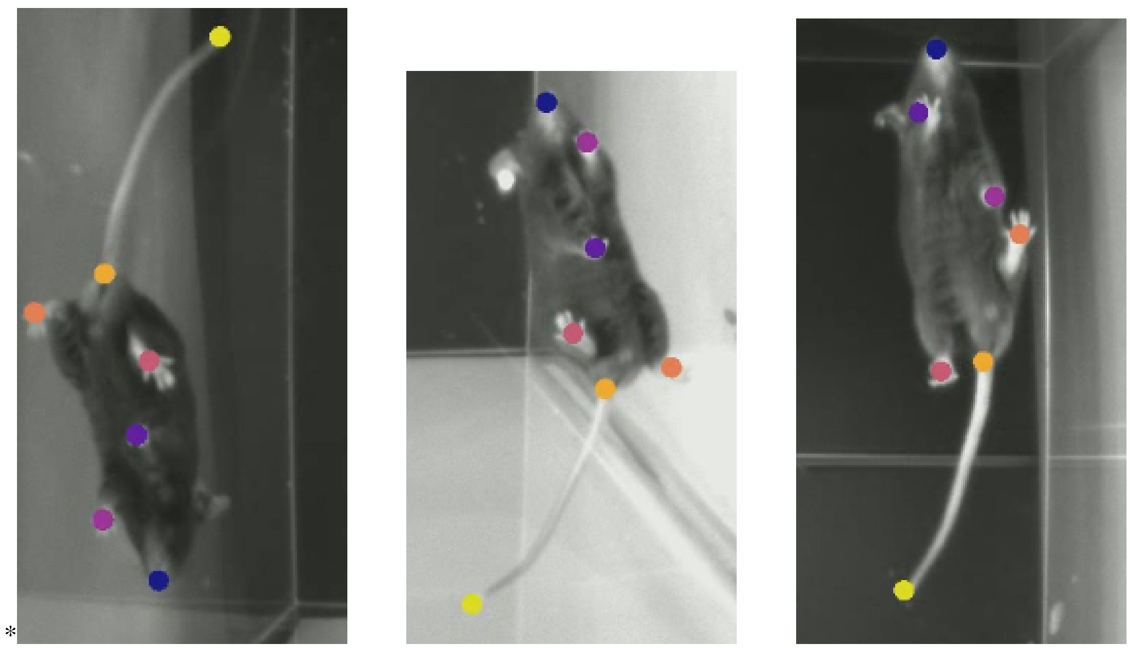
Gait Outlier Detection Methodology

Ameet Rahane
Feinberg School of Medicine
Northwestern University
ameet.rahane@northwestern.edu

1. Methods

1.1. DeepLabCut Paw Prediction



(a) (b) (c)

Figure 1: Sample Deeplabcut predictions

In order to quantitatively evaluate motion from video streams, we trained a DeepLabCut model [2]. DeepLabCut is a python package, built to do markerless pose estimation of user-defined body parts using a deep learning network. It allows the user to choose between a variety of deep learning infrastructures, including ResNet 50 [1], ResNet 101 [1], and a multitude of other topologies. For all of the analysis done in this work, we trained a ResNet101 network.

The network was initialized with weights trained on Imagenet, giving us the benefits of transfer training on an Imagenet based network, namely out of domain robustness for pose-estimation [3]. This allowed us to use the same trained network to predict all videos, across cohorts and genotypes, with little to no loss in accuracy.

Finally, we used this trained network to predict the snout, each paw, tail base, and tail tip for every video taken, which we proceeded to analyze for kinematics, as seen in Figure 1.

1.2. Manual Curation of Steps

I manually went through predicted video and picked out good exemplar steps for analysis, where the mouse was walking in a straight line with continuous ambulation and labeling looked good.

1.3. Outlier Detection

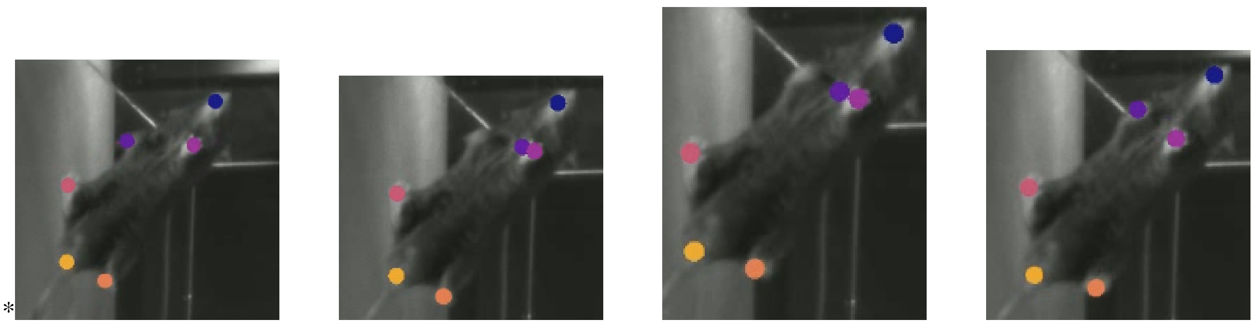
For this section, we'll walk through a single misprediction and the steps we take to fix this. Before we dive into the example, let's talk about the assumptions we're making.

1.3.1 Assumptions

We know that paw movements must be smooth and will very likely not contain more turns than a degree 3 polynomial. It also must be a smooth path. We also know that deep learning will sometimes make mispredictions.

1.3.2 Misprediction Example

To work on this, we'll be examining a severe misprediction from mouse 1058. From the example in Fig 2, it's clear that the left paw on the middle two frames are mispredicted.

*


(a) (b) (c) (d)

Figure 2: Four consecutive frames in a walking bout, where the middle two frames are clear mispredictions

To get a clearer idea of what's going on, let's plot the trajectory.

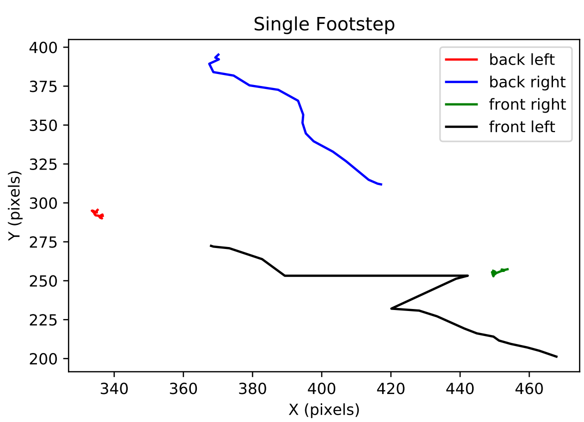


Figure 3: The trajectory of the same step that's shown in Figure 2. The back left and front right paws don't move, which makes sense as only the front left and the back right paws are moving here. It's also clear to clear to see where the back left paw is mispredicting for two frames. The back right path looks more correct.

1.3.3 Iterative Polynomial Sampling

How can we fix this? Going back to our assumptions, we know that the trajectory of a path must be smooth, continuous, and roughly look like a polynomial. Based on this, it would make sense to fit the trajectory to a polynomial. However, this reduces randomness and has the potential to cause issues with noise, which may cause issues down the line. To get around this, I propose an iterative sampling method that preserves randomness.

First, we fit two degree three polynomials: $p_x(t) : t \to x$ and $p_y(t) : t \to y$. That is, two maps from the time domain to x and y respectively.

Then, we find $p-$values of each data point as follows:

$$pvals = 2 * cdf(-abs(dist/stderr)) \tag{1}$$

where dist is the distance of each point from the fitted polynomial. This is the two tail test - the likelihood that each point actually lies on the polynomial.

Finally, we sample the data points using those p-values. Specifically, we replace the data point with it's polynomial counterpart according to a weighted coin flip, where the coin's weight is in accordance with the p-value. We repeat this process until the polynomial coefficients converge.

Great! Let's test this out now. From Figure 4, we see that this now looks much more correct, especially with the severe misprediction on the left paw.

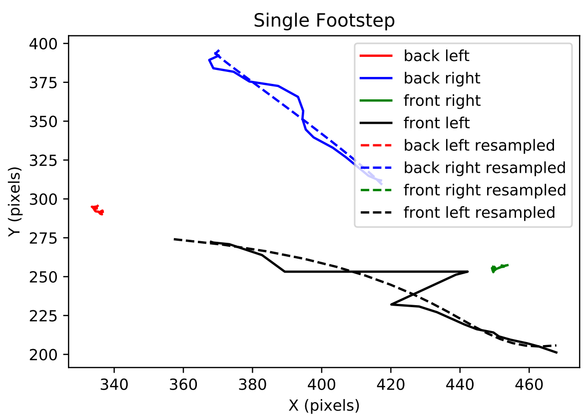


Figure 4: The trajectory of the same step as above, with fixed data overlaid. We see that this now looks much more correct, especially with the severe misprediction on the left paw.

Now, we can use this data for metrics.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, Aug. 2018.

[3] A. Mathis, M. Yüksekgönül, B. Rogers, M. Bethge, and M. W. Mathis. Pretraining boosts out-of-domain robustness for pose estimation, 2019.