Measuring Behavior 2024

13th International Conference on Methods and Techniques in Behavioral Research, Aberdeen, Scotland

15 – 17 May 2024

Proceedings

Multi-view triangulation-enabled annotation for multi-animal 3D pose in SLEAP

Liezl Maree[1], Shayan Afshar[1,2], Stefan Oline[2], Eric J. Leonardis[1], Annegret L. Falkner[2], and Talmo D. Pereira[1*]

[1]Salk Institute for Biological Studies, La Jolla, CA, USA.
2Princeton Neuroscience Institute, Princeton University, Princeton, NJ, USA.
***Address correspondence to: talmo@salk.edu**

Introduction

Deep learning-based markerless motion capture allows us to measure the kinematics of behavior at unprecedented resolution. SLEAP (Social LEAP Estimates Animal Poses) is a highly accessible open-source deep learning framework for markerless multi-animal 2D pose estimation that makes it particularly easy to track multiple interacting animals, making it useful for the study of social behaviors [1]. This type of behavior presents technical challenges, however, since social behaviors typically occur at close quarters, leading to frequent occlusions which result in loss of information and inferior tracking performance. To mitigate the effect of occlusions, we can use multiple cameras positioned at distinct viewpoints, affording practitioners the ability to track kinematics in 3D.

Multiple systems for multi-animal pose tracking are in development which project multi-view 2D pose into a 3D space [2]. Markerless deep learning based multi-animal 3D tracking systems have been developed for a variety of species, including rodents [3], monkeys [4], pigs [5], humans [6], and more [7]. A key challenge is generation of ground truth data for training and evaluating models. Here we present an extension to SLEAP, which enables the annotation of synchronized, multi-view, multi-animal pose, as well as 3D capabilities through integration with Anipose. Anipose is an open-source Python toolkit for camera calibration and markerless 3D pose triangulation. We propose practical solutions for the multi-view association problem and provide a usable pipeline for the multi-animal 3D pose tracking workflow. The utility of this approach is demonstrated through preliminary results on a large-scale multi-animal 3D dataset of freely-moving rodents in typical lab settings.

Methods

2D annotation pipeline

To understand the 3D annotation process, let us first review the existing 2D annotation pipeline. SLEAP currently ships with an easy-to-use annotation GUI which allows users to label body parts to track throughout a video. Users first create a "skeleton" to define which body parts to track and their connection to one another via "edges." After loading a video into SLEAP, users can begin annotating on a frame by frame basis. Within each frame, the annotator should create an "instance" for each animal in the frame. Each instance uses the skeleton to create visual markers (or "nodes") for the body parts to be labeled. The annotator then drags and drops each node to its corresponding body part location in the frame. The user should take care to annotate all animals in the frame or none at all, as leaving a frame half-labeled has the potential to teach the model to follow suit and also half-label frames.

Creating a few instances from scratch is the first step before entering the human-in-the-loop training cycle. The human-in-the-loop cycle starts by using a batch of frames annotated by the user to train a model that outputs predictions on unlabeled frames. The user then fixes these predictions and enters another round of this training cycle. Correcting the predictions output from a model is much faster than creating new annotations from scratch. Labeling and training the model in incremental batches also gives the user some feedback on which poses or lighting conditions may need more annotations.

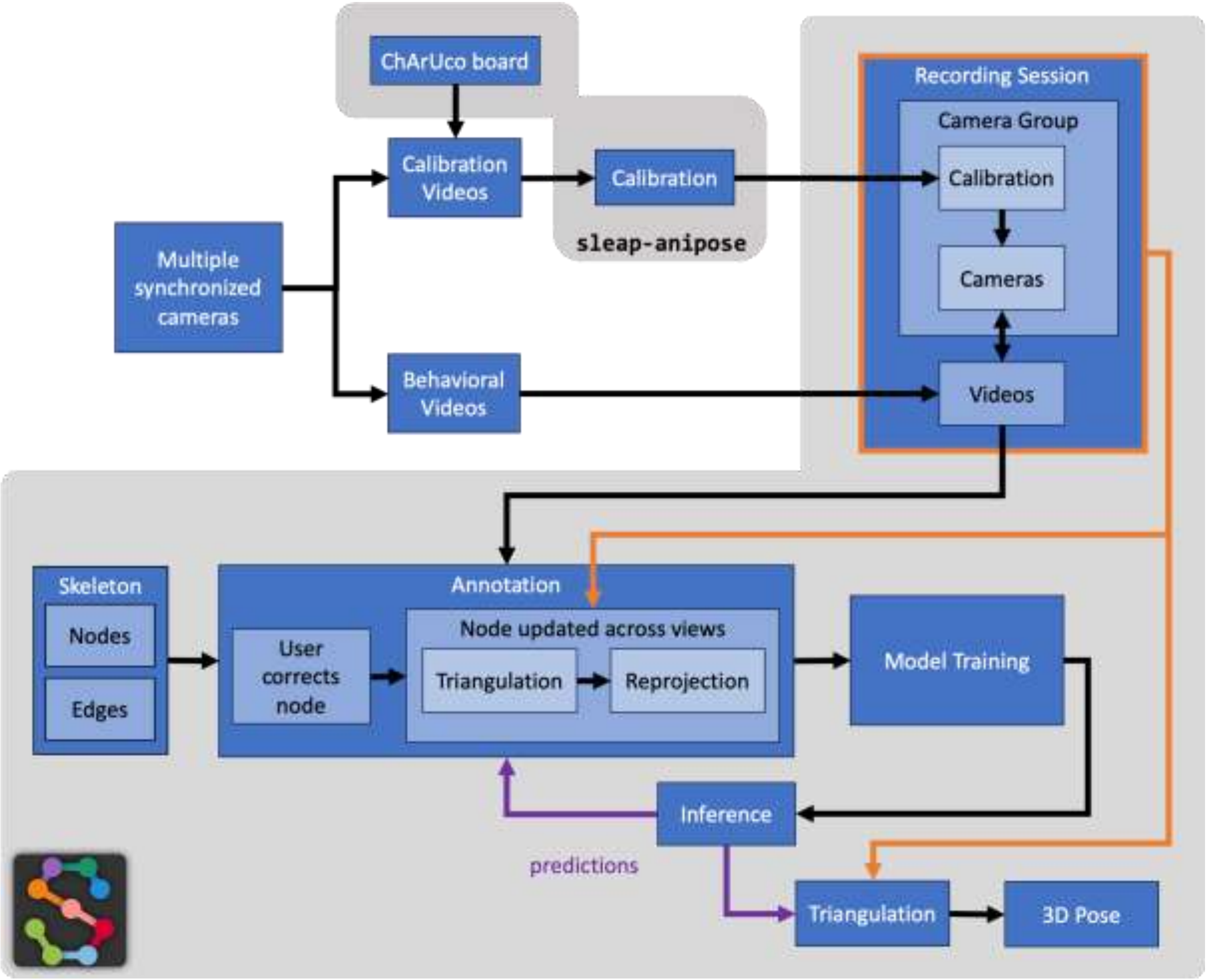

Figure 1. Flowchart of the 3D annotation pipeline using SLEAP and sleap-anipose.

3D annotation pipeline

The GUI for 3D annotation retains the usability of the original SLEAP GUI for 2D annotations with some additional improvements. When labeling, user's still only view and *directly* annotate one 2D frame at a time. However, after labeling the same temporal frame across at least two camera views, SLEAP is able to both correct and create new annotations on all camera views. The magic of 3D annotation starts with uploading a calibration file into SLEAP which parametrizes a multi-camera "recording session." The calibration file contains the extrinsic and intrinsic parameters of the cameras which allows SLEAP to take any points annotated across at least two views, triangulate them into 3D, and finally reproject the 3D points back onto all views as 2D points. The aforementioned calibration file can be created with a utility library sleap-anipose (version 0.1.7) described in further detail in the **Triangulation with anipose** section where we also outline triangulation and reprojection [8].

After uploading all related videos, the user can link videos to different cameras within a recording session. SLEAP assumes that these videos are synced, meaning that videos across different camera views start and end at the same time and have the same frame rate. Now, the user can start the multi-view annotation following a very similar user-in-the-loop training cycle with the only additional step being to annotate all frames across camera views before moving onto a new temporal frame. To easily switch between views, SLEAP has two shortcut keys to navigate forward and backwards between camera views at the same temporal frame index. After annotating the first two views, SLEAP uses these annotations to triangulate and reproject new annotations onto the remaining views. Each update made to a point in one view will also appear in related views. See Figure 1 for a visual depiction of the 3D annotation pipeline.

In the existing 2D SLEAP, annotations exist in two states. User-labeled annotations are "finalized" and ready for training, while model-generated annotations remain "pending" and require user validation before incorporation into training data. With the introduction of multi-view functionality, SLEAP now needs to account for an additional layer of annotation status. All annotations modified or created through reprojection will contribute to training, but users should have the ability to mark a point as “fixed” or “immutable,” preventing further adjustments during subsequent reprojections. SLEAP repurposes a feature initially designed for visual cues, changing a node's label color from green to red when a user manually positions a node, to indicate whether it should undergo updates during reprojection as noted in Figure 2 and Figure 3.

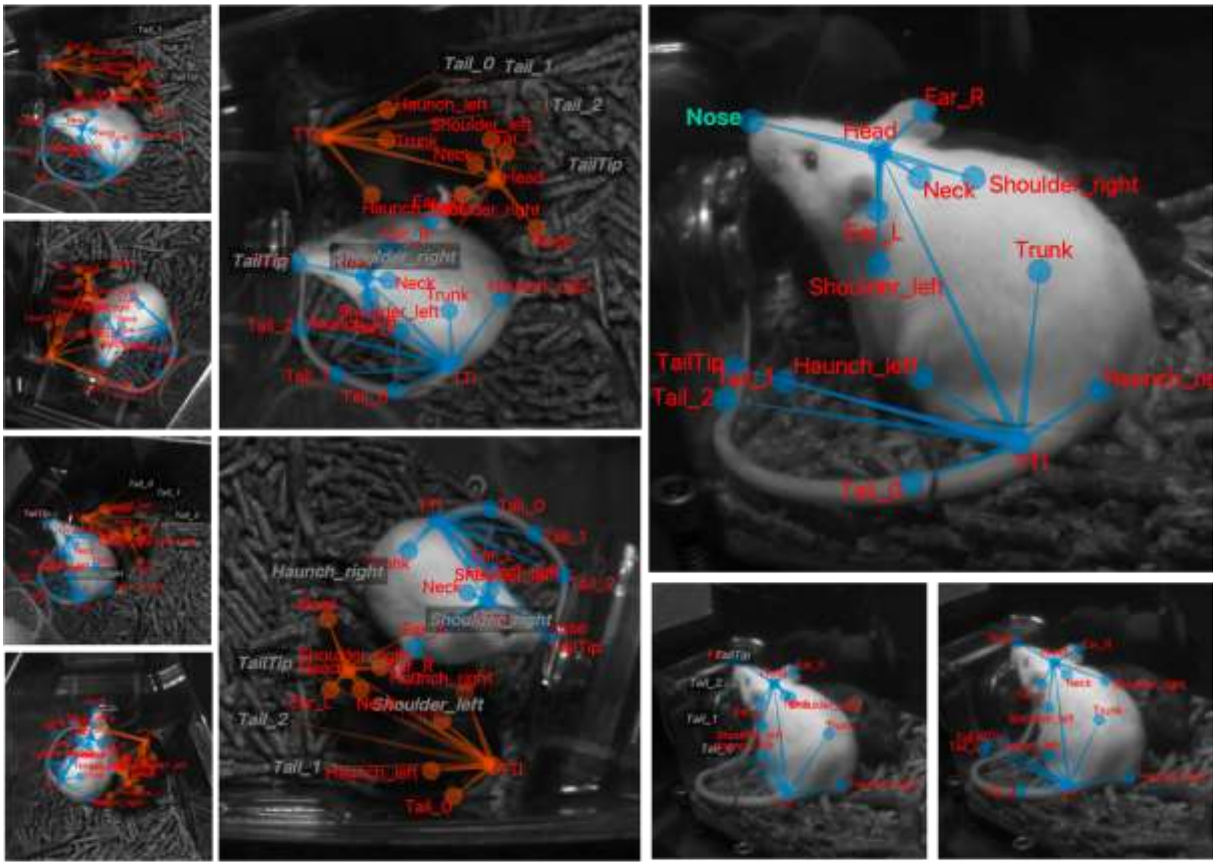

Figure 2. A collection of images from the same temporal frame index. All annotations seen above were created by double clicking predictions output from a trained model. Note that the “Nose” label in the upper right hand image is colored green to designate that the node should not be updated during reprojection.

Triangulation with anipose

Multi-view annotation within SLEAP is made possible through the utility library sleap-anipose. Extending off Aniposelib (version 0.4.3), sleap-anipose is designed for integration with multi-view SLEAP and exposes an API with convenient inputs and outputs for our needs [9]. SLEAP not only relies on the sleap-anipose library to create the calibration file needed for a recording session, but also depends on sleap-anipose to handle all aspects of triangulation and reprojection.

Calibration is performed by iteratively estimating camera parameters and 3D points then comparing them with some ground truth labels. The ground truth labels are usually estimated by using a highly recognizable stimulus, in this case, a checkerboard with Aruco patterns known as a ChArUco board. The bit encodings of the ChArUco patterns ensure that the board has a unique orientation. A calibration video is generated by pointing multiple synchronized camera views at the center of the arena, and the ChArUco board is placed in the center of the arena. Then an optimization procedure known as sparse bundle adjustment is performed to find the camera parameters and 3D points associated with the multiple views of the board. The calibration is stored as a toml file which stores the name, resolution, intrinsics, distortions, and extrinsics (rotation and translation) of each camera.

With the calibration file in hand, any point labeled across multiple overlapping views can be triangulated into 3D coordinates. Using just a single camera, we can draw a light ray from the camera’s focal point through a labeled node into 3D space. Without any other constraints on where that node is in 3D space, we can only guess that the node is somewhere along the light ray which unfortunately yields infinite possibilities. However, since the calibration file gives information on the relative positions of each camera to one another, we can narrow down where the 3D coordinate is located. If we draw light rays from a camera's focal point through a labeled node on the image plane for all cameras, we add a new constraint to the 3D point for each camera view considered. The most accurate 3D estimate of the node would be the coordinate where all light rays intersect. This process is known as triangulation, and although it sounds simple to implement, when taking into account lens distortion, which turns a light ray into something more akin to a light curve, the 3D estimate can easily lose accuracy without correct adjustment to the intrinsic camera parameters. Calibration effectively handles the intrinsic parameters, so we can confidently move onto reprojection.

Reprojection is essentially the node-update step for multi-view SLEAP. Once a node is moved, its 2D coordinates are triangulated to a 3D estimate, and finally the node coordinates are corrected to the reprojected 2D coordinates. In more detail, reprojection is when the estimations of the 3D points are multiplied by the camera parameters to transform the coordinates back into each 2D view. The reprojection error is calculated by taking the distance between the reprojected coordinates and the 2D ground truth labels. Reprojection provides a novel bootstrapping method where 3D estimates can be used to improve less reliable 2D camera views. This means that reprojection from 3D world coordinates can be used to recover poorly inferred keypoints and make pose estimation more robust. So, by adjusting a label in one view, we are able to correct the label across all views. This is particularly useful when a body part is difficult or impossible to see from a certain angle, but easily found in another view.

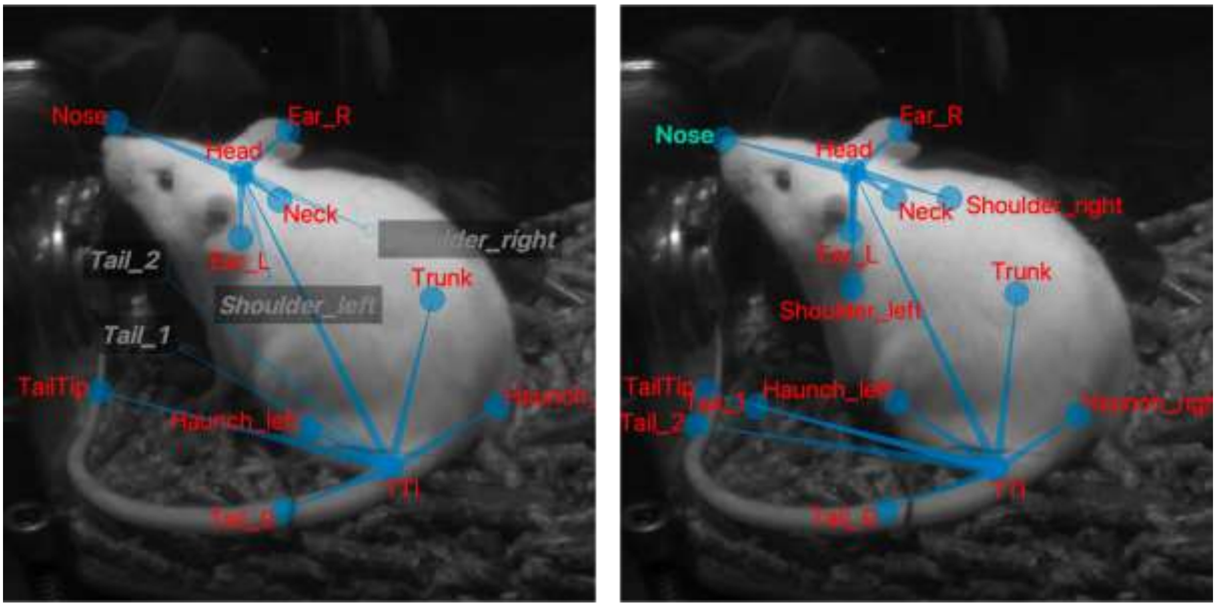

Figure 3. Two images from the same camera view and temporal frame showing a mouse before (left) and after (right) triangulation and reprojection. All nodes with green labels are considered "fixed" and are not updated during reprojection. In the above figure, the user updated the "Nose" node and the rest of the nodes were updated through reprojection. Most notably, the shoulders and "Tail_2" were updated to seemingly correct locations, while "Tail_1" still requires user adjustment.

Multi-view association

SLEAP is known for its ability to track *multiple* animals over time which can only be done by solving a temporal identity problem. SLEAP currently solves the temporal identity problem by either requiring the user to provide additional identity labels for training one of SLEAP's ID-based models, or by using a distance-based metric to determine identities across sequential frames.

Extending SLEAP to multiple views creates an additional across-view identity problem. To properly triangulate, SLEAP needs to be able to group together animals of a unique identity across camera views. While SLEAP could require that users manually label the identities of animals across views, to avoid additional strain to the annotator, SLEAP instead implements automated "hypothesis testing" to determine the correct identities for each animal as depicted in Figure 4 and Figure 5. Exhaustive hypothesis testing is a brute force approach for generating all possible groupings for animal identities, triangulating and reprojecting all groupings, and then measuring the reprojection error on every view for each grouping. This method of hypothesis testing finds the best grouping as the grouping with the lowest reprojection error, thereby solving the multi-view association problem. The major limitation of this approach is that the number of hypotheses generated is exponential in the number of cameras used (Figure 4). This limits the utility of this approach for realtime use depending on the scale of the setup, but is appropriate for most lab settings with relatively few animals and views.

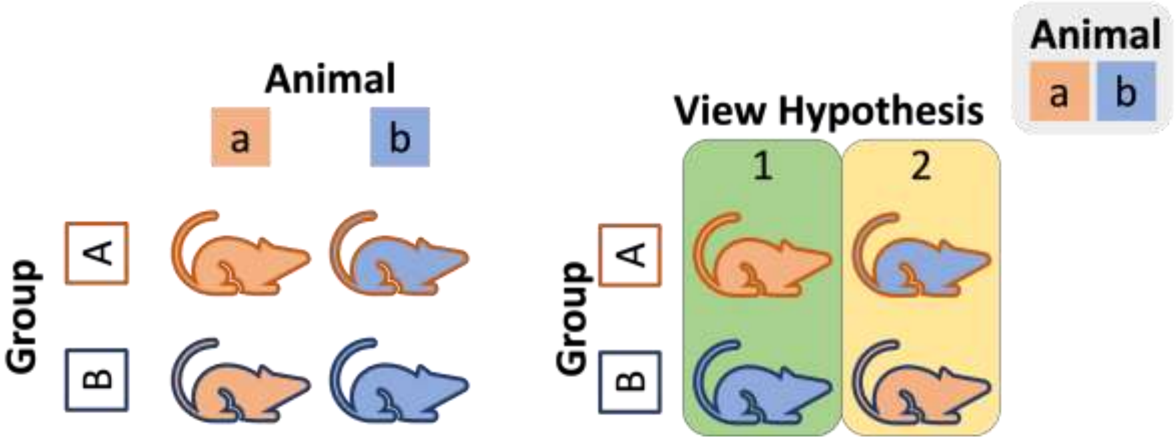

Figure 4. For a single view, hypotheses are generated by permuting the instances in the current view into each of the available instance groups. The number of available instance groups is equal to the number of unique animals in the entire video. This yields "the number of unique instances" factorial hypotheses for a single view. In this example there are 2 unique instances and, therefore, 2 grouping hypotheses with View Hypothesis 1 being the correct hypothesis.

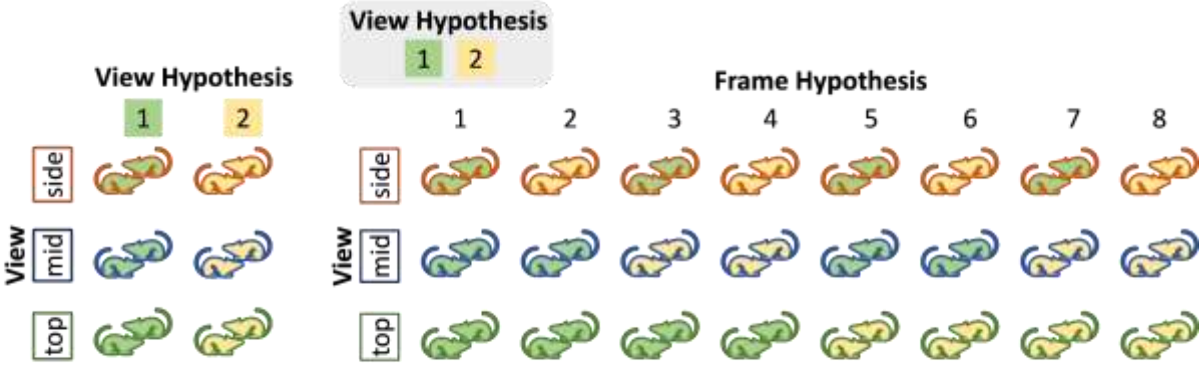

Figure 5. To generate frame-wide instance grouping hypotheses, we need to consider combinations of all possible view hypotheses. Thus, the number of frame-wide hypotheses created is "number view hypotheses" raised to the "number of views." From Figure 4, we know the correct view hypothesis is View Hypothesis 1, so the correct frame hypothesis would require all views to use View Hypothesis 1 (as is done in Frame Hypothesis 1).

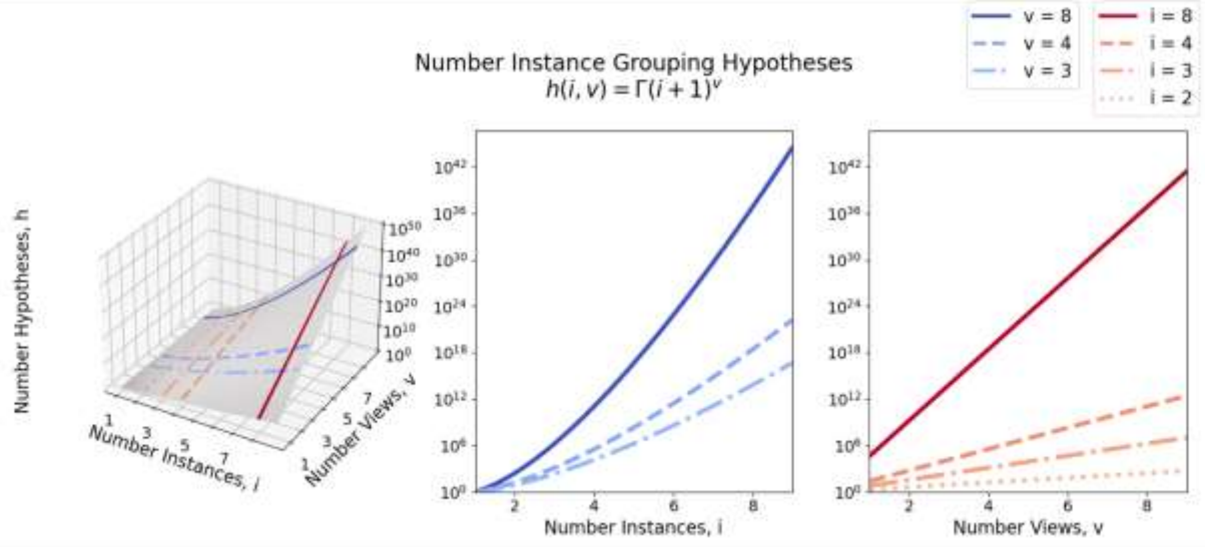

Figure 6. Plot showing the number of hypotheses generated is exponential in the number of camera views and factorial in the number of instances. The blue colored regions indicate values where the number of views are fixed while the red colored regions indicate values when the number of instances are fixed. These regions are plotted using a logarithmic scale to analyze the effects of views and instances on hypotheses separately. For visualization purposes, we use the gamma function *Γ* to interpolate the factorial function to non-integer values.

Future directions

Complete GUI integration

The GUI integration is the key piece that makes multi-view SLEAP a useful tool for its intended audience, i.e., everyone, regardless of programming ability. Multi-view SLEAP is still incomplete because users are unable to go through the entire 3D annotation pipeline via the GUI. So while the behind-the-scenes data structures are in place for a proof of concept, multi-view SLEAP still needs to add graphical elements for interacting with all adjustable aspects of these data structures.

Expose more functionality from sleap-anipose

While we aim to keep tasks modularized, i.e., sleap-anipose should handle all aspects related to calibration, triangulation and reprojection; it would be convenient if the user could access more of these utilities from within the SLEAP GUI. One direction would be to extend the 3D pipeline of SLEAP to include camera calibration. So, instead of expecting a calibration file as the starting point, the user could generate the calibration file from within SLEAP.

Add more functionality to sleap-anipose

Building on top of this idea of extending functionality, it would also be nice to take away some limiting factors such as requiring videos to be synchronized. While this requirement is fairly reasonable, providing a way to handle unsynchronized videos would make multi-view SLEAP easier to use in less constrained settings. One way to do this would be to permit manual or semi-automated resynchronization through the interface.

Faster multi-view association

More testing needs to be done on what assumptions can be made to lower the cost of automatic grouping across views. One idea is to allow users to manually set or verify a grouping for a unique instance across all views as a "hard assignment", then perform hypothesis testing by only permuting unassigned instances into groups. Building off this idea of hard assignments, SLEAP could perform a greedy version of hypothesis testing as follows. Imagine a user moves to a completely unlabeled temporal frame. The user then labels all instances for a single camera view before moving to the next view. As soon as the user labels an instance in the second view, a round of hypothesis testing will take place in which the new instance will be placed into an instance grouping. However, in the greedy version of hypothesis testing, whichever group the instance is assigned to will be treated as a hard assignment, unable to be changed. This would effectively lower the number of hypotheses generated down to "the number of existing instance groups plus one" (the plus one is for the case when the new instance is not in any existing instance group).

Conclusions

In this paper, we have provided an overview of multi-animal 3D pose estimation methodologies and extended SLEAP to allow for rapid annotations of multiple 2D views using 3D reprojection. Having more camera views requires more annotations, these issues can be ameliorated by bootstrapping annotations from other camera views. This can be achieved by properly calibrating the cameras to get intrinsic, extrinsic and distortion parameters which allow for the projection from each 2D view to a 3D world view and back. These reprojections from 3D to 2D can also be useful for tracking identities for a small number of animals but becomes more challenging when the number of animals increases. It is common for many labs to examine dyadic and triadic interactions, so despite the limitations, this tool should still be useful for many experimental contexts. Integrating aspects of the Anipose package into the SLEAP GUI will lead to an improved user experience for annotating multi-view multi-animal data. We expect these changes to the GUI to be integrated in the near future and expect it to have a significant impact on the labeling process for 3D multi-animal pipelines.

Ethical Statement

The acquisition of video for experiments depicted in Figure 2 and Figure 3 was approved by Princeton's IACUC. Methods consisted of placing mice into a wedge cage for 5-60 minute intervals and observing behavior.

References

1. 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., McKenzie-Smith, G.C., Mitelut, C.C., Castro, M.D., D'Uva, J., Kislin, M., Sanes, D.H., Kocher, S.D., S-H, S., Falkner, A.L., Shaevitz, J.W., Murthy, M. (2022). Sleap: A deep learning system for multi-animal pose tracking. *Nature Methods* **19**(4).

2. Marshall, J. D., Li, T., Wu, J. H., & Dunn, T. W. (2022). Leaving flatland: Advances in 3D behavioral measurement. Current opinion in neurobiology, 73, 102522.

3. Marshall, J. D., Klibaite, U., Gellis, A., Aldarondo, D. E., Ölveczky, B. P., & Dunn, T. W. (2021). The PAIR-R24M Dataset for Multi-animal 3D Pose Estimation. *Proceedings of the 35th Neural Information Processing Systems Conference* (NeurIPS 2021).

4. Marks, M., Qiuhan, J., Sturman, O., von Ziegler, L., Kollmorgen, S., von der Behrens, W., Mante, V., Bohacek, J., & Yanik, M. F. (2022). Deep-learning based identification, tracking, pose estimation, and behavior classification of interacting primates and mice in complex environments. Nature machine intelligence, 4(4), 331–340.

5. An, L., Ren, J., Yu, T., Hai, T., Jia, Y., & Liu, Y. (2023). Three-dimensional surface motion capture of multiple freely moving pigs using MAMMAL. Nature Communications, 14(1), 7727.

6. Long, C., Ai, H., Chen, R., Zhuang, Z., & Liu, S. (2020). Cross-view tracking for multi-human 3D pose estimation at over 100 fps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3279-3288).

7. Kevin Luxem, Jennifer J Sun, Sean P Bradley, Keerthi Krishnan, Eric Yttri, Jan Zimmermann, Talmo D Pereira, Mark Laubach (2023) Open-source tools for behavioral video analysis: Setup, methods, and best practices eLife 12:e79305.

8. Afshar, S., Pereira, T.D. (2023). sleap-anipose: SLEAP to Anipose triangulation pipeline for 3D multi-animal pose tracking [Computer software]. <https://github.com/talmolab/sleap-anipose>. Accessed 19 December 2023.

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