniques with an asterisk sign (*) will be used in final model evaluation. The parameter *n* represents the size of each training fold after applying the corresponding augmentation technique.

	n	Top-1 accuracy	Top-3 accuracy	MCA
No augmentation	1596	51.2 ± 1.0 [48.4, 54.0]	78.2 ± 0.7 [76.1, 80.3]	40.9 ± 2.3 [34.5, 47.3]
Geometrical				
rotation*	3189	51.0 ± 0.9 [48.5, 53.5]	76.4 ± 1.3 [72.7, 80.0]	46.1 ± 1.6 [41.8, 50.5]
affine*	3189	49.4 ± 0.8 [47.3, 51.5]	74.7 ± 0.7 [72.7, 76.7]	42.9 ± 2.2 [36.8, 49.1]
Non geo-metrical				
+2 crops	4217	54.3 ± 0.9 [51.7, 56.8]	80.3 ± 0.6 [78.6, 81.9]	47.2 ± 1.6 [42.7, 51.7]
+5 crops*	9548	54.4 ± 0.7 [52.4, 56.4]	78.2 ± 0.6 [76.4, 79.9]	48.7 ± 3.2 [39.9, 57.5]
noisy*	3189	50.3 ± 1.3 [46.8, 53.8]	76.0 ± 1.0 [73.3, 78.7]	44.7 ± 2.6 [37.6, 51.7]
Temporal				
stratified*	2925	51.9 ± 1.2 (48.7, 55.2)	76.4 ± 1.9 (71.1, 81.6)	46.5 ± 2.5 (39.6, 53.4)
random*	2925	50.4 ± 1.1 (47.4, 53.4)	77.0 ± 1.2 (73.5, 80.4)	43.7 ± 1.7 (39.1, 48.3)
sliding	2925	49.5 ± 0.6 (47.7, 51.3)	75.8 ± 0.8 (73.6, 78.0)	41.2 ± 1.6 (36.8, 45.6)

non-cropping techniques (i. e., *rotation*, *affine*, *noisy*, *stratified*, *random*, and *sliding*) tended to lower mean Top1-accuracy and Top3-accuracy but improve MCA. This suggests that these methods may better support underrepresented classes in the dataset, highlighting a potential tradeoff between Top*k*-accuracies and MCA. To illustrate this tradeoff, Fig. 5.8 visualizes the impact of selecting models based on MCA rather than Top1-accuracy. For all techniques, selecting models based on their best MCA results in higher MCA scores but lower Top1-accuracy, as indicated by the blue arrows pointing toward the upper-left corner.

For final model training, we included all augmentation techniques except *+2 crops* and *sliding*. The former is subsumed by *+5 crops*, while the latter's MCA confidence interval upper bound does not exceed the baseline.

5.5.3 Performance of Cross-View Generalization

As shown in Table 5.6, generalization performance varies across viewpoints. Cameras 0 and 4 demonstrate relatively similar metrics (Top1-accuracy 52.1/54.2, Top3-accuracy 76.5/76.0, MCA 35.3/38.0, respectively), as do Cameras 2 and 3 (Top1-accuracy 58.1/57.8, Top3-accuracy 81.1/81.0, MCA 58.5/56.5). In contrast, Camera 1 shows distinctive results (Top1-accuracy 44.0, Top3-accuracy 77.2, MCA 43.8).

This trend aligns with the camera setup (see Fig. 5.5), where Cameras 0 and 4 capture the scene from a greater distance and at similar angles, while Cameras 2 and 3 provide side views close to Camera 0, the main annotation source. Camera 1, positioned above the scene, offers a corner perspective, leading to fewer visible gestures. Consequently, the set of gesture classes to be considered by MCA metric is not identical for every camera: the missing class are *'bite', 'climb on', 'flap with object', 'pull other', 'stomp rhythmic object'* and *'throw object'* for Camera 1, *flap with object'* for Camera 2, and finally *flap with object'* and *'stomp rhythmic object* for Camera 3.

Similarly, the number of test items per camera reflects these patterns. Camera 4 has the highest number of test items (467), second only to Camera 0 (622), suggesting strong similarity between these views. Cameras 2 and 3 also have comparable test item counts (360 and 315, respectively). In contrast, Camera 1 has significantly fewer test items (232), consistent with its limited visibility of gestures.

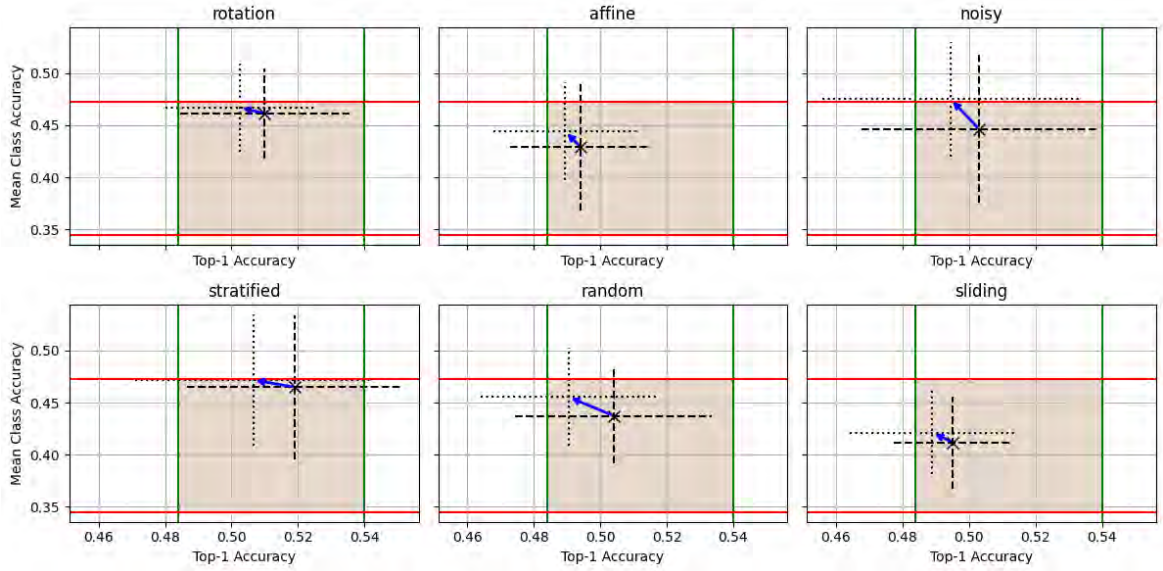

Fig. 5.8 **The impact of model selection based on MCA rather than Top1-accuracy**. For all six data augmentation techniques, selecting models based on MCA results in higher MCA but lower Top1-accuracy. Dashed segments indicate 95% confidence intervals for Top1-accuracy selection, dotted segments for MCA selection. Blue arrows represent shifts in score means. Red horizontal and green vertical lines show the confidence interval minimum and maximum of the baseline (without data augmentation).

Table 5.6 **Performance evaluation for cross-view generalization.** The columns n_{test} and $m_{classes}$ indicate the number of test items and missing classes respectively. MCA_{valid} means that missing classes were excluded for the calculus of MCA.

	n_{test}	$m_{classes}$	Top-1 accuracy	Top-3 accuracy	MCA_{valid}
Camera 0	622	-	52.1	76.5	35.3
Camera 1	232	6	44.0	77.2	43.8
Camera 2	360	1	58.1	81.1	58.5
Camera 3	315	2	57.8	81.0	56.5
Camera 4	467	-	54.2	76.0	38.0

5.5.4 Performance of multimodality: RGB + Optical Flow

In the following, we focus on the analysis of the effects of the two types of information sources (RGB and Optical flow) and of their fusion in a dual-stream architecture.

5.5.4.1 RGB stream

Based on the analysis of performance evaluation corresponding to the series of experiments presented in sections 5.5.1 and 5.5.2, we selected as the final, best model VideoMAEv2 architecture, pretrained on Kinetics 400, using a 16-frame input, and incorporating six data augmentation techniques: *+5 crops*, *rotation*, *affine*, *noisy*, *stratified*, and *random*. These augmentation techniques raise the number of training examples per fold to approximately 15,400.

Class-level metrics for the final model trained on RGB stream are shown in Fig. S16, with its normalized confusion matrix available in Fig. S18 in Appendix D. Two key observations can be made regarding the class-level metrics:

(i) **Performance Variability by Class**: The variability of the model's performance across classes can by measured by the coefficient of variation, a standardized measure of dispersion defined as the ratio between standard deviation and the mean. For the metrics precision, recall and F1-score, this coefficient varies between 25% and 29%, expressing the fact that these performance metrics have not a high variability across classes. On the other hand, for False Positive Rate (FPR), the coefficient of variation is 237%, expressing a huge variability, due particularly to few classes, as 'grab', 'touch' and 'grab pull' (the FPR values for these classes can be seen as *outliers* values). This effect could be partially attributed to the higher number of examples for these classes in the dataset, coupled with model selection based on Top1-accuracy. This selection

Table 5.7 **Performance evaluation of the final model trained on different data streams.** The model trained on RGB stream without data augmentation serves as the baseline, while combining data augmentation techniques leads to higher mean accuracy scores across all metrics (although not statistically significant). The model trained solely on optical flow performs significantly worse than the baseline model; however, fusing RGB and optical flow streams results in higher mean scores for Top1-accuracy and MCA (but not statistically significant).

	Top-1 accuracy	Top-3 accuracy	MCA
RGB			
no augmentation	61.4 ± 0.7 [59.4, 63.4]	84.3 ± 0.7 [82.3, 86.4]	57.4 ± 0.8 [55.1, 59.7]
with augmentation	64.2 ± 0.9 [61.6, 66.8]	**86.4** ± 0.9 [84.1, 88.8]	62.8 ± 2.7 [55.4, 70.1]
Optical Flow	48.9 ± 2.1 [46.4, 51.6]	71.5 ± 3.0 [67.8, 75.2]	41.8 ± 2.1 [39.2, 44.4]
RGB+Flow λ=0.65	**65.1** ± 2.2 [62.4, 67.8]	84.9 ± 1.9 [82.6, 87.2]	**63.3** ± 5.9 [56.0, 70.6]

process may favor models that overfit more frequent classes. Quantitatively, this tendency is further evidenced by the diminishing False Positive Rate as the number of examples per class decreases.

(ii) **Strong Performance on Rare Classes**: In contrast to other research on great ape action recognition [17, 50], we observe that certain classes with limited examples achieve high performance. For example, F1-scores for 'drum object' ($n = 31$), 'stomp object' ($n = 13$), 'shake object' ($n = 8$), and 'reach foot' ($n = 5$) are significantly higher than those for the five most frequent classes: 'grab' ($n = 462$), 'touch' ($n = 252$), 'grab pull' ($n = 134$), 'hand on' ($n = 121$), and 'directional push' ($n = 109$). This suggests that the model can perform well on rare gestures, presumably due to their distinct visual features that make them easier to differentiate.

5.5.4.2 Optical Flow stream

The final model trained solely on optical flow (OF) performs statistically significantly worse across all metrics than the same model trained exclusively on RGB, underscoring the importance of RGB information for this task (see Table 5.7). For more insights into class-level performance see Fig. S17 and Fig. S18 in Appendix D. While lower performance for Optical Flow streams on certain datasets is not uncommon, this result contradicts the assumption that the classification of fine-grained actions benefits more from motion features than appearance features [176].

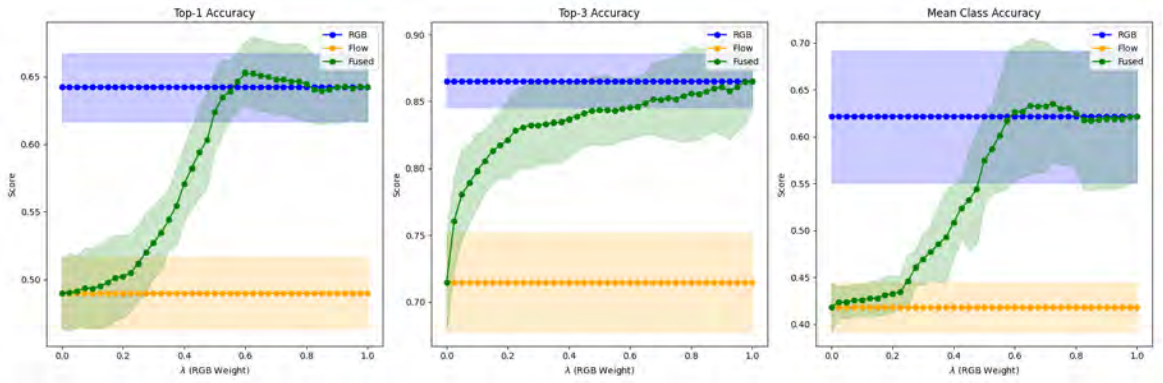

Fig. 5.9 **Performance of the fused RGB and optical flow (OF) streams as a function of λ (the weight assigned to the RGB stream), for values ranging from 0 to 1**. RGB alone (at $\lambda = 1$) consistently outperforms the OF stream (at $\lambda = 0$). An interval ($0.65 \leq \lambda \leq 0.8$) is observed where the fused RGB+OF stream achieves slightly higher Top1-accuracy and MCA than RGB alone, suggesting potential benefits of multimodal fusion.

Three factors may partially explain this discrepancy: (i) State-of-the-art video transformers, such as VideoMAEv2, are optimized for RGB stream and are highly capable of modeling and learning temporal representations directly from RGB data. (ii) The Optical Flow stream of VideoMAEv2 showed lower-than-expected performance during pretraining on Kinetics-400 (Top1-accuracy: 30.0), compared to previously reported results for CNN-based models (e.g., Top1-accuracy: 54.0 in [32]). This may be due to the inherent inductive bias of CNN architectures (e. g. translation equivariance) which Transformers lacks [38]. (iii) Unlike the Optical Flow stream, the RGB stream benefited from additional data augmentation techniques (see Sect. 5.4.4).

5.5.4.3 Fused RGB and Optical Flow streams

Figure 5.9 illustrates the model's performance when λ, the weight assigned to the RGB stream, varies between 0 and 1. For relatively low values of λ, the RGB stream alone significantly outperforms both the Optical Flow stream and the fused RGB + Optical Flow streams across all three metrics.

When $\lambda = 0.5$ (i.e., equal weights assigned to both RGB and OF streams), the mean scores for all metrics remain slightly higher for RGB alone than for the fused approach, although the difference is not statistically significant.

Interestingly, there exists an interval, $0.65 \leq \lambda \leq 0.8$, where the fused RGB+OF stream achieves higher mean scores for Top1-accuracy and MCA compared to the RGB stream alone. Although not statistically significant, this result suggests that FineChimp performance could potentially be improved with further exploration of multimodal architectures and fusion

techniques. The results of all experiments are shown in Table 5.7, where all three metrics showed improved means with data augmentation and/or fused streams, although without statistically significant differences.

5.6 Discussion and Future Work

The previous sections have presented our main findings, demonstrating that computer vision models can successfully learn to recognize chimpanzee communicative gestures. To the best of our knowledge, this is the first successful application of computer vision to the automatic recognition of great ape gestures—a highly meaningful source of information for understanding the evolution of human language. The creation of the *FineChimp* dataset was instrumental for this achievement, and we are committed to making this data publicly available. FineChimp offers researchers a novel resource: a dataset specifically designed for the recognition of animal gestural communication, featuring fine-grained chimpanzee actions labeled by domain experts and multiview videos of great apes. We hope these results inspire other researchers to improve classification accuracies on FineChimp and similar datasets, and, more importantly, to publish additional annotated video data of animals, leveraging domain knowledge to advance the field further.

Our findings highlight that some gesture classes are easier to recognize than others. This discrepancy may be influenced by factors such as the visual prominence of certain actions—e.g., a raised arm creates more contrast within the scene compared to a subtle hand touch—or the number of dataset examples per class. A notable limitation of FineChimp is the low number of examples for certain gesture classes. Additionally, the video resolution is relatively low in some cases, which can hinder fine-grained action classification. For instance, human action datasets like FineGym include videos with resolutions up to 1080p [176]. Future work should prioritize annotating less frequent gestures and aim to include high-definition video segments to enhance classification performance.

In primatology, identifying great ape gestures often requires assessing multiple criteria, such as indicators of intentionality and mutual gaze to determine if movements are part of communicative exchanges [57]. However, FineChimp's annotated gestures focus solely on the visual, mechanical characteristics typical of communication. For example, a juvenile grabbing its mother to climb on her back may be labeled as a 'grab,' even if it does not meet stricter definitions of communication. This deliberate choice was made to create a dataset tailored for computer vision algorithms. Future work could incorporate these additional criteria to expand FineChimp's utility for both computer vision and behavioral science applications.

Additionally, addressing the multimodality of great ape signals is a promising avenue for future research. FineChimp currently contains only video data, but chimpanzees often combine gestures with vocalizations [140, 84]. Furthermore, their likelihood of producing specific gestures with audible components may depend on the receiver's attentional state [57]. Including synchronized audio data alongside video could enhance the dataset's potential for capturing the full spectrum of chimpanzee communication.

In future work, we aim to provide perfectly synchronized frames across all viewpoints and extend the action recognition task to action detection, which could further reduce manual screening time for large video datasets. Inspired by developments in great ape behavior recognition, we anticipate that integrating additional modalities such as pose estimation [52], dense pose estimation [15], and text descriptions [16] could enhance gesture classification, particularly in out-of-distribution contexts such as in-the-wild videos [50].

The present work focuses exclusively on the recognition of great ape gestures as a supervised classification task, where gesture classes are predefined and annotated by humans. We envision that future AI-assisted research in great ape communication will also advance in unsupervised learning approaches. Such methods could enable systems to learn meaningful representations of gestures directly from raw video data in a purely data-driven manner, without the need for human supervision.

5.7 Conclusion

Artificial intelligence is increasingly transforming the study of animal behavior. Our hope is that these technologies will assist researchers in their areas of expertise by automating time-intensive, repetitive tasks, thereby enabling broader and deeper observational studies. Computer vision offers a wealth of opportunities to streamline the collection and annotation of video data, as demonstrated in this work. Models and public datasets like FineChimp can play a pivotal role in automating the recording of animal interactions with non-invasive technologies, processing large datasets with minimal human intervention, and preselecting footage of interest. Moreover, this study has shown that computer vision models can discern highly complex visual patterns in chimpanzee gestural communication.

We hope this achievement inspires further research to expand the availability of publicly accessible datasets featuring fine-grained animal actions. Such resources are critical for capturing meaningful insights into animal social interactions and communication, advancing our understanding of both animal behavior and the evolutionary roots of human language.

Chapter 6

Conclusion

6.1 Summary of Contributions

This dissertation is centered around the main research question: *How can deep learning action recognition methods be applied to animal behavior understanding, particularly in recognizing great ape behaviors and communicative gestures?*

In our research, we explored multiple machine learning tasks, including individual detection/tracking, pose estimation, and action recognition, using diverse approaches such as video-based and skeleton-based action recognition. Moreover, our work leveraged large-scale primate datasets, including OpenMonkeyChallenge [224], OpenApePose [36], ChimpACT [124], and PanAf500 [17].

The automated recognition of gestural signals of great apes represents one of the most significant contributions of this dissertation, both for computer vision and primatology. From a computer vision perspective, we demonstrated that large video transformers, equipped with dual-path architectures (RGB and Optical Flow), are particularly effective for fine-grained action recognition. In primatology, this marks the first automated classification of great ape gestures, a critical component of their communication system. Meanwhile, the study of great ape gestures continues to raise profound theoretical questions, such as whether gestures are learned through individual experience or form part of a species-wide repertoire with individual variations.

As emphasized in the introductory chapter, one of the most pressing challenges in advancing AI for animal research — particularly great ape action recognition — is the scarcity of publicly available datasets. To address this bottleneck, we contributed two large chimpanzee-specific datasets, annotated by expert primatologists: (i) *ChimpBehave*, a dataset for behavior recognition; (ii) *FineChimp*, a dataset for gestural signal classification. These

resources, openly available to the community, are expected to empower future researchers and foster further advances in both primatology and computer vision.

Skeleton-based action recognition, widely used in human-centric applications, holds significant promise for quantifying animal behaviors due to its lightweight architectures and robustness across individuals and environments. Despite its potential, its application to animal behavior recognition — particularly on large datasets and in natural settings — remains underexplored. This dissertation addresses this gap in two major ways: (i) we designed ASBAR (Animal Skeleton-Based Action Recognition), a species- and behavior-agnostic framework that integrates full model and data pipelines for skeleton-based action recognition; (ii) we validated the effectiveness of ASBAR on a large scale by successfully classifying great ape behaviors in their natural habitat, a particularly complex and challenging scenario for animal research.

Furthermore, while the robustness of skeleton-based approaches has been extensively studied for human-centric action recognition, their generalization capacity across visual domains has not been explored in animal studies. To this end, we compared the performances of skeleton-based and video-based methods in cross-dataset experiments. Our results demonstrated that skeleton-based methods exhibit superior generalization to visual domain changes, underscoring their robustness for great ape behavior recognition.

Finally, the automated multi-view camera setup remains operational to this day and has already facilitated video data collection for other teams of primatologists. Notably, it has been used to establish a blink rate baseline for great apes. In humans, blink rate has been related to cognitive load, and may well have implications for understanding stress in social groups, as well as assessing how apes process different stimuli in cognitive experiments.

Together, these contributions advance the intersection of deep learning and animal behavior research, laying a foundation for automated, scalable, and resource-efficient solutions to understand complex behaviors in non-human primates.

6.2 Future Research Directions

This dissertation primarily focuses on models for action recognition. While action recognition is inherently a classification task, future research could explore leveraging these models' feature representations to adapt them for spatiotemporal localization tasks, such as action detection [124]. Such extensions would be particularly valuable for primatologists, as they could significantly reduce the manual effort required for coding large video datasets.

The techniques employed in this work predominantly fall within the paradigm of supervised learning. However, unsupervised learning holds immense potential, particularly in

the study of great ape gestural communication. Unsupervised methods may uncover hidden patterns or relationships in the data that have so far remained unnoticed by human observers [123, 212]. This could open new avenues for understanding the complexity of great ape communication systems.

Our current methods rely solely on visual data, without incorporating sound. Audiovisual pipelines have already proven successful for chimpanzee behavior recognition [5], and future work could extend our models to include audio signals. This would enable the detection of multimodal behaviors, where gestures are often accompanied by audible components (e.g., 'single stomp,' see modalities in Table T4 in Appendix D). On a broader scale, a comprehensive machine learning study of great ape communicative systems would greatly benefit from integrating both gestural and vocal signals, reflecting their naturally multimodal communication strategies.

While this dissertation extensively investigated skeleton-based approaches for behavior recognition (Chapters 3 and 4), these methods were deliberately not explored for gesture recognition (Chapter 5). This decision was motivated by the lack of skeletal keypoints representing fine-grained finger movements in current public primate pose estimation datasets. Current skeleton annotations for primates are limited to limb extremities (wrists and ankles), which are insufficient for distinguishing gestures that depend on finger-level actions. Future work could focus on extending primate skeletons to include high-granularity keypoints, similar to whole-body pose annotations available for humans [89].

Additionally, while this work evaluated the generalization capacity of action recognition approaches across different visual and recording settings, a natural progression would be to examine how well these models generalize to distinct animal species. Investigating cross-species performance could offer further insights into the robustness and transferability of current action recognition techniques.

Finally, as part of our future research agenda, we aim to adapt human-centric gaze-following models [161] to chimpanzees. Mutual gaze is a critical marker of intentional communication in great apes [56, 198]. Developing such models could help quantify gaze behaviors and provide a deeper understanding of communicative intent in non-human primates.

References

[1] Agarap, A. (2018). Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv:1803.08375*.

[2] Aggarwal, J. and Ryoo, M. (2011). Human activity analysis. *ACM Computing Surveys*, 43(3):1–43.

[3] An, L., Ren, J., Yu, T., Hai, T., Jia, Y., and Liu, Y. (2023). Three-dimensional surface motion capture of multiple freely moving pigs using MAMMAL. *Nat. Commun.*, 14(1):7727.

[4] Anderson, D. J. and Perona, P. (2014). Toward a Science of Computational Ethology. *Neuron*, 84(1):18–31.

[5] Bain, M., Nagrani, A., Schofield, D., Berdugo, S., Bessa, J., Owen, J., Hockings, K. J., Matsuzawa, T., Hayashi, M., Biro, D., et al. (2021). Automated audiovisual behavior recognition in wild primates. *Science advances*, 7(46).

[6] Bain, M., Nagrani, A., Schofield, D., and Zisserman, A. (2019). Count, crop and recognise: Fine-grained recognition in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.

[7] Bala, P., Zimmermann, J., Park, H. S., and Hayden, B. Y. (2023). Self-supervised secondary landmark detection via 3D representation learning. *International Journal of Computer Vision*, 131(8):1980–1994.

[8] Bala, P. C., Eisenreich, B. R., Yoo, S. B. M., Hayden, B. Y., Park, H. S., and Zimmermann, J. (2020). Automated markerless pose estimation in freely moving macaques with OpenMonkeyStudio. *Nature communications*, 11(1):4560.

[9] Bengio, Y., Goodfellow, I., and Courville, A. (2017). *Deep learning*, volume 1. MIT Press Cambridge, MA, USA.

[10] Bergler, C., Smeele, S. Q., Tyndel, S. A., Barnhill, A., Ortiz, S. T., Kalan, A. K., Cheng, R. X., Brinkløv, S., Osiecka, A. N., Tougaard, J., et al. (2022). ANIMAL-SPOT enables animal-independent signal detection and classification using deep learning. *Scientific Reports*, 12(1):21966.

[11] Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4.

[12] Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., and Black, M. J. (2016). Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Computer Vision – ECCV 2016*, pages 561–578. Springer International Publishing.

[13] Bohnslav, J. P., Wimalasena, N. K., Clausing, K. J., Dai, Y. Y., Yarmolinsky, D. A., Cruz, T., Kashlan, A. D., Chiappe, M. E., Orefice, L. L., Woolf, C. J., and Harvey, C. D. (2021). DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels. *eLife*, 10:e63377.

[14] Brookes, O. and Burghardt, T. (2020). A dataset and application for facial recognition of individual gorillas in zoo environments. *arXiv preprint arXiv:2012.04689*.

[15] Brookes, O., Mirmehdi, M., Kühl, H., and Burghardt, T. (2023). Triple-stream deep metric learning of great ape behavioural actions. *arXiv preprint arXiv:2301.02642*.

[16] Brookes, O., Mirmehdi, M., Kuhl, H., and Burghardt, T. (2024a). Chim-pVLM: Ethogram-Enhanced Chimpanzee Behaviour Recognition. *arXiv preprint arXiv:2404.08937*.

[17] Brookes, O., Mirmehdi, M., Stephens, C., Angedakin, S., Corogenes, K., Dowd, D., Dieguez, P., Hicks, T. C., Jones, S., Lee, K., et al. (2024b). PanAf20K: a large video dataset for wild ape detection and behaviour recognition. *International Journal of Computer Vision*, pages 1–17.

[18] Brown, T. B. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

[19] Byrne, R. W., Cartmill, E., Genty, E., Graham, K. E., Hobaiter, C., and Tanner, J. (2017). Great ape gestures: intentional communication with a rich set of innate signals. *Animal cognition*, 20:755–769.

[20] Cao, J., Tang, H., Fang, H.-S., Shen, X., Lu, C., and Tai, Y.-W. (2019). Cross-domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9498–9507.

[21] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[22] Carreira, J., Noland, E., Hillier, C., and Zisserman, A. (2019). A short note on the Kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*.

[23] Carreira, J. and Zisserman, A. (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[24] Cartmill, E. A. and Byrne, R. W. (2007). Orangutans modify their gestural signaling according to their audience's comprehension. *Current Biology*, 17(15):1345–1348.

[25] Cauli, N. and Reforgiato Recupero, D. (2022). Survey on Videos Data Augmentation for Deep Learning Models. *Future Internet*, 14(3).

[26] Chen, C., Jafari, R., and Kehtarnavaz, N. (2017). A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools and Applications*, 76:4405–4425.

[27] Chen, J., Hu, M., Coker, D. J., Berumen, M. L., Costelloe, B., Beery, S., Rohrbach, A., and Elhoseiny, M. (2023). MammalNet: A large-scale video benchmark for mammal recognition and behavior understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13052–13061.

[28] Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T. S., and Zhang, L. (2020). HigherhrNet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5386–5395.

[29] Cheng, C., Huang, Z., Zhang, R., Huang, G., Wang, H., Tang, L., and Wang, X. (2024). A Real-time Multi-Subject Three Dimensional Pose Tracking System for Analyzing Social Behaviors of Non-human Primates. *bioRxiv*.

[30] Corballis, M. C. (2008). *The Gestural Origins of Language*, pages 11–23. Springer Japan, Tokyo.

[31] Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:273–297.

[32] Crasto, N., Weinzaepfel, P., Alahari, K., and Schmid, C. (2019). MARS: Motion-augmented rgb stream for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7882–7891.

[33] Crockford, C., Wittig, R. M., Mundry, R., and Zuberbühler, K. (2012). Wild chimpanzees inform ignorant group members of danger. *Current Biology*, 22(2):142–146.

[34] Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277.

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

[36] Desai, N., Bala, P., Richardson, R., Raper, J., Zimmermann, J., and Hayden, B. (2023). OpenApePose, a database of annotated ape photographs for pose estimation. *Elife*, 12:RP86873.

[37] Dezecache, G., Zuberbühler, K., Davila-Ross, M., and Dahl, C. D. (2021). A machine learning approach to infant distress calls and maternal behaviour of wild chimpanzees. *Animal Cognition*, 24:443–455.

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

[39] Duan, H., Zhao, Y., Chen, K., Lin, D., and Dai, B. (2022). Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978.

[40] Dunkley, K., Dunkley, A., Drewnicki, J., Keith, I., and Herbert-Read, J. E. (2023). A low-cost, long-running, open-source stereo camera for tracking aquatic species and their behaviours. *Methods in Ecology and Evolution*, 14(10):2549–2556.

[41] Erb, W. M., Ross, W., Kazanecki, H., Setia, T. M., Madhusudhana, S., and Clink, D. J. (2024). Vocal complexity in the long calls of bornean orangutans. *PeerJ*, 12:e17320.

[42] Ernst, A. and Küblbeck, C. (2011). Fast face detection and species classification of african great apes. In *2011 8th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 279–284. IEEE.

[43] Feichtenhofer, C. (2020). X3D: Expanding Architectures for Efficient Video Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[44] Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). SlowFast Networks for Video Recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210.

[45] Feichtenhofer, C., Li, Y., He, K., et al. (2022). Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958.

[46] Feng, L., Zhao, Y., Sun, Y., Zhao, W., and Tang, J. (2021). Action Recognition Using a Spatial-Temporal Network for Wild Felines. *Animals*, 11(2):485.

[47] Feng, L., Zhao, Y., Zhao, W., and Tang, J. (2022). A comparative review of graph convolutional networks for human skeleton-based action recognition. *Artificial Intelligence Review*, pages 1–31.

[48] Freytag, A., Rodner, E., Simon, M., Loos, A., Kühl, H. S., and Denzler, J. (2016). Chimpanzee faces in the wild: Log-euclidean CNNs for predicting identities and attributes of primates. In *Pattern Recognition: 38th German Conference, GCPR 2016, Hannover, Germany, September 12-15, 2016, Proceedings 38*, pages 51–63. Springer.

[49] Fuchs, M., Genty, E., Bangerter, A., Zuberbühler, K., and Cotofrei, P. (2024a). From Forest to Zoo: Domain Adaptation in Animal Behavior Recognition for Great Apes with ChimpBehave. 4th Workshop on CV4Animals: Computer Vision for Animal Behavior Tracking and Modeling, In conjunction with CVPR 2024.

[50] Fuchs, M., Genty, E., Bangerter, A., Zuberbühler, K., Odobez, J.-M., and Cotofrei, P. (2024b). From Forest to Zoo: Great Ape Behavior Recognition with ChimpBehave. Currently under review with minor revisions for the International Journal of Computers Vision.

[51] Fuchs, M., Genty, E., Zuberbühler, K., and Cotofrei, P. (2025). Automated Recognition of Great Ape Gestures. Manuscript in preparation.

[52] Fuchs, M., Genty, E., Zuberbühler, K., and Cotofrei, P. (2024c). ASBAR: an Animal Skeleton-Based Action Recognition framework. Recognizing great ape behaviors in the wild using pose estimation with domain adaptation. *eLife*, 13:RP97962.

[53] Gammelgård, F., Nielsen, J., Nielsen, E. J., Hansen, M. G., Alstrup, A. K. O., Perea-García, J. O., Jensen, T. H., and Pertoldi, C. (2024). Application of Machine Learning for Automating Behavioral Tracking of Captive Bornean Orangutans (Pongo Pygmaeus). *Animals*, 14(12):1729.

[54] Geldenhuys, C. M. and Niesler, T. R. (2024). Learning to rumble: Automated elephant call classification, detection and endpointing using deep architectures. *arXiv preprint arXiv:2410.12082*.

[55] Genty, E., Breuer, T., Hobaiter, C., and Byrne, R. W. (2009a). Gestural communication of the gorilla (Gorilla gorilla): repertoire, intentionality and possible origins. *Anim. Cogn.*, 12(3):527–546.

[56] Genty, E., Breuer, T., Hobaiter, C., and Byrne, R. W. (2009b). Gestural communication of the gorilla (Gorilla gorilla): repertoire, intentionality and possible origins. *Animal cognition*, 12:527–546.

[57] Genty, E. and Fuchs, M. (2023). *GApS: A Coding Scheme for Great Apes Signals in ELAN*. https://greatapesgestures.github.io.

[58] Genty, E. and Zuberbühler, K. (2014). Spatial reference in a bonobo gesture. *Current Biology*, 24(14):1601–1605.

[59] Ghani, B., Kalkman, V. J., Planqué, B., Vellinga, W.-P., Gill, L., and Stowell, D. (2024). Generalization in birdsong classification: impact of transfer learning methods and dataset characteristics. *arXiv preprint arXiv:2409.15383*.

[60] Graham, K. E. and Hobaiter, C. (2023). Towards a great ape dictionary: Inexperienced humans understand common nonhuman ape gestures. *PLoS Biology*, 21(1):e3001939.

[61] Graham, K. E., Hobaiter, C., Ounsley, J., Furuichi, T., and Byrne, R. W. (2018). Bonobo and chimpanzee gestures overlap extensively in meaning. *PLoS biology*, 16(2):e2004825.

[62] Greggor, A. L., Blumstein, D. T., Wong, B., and Berger-Tal, O. (2019). Using animal behavior in conservation management: a series of systematic reviews and maps.

[63] Gruber, T. and Zuberbühler, K. (2013). Vocal recruitment for joint travel in wild chimpanzees. *PloS one*, 8(9):e76073.

[64] Grund, C., Badihi, G., Graham, K. E., Safryghin, A., and Hobaiter, C. (2024). GesturalOrigins: A bottom-up framework for establishing systematic gesture data across ape species. *Behavior Research Methods*, 56(2):986–1001.

[65] Gu, N., Lee, K., Basha, M., Kumar Ram, S., You, G., and Hahnloser, R. H. R. (2024). Positive transfer of the whisper speech transformer to human and animal voice activity detection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7505–7509.

[66] Guo, S., Xu, P., Miao, Q., Shao, G., Chapman, C. A., Chen, X., He, G., Fang, D., Zhang, H., Sun, Y., et al. (2020). Automatic identification of individual primates with deep learning techniques. *Iscience*, 23(8).

[67] Hagiwara, M., Hoffman, B., Liu, J.-Y., Cusimano, M., Effenberger, F., and Zacarian, K. (2023). Beans: The benchmark of animal sounds. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

[68] Han, F., Reily, B., Hoff, W., and Zhang, H. (2017). Space-time representation of people based on 3D skeletal data: A review. *Computer Vision and Image Understanding*, 158:85–105.

[69] Hara, K., Kataoka, H., and Satoh, Y. (2017). Learning spatio-temporal features with 3D residual networks for action recognition. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 3154–3160.

[70] Hardin, A. and Schlupp, I. (2022). Using machine learning and DeepLabCut in animal behavior. *acta ethologica*, 25(3):125–133.

[71] Hauer, C., Nöth, E., Barnhill, A., Maier, A., Guthunz, J., Hofer, H., Cheng, R. X., Barth, V., and Bergler, C. (2023). Orca-spy enables killer whale sound source simulation, detection, classification and localization using an integrated deep learning-based segmentation. *Scientific Reports*, 13(1):11106.

[72] Hayden, B. Y., Park, H. S., and Zimmermann, J. (2022). Automated pose estimation in primates. *American journal of primatology*, 84(10):e23348.

[73] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

[74] He, K., Girshick, R., and Dollár, P. (2019). Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4918–4927.

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

[76] Heesen, R., Bangerter, A., Zuberbühler, K., Iglesias, K., Neumann, C., Pajot, A., Perrenoud, L., Guéry, J.-P., Rossano, F., and Genty, E. (2021a). Assessing joint commitment as a process in great apes. *IScience*, 24(8).

[77] Heesen, R., Bangerter, A., Zuberbühler, K., Rossano, F., Iglesias, K., Guéry, J.-P., and Genty, E. (2020). Bonobos engage in joint commitment. *Science Advances*, 6(51):eabd1306.

[78] Heesen, R., Zuberbühler, K., Bangerter, A., Iglesias, K., Rossano, F., Pajot, A., Guéry, J.-P., and Genty, E. (2021b). Evidence of joint commitment in great apes' natural joint actions. *Royal Society Open Science*, 8(12):211121.

[79] Heidari, N. and Iosifidis, A. (2020). Temporal Attention-Augmented Graph Convolutional Network for Efficient Skeleton-Based Human Action Recognition. *CoRR*, abs/2010.12221.

[80] Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., and Gilmer, J. (2021). The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8340–8349.

[81] Hinton, G. (2015). Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

[82] Hobaiter, C. and Byrne, R. W. (2011). The gestural repertoire of the wild chimpanzee. *Animal cognition*, 14:745–767.

[83] Hobaiter, C. and Byrne, R. W. (2014). The meanings of chimpanzee gestures. *Current Biology*, 24(14):1596–1600.

[84] Hobaiter, C., Byrne, R. W., and Zuberbühler, K. (2017). Wild chimpanzees' use of single and combined vocal and gestural signals. *Behav. Ecol. Sociobiol.*, 71(6):96.

[85] Huang, K., Han, Y., Chen, K., Pan, H., Zhao, G., Yi, W., Li, X., Liu, S., Wei, P., and Wang, L. (2021). A hierarchical 3D-motion learning framework for animal spontaneous behavior mapping. *Nat. Commun.*, 12(1):2784.

[86] Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. (2016). DeeperCut: A deeper, stronger, and faster multi-person pose estimation model. In *European conference on computer vision*, pages 34–50. Springer.

[87] International Union for Conservation of Nature (2024). IUCN Red List. *iucnredlist.org*.

[88] Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231.

[89] Jin, S., Xu, L., Xu, J., Wang, C., Liu, W., Qian, C., Ouyang, W., and Luo, P. (2020). Whole-body human pose estimation in the wild. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 196–214. Springer.

[90] Jocher, G., Qiu, J., and Chaurasia, A. (2023). Ultralytics YOLO. *https://github.com/ultralytics/ultralytics*.

[91] Joska, D., Clark, L., Muramatsu, N., Jericevich, R., Nicolls, F., Mathis, A., Mathis, M. W., and Patel, A. (2021). AcinoSet: A 3D Pose Estimation Dataset and Baseline Models for Cheetahs in the Wild. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13901–13908.

[92] Kaneko, T., Matsumoto, J., Lu, W., Zhao, X., Ueno-Nigh, L. R., Oishi, T., Kimura, K., Otsuka, Y., Zheng, A., Ikenaka, K., Baba, K., Mochizuki, H., Nishijo, H., Inoue, K.-I., and Takada, M. (2024). Deciphering social traits and pathophysiological conditions from natural behaviors in common marmosets. *Curr. Biol.*, 34(13):2854–2867.e5.

[93] Kaneko, T., Matsumoto, J., Lu, W., Zhao, X., Ueno-Nigh, L. R., Oishi, T., Kimura, K., Otsuka, Y., Zheng, A., Ikenaka, K., et al. (2023). Establishing an AI-based evaluation system that quantifies social/pathophysiological behaviors of common marmosets. *bioRxiv*, pages 2023–10.

[94] Karashchuk, P., Rupp, K. L., Dickinson, E. S., Walling-Bell, S., Sanders, E., Azim, E., Brunton, B. W., and Tuthill, J. C. (2021). Anipose: a toolkit for robust markerless 3D pose estimation. *Cell reports*, 36(13).

[95] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

[96] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., and Zisserman, A. (2017). The Kinetics Human Action Video Dataset.

[97] Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota.

[98] Kholiavchenko, M., Kline, J., Ramirez, M., Stevens, S., Sheets, A., Babu, R., Banerji, N., Campolongo, E., Thompson, M., Van Tiel, N., et al. (2024). KABR: In-situ dataset for kenyan animal behavior recognition from drone videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 31–40.

[99] Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes.

[100] Kipf, T. N., Fetaya, E., Wang, K., Welling, M., and Zemel, R. S. (2018). Neural Relational Inference for Interacting Systems. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2693–2702. PMLR.

[101] Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17.

[102] Krakauer, J. W., Ghazanfar, A. A., Gomez-Marin, A., MacIver, M. A., and Poeppel, D. (2017). Neuroscience needs behavior: correcting a reductionist bias. *Neuron*, 93(3):480–490.

[103] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

[104] Kumar, P. and Shingala, M. (2021). Native monkey detection using deep convolution neural network. In *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020*, pages 373–383. Springer.

[105] Kumar, T., Brennan, R., Mileo, A., and Bendechache, M. (2024). Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions. *IEEE Access*, pages 1–1.

[106] Labuguen, R., Matsumoto, J., Negrete, S. B., Nishimaru, H., Nishijo, H., Takada, M., Go, Y., Inoue, K.-i., and Shibata, T. (2021). MacaquePose: a novel "in the wild" macaque monkey pose dataset for markerless motion capture. *Frontiers in behavioral neuroscience*, 14:581154.

[107] Laskowski, L., Sawahn, R., Schall, M., Wasmuht, D., Bermejo, M., and de Melo, G. (2023). GorillaVision–Open-Set Re-Identification of Wild Gorillas.

[108] Lauer, J., Zhou, M., Ye, S., Menegas, W., Schneider, S., Nath, T., Rahman, M. M., Santo, V. D., Soberanes, D., Feng, G., Murthy, V. N., Lauder, G., Dulac, C., Mathis, M., and Mathis, A. (2022). Multi-animal pose estimation, identification and tracking with DeepLabCut. *Nature Methods*, 19:496 – 504.

[109] Le, V.-T., Tran-Trung, K., and Hoang, V. T. (2022). A comprehensive review of recent deep learning techniques for human activity recognition. *Computational Intelligence and Neuroscience*, 2022(1):8323962.

[110] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2.

[111] Li, C. and Lee, G. H. (2021). From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1482–1491.

[112] Li, C., Mellbin, Y., Krogager, J., Polikovsky, S., Holmberg, M., Ghorbani, N., Black, M. J., Kjellström, H., Zuffi, S., and Hernlund, E. (2024). The poses for equine research dataset (PFERD). *Sci. Data*, 11(1):497.

[113] Li, C., Xiao, Z., Li, Y., Chen, Z., Ji, X., Liu, Y., Feng, S., Zhang, Z., Zhang, K., Feng, J., et al. (2023). Deep learning-based activity recognition and fine motor identification using 2d skeletons of cynomolgus monkeys. *Zoological Research*, 44(5):967.

[114] Li, J., Liu, X., Zhang, W., Zhang, M., Song, J., and Sebe, N. (2020a). Spatio-Temporal Attention Networks for Action Recognition and Detection. *IEEE Transactions on Multimedia*, 22(11):2990–3001.

[115] Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2019). Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[116] Li, W., Swetha, S., and Shah, M. (2020b). Wildlife action recognition using deep learning.

[117] Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., and Feichtenhofer, C. (2022). MViTv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4804–4814.

[118] Lin, J., Gan, C., and Han, S. (2019). TSM: Temporal Shift Module for Efficient Video Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[119] Liu, D., Hou, J., Huang, S., Liu, J., He, Y., Zheng, B., Ning, J., and Zhang, J. (2023). LoTE-Animal: A long time-span dataset for endangered animal behavior understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20064–20075.

[120] Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.-Y., and Kot, A. C. (2020). NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701.

[121] Loos, A. and Ernst, A. (2013). An automated chimpanzee identification system using face detection and recognition. *EURASIP Journal on Image and Video Processing*, 2013:1–17.

[122] Loos, A. and Pfitzer, M. (2012). Towards automated visual identification of primates using face recognition. In *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 425–428. IEEE.

[123] Luxem, K., Mocellin, P., Fuhrmann, F., Kürsch, J., Miller, S. R., Palop, J. J., Remy, S., and Bauer, P. (2022). Identifying behavioral structure from deep variational embeddings of animal motion. *Communications Biology*, 5(1):1267.

[124] Ma, X., Kaufhold, S., Su, J., Zhu, W., Terwilliger, J., Meza, A., Zhu, Y., Rossano, F., and Wang, Y. (2023). Chimpact: A longitudinal dataset for understanding chimpanzee behaviors. *Advances in Neural Information Processing Systems*, 36:27501–27531.

[125] Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables.

[126] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

[127] Marks, M., Jin, Q., Sturman, O., von Ziegler, L., Kollmorgen, S., von der Behrens, W., Mante, V., Bohacek, J., and Yanik, M. F. (2022). Deep-learning-based identification, tracking, pose estimation and behaviour classification of interacting primates and mice in complex environments. *Nature machine intelligence*, 4(4):331–340.

[128] Marshall, J. D., Klibaite, U., Gellis, A., Aldarondo, D. E., Ölveczky, B. P., and Dunn, T. W. (2021). The PAIR-R24M Dataset for Multi-animal 3D Pose Estimation. *bioRxiv*.

[129] Martin, P.-E. (2023). Dataset Generation and Bonobo Classification from Weakly Labelled Videos. In *Proceedings of SAI Intelligent Systems Conference*, pages 689–700. Springer.

[130] Martin, P.-E., Kachel, G., Wieg, N., Eckert, J., and Haun, D. B. (2024). ApeTI: A Thermal Image Dataset for Face and Nose Segmentation with Apes. *Signals*, 5(1).

[131] Martini, L. M., Bognár, A., Vogels, R., and Giese, M. A. (2024). MacAction: Realistic 3D macaque body animation based on multi-camera markerless motion capture. *bioRxiv*, pages 2024–01.

[132] Mathis, A., Biasi, T., Schneider, S., Yuksekgonul, M., Rogers, B., Bethge, M., and Mathis, M. W. (2021). Pretraining boosts out-of-domain robustness for pose estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1859–1868.

[133] Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., and Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289.

[134] Mathis, M. W. and Mathis, A. (2020). Deep learning tools for the measurement of animal behavior in neuroscience. *Current opinion in neurobiology*, 60:1–11.

[135] Matsumoto, J., Kaneko, T., Kimura, K., Negrete, S. B., Guo, J., Suda-Hashimoto, N., Kaneko, A., Morimoto, M., Nishimaru, H., Setogawa, T., et al. (2023). Three-dimensional markerless motion capture of multiple freely behaving monkeys for automated characterization of social behavior. *bioRxiv*, pages 2023–09.

[136] Max Planck Institute for Evolutionary Anthropology (2024). Pan African programme: The Cultured Chimpanzee. *http://panafrican.eva.mpg.de/index.php*.

[137] Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.-P., Xu, W., Casas, D., and Theobalt, C. (2017). VNect: Real-Time 3D Human Pose Estimation with a Single RGB Camera. *ACM Trans. Graph.*, 36(4).

[138] Mielke, A., Badihi, G., Graham, K. E., Grund, C., Hashimoto, C., Piel, A. K., Safryghin, A., Slocombe, K. E., Stewart, F., Wilke, C., et al. (2024). Many morphs: Parsing gesture signals from the noise. *Behavior research methods*, pages 1–18.

[139] Mimura, K., Matsumoto, J., Mochihashi, D., Nakamura, T., Nishijo, H., Higuchi, M., Hirabayashi, T., and Minamimoto, T. (2024). Unsupervised decomposition of natural monkey behavior into a sequence of motion motifs. *Communications Biology*, 7(1):1080.

[140] Mine, J. G., Wilke, C., Zulberti, C., Behjati, M., Bosshard, A. B., Stoll, S., Machanda, Z. P., Manser, A., Slocombe, K. E., and Townsend, S. W. (2024). Vocal-visual combinations in wild chimpanzees. *Behavioral Ecology and Sociobiology*, 78(10):1–13.

[141] MMAction2 Contributors (2020). OpenMMLab's Next Generation Video Understanding Toolbox and Benchmark. *https://github.com/open-mmlab/mmaction2*.

[142] Mu, J., Qiu, W., Hager, G. D., and Yuille, A. L. (2020). Learning from synthetic animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12386–12395.

[143] Mutanu, L., Gohil, J., Gupta, K., Wagio, P., and Kotonya, G. (2022). A review of automated bioacoustics and general acoustics classification research. *Sensors*, 22(21):8361.

[144] Myagmar-Ochir, Y. and Kim, W. (2023). A survey of video surveillance systems in smart city. *Electronics*, 12(17):3567.

[145] Naik, H., Chan, A. H. H., Yang, J., Delacoux, M., Couzin, I. D., Kano, F., and Nagy, M. (2023). 3D-POP - An Automated Annotation Approach to Facilitate Markerless 2D-3D Tracking of Freely Moving Birds With Marker-Based Motion Capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21274–21284.

[146] Nakamura, T., Matsumoto, J., Nishimaru, H., Bretas, R. V., Takamura, Y., Hori, E., Ono, T., and Nishijo, H. (2016). A markerless 3D computerized motion capture system incorporating a skeleton model for monkeys. *PloS one*, 11(11):e0166154.

[147] Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., and Mathis, M. W. (2019). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature protocols*, 14(7):2152–2176.

[148] Nellissen, L., Fuh, T., Zuberbühler, K., and Masi, S. (2024). Vocal consensus building for collective departures in wild western gorillas. *Proceedings B*, 291(2033):20240597.

[149] Ng, X. L., Ong, K. E., Zheng, Q., Ni, Y., Yeo, S. Y., and Liu, J. (2022). Animal kingdom: A large and diverse dataset for animal behavior understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19023–19034.

[150] Nishida, T., Kano, T., Goodall, J., McGrew, W. C., and Nakamura, M. (1999). Ethogram and ethnography of Mahale chimpanzees. *Anthropological Science*, 107(2):141–188.

[151] Obinata, Y. and Yamamoto, T. (2021). Temporal Extension Module for Skeleton-Based Action Recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE.

[152] Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., and Yu, F. (2021). Quasi-dense similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 164–173.

[153] Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S.-H., Murthy, M., and Shaevitz, J. W. (2018). Fast animal pose estimation using deep neural networks. *Nature Methods*, 16(1):117–125.

[154] Pereira, T. D., Shaevitz, J. W., and Murthy, M. (2020). Quantifying behavior to understand the brain. *Nature Neuroscience*, 23(12):1537–1549.

[155] Pereira, T. D., Tabris, N., Matsliah, A., Turner, D. M., Li, J., Ravindranath, S., Papadoyannis, E. S., Normand, E., Deutsch, D. S., Wang, Z. Y., et al. (2022). SLEAP: A deep learning system for multi-animal pose tracking. *Nature methods*, 19(4):486–495.

[156] Phaniraj, N., Wierucka, K., Zürcher, Y., and Burkart, J. M. (2023). Who is calling? optimizing source identification from marmoset vocalizations with hierarchical machine learning classifiers. *Journal of the Royal Society Interface*, 20(207):20230399.

[157] Pichler, M. and Hartig, F. (2023). Machine learning and deep learning—a review for ecologists. *Methods in Ecology and Evolution*, 14(4):994–1016.

[158] Pillai, R., Gupta, R., Sharma, N., and Bansal, R. K. (2023). A deep learning approach for detection and classification of ten species of monkeys. In *2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES)*, pages 1–6.

[159] Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., and Schiele, B. (2016). DeepCut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937.

[160] Prieur, J., Barbu, S., Blois-Heulin, C., and Lemasson, A. (2020). The origins of gestures and language: history, current advances and proposed theories. *Biological Reviews*, 95(3):531–554.

[161] Recasens, A., Vondrick, C., Khosla, A., and Torralba, A. (2017). Following gaze in video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1435–1443.

[162] Reddy, P. R., Kumar, M. V., Kumari, K. V., Prathima, T., and Katta, S. (2023). Preventing Monkey Menace Using YOLO Based Object Detection Model. In *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)*, pages 1–7. IEEE.

[163] Roberts, A. I., Roberts, S. G. B., and Vick, S.-J. (2014). The repertoire and intentionality of gestural communication in wild chimpanzees. *Animal Cognition*, 17:317–336.

[164] Romero-Mujalli, D., Bergmann, T., Zimmermann, A., and Scheumann, M. (2021). Utilizing DeepSqueak for automatic detection and classification of mammalian vocalizations: a case study on primate vocalizations. *Scientific reports*, 11(1):24463.

[165] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[166] Rüttimann, L., Wang, Y., Rychen, J., Tomka, T., Hörster, H., Rocha, M. D., and Hahnloser, R. (2024). Multimodal system for recording individual-level behaviors in songbird groups. *bioRxiv*.

[167] Sakib, F. and Burghardt, T. (2021). Visual Recognition of Great Ape Behaviours in the Wild. In *Proc. ICPR Workshop on VAIB*.

[168] Sanakoyeu, A., Khalidov, V., McCarthy, M. S., Vedaldi, A., and Neverova, N. (2020). Transferring dense pose to proximal animal classes. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 5233–5242.

[169] Sarkar, E., Wierucka, K., Bosshard, A., Burkart, J. M., and Magimai-Doss, M. (2024). On Feature Representations for Marmoset Vocal Communication Analysis. *Available at SSRN 5003381*.

[170] Schäfer-Zimmermann, J. C., Demartsev, V., Averly, B., Dhanjal-Adams, K., Duteil, M., Gall, G., Faiß, M., Johnson-Ulrich, L., Stowell, D., Manser, M. B., et al. (2024). Animal2VEC and MeerKAT: A self-supervised transformer for rare-event raw audio input and a large-scale reference dataset for bioacoustics. *arXiv preprint arXiv:2406.01253*.

[171] Schel, A. M., Townsend, S. W., Machanda, Z., Zuberbühler, K., and Slocombe, K. E. (2013). Chimpanzee alarm call production meets key criteria for intentionality. *PLoS One*, 8(10):e76674.

[172] Schofield, D., Nagrani, A., Zisserman, A., Hayashi, M., Matsuzawa, T., Biro, D., and Carvalho, S. (2019). Chimpanzee face recognition from videos in the wild using deep learning. *Science advances*, 5(9):eaaw0736.

[173] Schofield, D. P., Albery, G. F., Firth, J. A., Mielke, A., Hayashi, M., Matsuzawa, T., Biro, D., and Carvalho, S. (2023). Automated face recognition using deep neural networks produces robust primate social networks and sociality measures. *Methods in Ecology and Evolution*, 14(8):1937–1951.

[174] Shah, K., Shah, A., Lau, C. P., de Melo, C. M., and Chellappa, R. (2023). Multi-View Action Recognition Using Contrastive Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3381–3391.

[175] Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. (2016). NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[176] Shao, D., Zhao, Y., Dai, B., and Lin, D. (2020). FineGym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2616–2625.

[177] Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019). Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[178] Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2020). Skeleton-Based Action Recognition With Multi-Stream Adaptive Graph Convolutional Networks. *IEEE Transactions on Image Processing*, 29:9532–9545.

[179] Shiu, Y., Palmer, K., Roch, M. A., Fleishman, E., Liu, X., Nosal, E.-M., Helble, T., Cholewiak, D., Gillespie, D., and Klinck, H. (2020). Deep neural networks for automated detection of marine mammal species. *Scientific reports*, 10(1):607.

[180] Shukla, A., Cheema, G. S., Anand, S., Qureshi, Q., and Jhala, Y. (2019). Primate face identification in the wild. In *PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26-30, 2019, Proceedings, Part III 16*, pages 387–401. Springer.

[181] Simonyan, K. and Zisserman, A. (2014). Two-Stream Convolutional Networks for Action Recognition in Videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, page 568–576, Cambridge, MA, USA. MIT Press.

[182] Sofroniew, N., Lambert, T., Evans, K., Nunez-Iglesias, J., Bokota, G., Winston, P., Peña-Castellanos, G., Yamauchi, K., Bussonnier, M., Doncila Pop, D., Can Solak, A., Liu, Z., Wadhwa, P., Burt, A., Buckley, G., Sweet, A., Migas, L., Hilsenstein, V., Gaifas, L., Bragantini, J., Rodríguez-Guerra, J., Muñoz, H., Freeman, J., Boone, P., R Lowe,

A., Gohlke, C., Royer, L., Pierré, A., Har-Gil, H., and McGovern, A. (2024). napari: a multi-dimensional image viewer for Python. *https://github.com/napari/napari*.

[183] Stern, U., He, R., and Yang, C.-H. (2015). Analyzing animal behavior via classifying each video frame using convolutional neural networks. *Scientific reports*, 5(1):14351.

[184] Stowell, D. (2022). Computational bioacoustics with deep learning: a review and roadmap. *PeerJ*, 10:e13152.

[185] Stroud, J., Ross, D., Sun, C., Deng, J., and Sukthankar, R. (2020). D3d: Distilled 3d networks for video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 625–634.

[186] Sturman, O., von Ziegler, L., Schläppi, C., Akyol, F., Privitera, M., Slominski, D., Grimm, C., Thieren, L., Zerbi, V., Grewe, B., et al. (2020). Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. *Neuropsychopharmacology*, 45(11):1942–1952.

[187] Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

[188] Sun, J. J., Zhou, H., Zhao, L., Yuan, L., Seybold, B., Hendon, D., Schroff, F., Ross, D. A., Adam, H., Hu, B., et al. (2024). Video Foundation Models for Animal Behavior Analysis. *bioRxiv*, pages 2024–07.

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

[190] Sun, Z., Zhu, X., Lei, Z., and Ma, X. (2022). Caged Monkey Dataset: A New Benchmark for Caged Monkey Pose Estimation. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 694–706. Springer.

[191] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

[192] Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

[193] Tinbergen, N. (1963). On aims and methods of ethology. *Zeitschrift für tierpsychologie*, 20(4):410–433.

[194] Tomasello, M. and Call, J. (2019). Thirty years of great ape gestures. *Anim. Cogn.*, 22(4):461–469.

[195] Tomasello, M. and Call, J. (2020). Ape gestures and the origins of language. In *The gestural communication of apes and monkeys*, pages 221–239. Psychology Press.

[196] Tong, Z., Song, Y., Wang, J., and Wang, L. (2022). VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093.

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

[198] Townsend, S. W., Koski, S. E., Byrne, R. W., Slocombe, K. E., Bickel, B., Boeckle, M., Braga Goncalves, I., Burkart, J. M., Flower, T., Gaunet, F., et al. (2017). Exorcising Grice's ghost: An empirical approach to studying intentional communication in animals. *Biological Reviews*, 92(3):1427–1433.

[199] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning Spatiotemporal Features with 3D Convolutional Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE.

[200] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

[201] Tuia, D., Kellenberger, B., Beery, S., Costelloe, B. R., Zuffi, S., Risse, B., Mathis, A., Mathis, M. W., Van Langevelde, F., Burghardt, T., et al. (2022). Perspectives in machine learning for wildlife conservation. *Nature communications*, 13(1):1–15.

[202] Vaswani, A. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*.

[203] von Ziegler, L., Sturman, O., and Bohacek, J. (2021). Big behavior: challenges and opportunities in a new era of deep behavior profiling. *Neuropsychopharmacology*, 46(1):33–44.

[204] Wang, C. and Yan, J. (2023). A Comprehensive Survey of RGB-Based and Skeleton-Based Human Action Recognition. *IEEE Access*, 11:53880–53898.

[205] Wang, D. (2018). *Stacked dense-hourglass networks for human pose estimation*. PhD thesis, University of Illinois at Urbana-Champaign.

[206] Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., and Qiao, Y. (2023). VideoMAE v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560.

[207] Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153.

[208] Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., and Cottrell, G. (2018). Understanding convolution for semantic segmentation. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 1451–1460. Ieee.

[209] Wang, S., Li, Z., Zhao, Y., Xiong, Y., Wang, L., and Lin, D. (2020). denseflow. https://github.com/open-mmlab/denseflow.

[210] Wang, T., Li, X., Zhang, C., Wu, M., and Zhu, K. (2024). Phonetic and Lexical Discovery of Canine Vocalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13972–13983.

[211] Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732.

[212] Weinreb, C., Pearl, J. E., Lin, S., Osman, M. A. M., Zhang, L., Annapragada, S., Conlin, E., Hoffmann, R., Makowska, S., Gillis, W. F., et al. (2024). Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics. *Nature Methods*, 21(7):1329–1339.

[213] Wiltshire, C., Lewis-Cheetham, J., Komedová, V., Matsuzawa, T., Graham, K. E., and Hobaiter, C. (2023a). DeepWild: Application of the pose estimation tool DeepLabCut for behaviour tracking in wild chimpanzees and bonobos. *Journal of Animal Ecology*, 92(8):1560–1574.

[214] Wiltshire, C., Lewis-Cheetham, J., Komedová, V., Matsuzawa, T., Graham, K. E., and Hobaiter, C. (2023b). DeepWild: Application of the pose estimation tool DeepLabCut for behaviour tracking in wild chimpanzees and bonobos. *Journal of Animal Ecology*.

[215] Witham, C. L. (2018). Automated face recognition of rhesus macaques. *Journal of neuroscience methods*, 300:157–165.

[216] Wu, B., Sakti, S., Takamichi, S., and Nakamura, S. (2024). A Neural Transformer Framework for Simultaneous Tasks of Segmentation, Classification, and Caller Identification of Marmoset Vocalization. *arXiv preprint arXiv:2410.23279*.

[217] Xin, W., Liu, R., Liu, Y., Chen, Y., Yu, W., and Miao, Q. (2023a). Transformer for skeleton-based action recognition: A review of recent advances. *Neurocomputing*, 537:164–186.

[218] Xin, W., Liu, R., Liu, Y., Chen, Y., Yu, W., and Miao, Q. (2023b). Transformer for skeleton-based action recognition: A review of recent advances. *Neurocomputing*, 537:164–186.

[219] Yan, S., Xiong, Y., and Lin, D. (2018). Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *CoRR*, abs/1801.07455.

[220] Yang, X., Burghardt, T., and Mirmehdi, M. (2023a). Dynamic curriculum learning for great ape detection in the wild. *International Journal of Computer Vision*, 131(5):1163–1181.

[221] Yang, X., Mirmehdi, M., and Burghardt, T. (2019). Great ape detection in challenging jungle camera trap footage via attention-based spatial and temporal feature blending. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.

[222] Yang, Y., Deng, Y., Xu, Y., and Zhang, J. (2023b). APTv2: Benchmarking Animal Pose Estimation and Tracking with a Large-scale Dataset and Beyond. *arXiv preprint arXiv:2312.15612*.

[223] Yang, Y., Yang, J., Xu, Y., Zhang, J., Lan, L., and Tao, D. (2022). APT-36K: A large-scale benchmark for animal pose estimation and tracking. *Advances in Neural Information Processing Systems*, 35:17301–17313.

[224] Yao, Y., Bala, P., Mohan, A., Bliss-Moreau, E., Coleman, K., Freeman, S. M., Machado, C. J., Raper, J., Zimmermann, J., Hayden, B. Y., et al. (2023). OpenMonkeyChallenge: dataset and benchmark challenges for pose estimation of non-human primates. *International Journal of Computer Vision*, 131(1):243–258.

[225] Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L., Sang, N., and Wang, J. (2021a). Lite-hrnet: A lightweight high-resolution network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10440–10450.

[226] Yu, H., Xu, Y., Zhang, J., Zhao, W., Guan, Z., and Tao, D. (2021b). AP-10K: A benchmark for animal pose estimation in the wild. *arXiv preprint arXiv:2108.12617*.

[227] Yue, R., Tian, Z., and Du, S. (2022). Action recognition based on RGB and skeleton data sets: A survey. *Neurocomputing*, 512:287–306.

[228] Yurimoto, T., Kumita, W., Sato, K., Kikuchi, R., Oka, G., Shibuki, Y., Hashimoto, R., Kamioka, M., Hayasegawa, Y., Yamazaki, E., Kurotaki, Y., Goda, N., Kitakami, J., Fujita, T., Inoue, T., and Sasaki, E. (2024). Development of a 3D tracking system for multiple marmosets under free-moving conditions. *Commun. Biol.*, 7(1):216.

[229] Zach, C., Pock, T., and Bischof, H. (2007). A duality based approach for realtime TV-L1 optical flow. In *Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007. Proceedings 29*, pages 214–223. Springer.

[230] Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. (2022). Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113.

[231] Zhan, W., Zou, Y., He, Z., and Zhang, Z. (2021). Key points tracking and grooming behavior recognition of Bactrocera minax (Diptera: Trypetidae) via DeepLabCut. *Mathematical problems in engineering*, 2021:1–15.

[232] Zhang, F., Zhu, X., Dai, H., Ye, M., and Zhu, C. (2020). Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7093–7102.

[233] Zhang, J., Huang, J., Jin, S., and Lu, S. (2024). Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[234] Zhao, J., Zhao, W., Deng, B., Wang, Z., Zhang, F., Zheng, W., Cao, W., Nan, J., Lian, Y., and Burke, A. F. (2024). Autonomous driving system: A comprehensive survey. *Expert Systems with Applications*, 242:122836.

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

[236] Zhu, Y., Lan, Z., Newsam, S., and Hauptmann, A. (2019). Hidden Two-Stream Convolutional Networks for Action Recognition. In *Computer Vision – ACCV 2018*, pages 363–378. Springer International Publishing.

[237] Zhu, Y., Li, X., Liu, C., Zolfaghari, M., Xiong, Y., Wu, C., Zhang, Z., Tighe, J., Manmatha, R., and Li, M. (2020). A Comprehensive Study of Deep Video Action Recognition. *CoRR*, abs/2012.06567.

Appendix A

Supporting Information of Chapter 2

A.1 Multi-stream GCN

A two-stream adaptive GCN is introduced in [177], building upon the spatio-temporal GCN (ST-GCN) proposed by [219]. This new architecture is motivated by some shortcomings of ST-GCN, mostly due to its fixed, predefined skeleton structure, its lack of data-dependent adaptiveness and absence of bone-related information.

They first reformulate the ST-GCN forward propagation rule f_{out} which includes the adjacency matrix, A_k, defining the fixed edges between each joint and the learnable mask matrix, M_k, indicating the edge importance weighting. The expression of f_{out} is:

$$f_{out} = \sum_{k}^{K_v} W_k (f_{in} A_k) \odot M_k$$

where K_v represents the kernel size. To make the graph topology adaptive in each layer, a new forward propagation rule is presented and constitutes the first stream of the network (so-called joint-stream):

$$f_{out} = \sum_{k}^{K_v} W_k f_{in} (A_k + B_k + C_k)$$

where A_k is an $N \times N$ matrix representing the adjacency matrix, and B_k is also an $N \times N$ matrix, whose function is similar to the mask matrix and whose parameters are learned during training. The major difference with ST-GCN, however, is that it is not part of a dot product operation but an addition to A_k. In other words, ST-GCN edges importance is only part of the features propagation if its structural link preexists, i. e. it is conditional to its skeleton's fix structure. In short, if a connection between two nodes has a value of zero in A_k, its element-wise multiplication with any M_k coefficient will be 0. In opposition, in this work,

non-structural edges will partially infer on the final action prediction. This allows actions that relate physically distant body parts, to now have a learnable edge representation on the graph. For example, actions such as "clapping both hands" will therefore be predicted more easily. The third matrix C_k is a data-dependent graph that predicts a connection between two nodes if they are similar.

Similarity between nodes is computed as:

$$f(v_i, v_j) = \frac{e^{\theta(v_i)^T \phi(v_j)}}{\sum_{j=1}^{N} e^{\theta(v_i)^T \phi(v_j)}}$$

where θ and ϕ are two embedding functions with learnable parameters. In practice, due to its softmax operation, C_k can be computed using:

$$C_k = \text{softmax}(f_{in}^T W_{\theta k}^T W_{\phi k} f_{in})$$

Here $W_{\theta k}$ and $W_{\phi k}$ are both learnable weights of the embedding functions. The output of this first stream is an action class prediction using a final softmax classifier.

The second stream of the network is called the second-order information, or bone-stream. For each structural bone, or edge, between two adjacent nodes, the length and spatial direction are coded as features of the node furthest away from the central joint. The resulting graph is fed into a separate stream, called B-stream, with similar architecture to the first stream. The output predicted by the second stream is a similar softmax classifier as the one from the first stream. Both predictions are added together, resulting in the overall network class prediction. The model outperforms previous state-of-the art architecture.

Building on their previous success, [178] offers further developments of their model via a novel multi-stream attention-enhanced adaptive GCN (MS-AAGCN). Very similar to their prior work, the new stream architecture encompasses two additional streams for joint and bone movements. The nodes' features are equal to the spatial Euclidean distance they travelled in-between frames. Final model prediction is the addition of all four stream-specific softmax predictions. Furthermore, to improve the model's performance, spatial, temporal and channel-related attention modules are integrated. The overall result exceeds the performance of other state-of-the-art models.

More recent works obtain even higher performance, such as [151], which improves MS-AAGCN, further adding a temporal extension module that enables inter-frames link prediction between nodes representing different anatomical joints.

Finally, in an attempt to reduce the number of computations, [79] makes use of a learnable spatial attention map, similar to the matrix B_k developed by [177]. They additionally integrate

a temporal attention module and optimize the overall architecture to increase computational efficiency. This work achieves state-of-the-art performance while reducing the number of computations by a factor of 9.6.

A.2 Actional-Structural GCN

When dealing with interactive systems, [100] introduces a way to predict implicit relationships in an unsupervised way by only using observational data (see Appendix A.3 for details). This approach is implemented in [115] for skeleton-based single human action recognition, where each joint is considered as an independent system. The resulting graph is used conjointly with an anatomic structural graph for a global action classification.

To implement the model of [100] into single human action recognition, the authors pre-train the ELBO loss function for a few epochs to obtain a warmed-up adjencency matrix A. This allows them to develop an actional graph convolution that incorporates the actional dependencies between each body joint.

Given the feature vector X_{in}, the model output of the actional stream is:

$$X_{\text{act}} = \sum_{k=1}^{K} \hat{A}_{\text{act}}^{k} X_{\text{in}} W_{\text{act}}^{k} \in \mathbb{R}^{N\text{x}d}$$

where $\hat{A}_{\text{act}}^{k} = D_{\text{act}}^{k^{-1}} A_{\text{act}}^{k}$ is used as the graph convolutional kernel. The matrix $A_{\text{act}}^{k} = A_{:,:,k} \in [0,1]^{N\mathbf{x}N}$ is the linking probability of the k-th edge type, W_{act}^{k} is the learnable weight matrix for the edge type k and d is the dimension of the output features.

On the other hand, somewhat similar to the forward propagation rule of [219], the output of the structural stream is:

$$X_{\text{struc}} = \sum_{l=1}^{L} \sum_{p \in \mathscr{P}} M_{\text{struc}}^{(p,l)} \odot \hat{A}^{(p)l} X_{\text{in}} W_{\text{struc}}^{(p,l)}$$

where L is the polynomial order that determines the receptive field of each node and allows messages from L-hop neighbors to be passed through the successive graph convolutions. $\hat{A}^{(p)}$ is the graph transition matrix of the p-th parted graph, whereas W and M are the weights and edge importance matrix respectively. The element-wise product is symbolized by $\odot$.

The final model output is therefore given by:

$$X_{\text{out}} = X_{\text{struc}} + \lambda X_{\text{act}} \in \mathbb{R}^{N\text{x}d}$$

where λ is a hyperparameter giving more or less importance to the actional stream. After adding a temporal convolution to capture inter-frame action features, the network's ouptut is fed into a softmax action classifier to obtain a class prediction ($\hat{y}$). The loss function for action prediction is a standard cross entropy loss:

$$\mathscr{L} = -y^T \log(\hat{y})$$

with y^T the transposed ground truth vector of the action class.

A.3 Relational Inference of Interacting Systems

Based strictly on their motion data, the model presented by [100] can infer interactions between interactive systems. At the center of the model is a variational auto-encoder [99], whose latent representation is the interaction graph and whose reconstruction is achieved via graph neural networks. The encoding part of the network inputs the observed trajectories $x = (x^1, ..., x^T)$, i. e. the trajectories of N objects over T time steps. The features of the object v_i (its location and velocity) at time t is denoted by x_i^t, the set of all features of N objects at time t by $x^t = \{x_1^t, \ldots, x_N^t\}$, and the trajectory over T time steps of the i-th object by $x_i = (x_i^1, \ldots, x_i^T)$.

The encoder's goal is to infer pairwise discrete edge types z_{ij}, representing the relationship between objects v_i and v_j, for each pair of objects. The encoder is given as

$$q_\phi(z_{ij}|x) = \text{softmax}(f_{enc,\phi}(x)_{ij,1:K})$$

where $f_{enc,\phi}(x)$ is a GNN of the fully-connected graph with learnable parameters ϕ and K the number of interaction types.

Because the variables in the latent space are discrete, the model can not backpropagate properly. Therefore, the model will instead sample from a continuous approximation of its discrete distribution as introduced by [125] to allow gradient-based learning.

The decoder's goal is to predict the location of each object at time $t+1$, i. e. $p_\theta(x^{t+1}|x^t, \ldots, x_1, z)$ or in the case of a Markovian dynamics simulation, as it is here, $p_\theta(x^{t+1}|x^t, z)$).

Learning occurs when the model maximizes the variational autoencoder ELBO loss function from [99]:

$$\mathscr{L} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}[q_\phi(z|x)||p_\theta(z)]$$

where the first term is called the reconstruction error, estimated as:

$$-\sum_{j}\sum_{t=2}^{T}\frac{\|x_j^t-\mu_j^t\|}{2\sigma^2}+\text{const}$$

whereas the latter term is called the KL divergence and is computed as the sum of entropies H:

$$\sum_{i\neq j}H(q_\phi(z_{ij}|x)+\text{const}$$

Appendix B

Supporting Information of Chapter 3

B.1 PCK Nasal Dorsum

Fig. S1 **PCK nasal dorsum**. The turquoise segment represents the length between the center of the eyes and the tip of the nose, i.e., the nasal dorsum. Any model prediction (represented in green) that falls within this distance of the ground-truth location (indicated in red) is considered as detected. In this case, all keypoints are detected except for the shoulders, neck, left wrist, and the hip (circled in purple). Hence, for this image, the detection rate would be $12/17 = 0.706 = 70.56\%$.

B.2 Prediction Comparison of Pose Estimation Models

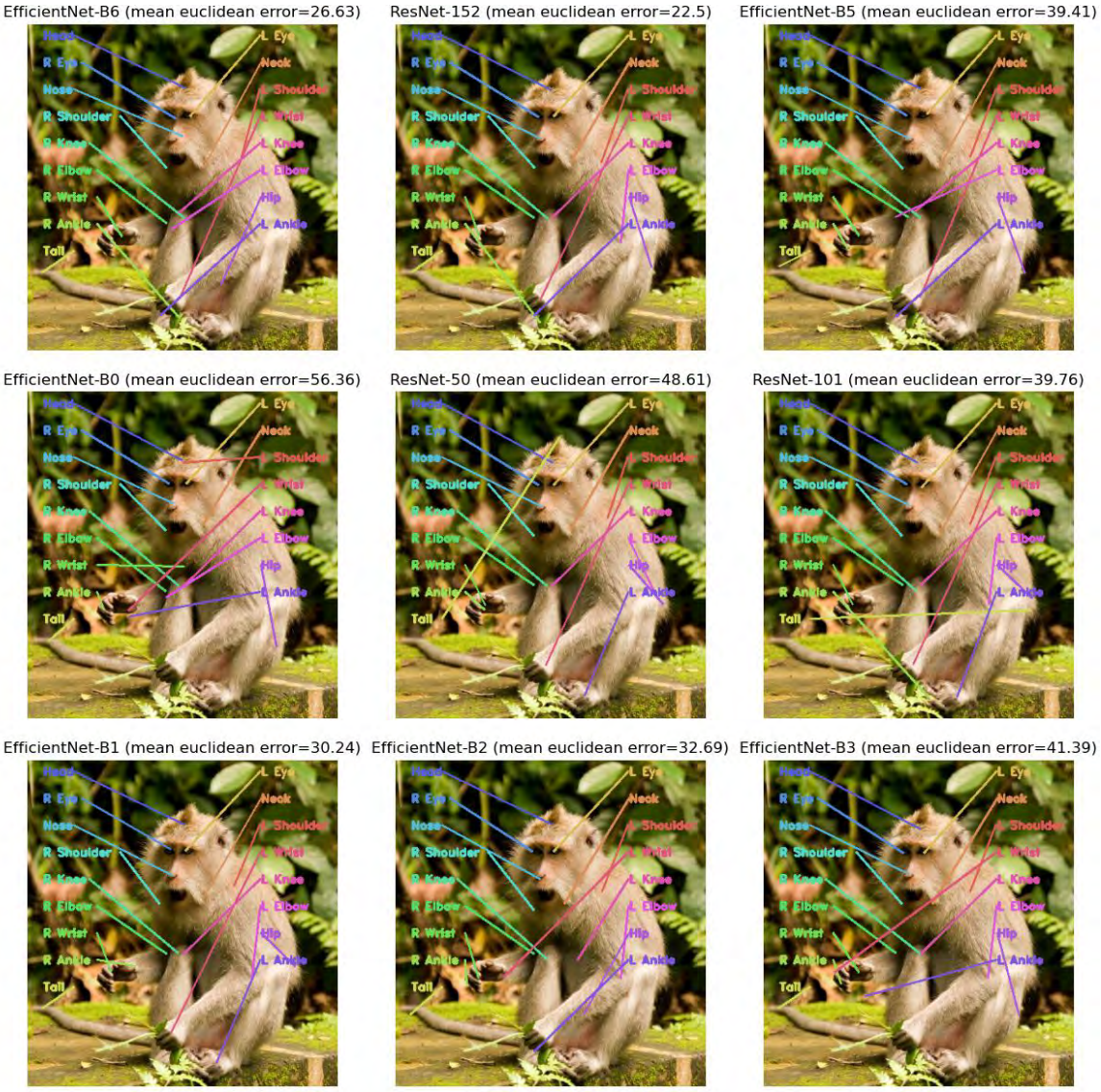

Fig. S2 **Prediction comparison of the nine models at test time.** After 40,000 training iterations, the models' test predictions are visually compared to one example of the test set. Note for example that i) ResNet-50 (*center*) wrongly predicts the top of the head as the tail's position, ii) only three models can predict the left ankle's position accurately (ResNet-50 (*center*), ResNet-101 (*center right*), and EfficientNet-B1 (*bottom left*)) and iii) no model correctly detects the left knee's location.

B.3 NMER by Families, Species and Keypoints

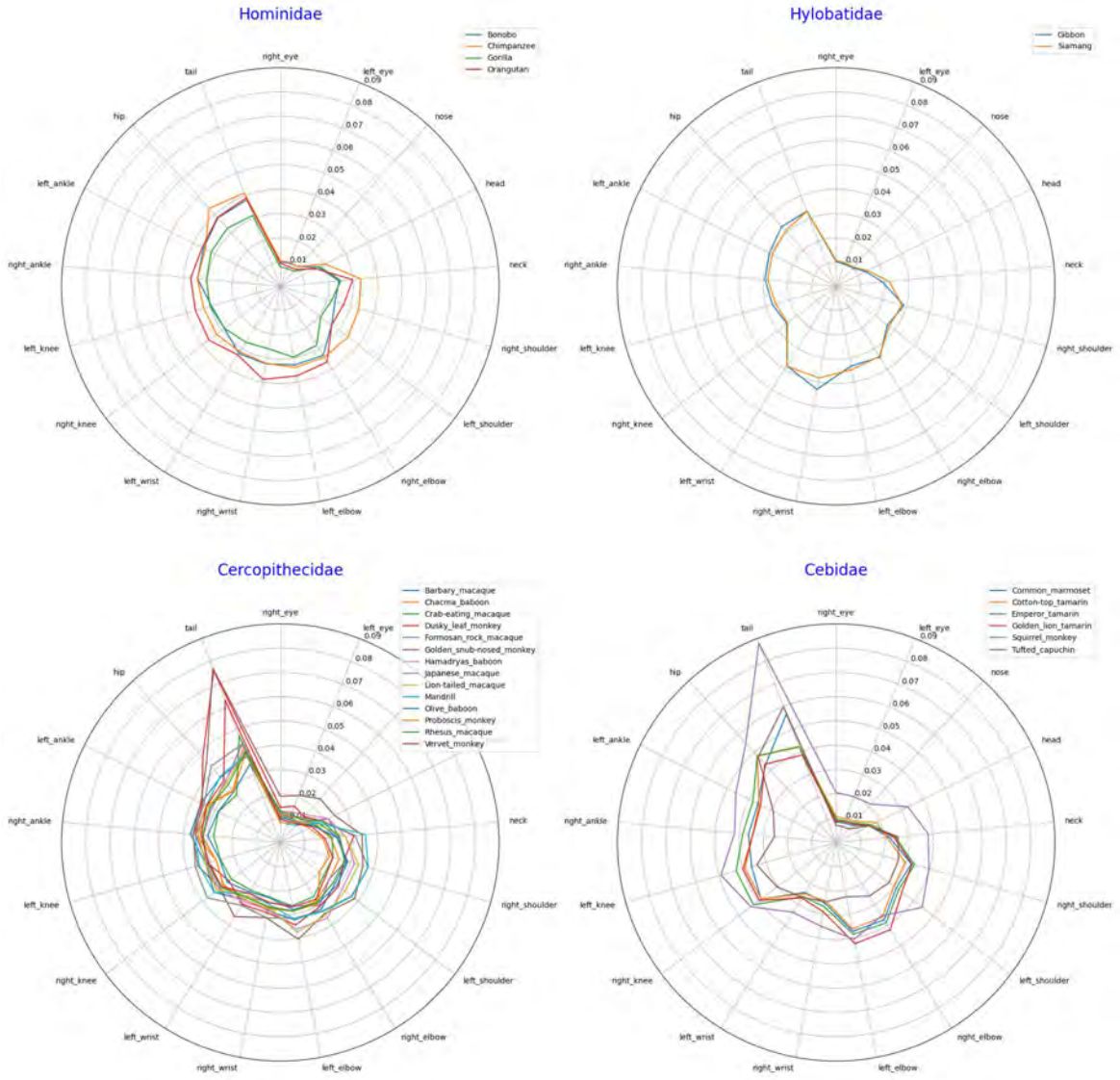

Fig. S3 **Normalized error rate by families, species and keypoints.** For all OMC images at test time, we visualize the normalized error rate (NMER) for each species.

B.4 Examples of Elements of the ASBAR GUI

Fig. S4 **Examples of UI elements of the ASBAR graphical user interface**. The GUI is terminal-based and therefore can be rendered even when accessed on a distant machine, such as a cloud-based platform or a remote high-performance computer.

Appendix C

Supporting Information of Chapter 4

C.1 Image Examples of ChimpBehave

Fig. S5 **Image examples of ChimpBehave.** ChimpBehave consists of 1,362 annotated video segments of chimpanzees, derived from a collection of 50 longer videos recorded in 2016 at the Basel Zoo indoor enclosure.

C.2 Image Examples of PanAf500

Fig. S6 **Image examples of PanAf500.** PanAf documents the behaviors of chimpanzees and gorillas in their natural habitats, using static cameras deployed in African forests to capture a wide variety of ape populations.

C.3 Pose Estimation Examples on ChimpBehave

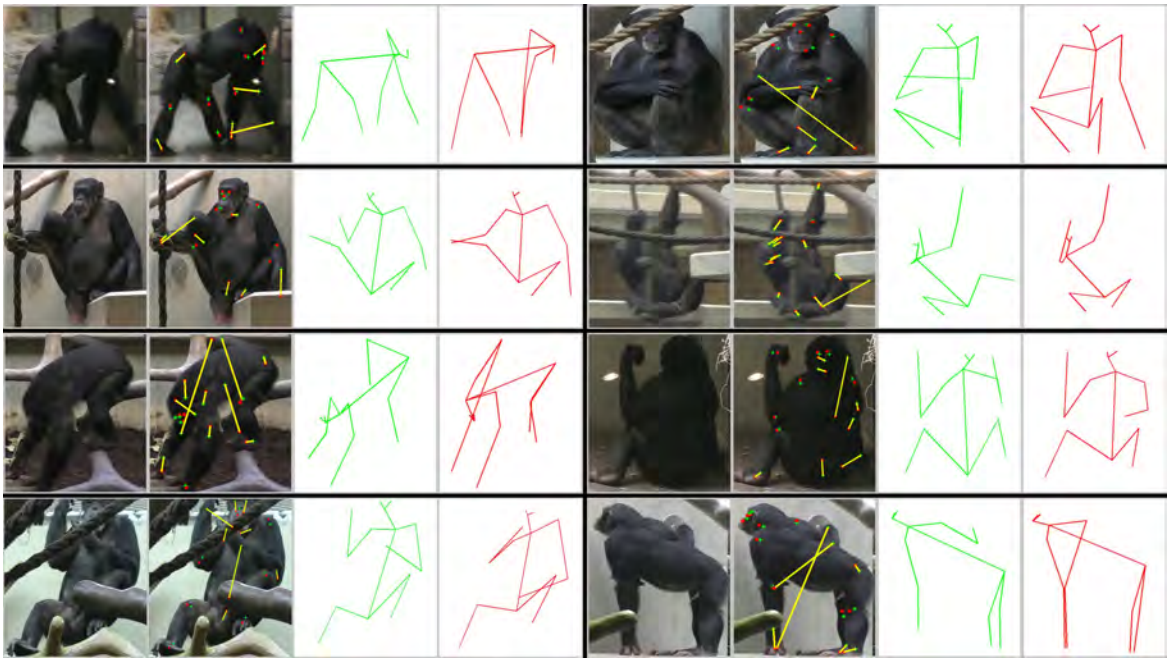

Fig. S7 **Pose estimation examples on ChimpBehave in which one of the limbs is incorrectly detected**. For example, both examples on the top row show one of the upper limb being mixed up with other visual elements. See Fig. 4.6 for more representation details.

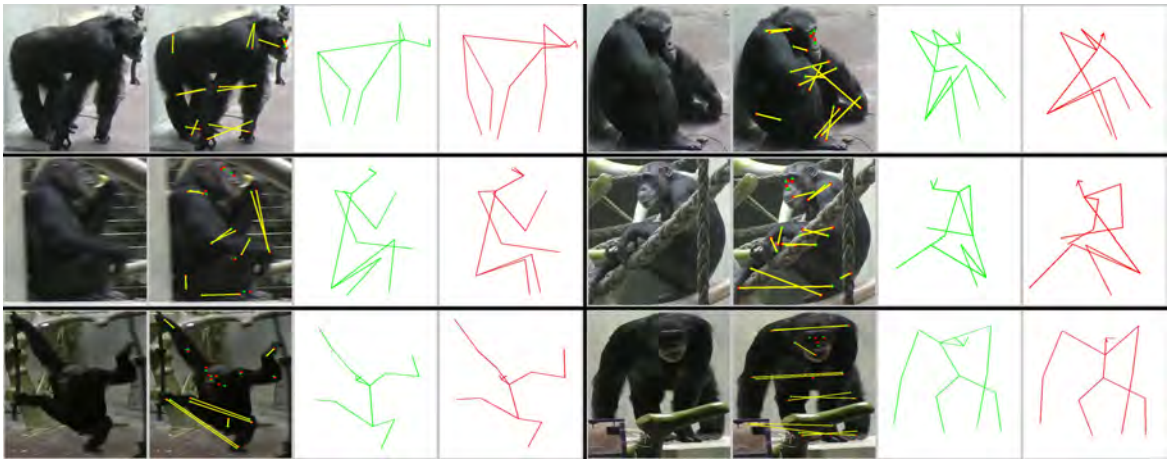

Fig. S8 **Pose estimation examples on ChimpBehave in which the overall skeleton representation is correct but HRNet inverted left/right limbs**. Such inversions may still lead to accurate behavior classification as PoseConv3D can flip symmetrical joints as a data augmentation strategy. See Fig. 4.6 for more representation details.

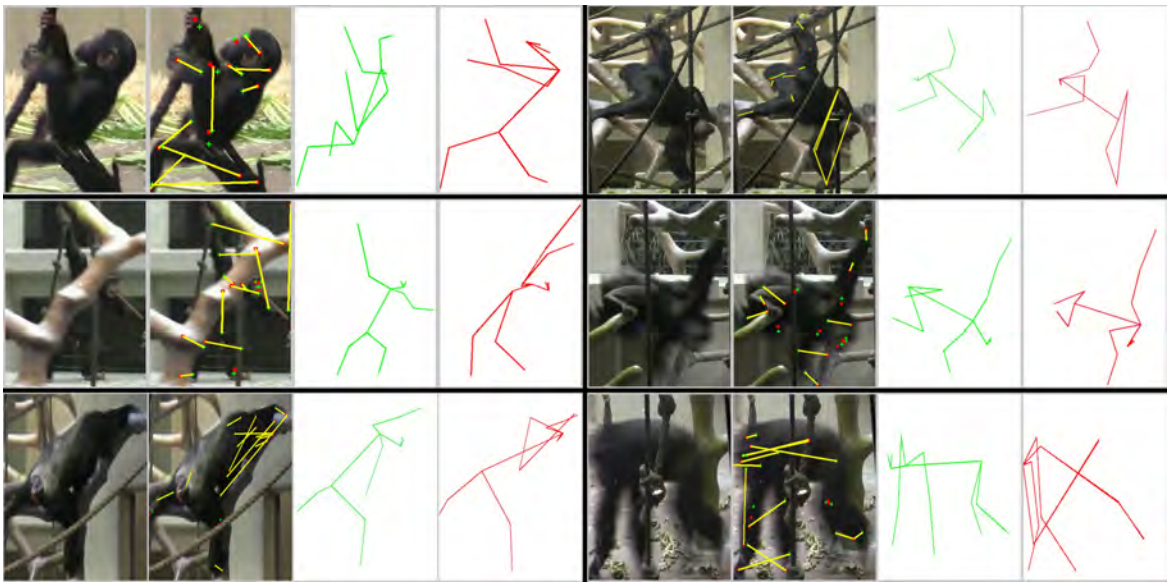

Fig. S9 **Pose estimation examples on ChimpBehave in which the overall predicted skeleton fails to capture an accurate representation of the individual's pose.** The pose estimation network may easily struggle due to various factors, such as the presence of other individuals (top left), partial occlusion (center left), less frequent postures (bottom left) or fast moving individuals and blurriness (bottom right). See Fig. 4.6 for more representation details.

Fig. S10 **Pose estimation examples on ChimpBehave in which the overall predicted skeleton seems accurate but the number of correctly detected keypoints may be low due to the individual's head orientation.** As PCK-nasal dorsum accounts for the rate of detected keypoints within the threshold of the nasal dorsum's length, this metrics is sensitive to the individual's head position, when looking down (left, PCK-ND = 26,6 %) or away from the camera (right, PCK-ND = 6,6 %) for example. See Fig. 4.6 for more representation details.

C.4 Pose Estimation Examples on PanAf500

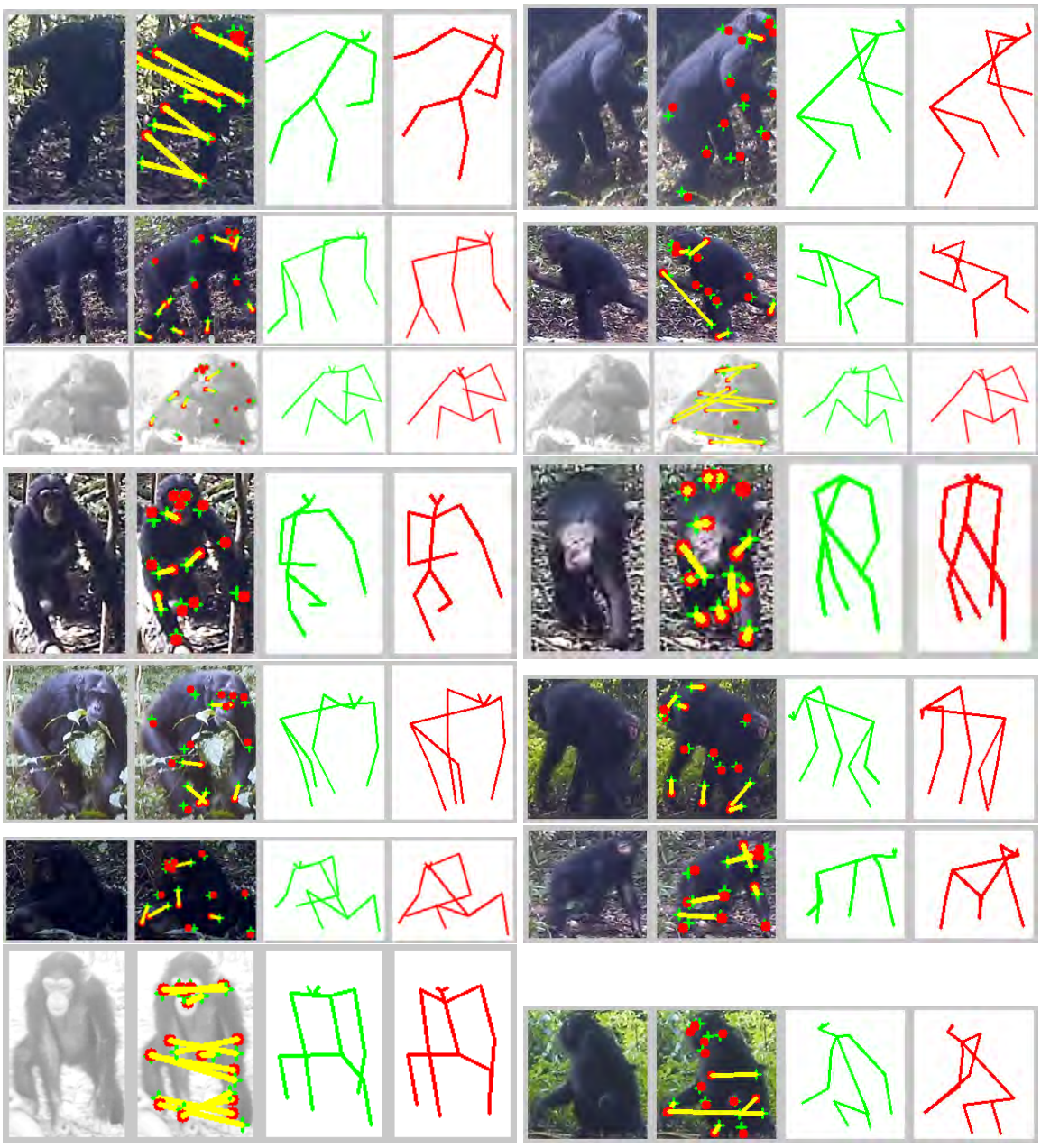

Fig. S11 **Pose estimation examples on PanAf500**. From left to right: 1) original image cropped around the individual bounding box. 2) keypoint ground truth (green crosses) and HRNet prediction (red dots), the prediction error is highlighted when greater than the nasal dorsum length (yellow segments). 3) ground truth skeleton (in green). 4) predicted skeleton (in red). See Fig. 4.6 for more representation details.

C.5 Examples of Miniclips

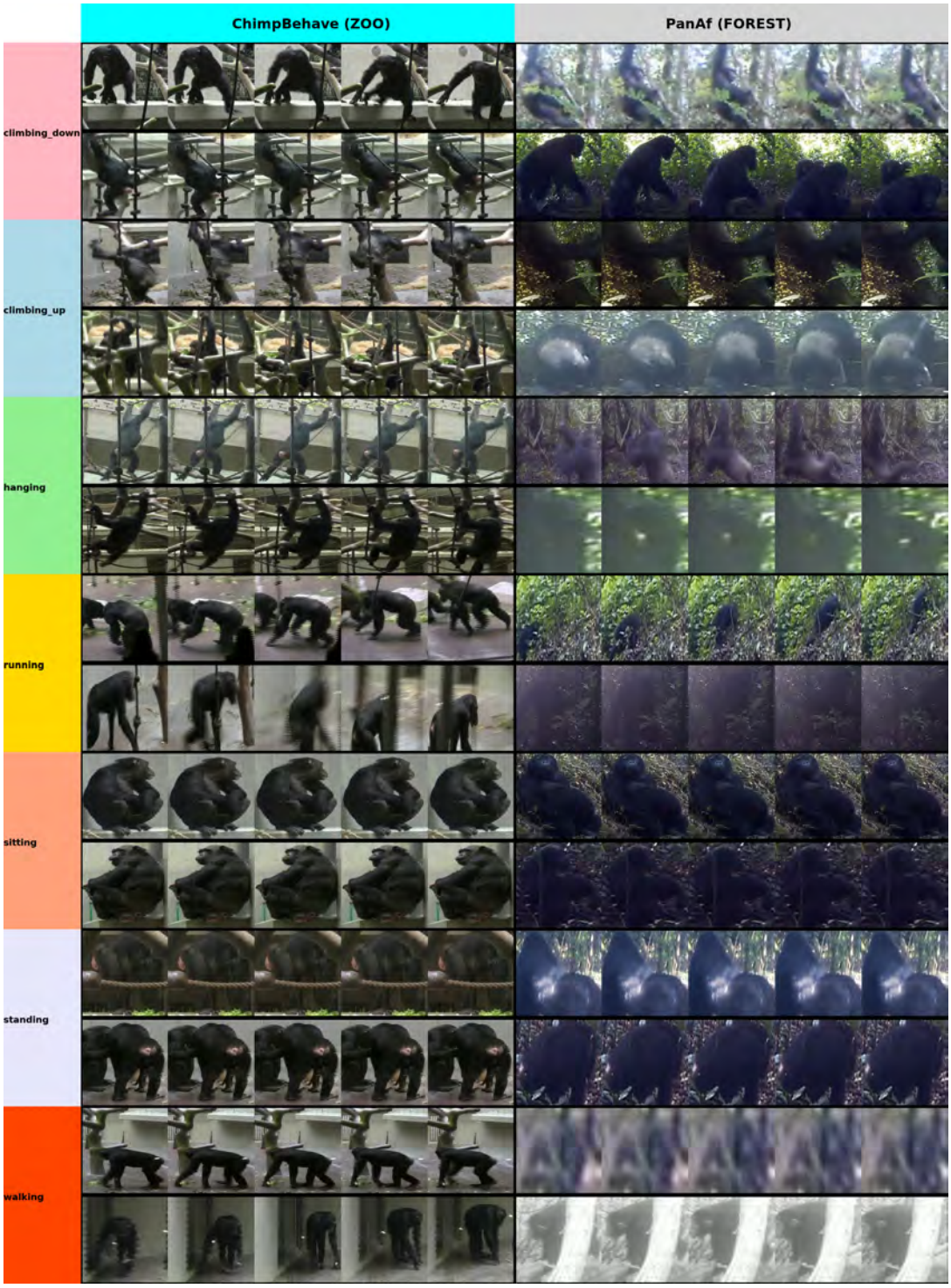

Fig. S12 **Examples of miniclips between datasets and behavior classes.** Note that we sampled 4 out of 20 frames from each miniclip for visualization purposes.

C.6 UMAP Visualization of ChimpBehave

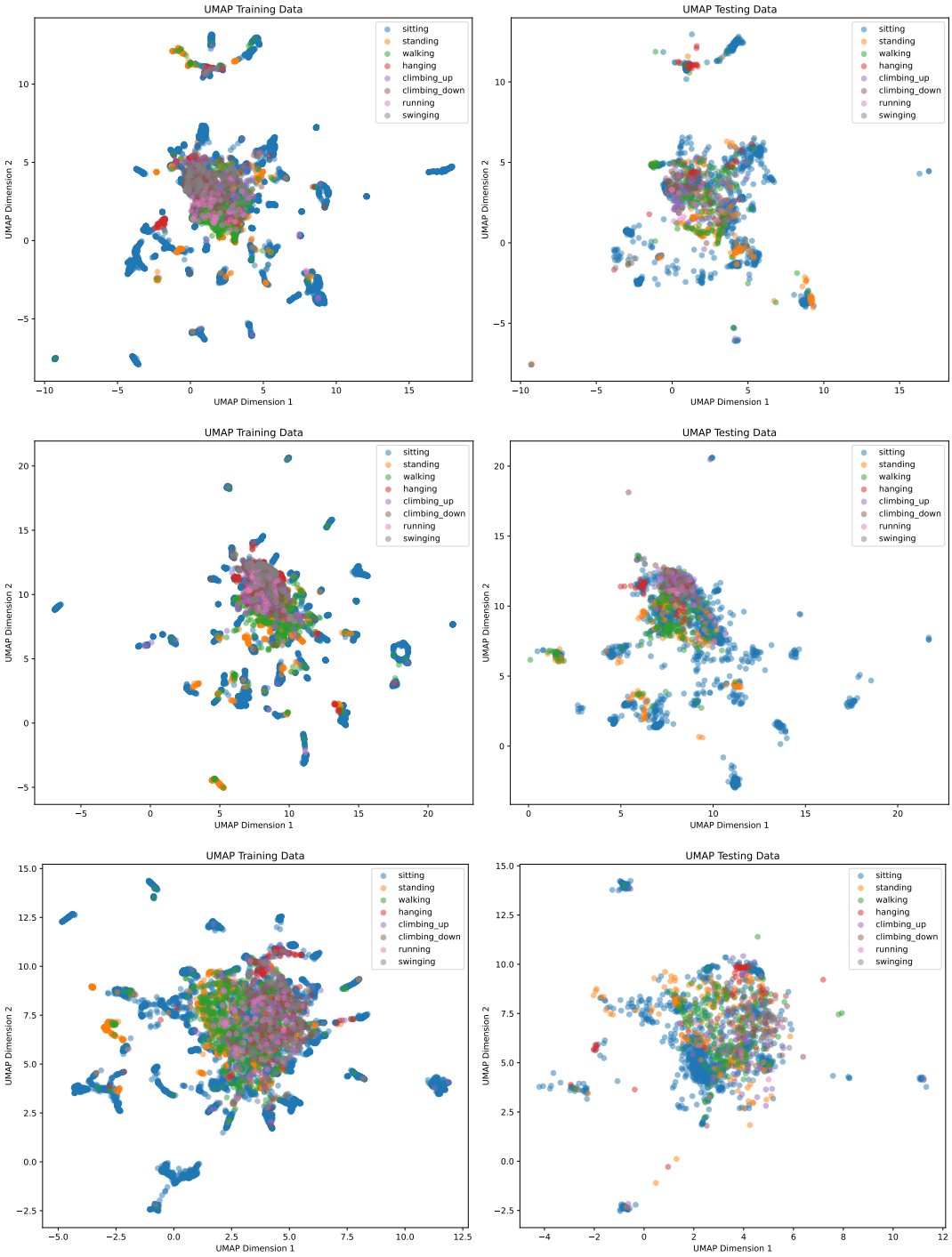

Fig. S13 **UMAP visualization of ChimpBehave for folds 1 to 3 (top to bottom) using the video-based approach.**

C.7 Confusion Matrices

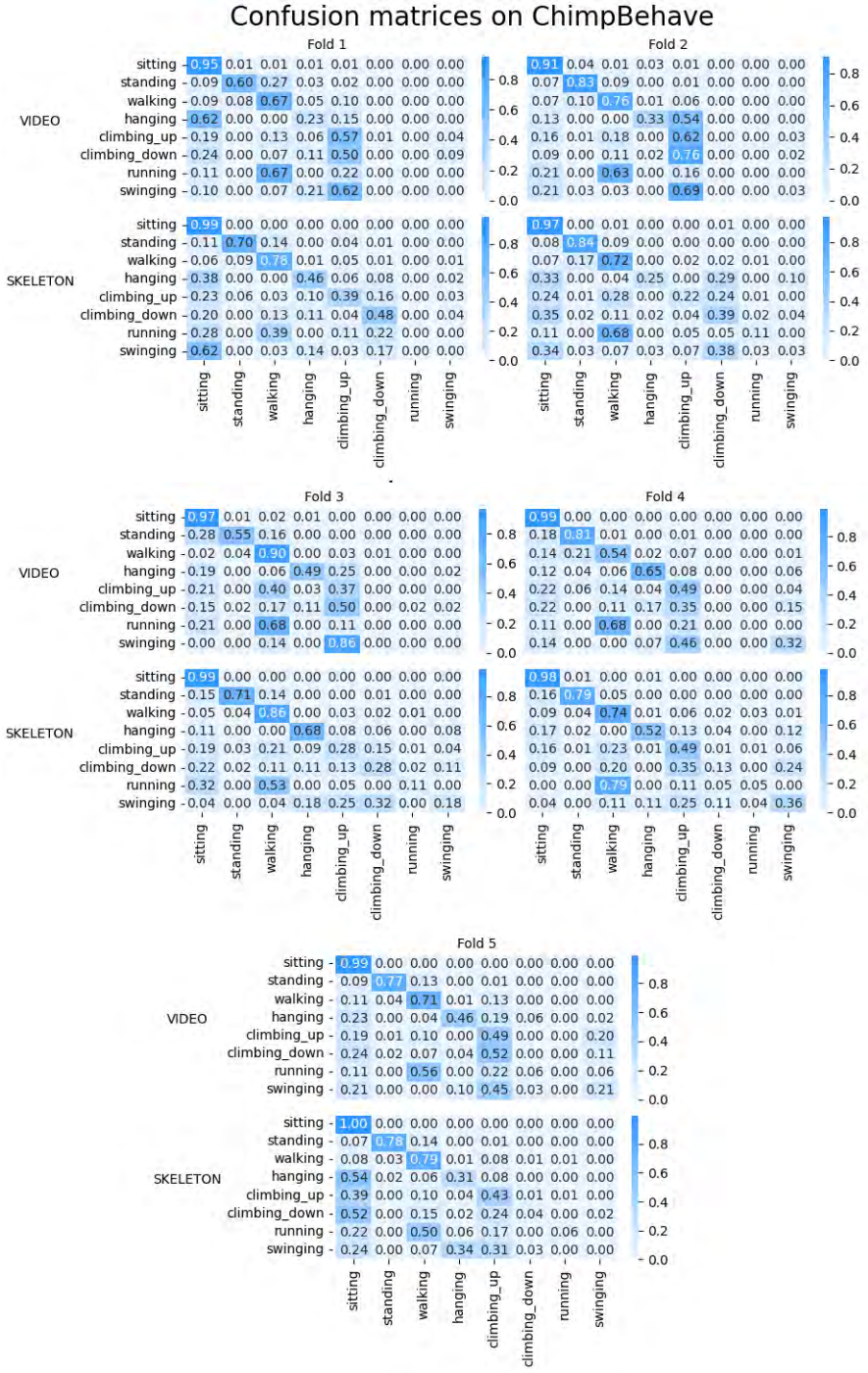

Fig. S14 **Confusion matrices of ChimpBehave (8 classes) for all cross-validation folds.**

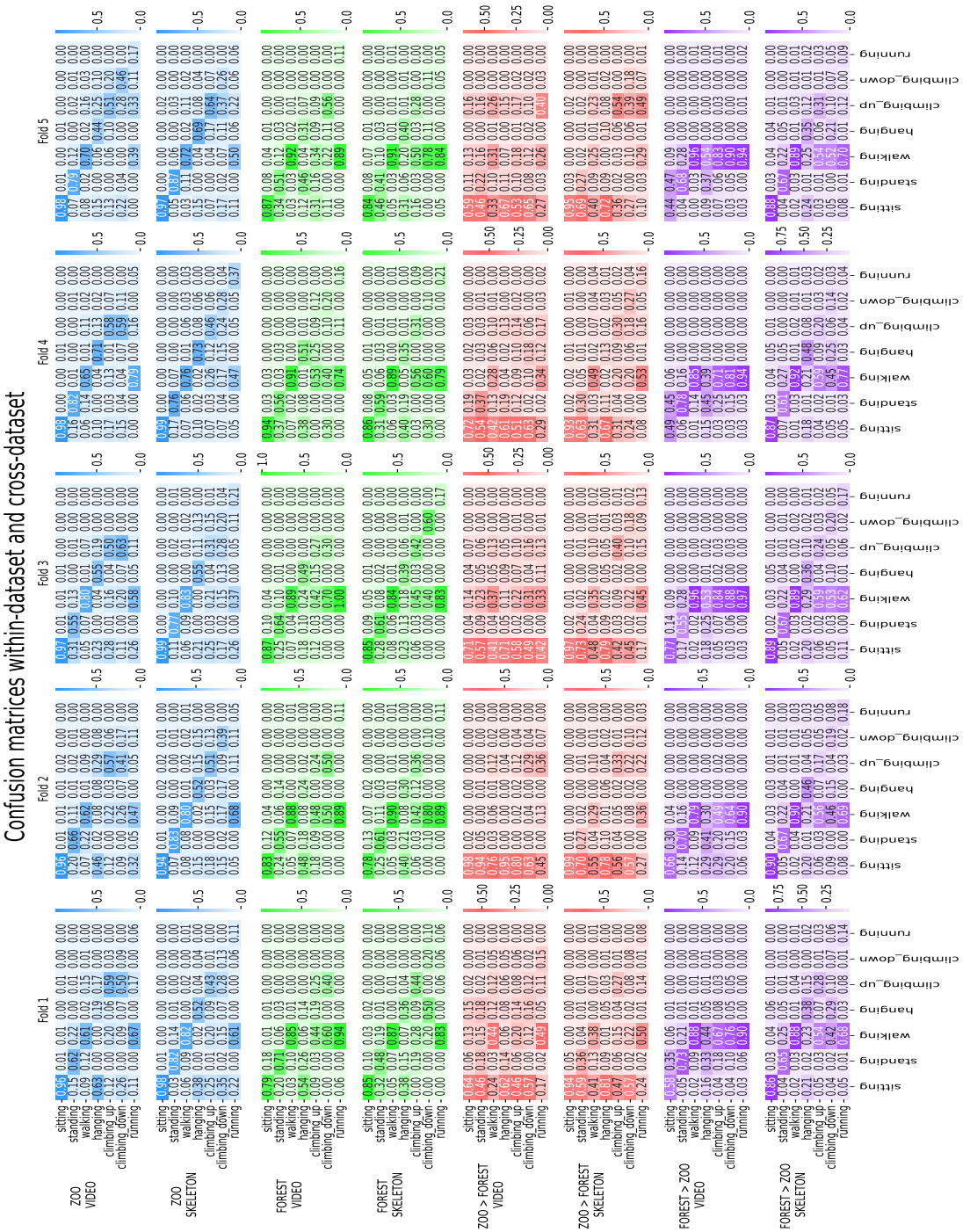

Fig. S15 **Confusion matrices of ZOO (ChimpBehave, 7 classes) and FOREST (PanAf, 7 classes) for all cross-validation folds**.

C.8 Tracking Model Fine-Tuning

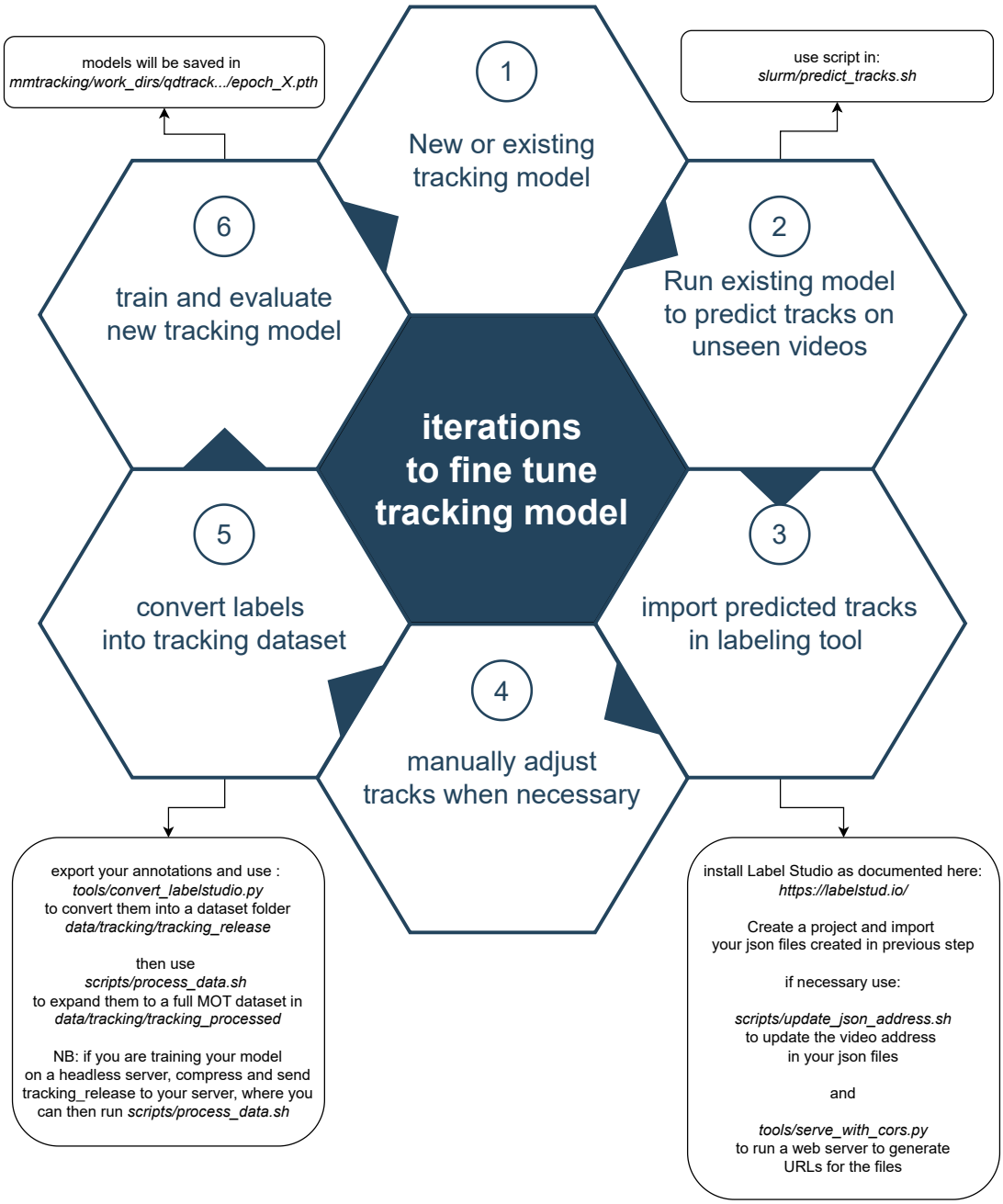

Fig. S16 **Description of steps to fine-tune a tracking model**

C.9 Behavioral Ethogram

Table T1 **Behavioral ethogram, adapted from [150]** .

Behavioral Class	**Description**
Sitting	Sitting with the rump on the ground, branch, or another surface, with the body more or less vertical.
Standing	Standing quadrupedally, e.g., pausing to scan the environment or waiting for another individual.
Walking	Walking quadrupedally.
Running	Locomoting quickly.
Hanging	Hanging by one or both hands from a structural element.
Swinging	Hanging from ropes by the arms and swinging back and forth.
Climbing down	Climbing down from an elevated site.
Climbing up	Climbing up a structural element using all four limbs.

C.10 Within-Dataset Class-level Metrics

Table T2 **Within-Dataset class-level metrics** (Mean ± Std) for ZOO (ChimpBehave, 7 classes) and FOREST (PanAf, 7 classes) across 5-fold cross-validation

	Precision	Recall	F1 Score	FPR	FNR
ZOO Video					
Sitting	0.942 ± 0.014	0.969 ± 0.008	0.955 ± 0.011	0.151 ± 0.038	0.031 ± 0.008
Standing	0.801 ± 0.061	0.690 ± 0.102	0.739 ± 0.079	0.019 ± 0.006	0.310 ± 0.102
Walking	0.671 ± 0.086	0.677 ± 0.069	0.671 ± 0.061	0.035 ± 0.012	0.323 ± 0.069
Hanging	0.462 ± 0.170	0.394 ± 0.231	0.417 ± 0.206	0.010 ± 0.001	0.606 ± 0.231
Climbing Up	0.351 ± 0.024	0.551 ± 0.039	0.428 ± 0.023	0.037 ± 0.004	0.449 ± 0.039
Climbing Down	0.328 ± 0.190	0.165 ± 0.156	0.197 ± 0.144	0.006 ± 0.005	0.835 ± 0.156
Running	0.600 ± 0.490	0.055 ± 0.061	0.098 ± 0.104	0.000 ± 0.000	0.945 ± 0.061
ZOO Skeleton					
Sitting	0.958 ± 0.009	0.973 ± 0.018	0.965 ± 0.009	0.107 ± 0.024	0.027 ± 0.018
Standing	0.861 ± 0.022	0.807 ± 0.041	0.832 ± 0.019	0.014 ± 0.003	0.193 ± 0.041
Walking	0.738 ± 0.042	0.784 ± 0.040	0.759 ± 0.017	0.028 ± 0.007	0.216 ± 0.040
Hanging	0.583 ± 0.096	0.602 ± 0.091	0.586 ± 0.074	0.012 ± 0.005	0.398 ± 0.091
Climbing Up	0.472 ± 0.082	0.472 ± 0.107	0.461 ± 0.056	0.020 ± 0.010	0.528 ± 0.107
Climbing Down	0.420 ± 0.117	0.252 ± 0.088	0.295 ± 0.064	0.010 ± 0.006	0.748 ± 0.088
Running	0.627 ± 0.199	0.160 ± 0.119	0.220 ± 0.120	0.001 ± 0.001	0.840 ± 0.119
FOREST Video					
Sitting	0.794 ± 0.016	0.860 ± 0.047	0.825 ± 0.017	0.156 ± 0.023	0.140 ± 0.047
Standing	0.655 ± 0.101	0.596 ± 0.072	0.617 ± 0.047	0.086 ± 0.040	0.404 ± 0.072
Walking	0.835 ± 0.029	0.887 ± 0.025	0.860 ± 0.017	0.076 ± 0.017	0.113 ± 0.025
Hanging	0.497 ± 0.201	0.338 ± 0.143	0.400 ± 0.166	0.018 ± 0.008	0.662 ± 0.143
Climbing Up	0.453 ± 0.201	0.191 ± 0.080	0.264 ± 0.112	0.005 ± 0.003	0.809 ± 0.080
Climbing Down	0.057 ± 0.114	0.040 ± 0.080	0.047 ± 0.094	0.001 ± 0.001	0.960 ± 0.080
Running	0.363 ± 0.318	0.075 ± 0.064	0.123 ± 0.105	0.001 ± 0.001	0.925 ± 0.064
FOREST Skeleton					
Sitting	0.771 ± 0.021	0.834 ± 0.028	0.801 ± 0.016	0.173 ± 0.022	0.166 ± 0.028
Standing	0.645 ± 0.045	0.540 ± 0.082	0.585 ± 0.060	0.076 ± 0.016	0.460 ± 0.082
Walking	0.789 ± 0.018	0.882 ± 0.023	0.833 ± 0.011	0.102 ± 0.012	0.118 ± 0.023
Hanging	0.557 ± 0.052	0.361 ± 0.038	0.438 ± 0.042	0.015 ± 0.002	0.639 ± 0.038
Climbing Up	0.646 ± 0.085	0.364 ± 0.061	0.462 ± 0.058	0.004 ± 0.001	0.636 ± 0.061
Climbing Down	0.500 ± 0.105	0.222 ± 0.193	0.283 ± 0.180	0.001 ± 0.000	0.778 ± 0.193
Running	0.430 ± 0.182	0.119 ± 0.062	0.177 ± 0.079	0.002 ± 0.001	0.881 ± 0.062

C.11 Cross-Dataset Class-level Metrics

Table T3 **Cross-Dataset class-level metrics** (Mean ± Std) for ZOO (ChimpBehave, 7 classes) and FOREST (PanAf, 7 classes) across 5-fold cross-validation

	Precision	Recall	F1 Score	FPR	FNR
ZOO>FOREST Video					
Sitting	0.503 ± 0.034	0.728 ± 0.134	0.587 ± 0.025	0.516 ± 0.167	0.272 ± 0.134
Standing	0.385 ± 0.046	0.181 ± 0.111	0.224 ± 0.099	0.076 ± 0.053	0.819 ± 0.111
Walking	0.634 ± 0.151	0.290 ± 0.130	0.362 ± 0.135	0.098 ± 0.065	0.710 ± 0.130
Hanging	0.033 ± 0.030	0.039 ± 0.038	0.032 ± 0.032	0.049 ± 0.051	0.961 ± 0.038
Climbing Up	0.033 ± 0.013	0.123 ± 0.030	0.050 ± 0.017	0.087 ± 0.055	0.877 ± 0.030
Climbing Down	0.019 ± 0.023	0.016 ± 0.015	0.015 ± 0.015	0.008 ± 0.008	0.984 ± 0.015
Running	0.200 ± 0.400	0.004 ± 0.009	0.009 ± 0.017	0.000 ± 0.000	0.996 ± 0.009
ZOO>FOREST Skeleton					
Sitting	0.556 ± 0.023	0.956 ± 0.020	0.703 ± 0.013	0.534 ± 0.060	0.044 ± 0.020
Standing	0.633 ± 0.078	0.287 ± 0.041	0.390 ± 0.030	0.045 ± 0.020	0.713 ± 0.041
Walking	0.864 ± 0.024	0.352 ± 0.081	0.494 ± 0.075	0.025 ± 0.011	0.648 ± 0.081
Hanging	0.484 ± 0.129	0.077 ± 0.034	0.126 ± 0.042	0.006 ± 0.005	0.923 ± 0.034
Climbing Up	0.145 ± 0.025	0.365 ± 0.095	0.201 ± 0.021	0.047 ± 0.025	0.635 ± 0.095
Climbing Down	0.142 ± 0.043	0.163 ± 0.066	0.139 ± 0.034	0.007 ± 0.005	0.837 ± 0.066
Running	0.237 ± 0.077	0.083 ± 0.057	0.100 ± 0.053	0.005 ± 0.005	0.917 ± 0.057
FOREST>ZOO Video					
Sitting	0.953 ± 0.023	0.588 ± 0.117	0.720 ± 0.086	0.078 ± 0.049	0.412 ± 0.117
Standing	0.219 ± 0.056	0.689 ± 0.079	0.324 ± 0.049	0.298 ± 0.099	0.311 ± 0.079
Walking	0.382 ± 0.030	0.887 ± 0.068	0.532 ± 0.017	0.147 ± 0.029	0.113 ± 0.068
Hanging	0.044 ± 0.055	0.011 ± 0.019	0.017 ± 0.028	0.003 ± 0.003	0.989 ± 0.019
Climbing Up	0.285 ± 0.180	0.020 ± 0.011	0.037 ± 0.019	0.002 ± 0.001	0.980 ± 0.011
Climbing Down	0.250 ± 0.387	0.002 ± 0.002	0.003 ± 0.004	0.000 ± 0.000	0.998 ± 0.002
Running	0.033 ± 0.067	0.004 ± 0.009	0.008 ± 0.015	0.000 ± 0.000	0.996 ± 0.009
FOREST>ZOO Skeleton					
Sitting	0.976 ± 0.001	0.877 ± 0.014	0.924 ± 0.007	0.056 ± 0.003	0.123 ± 0.014
Standing	0.719 ± 0.031	0.654 ± 0.025	0.685 ± 0.022	0.028 ± 0.004	0.346 ± 0.025
Walking	0.470 ± 0.017	0.896 ± 0.012	0.616 ± 0.015	0.102 ± 0.007	0.104 ± 0.012
Hanging	0.199 ± 0.044	0.404 ± 0.053	0.266 ± 0.047	0.045 ± 0.009	0.596 ± 0.053
Climbing Up	0.378 ± 0.033	0.241 ± 0.048	0.289 ± 0.031	0.015 ± 0.004	0.759 ± 0.048
Climbing Down	0.365 ± 0.042	0.135 ± 0.057	0.191 ± 0.064	0.006 ± 0.002	0.865 ± 0.057
Running	0.203 ± 0.079	0.125 ± 0.053	0.151 ± 0.059	0.005 ± 0.002	0.875 ± 0.053

C.12 Classes in Great Ape behavior Datasets

Table T4 **Comparison of great ape behavior datasets**

Category	ChimpBehave	PanAf500	PanAf20K	ChimpACT
locomotion	walking	walking	travel	moving
	running	running		
	standing	standing	resting	resting
	sitting	sitting		
				sleeping
	climbing up	climbing up	climbing	climbing
	climbing down	climbing down		
	hanging	hanging		
	swinging			
object interaction			feeding	eating
			tool use	
			object carrying	
				solitary object playing
				manipulating object
social interaction		sitting on back	chimp carrying	being carried
				carrying
			grooming	being groomed
				grooming
			aggression	aggressing
				losing object
				taking object
				begging
				being begging from

Category	**ChimpBehave**	**PanAf500**	**PanAf20K**	**ChimpACT**
				nursing
				being nursed
				playing
				embracing
				touching
			social interaction	
			sex	
other		camera interaction	camera reaction	
			display	displaying
			bipedal	
			vocalisation	
			piloerrection	
			cross species	
			no behavior	
				erection

Appendix D

Supporting Information of Chapter 5

D.1 Gesture Class Description

Table T4 **Descriptions of signals and their corresponding modalities from [57].**

Signal	**Modality**	**Description**
Beckoning arm	S	Stretching arm toward another, followed by a sideways sweeping movement of the arm toward the self and ending with a twirl of the wrist from palm upward to downward, indicating an invitation to approach and follow
Beckoning hand	S	Stretching arm toward another, followed by a twirl of the wrist from palm upward to downward
Bite	T	Holding another's body part between lips or teeth without pressure (mock bite)
Climb-on	S	Sweeping movement of the arm over own opposite shoulder, invitation to climb on back
Directional hand-on	T	Touching other's body part (usually back) with palm of hand and maintaining touch for more than 2s to orient other in a desired direction
Directional push	T	Gentle push on another's body part with hand(s), arm(s), feet, or head to orient partner's body in the desired direction
Drum object	A	Drumming an object with fists

Embrace	T	Signaller wraps arm(s) around recipient and maintains physical contact
Flap	S	Raising one arm and hand and making a downward slapping movement of the arm in front of another, in the air
Flap with object	S	Raising one arm and hand and making a downward slapping movement of the arm in front of another with object held in hand, in the air
Grab	T	Grabbing gently another's body part with closed hand(s)
Grab-pull	T	Grabbing gently another's body part with closed hand(s) and pulling towards self
Hand on	T	Touching head (or other body part) of another with palm(s) of hand(s) and maintaining touch for more than 2s
Hit object	A	Hitting object with closed fist
Knock object	A	Hitting an object forcefully and multiply with fist or wrist
Open arm	S	Subtle opening of the arm, invitation to body contact
Open leg	S	Subtle opening of the leg, invitation to body contact
Poke other	T	Touching firmly and briefly another's body part with finger, may be repetitive
Punch other	T	Hitting another forcefully and singly with fist(s) or wrist(s)
Pull other	T	Pulling gently another with palm of hand towards self
Push other	T	Pushing away gently another with hand(s) or arm(s)
Raise arm	S	Raising one (or both) arm(s) above head level
Reach foot	S	Holding a foot toward another by extending the leg and foot
Reach hand	S	Holding a hand toward another by extending the arm and hand
Shake head	S	Shaking head from side to side on a horizontal axis
Shake object	S	Shaking fixed object forcefully with one or both hands

Slap other	T	Slapping forcefully and singly another with palm of hand
Slap object	A	Slapping forcefully and singly object with palm of hand
Stomp multiple	A	Stamping ground forcefully with sole of foot repeatedly (more than once)
Stomp object	A	Stamping object forcefully with sole of foot once
Stomp rhythmic	A	Stamping ground forcefully and alternatively with one foot then the other rapidly
Stomp single	A	Stamping ground forcefully with sole of foot once
Stretch over	S	Stretching and raising arm till about head level with the palm facing downwards, like embracing another's body without touching, sexual invitation
Stroke other	T	Stroking another individual's body with gentle back-and-forth movement of palm of hand or fingers
Swing arm	S	Swinging arm back and forth on the side of the body, either once or repetitively
Tap other	T	Tapping repetitively another with palm of hand, with firm short contact of the fingers to the other's body (may include rhythmic repetition or single movement)
Throw object	S	Throwing an object in the direction of another
Touch	T	Touching gently another individual's body part with palm of hand, for under 2s

D.2 Class-level Metrics for RGB Stream

Fig. S16 **Class-level metrics for the RGB stream**. The classes are ordered by the number of examples in the dataset, from most to least frequent.

D.3 Class-level Metrics for Optical Flow Stream

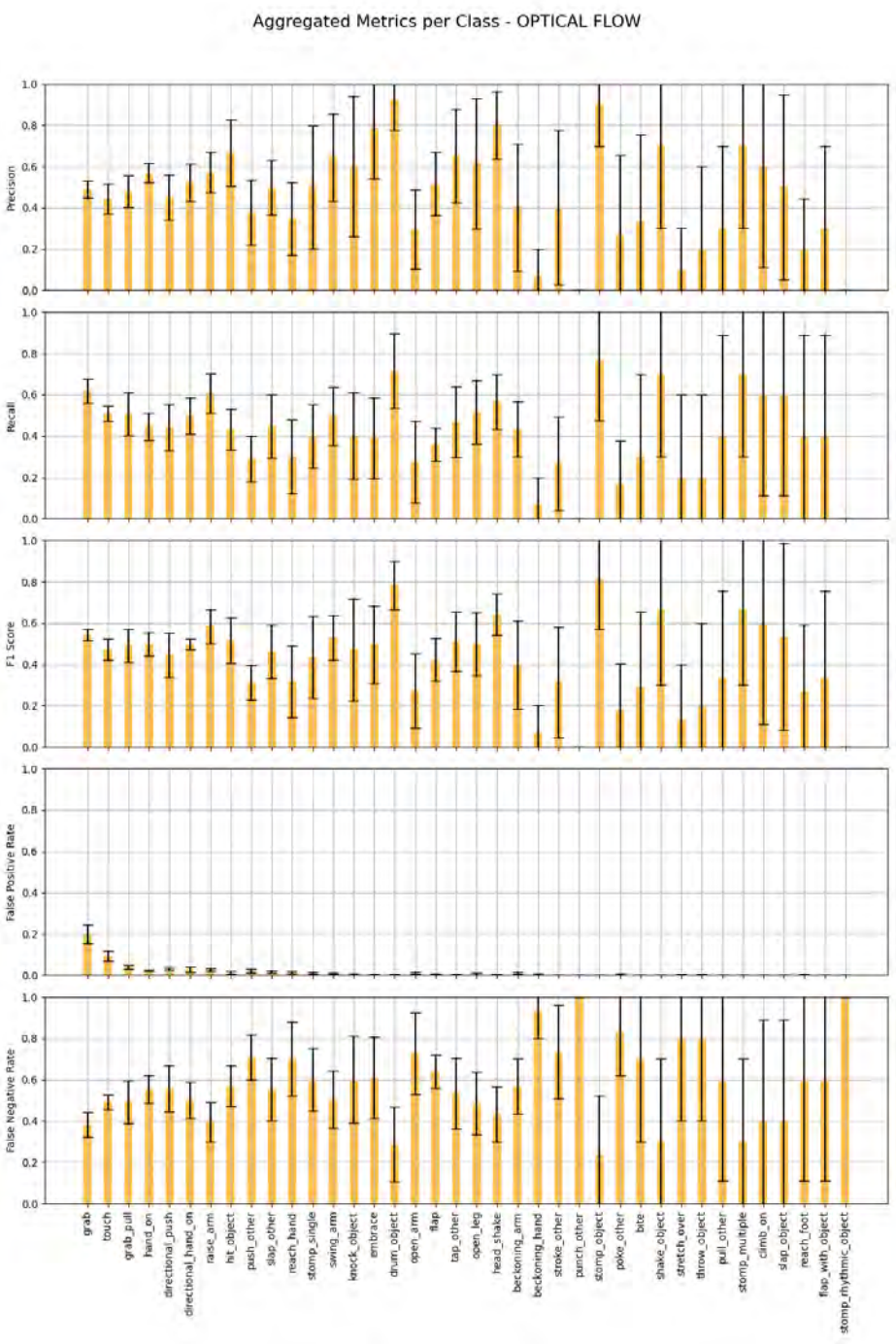

Fig. S17 **Class-level metrics for the optical flow stream.** The classes are ordered by the number of examples in the dataset, from most to least frequent.

D.4 Confusion Matrices

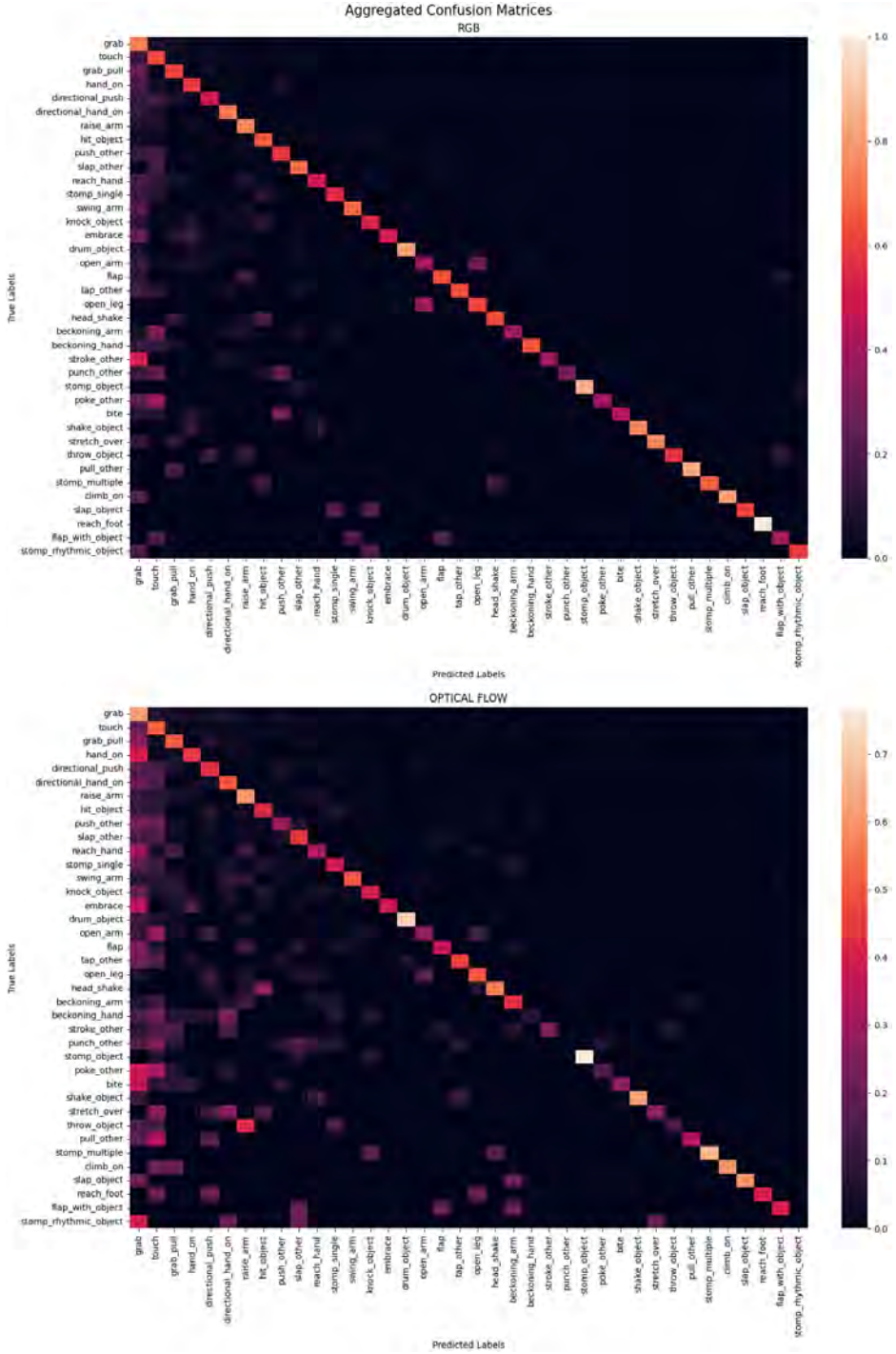

Fig. S18 **Confusion matrices for RGB (top) and optical flow (bottom).** The classes are ordered by the number of examples in the dataset, from most to least frequent.

Appendix E

Evaluation Metrics Formulas

We present here the mathematical formulas of used metrics.

$$\text{Top-1 Accuracy} = \frac{\sum_{i=1}^{N} \delta(y_i = \hat{y}_i)}{N} \tag{E1}$$

$$\text{Top-3 Accuracy} = \frac{\sum_{i=1}^{N} \delta(y_i \in \{\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \hat{y}_i^{(3)}\})}{N} \tag{E2}$$

$$\text{Mean Class Accuracy} = \frac{1}{C} \sum_{i=1}^{C} \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{E3}$$

$$\text{Mean Average Precision (mAP)} = \frac{1}{C} \sum_{i=1}^{C} \text{AP}_i \tag{E4}$$

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{E5}$$

$$\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{E6}$$

$$\text{F1 Score}_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{E7}$$

$$\text{F1}_{\text{weighted}} = \frac{\sum_{i=1}^{C} s_i \times \text{F1}_i}{\sum_{i=1}^{C} s_i} \tag{E8}$$

$$\text{False Positive Rate (FPR}_i) = \frac{\text{FP}_i}{\text{FP}_i + \text{TN}_i} \tag{E9}$$

$$\text{False Negative Rate (FNR}_i) = \frac{\text{FN}_i}{\text{FN}_i + \text{TP}_i} \tag{E10}$$

Where,

- N: Total number of predictions/instances
- C: Total number of classes
- y_i: True class for the i-th instance
- $\hat{y}_i$: Predicted class for the i-th instance (the class with the highest confidence score)
- $\{\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \hat{y}_i^{(3)}\}$: set of top-3 predicted classes for the i-th instance, ranked by confidence scores
- s_i: Support for class i (number of true instances in class i)
- TP_i: True positives for class i (number of instances from class i correctly predicted as class i)
- FP_i: False positives for class i (number of instances not belonging to class i, incorrectly predicted as class i)
- FN_i: False negatives for class i (number of instances from class i not predicted as such)
- TN_i: True negatives for class i (number of instances not belonging to class i and not predicted as class i)
- AP_i: Average Precision for class i (the area under the precision-recall curve for class i)
- $\delta(\cdot)$: Indicator function, returns 1 if the condition is true, 0 otherwise