DeepLabCut workshop
Session 1: Project management, labeling, & data set curation

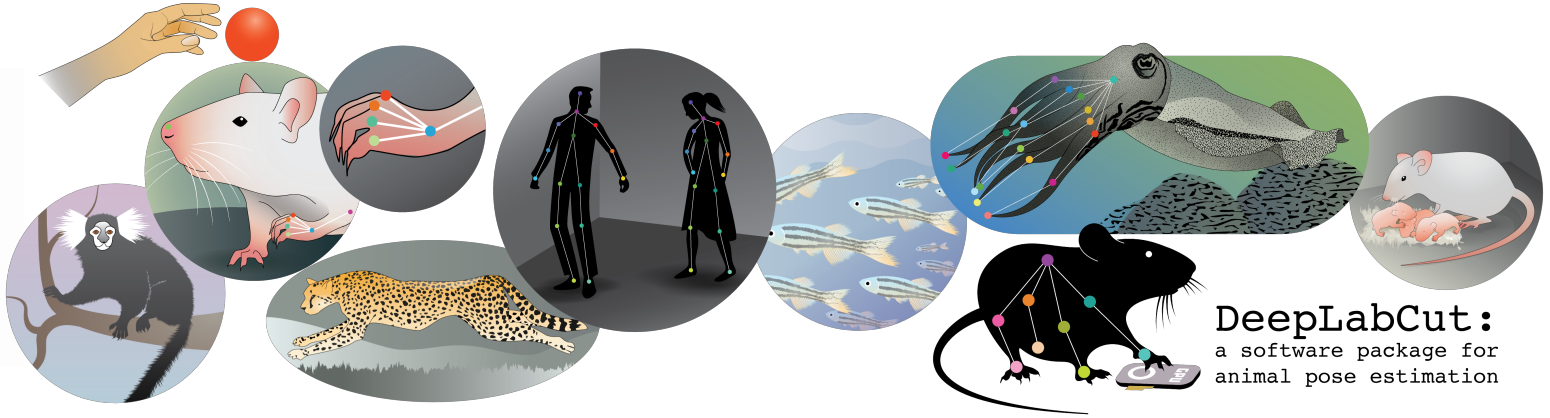
16th and 17th of January 2020

Rowland Institute, Harvard University

Alexander Mathis
Mackenzie Mathis
Jessy Lauer

Chan Zuckerberg Initiative

HARVARD UNIVERSITY



“Software 2.0” – integration of annotation, training and inference

Train DNN

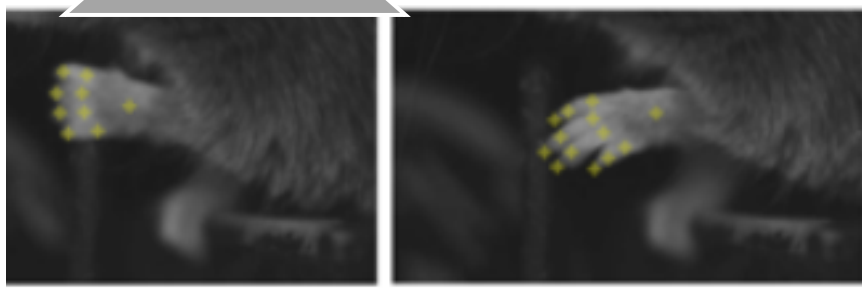
Create a project, extract frames, + GUIs to label your data

Select + Train your deep neural network

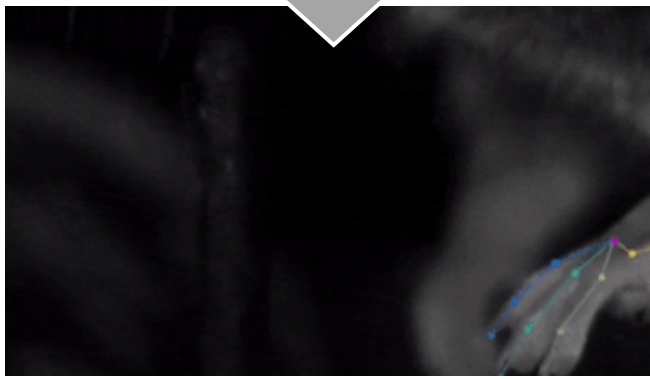
Evaluate network performance

(active learning + GUIs if improvement needed)

Run inference on new videos, create labeled videos, + plot your results!



refine?



Mathis et al. Nature Neurosci, 2018
Nath* & Mathis* et al. Nature Protocols 2019

Step-by-step user guide

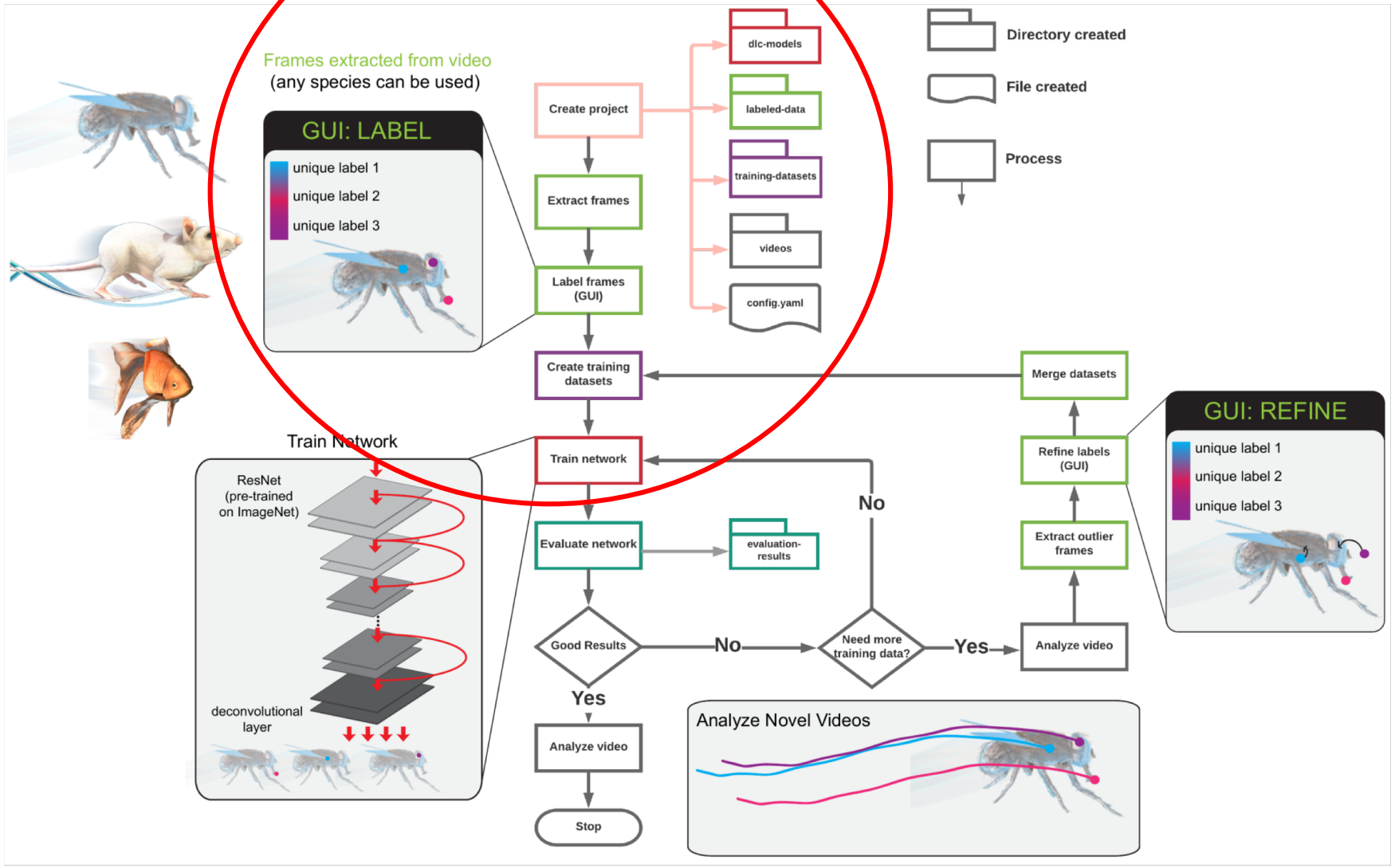
NATURE PROTOCOLS PROTOCOL

Table 1 | Summary of commands

Operation	Command
Open IPython and import DeepLabCut (Step 1)	`ipython` `import deeplabcut`
Create a new project (Step 2)	`deeplabcut.create_new_project('project_name','experimenter',['path of video 1','path of video2',...])`
Set a `config_path` variable for ease of use (Step 3)	`config_path = '/yourdirectory/project_name/config.yaml'`
Extract frames (Step 4)	`deeplabcut.extract_frames(config_path)`
Label frames (Steps 5 and 6)	`deeplabcut.label_frames(config_path)`
Check labels (optional)(Step 7)	`deeplabcut.check_labels(config_path)`
Create training dataset (Step 8)	`deeplabcut.create_training_dataset(config_path)`
Train the network (Step 9)	`deeplabcut.train_network(config_path)`
Evaluate the trained network (Step 11)	`deeplabcut.evaluate_network(config_path)`
Video analysis and plotting results (Step 11)	`deeplabcut.analyze_videos(config_path,['path of video 1 or folder','path of video2',…])`
Video analysis and plotting results (Step 12)	`deeplabcut.plot_trajectories(config_path,['path of video 1','path of video2',...])`
Video analysis and plotting results (Step 13)	`deeplabcut.create_labeled_video(config_path,['path of video 1','path of video2',...])`
Refinement: extract outlier frames (Step 14)	`deeplabcut.extract_outlier_frames(config_path,['path of video 1','path of video 2'])`
Refine labels (Step 15)	`deeplabcut.refine_labels(config_path)`
Combine datasets (Step 16)	`deeplabcut.merge_datasets(config_path)`

Nath*, Mathis* et al. Nature Protocols (>8k downloads since June) and preprint on BioRxiv (>15k downloads since Nov 2018)

DeepLabCut 2.0 workflow



Nath*, Mathis* Nature Protocols, 2019

2D Project Folders

▶ training data,
trained networks,
`config.yaml` file

This is the master folder created when you create a project (Step 1)

New (2D) videos for analysis

▶ batch process videos
▶ analyzed data

you can place new videos here, and then run:

```
deeplabcut.analyze_videos(config_path,
    folderpath, videotype='.mp4')
```

Your 2D or 3D project “entry point” is through the `config.yaml` file
When you want to work on your project:

```
activate DLCenvName
ipython
import deeplabcut
config_path ='/home/yourprojectfolder/config.yaml'
```

3D Project Folder

▶ 3D `config.yaml`
tools for calibration,
camera corrections

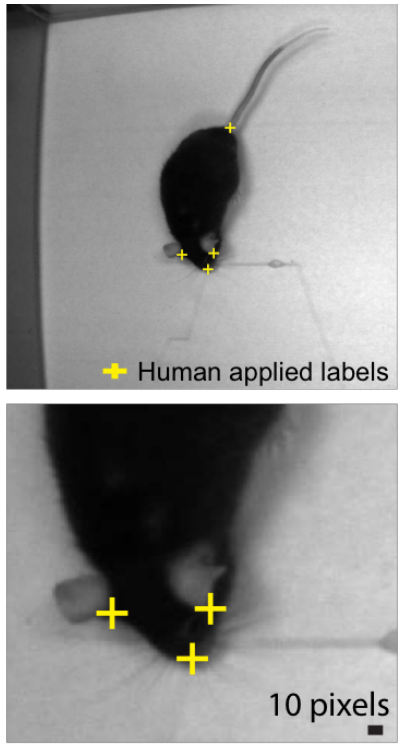
This is the 3D master folder created when you create a 3D project (Step 1)

▶ batch process 3D video analysis:

```
deeplabcut.triangulate(
    config_path3d,video_folder)
```

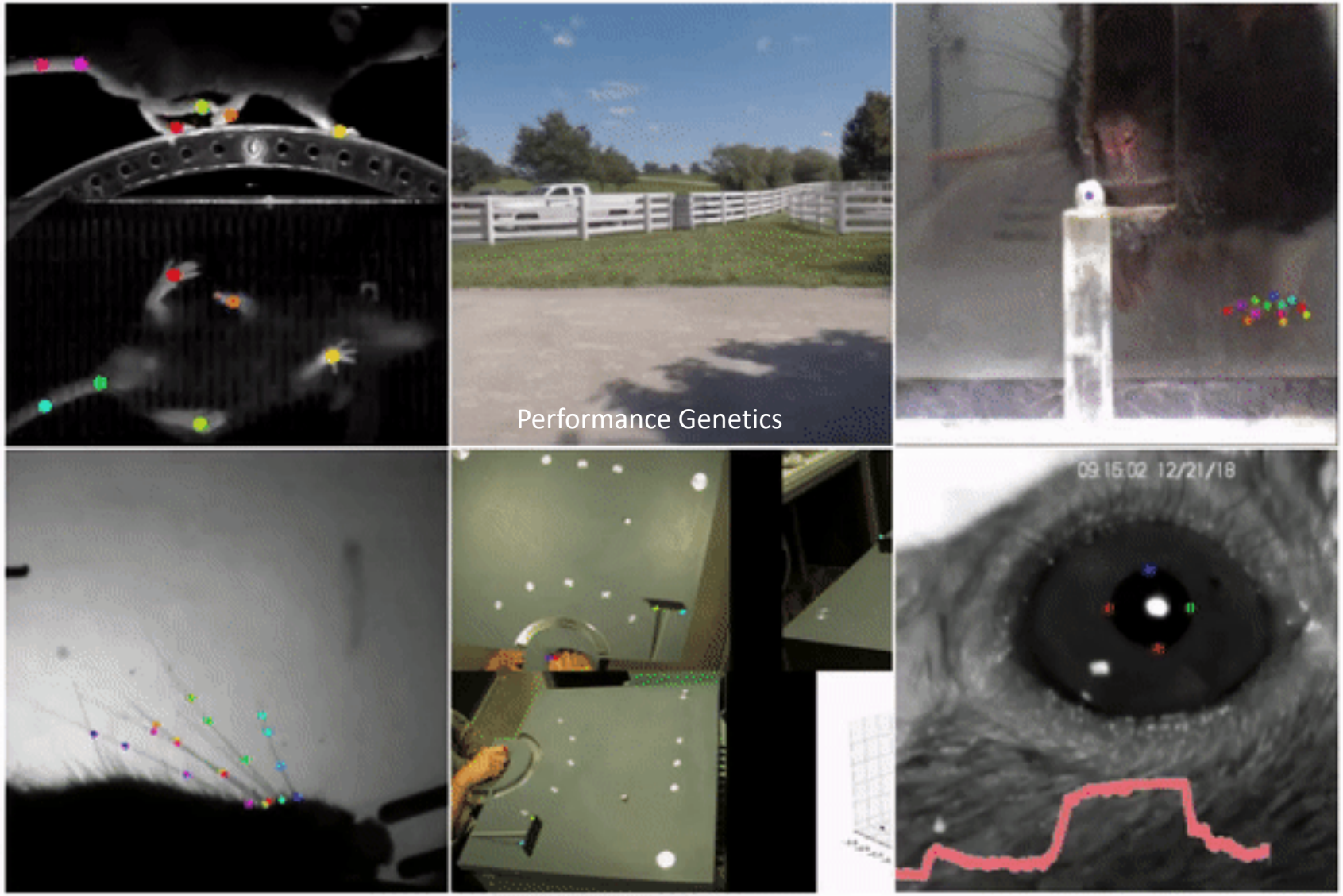
What bodyparts can be detected?

- Anything you can reliably label, you can use. Does not need to be a moving object (can be corners of box, LED, etc).



Locomotion
3D tracking of limbs

R. Warren (Sawtell lab)
Columbia University



Pellet reaching task

D. Leventhal Lab
University of Michigan

Whisker tracking

A Erskine (Hires lab)
USC

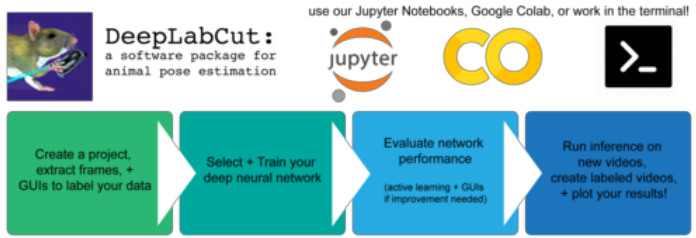
Pupil tracking

T. Vaissiere
Scripps

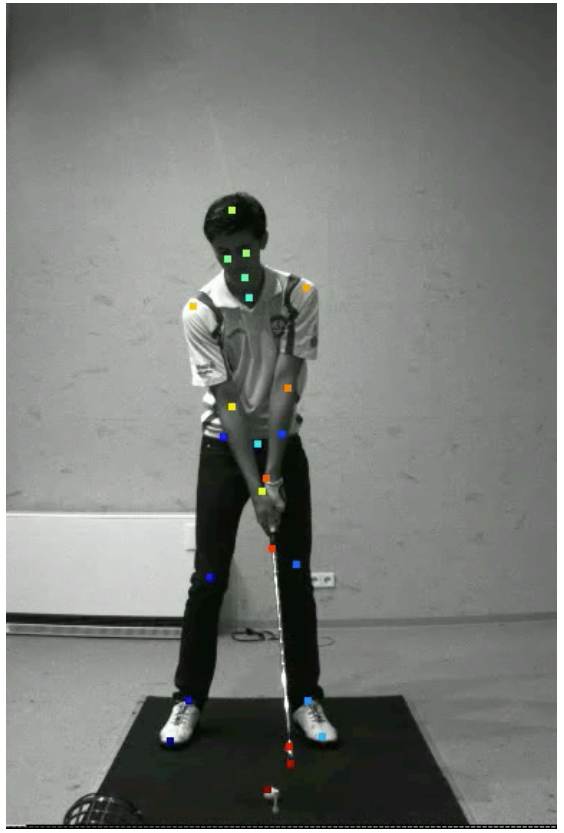
J Bonaiuto (Ferrari lab) CNRS

DeepLabCut – Deep Learning toolbox for pose-estimation

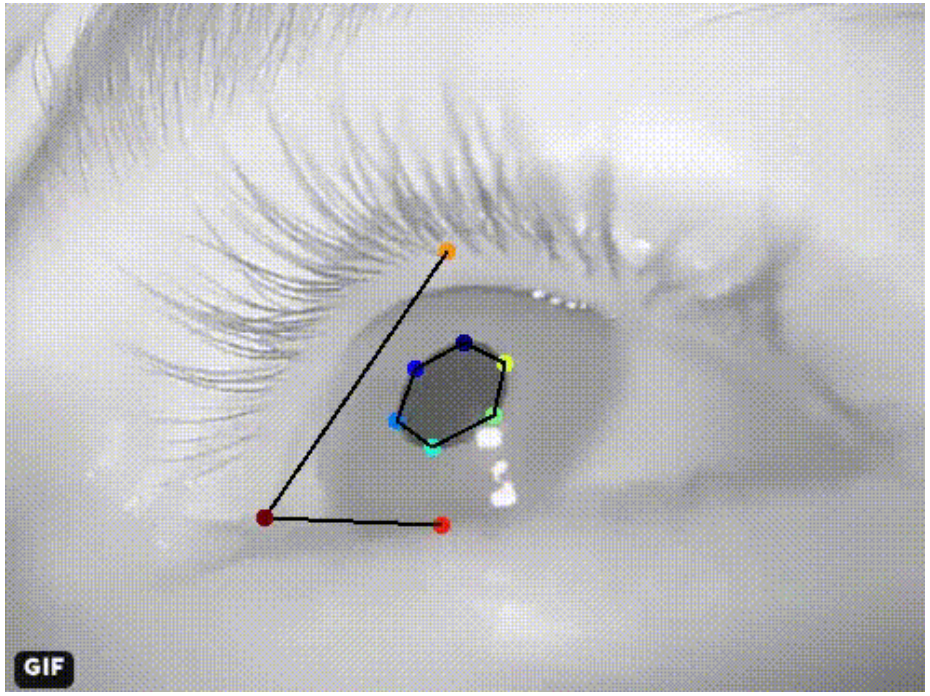
Open source & free, > 500 labs, >30,000 installations



Cronin et al, J Biomechanics 2019



Sebastian Riegelbauer, TU Munich



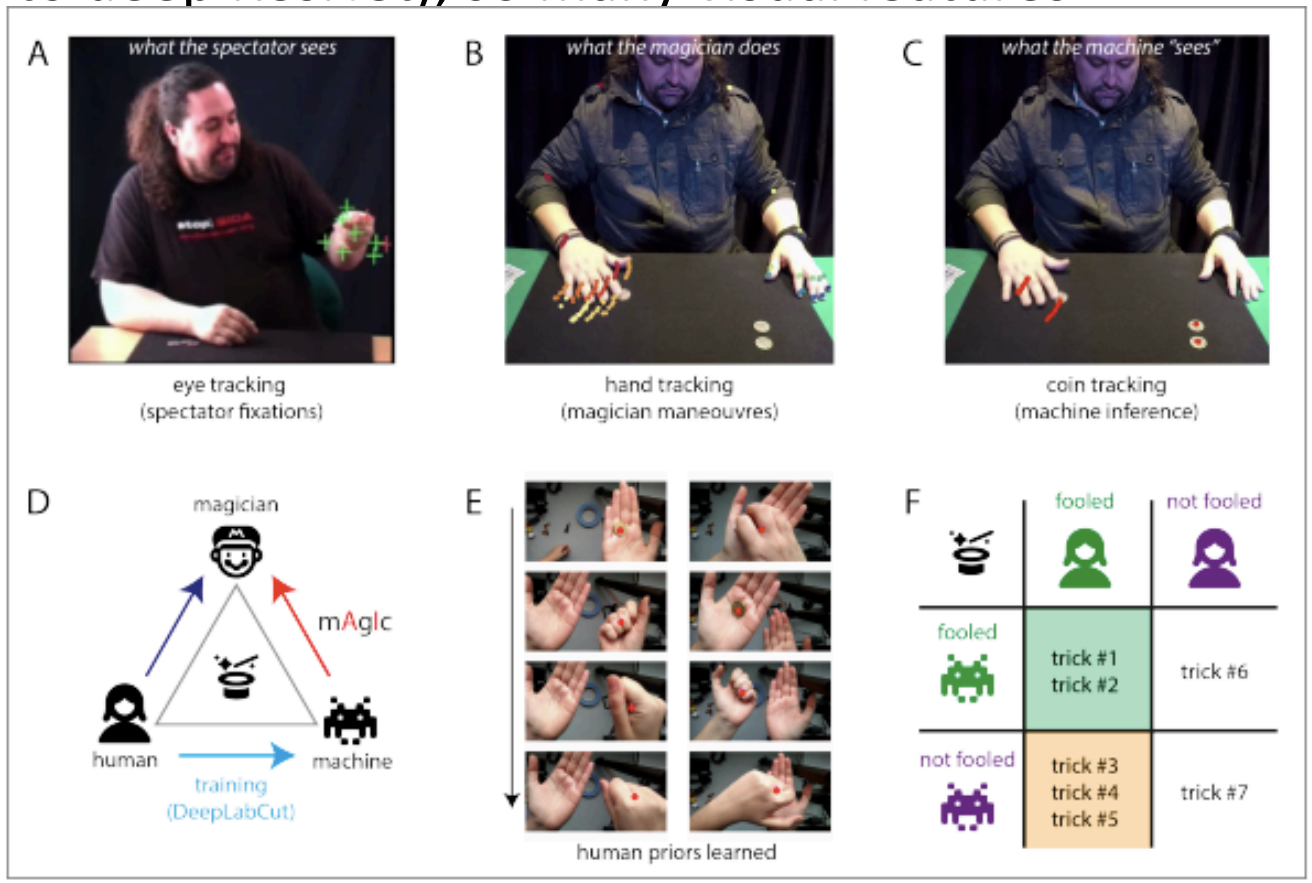
18 labeled frames; Data: Ehinger et al. PeerJ 2019

J. Saunders, U of Oregon

What bodyparts can be detected?

DLC has very large receptive fields (due to deep ResNet), so many visual features + geometric layout can be considered:

- Salient points (e.g. snout/tail base)
- Texture of objects
- Can teach the network to guess
- Geometric relationship
  - (e.g. left/right ear; whiskers!)

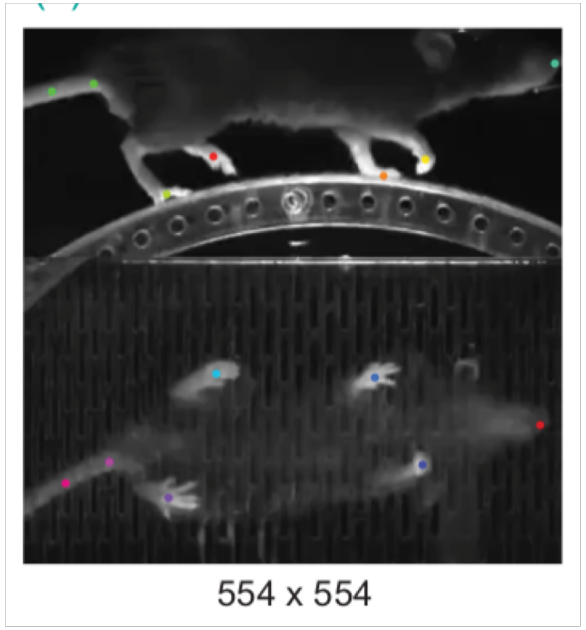


Playing magic tricks to deep neural networks untangles human deception
by Zaghi-Lara et al. arxiv 2019

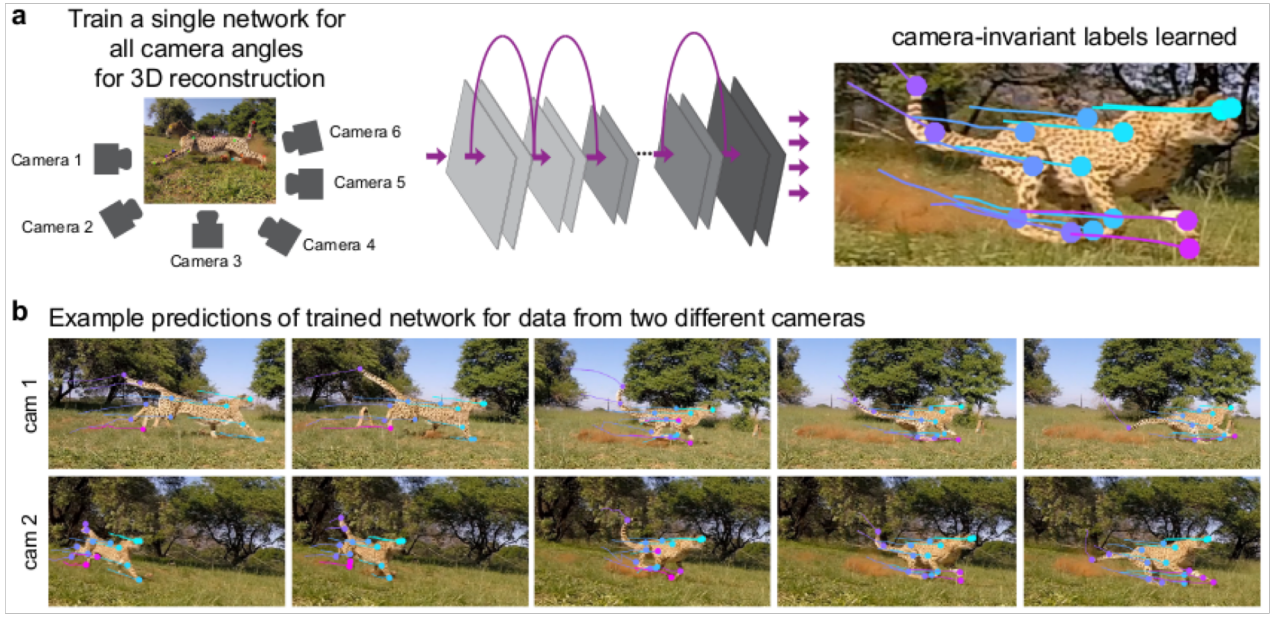
How should a training set be created?

- Reflect **variability of experiments**

Labels can be **from/in** different views (cameras)



Mathis & Warren 2018



Nath*, Mathis* et al. 2019

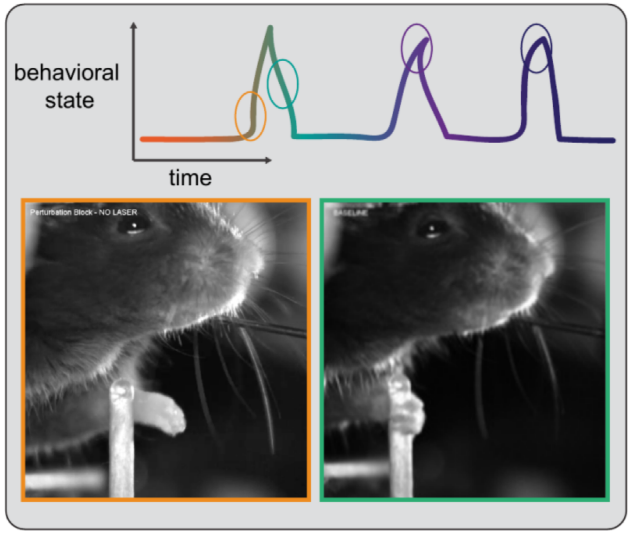
Frame extraction

Select videos to grab frames:

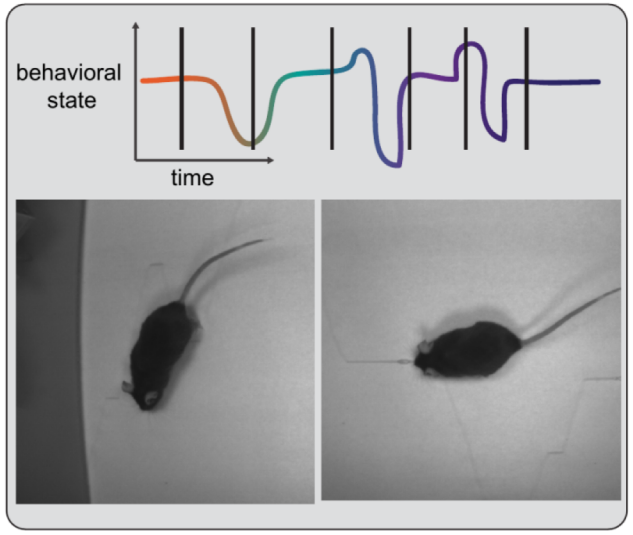
Use videos with image from:
- different sessions reflecting (if the case) varying light conditions, backgrounds, setups, and camera angles (etc).
- different individuals, especially if they look different (i.e. brown + black mice)

3 methods for frame extraction to create a labeled train/test set

Image based clustering (k-means)



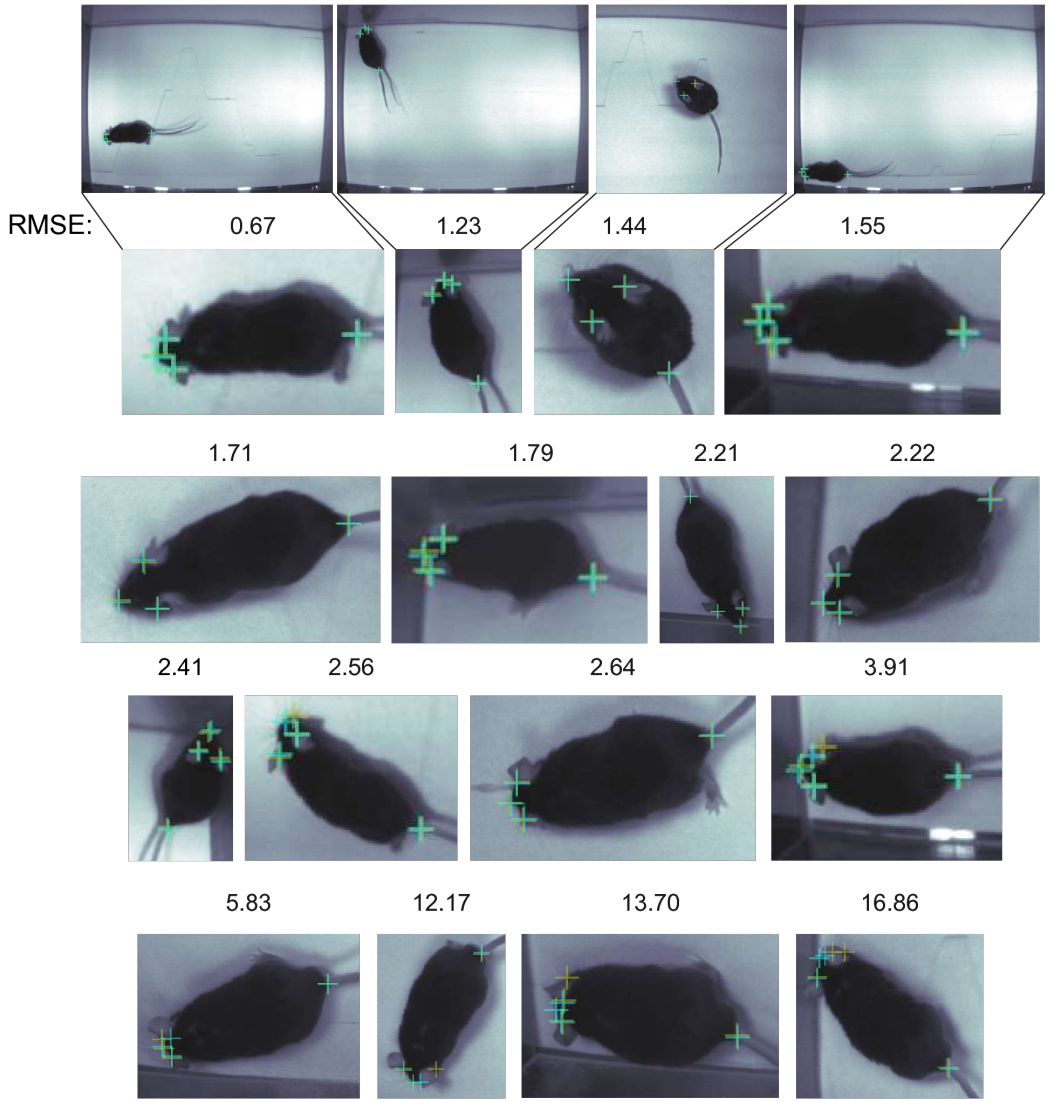
Random temporal sampling (uniform)

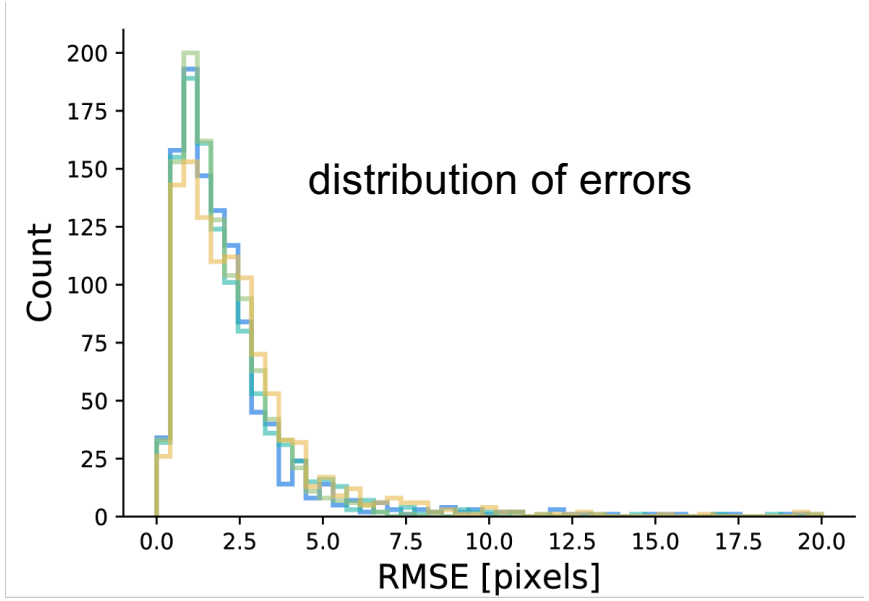


GUI for manual frame grabbing

Nath*, Mathis* Nature Protocols, 2019

Benchmarking: human accuracy



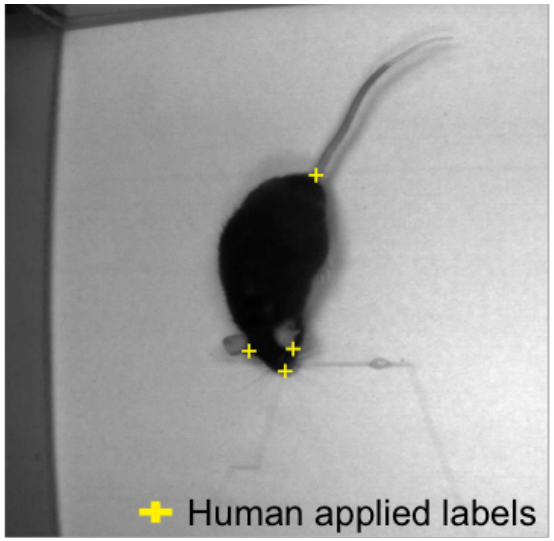


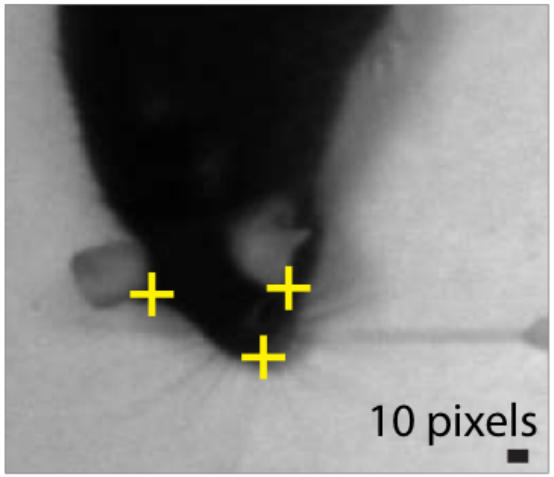
Mean: 2.7 pixels

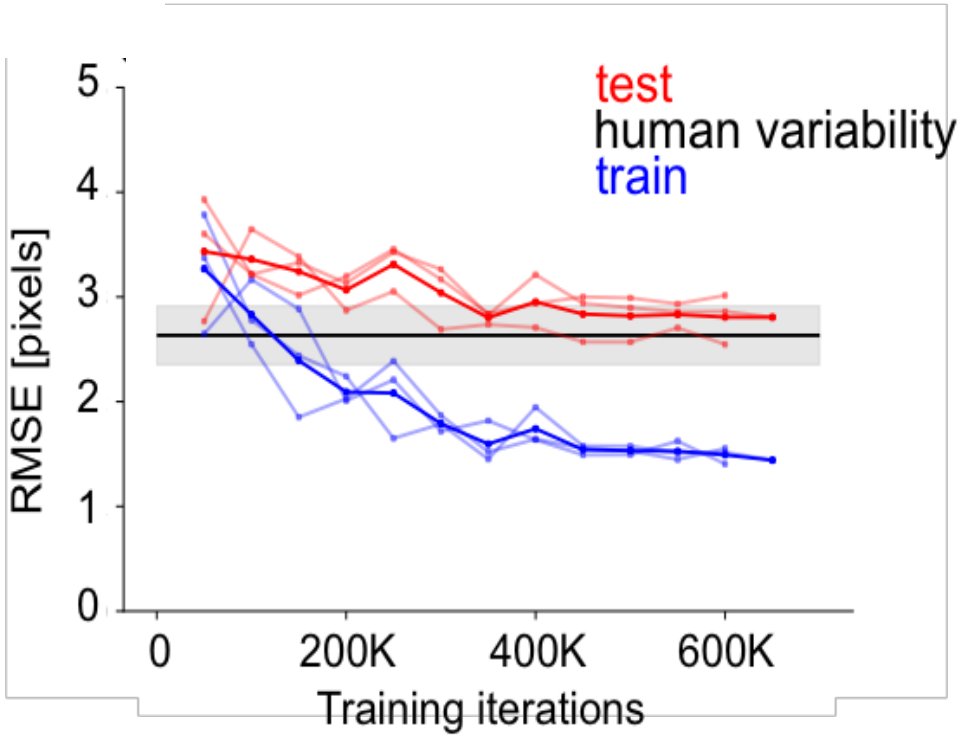
1,080 distinct frames, 8 different mice, 2 cameras (different resolution),
One labeler [one twice; >1 month apart] for 4 body parts

Mathis et al. Nature Neurosci, 2018
Data: Murthy lab (Harvard)

Human-level accuracy on test set

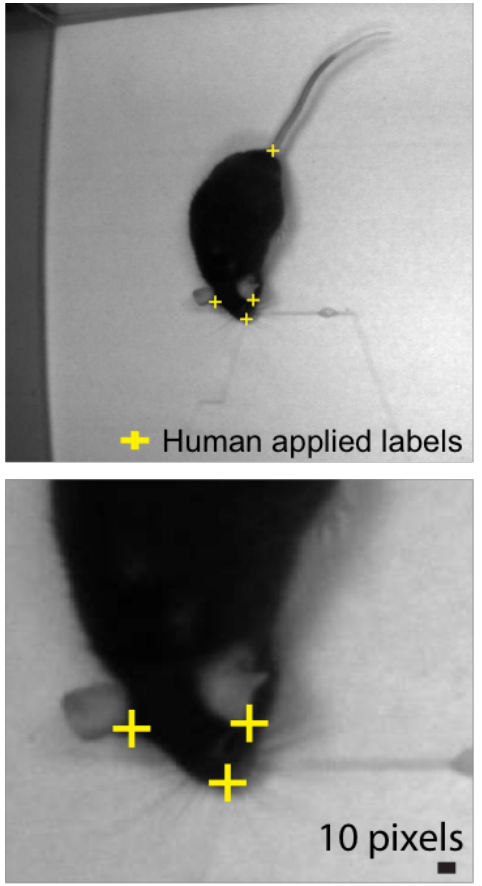




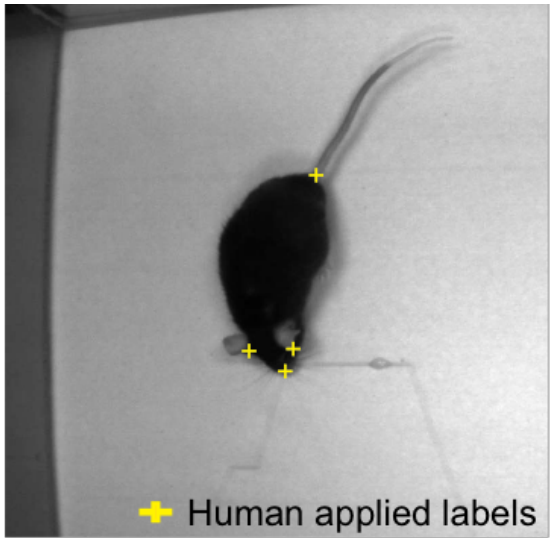


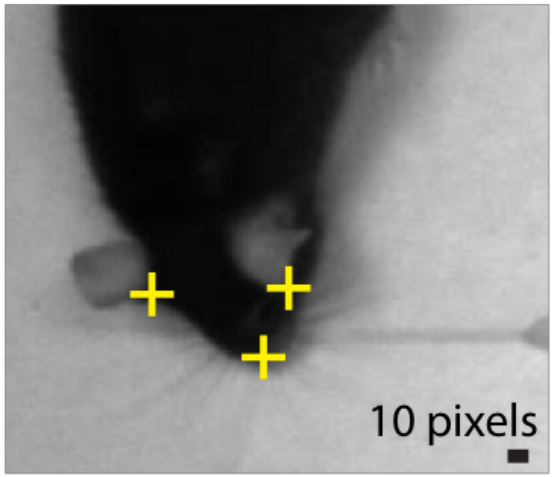
Mathis et al.
Nature Neurosci, 2018

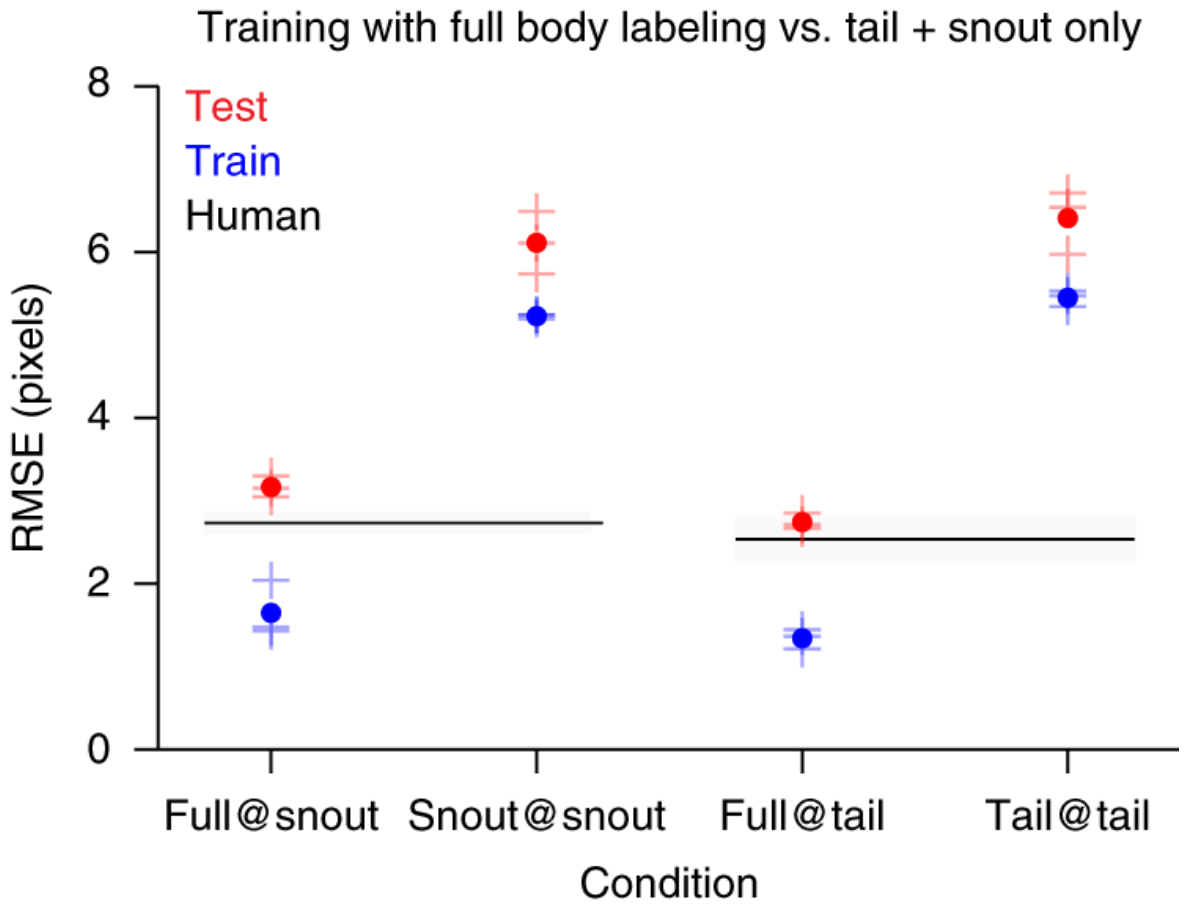
How little training data is required?



More labels are better







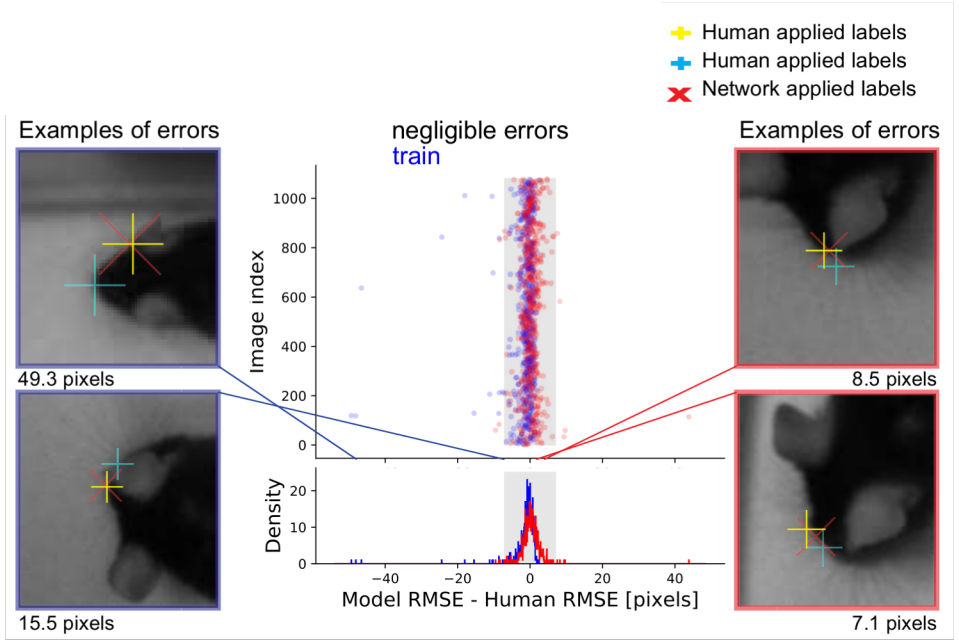
Mathis et al.
Nature Neurosci, 2018

Labeling GUI features

- Zooming, Pan, auto advance (some hot keys)
- Only visible features have to be labeled
- The label (names) are defined in the config file
- The meaning of features (*bodyparts, objects*) is provided by the user
- Additional (new) labels can be added later to a project

Labeling considerations

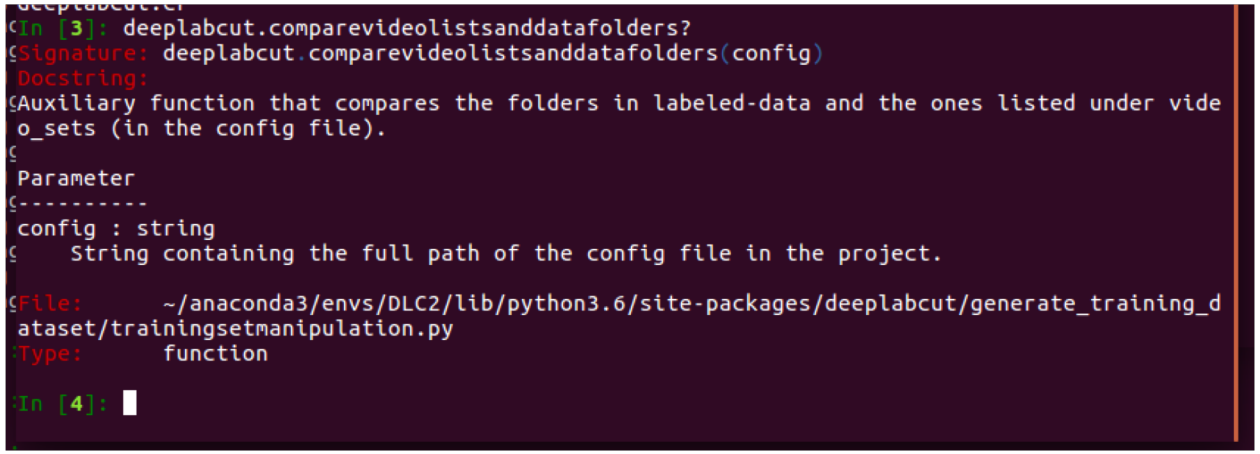
- Label (feature locations) **consistently**
- Errors are bad! (especially when only a few images are labeled)
- Left vs. right body parts can be if full animal visible (due to large rec. fields), but flipped labels are problematic



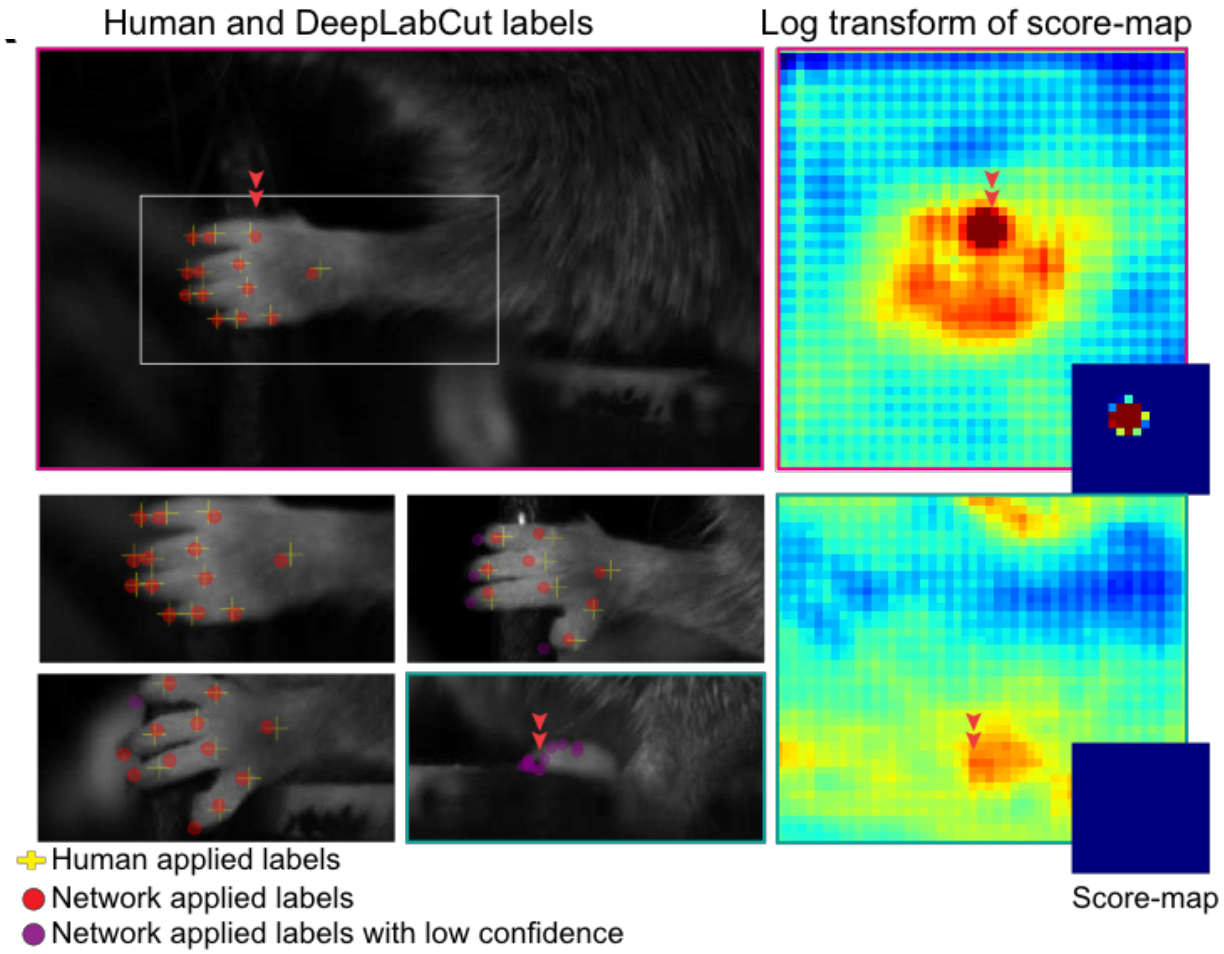
- You can check labels by plotting all the figures (always do this before training!)
- You can correct wrong labels manually with the labeling GUI!

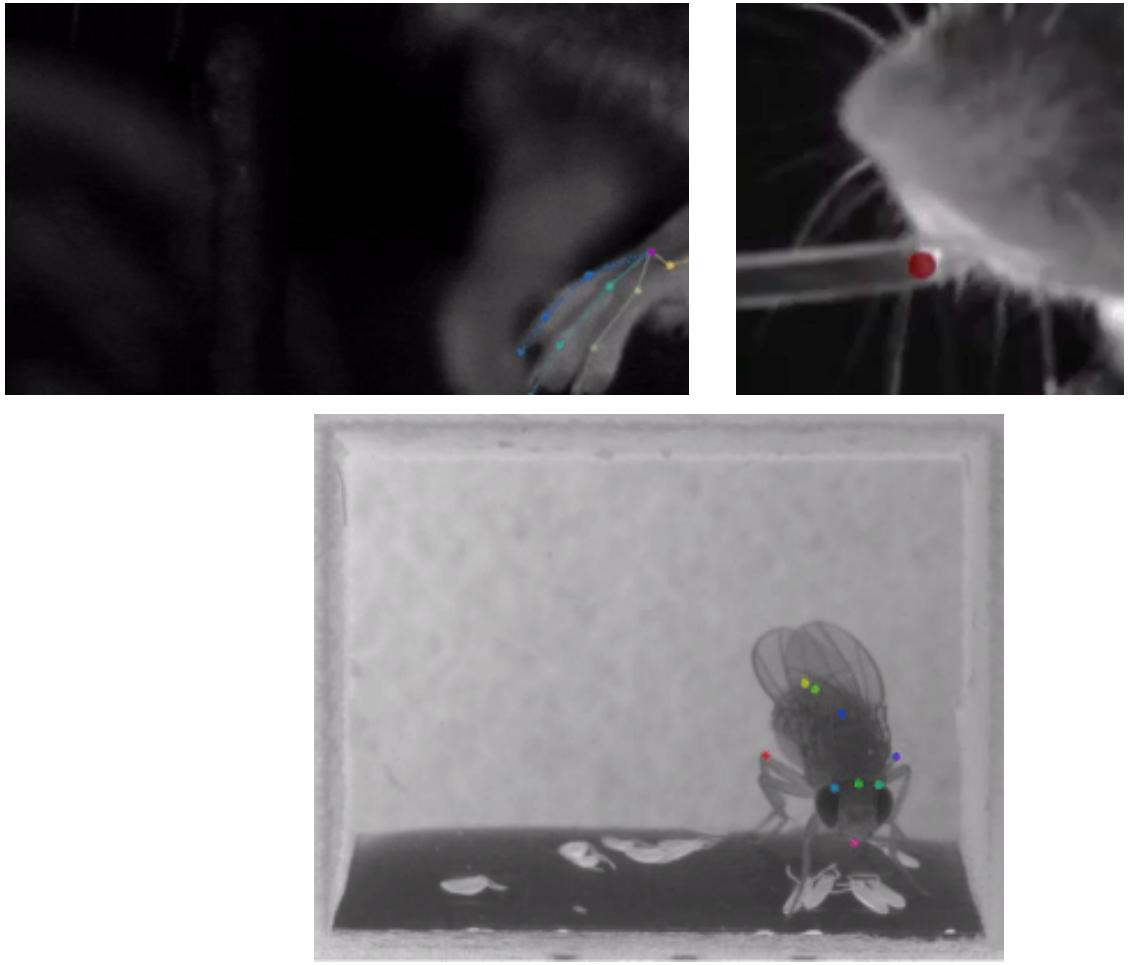
Auxiliary functions to manage your annotation datasets

- comparevideolistsanddatafolders
- datasetstovideolistandviceversa
- dropannotationfileentriesduetodeletedimages



Score-maps provide network confidence readout





Mathis et al, Nature Neuroscience 2018

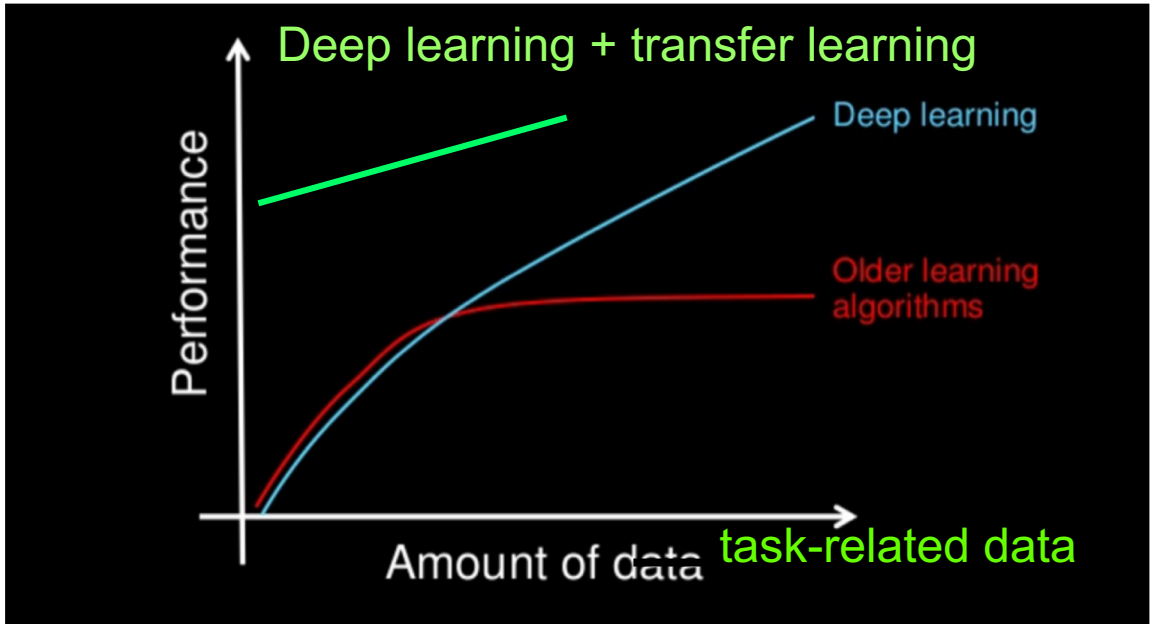
Transfer learning enables deep learning in many pose estimation settings

Pre-trained! (i.e. on ImageNet)

image → Predictor → pose

train

Only a few examples (10-200) for most applications



Adapted from Andrew Ng