refineDLC: an advanced post-processing pipeline for DeepLabCut outputs

Weronika Klecel[1], Hadley Rahael[2], Samantha A. Brooks[2*]

1 – Department of Animal Genetics and Conservation, Institute of Animal Sciences, Warsaw University of Life Sciences, Ciszewskiego 8, 02-786, Warsaw, Poland

2 – Department of Animal Sciences, Genetics Institute, University of Florida, Gainesville, FL 32611-0910, USA

* Corresponding author

Running headline: DeepLabCut outputs post-processing pipeline

Data availability: Unprocessed validation datasets have been deposited in Dryad.

Conflict of interest: The authors declare no conflict of interest.

Author contributions: Weronika Klecel and Samantha A. Brooks conceived the ideas and designed methodology; Hadley Rahael and Weronika Klecel collected the data; Hadley Rahael and Weronika Klecel analysed the data; Weronika Klecel and Samantha A. Brooks led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

Abstract

1. DeepLabCut has notably transformed behavioral and locomotor research by enabling precise, markerless pose estimation through advanced deep learning methods. Despite its broad adoption across species and behaviors, current usage frequently overlooks quantitative-oriented kinematic analyses due to the complexity and noise present in raw model outputs. Researchers are often challenged by the computational expertise required to clean, refine, and interpret the generated data effectively. Here, we introduce refineDLC, an innovative and comprehensive post-processing pipeline explicitly designed to streamline the conversion of noisy DeepLabCut outputs into robust, analytically reliable kinematic data.

2. Our user-friendly pipeline incorporates critical data cleaning steps – including inversion of y-coordinates for intuitive spatial interpretation, removal of zero-value frames, and exclusion of irrelevant body part labels. Additionally, it integrates robust, dual-stage filtering methods based on likelihood scores and positional changes, significantly enhancing the accuracy and consistency of data. Furthermore, our approach offers multiple interpolation strategies, effectively managing missing values to maintain data continuity and integrity. We evaluated the performance and versatility of our pipeline across two distinct datasets: controlled locomotion in cattle and field-recorded, highly variable trotting horse videos.

3. Results demonstrated substantial improvements in the quality and interpretability of processed outputs, transforming initially noisy and inconsistent data into physiologically meaningful kinematic patterns. RefineDLC successfully reduced data variability and eliminated false-positive labeling errors, offering reliable and ready-to-analyze datasets irrespective of recording conditions and animal species. By simplifying the transformation from raw DeepLabCut outputs to meaningful kinematic insights, refineDLC significantly enhances accessibility and usability, enabling a broader range of researchers – especially those with limited programming expertise – to perform precise quantitative analyses.

4. Looking forward, adaptive filtering algorithms and real-time quality assessment features are potential enhancements that could further optimize performance, expand pipeline applicability, and automate analysis. Thus, RefineDLC not only addresses the current limitations in markerless tracking technologies but also sets the stage for future advancements in precision phenotyping, behavioral ecology, animal science, and conservation biology.

Keywords: animal locomotion, behavioral analysis, DeepLabCut, interpolation methods, kinematic analysis, markerless tracking, pose estimation, post-processing pipeline

Introduction

The DeepLabCut software package revolutionized wide-field behavioral and locomotor studies by enabling precise, markerless pose estimation using deep learning techniques (Mathis et al., 2018). The tool, originally developed to be used in neuroscience and behavioral studies, has also been implemented in the field of animal science and conservation biology (Pereira et al., 2019). By providing an accessible platform for tracking and analyzing movement without the need for the physical contact with the animal, it enhanced the possibilities of conducting behavioral and kinematic studies on big samples as well as on feral populations. To this day, DeepLabCut has been successfully applied in research on numerous species, including horses (Bucci et al., 2025), dogs (Gill et al., 2024), cattle (Zhao et al., 2023) or non-human primates (Wiltshire et al., 2023). However, most of these studies focused on the specific behaviors rather than quantitative kinematic traits and didn’t develop a robust pipeline for further quantitative analysis.

Therefore, despite its widespread adoption and the availability of accessible Graphical User Interface (GUI), the usability of DeepLabCut is somewhat limited by the computational expertise required to interpret, and refine, the outputs. The raw bodypart coordinate data can be noisy and challenging to process, especially when the accuracy of the model is low. Translating these raw outputs into meaningful kinematic parameters often requires additional processing steps, which can be a barrier for researchers without a strong background in programming and data analysis.

Thus, there is a need for a comprehensible and user-friendly post-processing pipeline to enhance the usability of DeepLabCut. Such a pipeline should streamline the transformation of raw coordinate data, limiting noisiness, as well as false positive and false negative labelling. By simplifying data cleaning, filtering and interpolation, a standardized post-processing framework can facilitate more accurate and efficient analyses.

This paper addresses this need by presenting an accessible post-processing pipeline designed to facilitate kinematic analysis from DeepLabCut outputs. We aim to provide researchers with a tool that bridges the gap between raw noisy outputs and ready-to-interpret datasets, enhancing the overall utility of DeepLabCut in locomotor studies and beyond.

Methods

Material

We gathered two diverse datasets: a small set of 17 videos of walking dairy cows recorded in a controlled environment (MP4, 1080p), and large set of 524 videos of trotting horses obtained from YouTube (AVI, 720p) (www.youtube.com/arabian-essence, accessed September – October 2023). The videos of horses were not recorded for experimental purposes and therefore contained many issues such as partial occlusion of the animal, inconsistent camera panning, low video quality etc. Both sets of videos were labeled with a custom-trained resnet-50 model with 1M+ iterations, 2041 training frames for horse and 217 for cow, with the same 22 body parts labeled in both models.

In the cow model, train error reached 1.52 px with test error = 6.68 px after 1.03M training iterations. In the horse model, after 1M iterations, train error was 8.77 px while test error = 12.68 px. For this research, we selected random videos from both datasets and plotted data related to six specific body parts: right front hoof, left hind fetlock, elbow, stifle, withers, and croup (Figure 1).

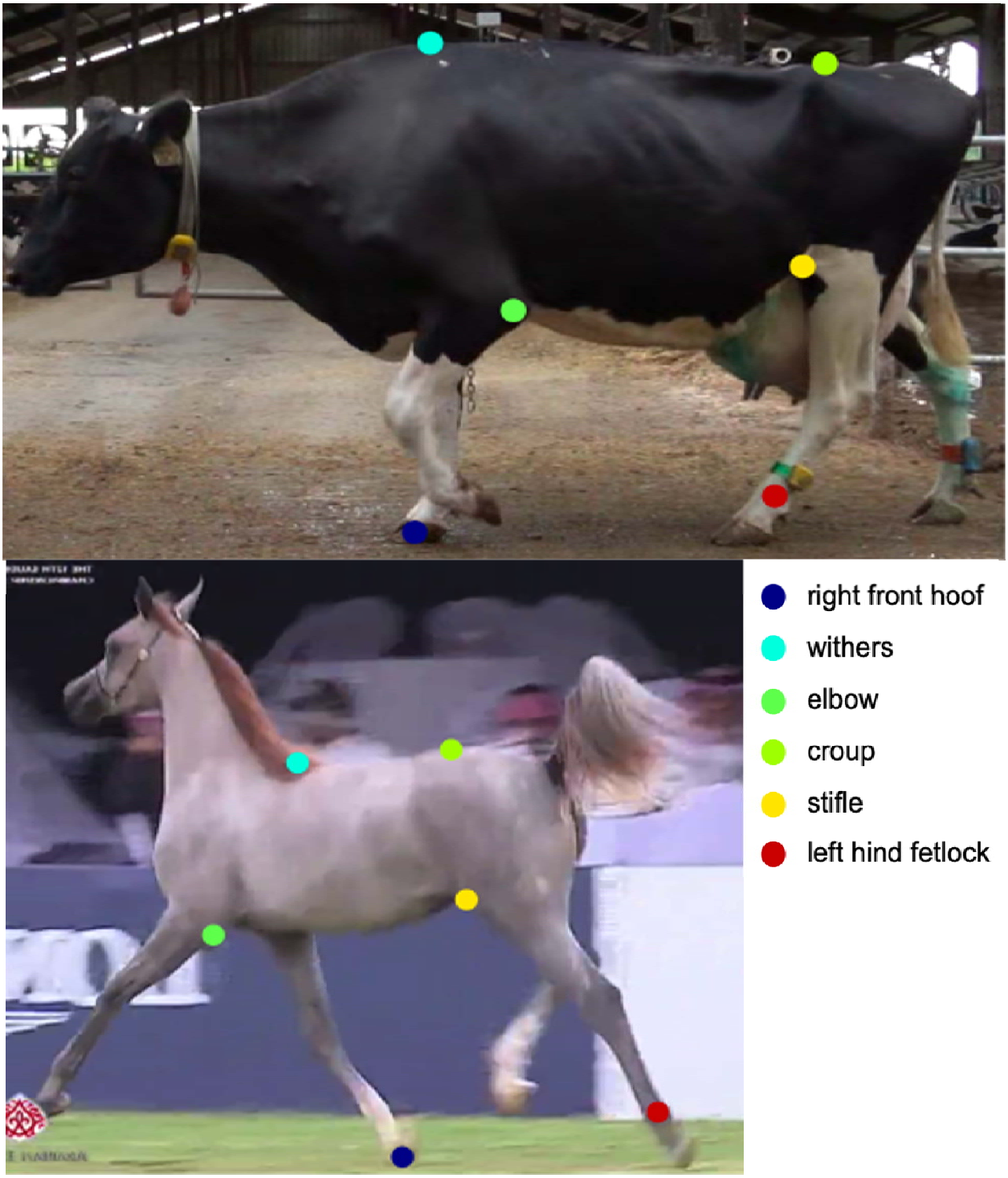


Figure 1: The six body parts selected for pilot analysis and visualization in this study: withers, croup, elbow, stifle, right front hoof, and left hind fetlock.

Data processing

Our data processing pipeline begins with essential data cleaning steps to prepare the output from DeepLabCut for further analysis. These preprocessing procedures address specific issues that have been identified during analysis of DLC output files: inverting the values of y coordinates so that the origin is at the bottom left of the frame, deleting non-processed rows, and discarding body part labels that are not relevant for further analysis. These steps are followed by filtering of the coordinate values based on the position changes of each ‘body part’ and the likelihood assigned by the DeepLabCut model, and subsequent interpolation to replace removed values. The full workflow is presented in Figure 1. Additionally, we present the script for the visualization of position changes and the distributions of relevant values, a useful feature for determining the optimum filtering parameters for unique datasets.

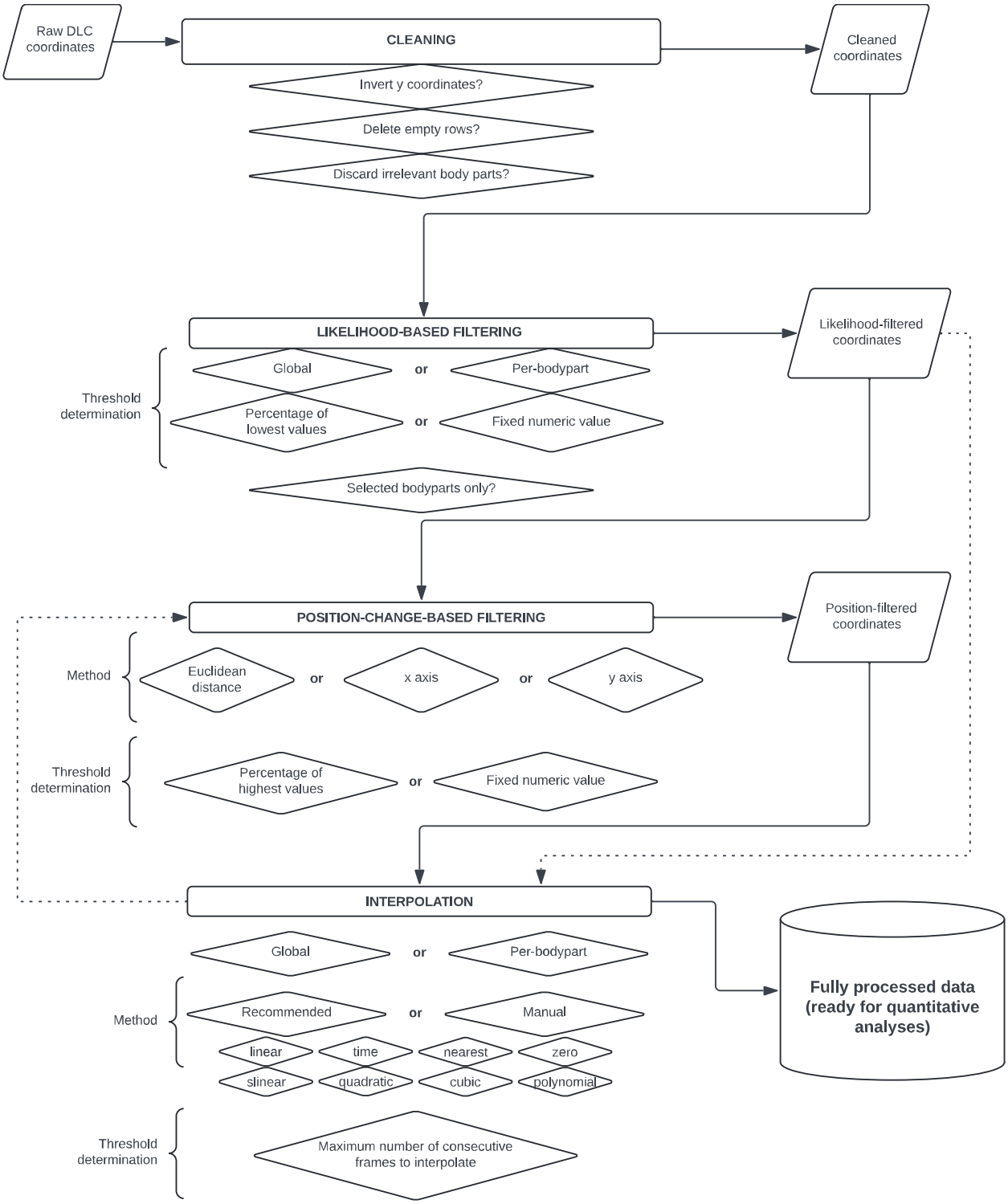


Figure 2: The workflow presented in this study. The assumed input is the raw coordinates obtained with DeepLabCut analysis contained in one directory. The final output retains the input data structure.

Data Cleaning Steps

Inversion of Y Coordinates

In computer vision and image processing, the coordinate system of digital images typically places the origin point (0,0) at the top-left corner of the image. In this system, the x-coordinate increases to the right, and the y-coordinate increases downward. This convention stems from the way images are stored in memory and displayed on screens, where pixels are mapped starting from the top-left corner moving row by row.

DeepLabCut follows this standard image coordinate system when outputting the tracked positions, resulting in y-coordinate values that increase as the position moves downward in the frame. However, for many analyses, especially those involving physical movement in real-world space, it is more intuitive and conventional to have a coordinate system where the origin is at the bottom-left, and the y-coordinate increases upward.

To align the tracking data with a more traditional Cartesian coordinate system and facilitate intuitive interpretation of vertical movements (where upward movement corresponds to increasing y-values), we included the option of inverting the y-coordinate values in the dataset. This transformation effectively mirrors the y-axis, reorienting the coordinate system so that higher y-values represent positions higher in the frame. By doing so, analyses that rely on vertical displacement – such as calculating the height of jumps or any other upward movements – become more straightforward and align with conventional mathematical representations.

Removal of Zero-Value Rows

During processing the data, we observed that some frames contained zero values across all coordinates and body parts (always in the end of the video). This issue can be caused by corrupted video frames, variable frame rates, or video encoding issues. This anomaly also arises from an occasional error within the DeepLabCut processing pipeline, which has yet to be fully addressed by the software developers (https://github.com/DeepLabCut/DeepLabCut/issues/982). These zero-value rows do not represent valid tracking data and can introduce significant inaccuracies, especially while calculating descriptive statistics, if left uncorrected. The issue can be resolved by re-encoding the video using H.264 codec with a constant frame rate and standard pixel format with tools such as FFmpeg or converting video format from AVI to MP4. However, if the rate of the non-analyzed frames is small, the simplest solution can be to remove the zero values for non-processed frames. We implemented a procedure to identify and remove any rows where all coordinate values are zero. By excluding these erroneous data points, we enhance the overall integrity of the dataset and prevent potential distortions in the analysis. Additionally, we implemented a function that outputs the total number of frames, and the number of frames corrupted in every file of the dataset to facilitate calculation of summary statistics for this issue.

Exclusion of selected body parts

In certain instances, it is helpful to construct a model that includes body parts that are not relevant to the analysis but are helpful for improving the model performance. For instance, in the horse model, the handler's legs were annotated to help the model distinguish them from the animal's legs, reducing misidentification during tracking

(Figure 3). Meanwhile, data corresponding to the handler's legs are irrelevant for analyzing the animal's movement patterns. Therefore, we opted to remove such body parts from the dataset after initial processing. This step is optional and can be tailored based on the specific requirements of the analysis. By focusing solely on the relevant body parts, we streamline the dataset and, most importantly, reduce computational complexity.

Figure 3: Handler's legs labeled in order to minimize the confusion between handler's and horse's limbs. (The predicted labels are marked with pluses, and the human labels are marked with discs).

Data filtering

Following the initial steps of data cleaning, we filtered the data based on various criteria, excluding not only low-likelihood labels, but also capturing “false positive” values. Two main filtering steps are implemented: firstly, we filter out the lowest-likelihood frames, based on the values assigned by the model. Keeping in mind that some of the erroneous labels, dubbed “false positives”, may evade the likelihood filter, we then focus

on position change-based filtering. This method allows identification of sudden and physiologically improbable label movements.

Likelihood-Based Filtering

Each frame in the DeepLabCut output is assigned with the probability of a correct labeling. Based on the value of this likelihood, the user can filter the output with the low likelihood. This can be done by either setting an absolute liklihood threshold (e.g., the default value of DeepLabCut is 0.6), or by defining a proportion of the lowest likelihood values to be discarded, based on the distribution of values. The user can also plot the distributions of the likelihood values globally, or for the selected body parts, beforehand so they can decide on the correct filter threshold tailored to their data.

The distributions of the likelihood values for each body part in both example datasets are presented in the Supplementary File (Figures S2-S7). Based on the obtained results, we filtered 5% of the lowest values globally. In addition to outputting the filtered coordinate files, the script tracks the number of discarded frames and provides an aggregate summary.

Outlier identification criteria for position-based filtering

The user can choose to filter “false positive” values based either on the Euclidean distance for the same bodypart label in consecutive or just on the change along x or y axis. The Euclidean-distance-based option is in general more robust. The x or y axis option is useful if the main goal is to calculate some speed-related parameters.

Similarly to the likelihood-filter, two filtering modes are available. In the “absolute threshold” mode, the script calculates the Euclidean distance or the difference in x- or y-

values between consecutive frames for each body part. If the computed distance exceeds a user-specified absolute threshold (in pixels), coordinates of the subsequent frame are removed from further analysis. In the percentage-based filtering, instead of absolute value, the user defines a percentage beyond which the highest distances are filtered out. When applied globally, the script aggregates all distances across all scorers and body parts to compute a single cutoff value. Alternatively, the script can calculate individual cutoff values for each body part separately, allowing for more tailored filtering when different body parts exhibit varying motion dynamics.

The last option determines how many consecutively missing frames should be interpolated. The determination of this parameter depends both on the nature of the movement and the video frequency.

Interpolation of filtered data

The filtering steps naturally result in missing data points where coordinates have been set to NaN (not a number). To address gaps in the dataset and maintain continuity, the user can interpolate missing data using one of eight methods: linear, time, nearest neighbor, zero order hold, spline linear, quadratic, cubic, or polynomial. A maximum number of consecutive frames to interpolate can be set by the user as well.

Interpolation can be performed after each filtering step separately or just once in the end. Either way, the choice of the algorithm should be chosen based on the shape of the trajectory function, so plotting the trajectories before interpolation is highly recommended. For the user convenience, the script suggests either a linear or cubic method, based on the structure of the input data. The maximum number of consecutive

frames to interpolate depends both on the video recording frequency and the type of the studied locomotion (e.g., the speed and swiftness of the animal).

Software implementation

The refineDLC post-processing pipeline is implemented as a modular Python package (Python version 3.10 or higher) consisting of four standalone scripts: `clean_coordinates.py`, `likelihood_filter.py`, `position_filter.py`, and `interpolate.py`. Each script is designed to be executed via a command-line interface (CLI), facilitating reproducibility, automation, and integration into larger data-processing pipelines.

The first step (`clean_coordinates.py`) addresses initial data preparation, including inverting y-coordinate values, removing zero-value frames, and optionally discarding irrelevant body parts. Users specify input and output file paths as command-line arguments, along with an optional parameter listing body parts to exclude from further analysis.

Subsequently, the second step (`likelihood_filter.py`) filters the cleaned data based on user-defined likelihood thresholds. Low-confidence data points identified by DeepLabCut likelihood scores below this threshold are replaced with NaN (not a number), and corresponding coordinate values are excluded. This step requires specifying the input and output file paths and the likelihood threshold value.

The third step (`position_filter.py`) applies filtering based on positional changes between consecutive frames. Users select from three filtering methods (`euclidean`, `x`, or `y`) and set thresholds indicating the maximum allowed positional change. Data points exceeding this threshold are marked as NaN to denote exclusion.

Finally, the `interpolate.py` script addresses gaps resulting from the previous filtering processes. Users can choose from several interpolation methods (linear, nearest neighbor, zero-order hold, spline linear, quadratic, or cubic) and set the maximum allowable gap size for interpolation. This maintains data continuity while respecting the temporal dynamics of motion.

Each script provides detailed logging, facilitating debugging and data tracking. Comprehensive documentation and example command-line usage are included in the repository's README.md file, promoting ease of adoption and reproducibility across research contexts.

The entire RefineDLC pipeline, along with installation instructions, usage examples, and source code, is publicly available via GitHub (github.com/wer-kle/refineDLC).

Results

Distributions of the position change values

Raw data

Across all body parts, horses at the trot consistently exhibited broader and more variable movements than the walking cattle. The average and maximum differences position changes in both datasets are presented in Table 1.

Table 1: Average and maximum position changes between neighboring frames before filtering, shown in the percentage of the video frame height (1080p for cattle, 720p for horses).

	Cattle		Horses	
Body Part	Average (%)	Max (%)	Average (%)	Max (%)
Withers	1.15	10.5	5.41	34.84
Croup	1.74	40.68	9.38	162.07
Elbow	0.64	45.37	2.41	191.01
Stifle	3.31	44.5	16.72	136.13
Right front hoof	1.32	18.36	9.43	108.4
Left hind fetlock	1.02	19.54	6.91	81.97

Filtered data

The threshold likelihood value marking the 5% of the lowest values in the whole datasets was 0.6562 for cattle dataset and 0.3655 for horse dataset. The numbers of discarded frames by body part are presented in Table 2.

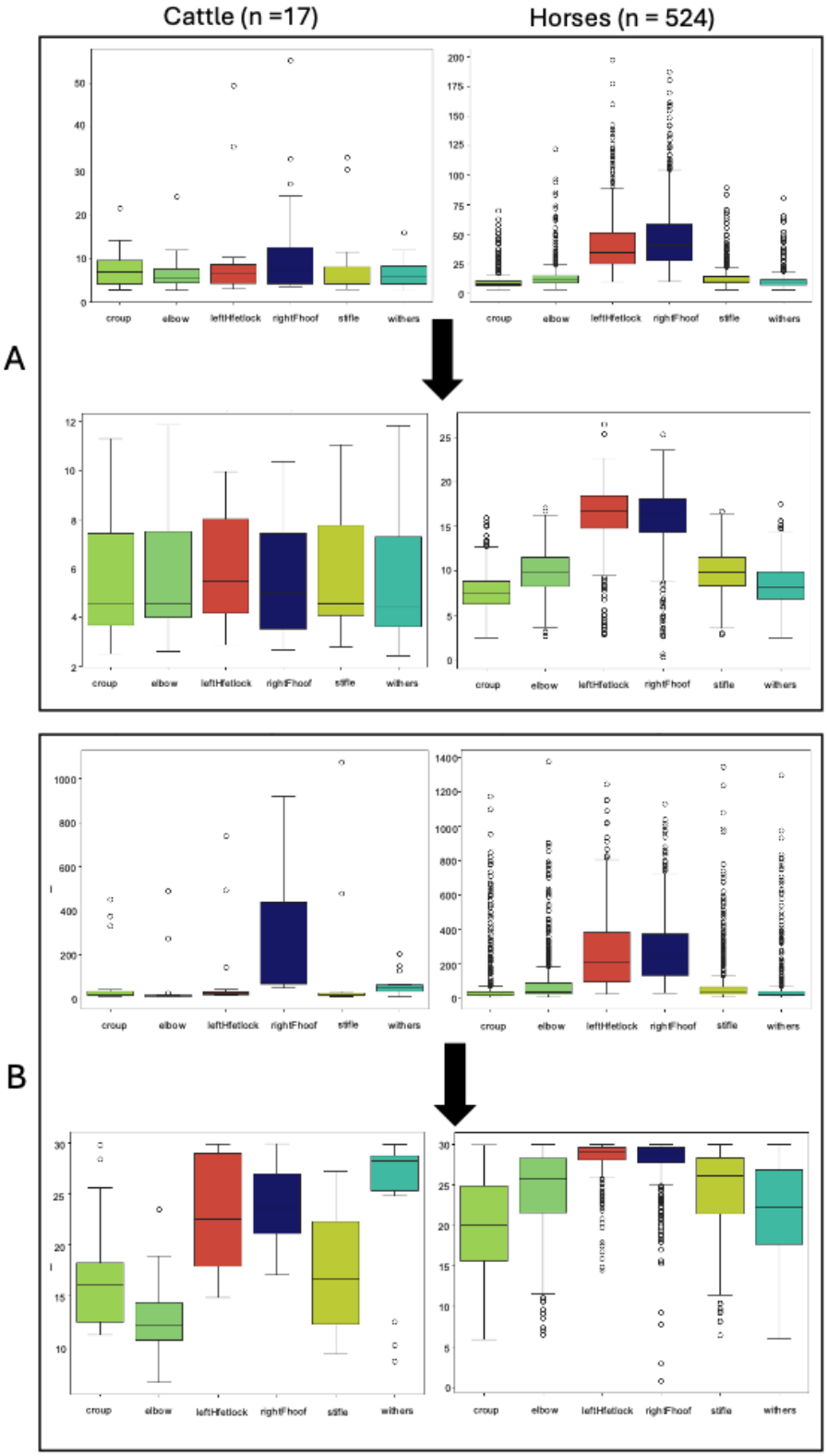
Table 2: The number of discarded frames per body part in both datasets after the application of 5% global likelihood filter.

Body Part	Cattle	Horses
Withers	138	1069
Croup	131	559
Elbow	45	1483
Stifle	101	1441
Right front hoof	316	3924
Left hind fetlock	350	4494

After applying a 30-pixel position-change threshold, both cattle and horse datasets exhibited reduced variability and narrower, more realistic, ranges compared to the unfiltered data. Post-filtering average and maximum values are presented in Table 3.

Table 3: Average and maximum position changes between neighboring frames after applying position-change threshold filter = 30 px.

Body Part	Cattle		Horses	
	Average (%)	Max (%)	Average (%)	Max (%)
Withers	0.51	2.76	1.16	4.17
Croup	0.52	2.76	1.07	4.16
Elbow	0.54	2.17	1.36	4.17
Stifle	0.55	2.52	1.37	4.17
Right front hoof	0.54	2.77	2.19	4.17
Left hind fetlock	0.56	2.77	2.24	4.17


Cattle (n =17)
Horses (n = 524)
A
B
croup
elbow
leftHfetlock
rightFhoof
stifle
withers

265

Figure 4: Distributions of the average (A) and maximum (B) position changes before and after filtering in both datasets.

Readability of the locomotion patterns

After interpolating the values that were missing from the filtering, the efficiency of the pipeline have been tested by plotting the Euclidean position changes across frames from the raw data and from the coordinate files obtained after implementing all steps. The trajectory plots drawn from a very noisy sample are presented in the Figure 5. Despite the loss of multiple data points, physiologically probable patterns in the motion of each point are now observable.

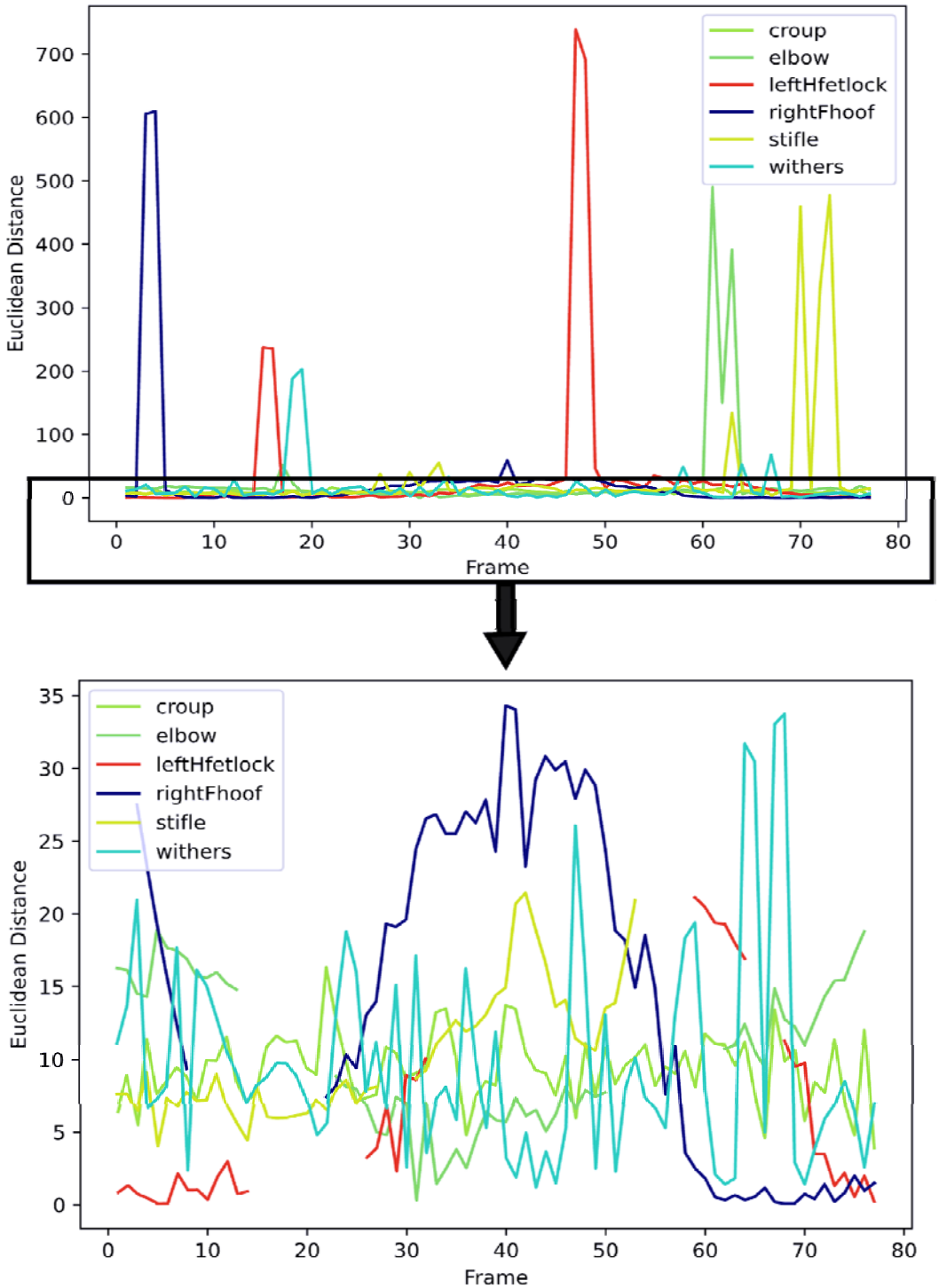


Figure 5: Movement trajectories, presented as the Euclidean position changes, drawn from the raw data (above) and post-processed data (below).

Discussion

Post-processing pipeline presented in this paper addresses an important gap in markerless tracking technologies by streamlining the transition from raw coordinate data outputs of DeepLabCut to robust, reliable, and analytically meaningful kinematic insights. Previous applications of DeepLabCut, while groundbreaking (Mathis et al., 2018; Nath et al., 2019), were often limited by the computational expertise required to manage raw outputs effectively and by the need for extensive manual post-processing to mitigate errors arising from occlusion and low-confidence predictions (Pereira et al., 2019). Our proposed approach effectively bridges this gap by providing an intuitive, user-friendly, and versatile framework, which notably enhances the accessibility of precise quantitative analyses in behavioral and locomotion studies.

A critical advancement of our pipeline is the targeted refinement of DLC outputs through sequential data cleaning, filtering, and interpolation. The initial data cleaning stage, including inversion of Y coordinates, addresses a fundamental yet frequently overlooked inconsistency between image processing conventions and the intuitive Cartesian system used in kinematic analyses. This seemingly minor adjustment notably improves interpretability, especially in vertical displacement analyses common in biomechanics research. Moreover, the systematic removal of erroneous zero-value frames and irrelevant body parts significantly enhances the integrity and efficiency of subsequent analyses, ensuring that only valid, meaningful data are processed further.

Prior approaches to filtering raw DeepLabCut outputs have largely relied on likelihood-based criteria to remove low-confidence predictions (Hayakawa et al., 2024;

Wiltshire et al., 2023). However, such methods alone may not catch spurious “false positives” that manifest rapid, non-physiological positional changes with assigned high likelihood. Our pipeline overcomes these limitations by uniquely combining likelihood-based filtering with a positional-change filtering approach. This dual strategy not only enhances the robustness of the data by mitigating errors from both low-confidence scores and abrupt, anomalous movements but also provides a more comprehensive and automated solution particularly suited to datasets of varying quality and recording conditions.

Interpolation of filtered data points is another essential component that maintains data continuity. Previous studies have explored both linear and spline-based interpolation methods to reconstruct missing data (Skjelbred & Kong, 2019). Our pipeline extends these efforts by offering a spectrum of interpolation techniques that are automatically selected based on the trajectory characteristics of the tracked body parts. This adaptive approach supported by visual trajectory plotting ensures that the reconstructed kinematic profiles preserve the temporal dynamics of the movement, aligning with recent recommendations for enhancing kinematic analyses through data-driven decision-making.

Evaluation on two distinctly different datasets, controlled cattle locomotion and uncontrolled horse movements, further validates the broad applicability of our method. Despite marked differences in recording quality and environmental control, substantial improvements in data reliability and analytical consistency were observed across both datasets. These results are consistent with recent studies emphasizing the importance of method generalizability across diverse experimental conditions (Pereira et al., 2022)

and indicate that our pipeline may offer performance benefits comparable to or exceeding those reported by other markerless tracking systems.

Nevertheless, several considerations warrant further investigation. The current threshold values for likelihood and positional-change filtering were empirically determined; future work should explore adaptive optimization strategies, potentially employing machine learning techniques, to dynamically tailor these thresholds to specific species, movement types, or recording setups, thereby further minimizing manual intervention. Additionally, integrating trajectory-based machine learning algorithms such as recurrent neural networks or reinforcement learning models may offer further refinements in prediction accuracy and robustness.

In conclusion, the presented RefineDLC pipeline significantly advances the usability and analytic accuracy of DeepLabCut outputs, offering a standardized method for precise kinematic analyses in behavioral and locomotor research. Future research directions will benefit from incorporating adaptive algorithms and additional automation features, thereby enhancing efficiency and broadening the applicability of markerless tracking technologies across a wider range of species and experimental conditions.

References

Bucci, M. P., Dewberry, L. S., Staiger, E. A., Allen, K., & Brooks, S. A. (2025). AI-assisted digital video analysis reveals changes in gait among three-day event horses during competition. *Journal of Equine Veterinary Science*, *146*, 105344.

Gill, H., Grant, R., Gardiner, J., Bates, K., Charles, J., & Brassey, C. (2024). Enhancing Understanding Of An AI-based Markerless Tracking Approach For Gait Analysis In Domestic Dogs (Canis lupis familiaris). *Proceedsings of Measuring Behavior 2024*, 155–157.

Hayakawa, S., Kataoka, K., Yamamoto, M., Asahi, T., & Suzuki, T. (2024). DeepLabCut-based daily behavioural and posture analysis in a cricket. *Biology Open*, *13*(4), bio060237.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, *21*(9), Article 9. https://doi.org/10.1038/s41593-018-0209-y

Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., & Mathis, M. W. (2019). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature Protocols*, *14*(7), Article 7. https://doi.org/10.1038/s41596-019-0176-0

Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S.-H., Murthy, M., & Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks. *Nature Methods*, *16*(1), 117–125.

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. (2022).

SLEAP: A deep learning system for multi-animal pose tracking. *Nature Methods*, *19*(4), 486–495.

Skjelbred, H. I., & Kong, J. (2019). *A comparison of linear interpolation and spline interpolation for turbine efficiency curves in short-term hydropower scheduling problems*. *240*, 042011.

Wiltshire, C., Lewis-Cheetham, J., Komedová, V., Matsuzawa, T., Graham, K. E., & Hobaiter, C. (2023). DeepWild: Application of the pose estimation tool DeepLabCut for behaviour tracking in wild chimpanzees and bonobos. *Journal of Animal Ecology*, *92*(8), 1560–1574.

Zhao, K., Zhang, M., Ji, J., Zhang, R., & Bewley, J. M. (2023). Automatic lameness scoring of dairy cows based on the analysis of head-and back-hoof linkage features using machine learning methods. *Biosystems Engineering*, *230*, 424–441.