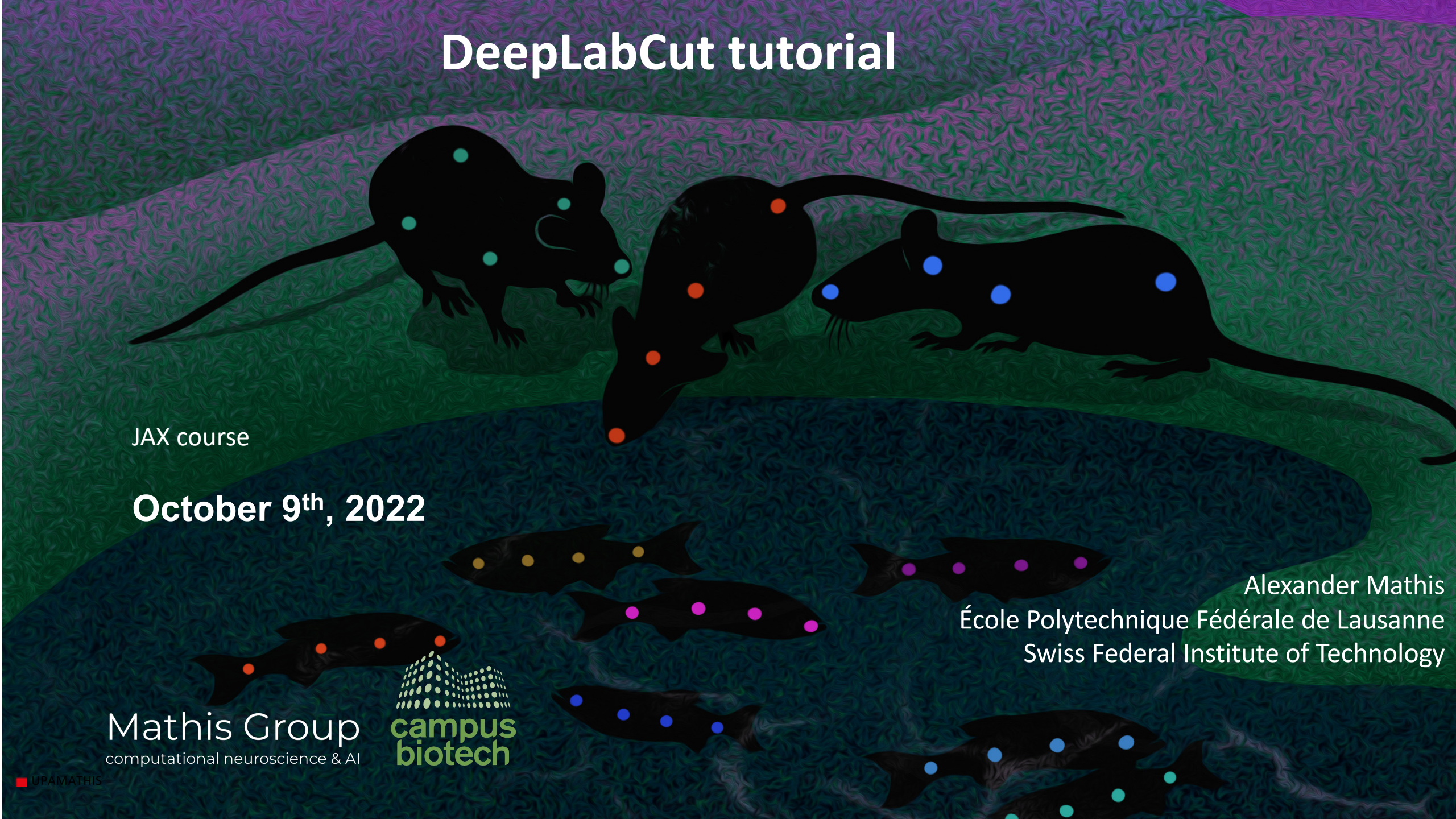
DeepLabCut tutorial



JAX course

October 9th, 2022

Alexander Mathis
École Polytechnique Fédérale de Lausanne
Swiss Federal Institute of Technology

Mathis Group
computational neuroscience & AI

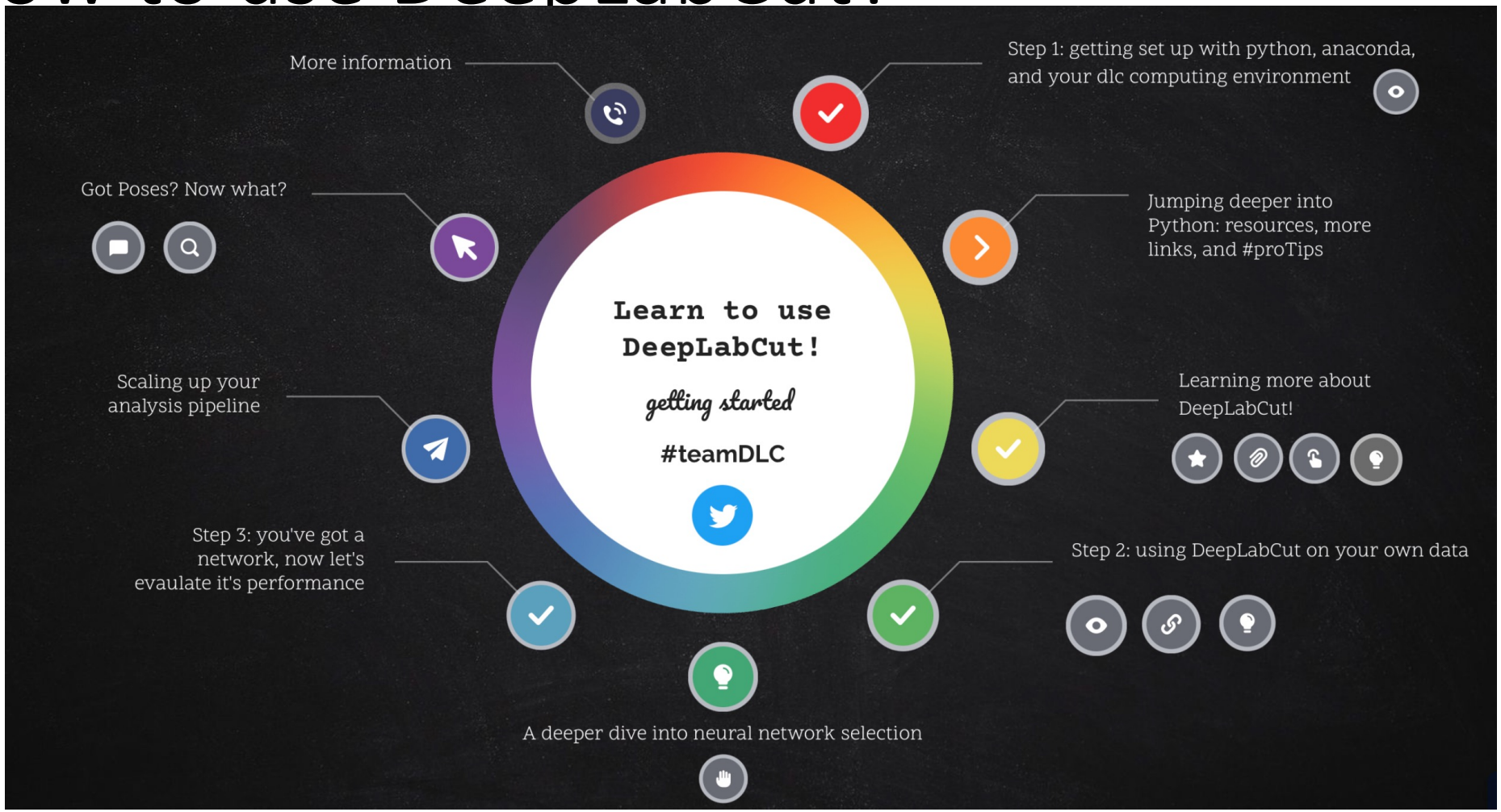
campus biotech

UPAMATHIS

Today’s sunrise at JAX.

Happy DeepLabCutting!

How to use DeepLabCut?



https://github.com/DeepLabCut/DeepLabCut-Workshop-Materials

DeepLabCut workflow

Train DNN

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

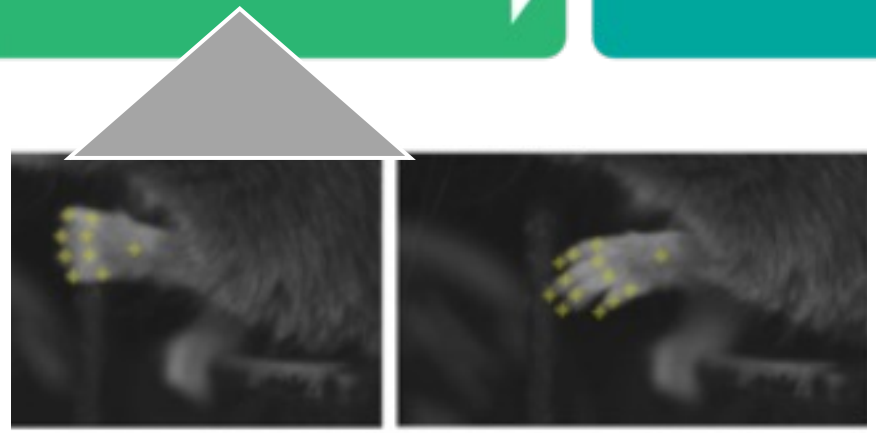
Select + Train your deep neural network

Evaluate network performance

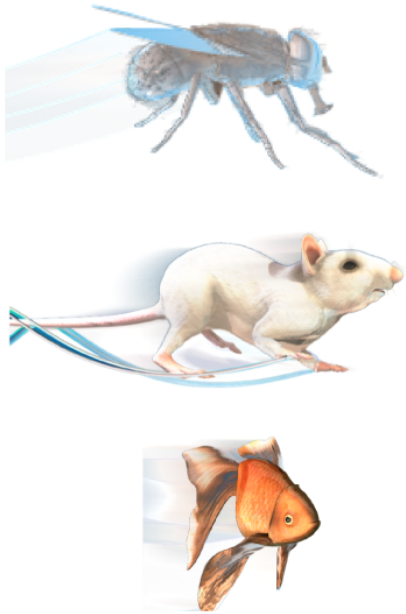
(active learning + GUIs if improvement needed)

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

refine?



Nath*, Mathis* et al. Nature Protocols



Steps 7 and 8

GUI: labeling

Unique label 1
Unique label 2
Unique label 3

Step 9: check labels!

Frames extracted from video
(any species can be used)

Steps 1–5

Create project

Step 6

Extract frames

Label frames (GUI)

dlc-models
Labeled-data
Training-datasets
Videos
Config.yaml

Directory created
File created
Process

Step 10

Create training datasets

Step 11

Train network

Train network

ResNet (pre-trained on ImageNet)

Deconvolutional layer

Step 12

Evaluate network

Evaluation-results

Good results

No

Need more training data?

Yes

No

Yes

Analyze video

Stop

Steps 13–15

Analyze novel videos

Merge datasets

Refine labels (GUI)

Extract outlier frames

Analyze video

Steps 16–18

GUI: refinement

Unique label 1
Unique label 2
Unique label 3

nature protocols

PROTOCOL

https://doi.org/10.1038/s41596-019-0176-0

Using DeepLabCut for 3D markerless pose estimation across species and behaviors

Tanmay Nath[1,5], Alexander Mathis[1,2,5], An Chi Chen[3], Amir Patel[3], Matthias Bethge[4] and Mackenzie Weygandt Mathis[1*]

We are working on un expanded protocol that also includes maDLC
But also see the online docs (which are already up-to-date) →
https://deeplabcut.github.io/DeepLabCut/README.html

Nath*, Mathis* et al. Nature Protocols

Project GUI demo!

We have a user-friendly GUI that allows you to carry out the whole workflow.

Note completely updated GUI forthcoming in DLC version 2.3; rc1 already out.

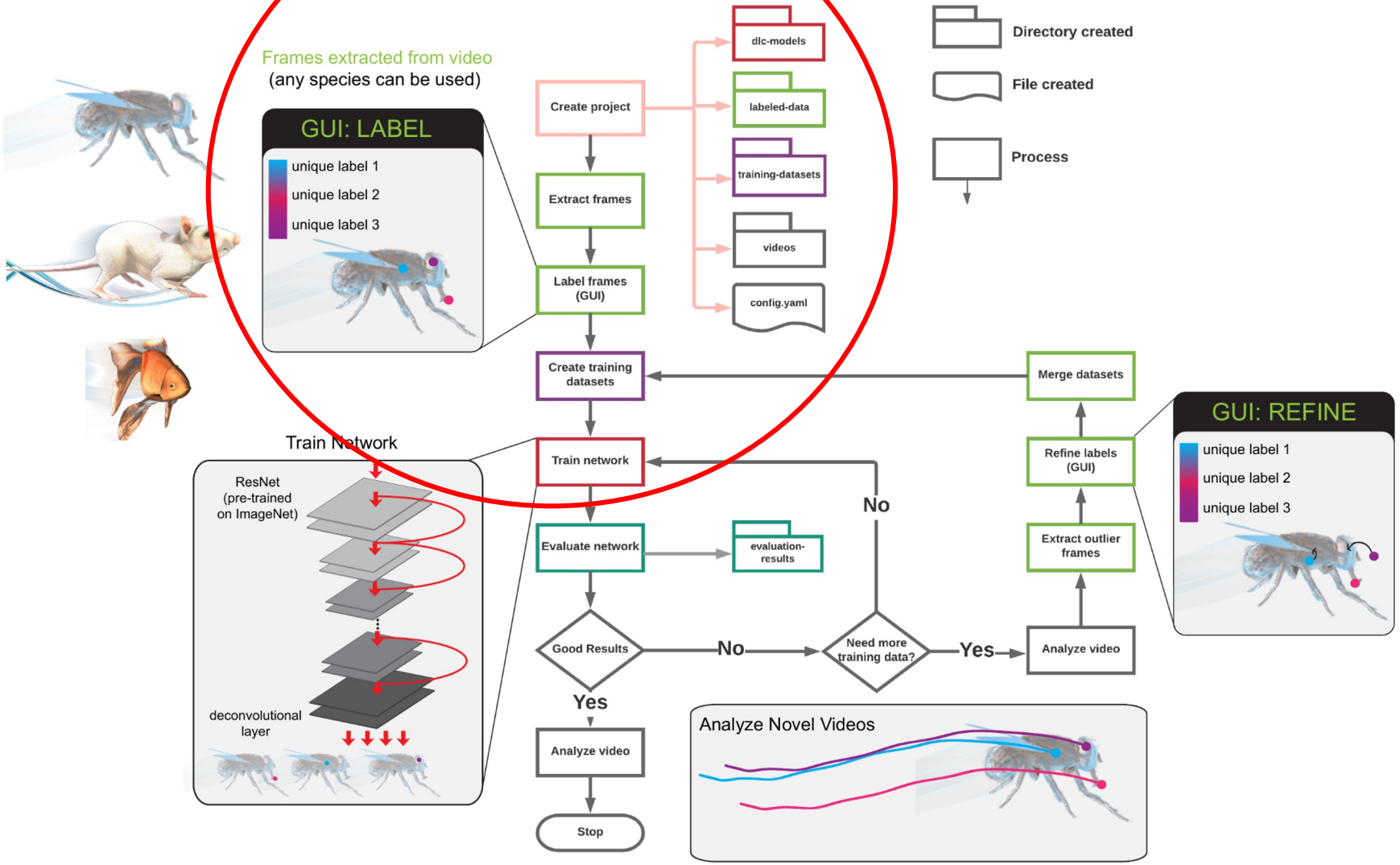
Nath*, Mathis* et al. Nature Protocols

Key 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',...])`

Nath*, Mathis* et al. Nature Protocols

DeepLabCut 2.0 workflow



Nath*, Mathis* Nature Methods 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'
```

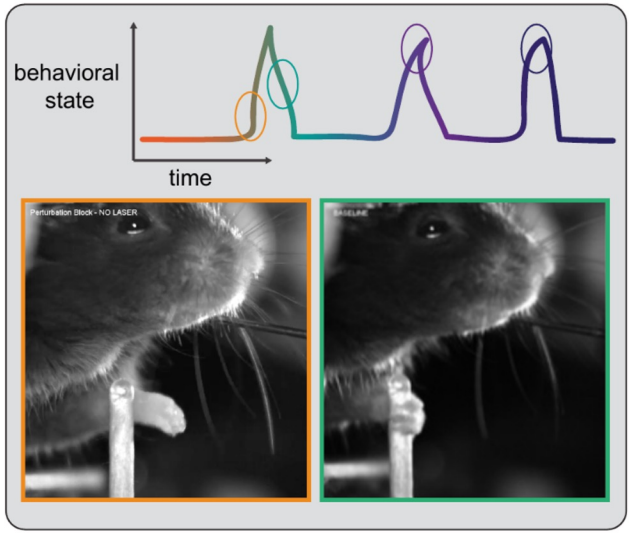
Frame extraction: deeplabcut.extract_frames(..)

Select videos to grab frames:

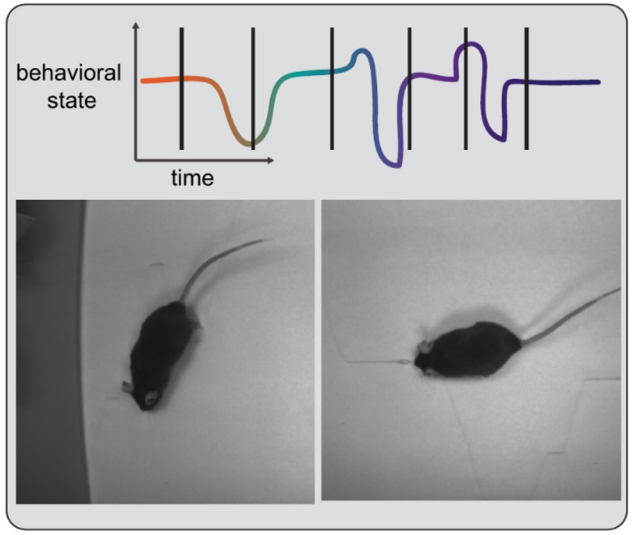
Use videos with image from:
- different sessions reflecting (if the case) varying light conditions, backgrounds, setups, and camera angles (etc).
- different individuals, especially if they look different (i.e. brown + black mice)

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

Image based clustering (k-means)



Random temporal sampling (uniform)



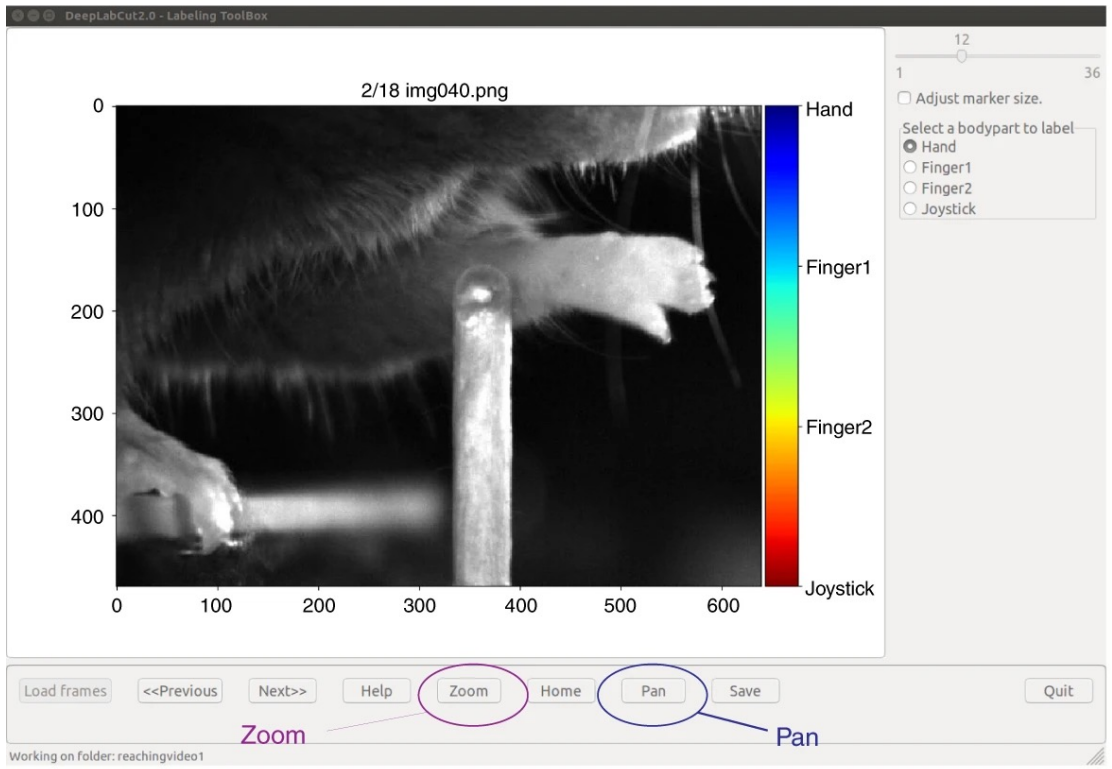
GUI for manual frame grabbing

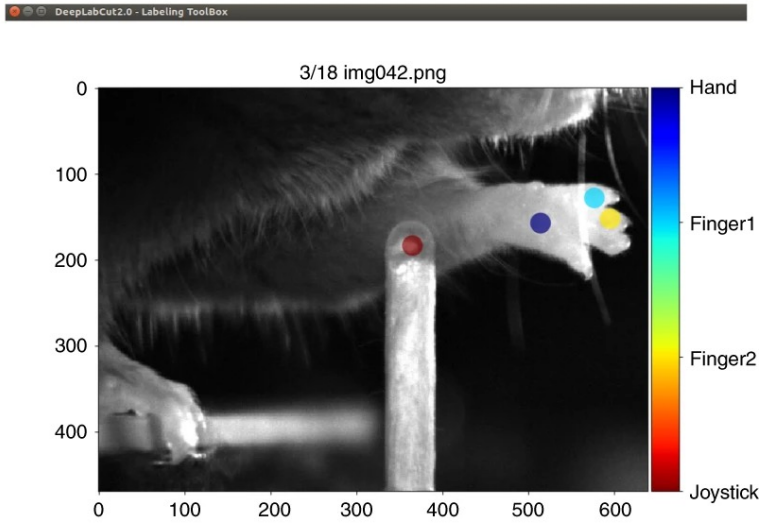
Nath*, Mathis* Nature Methods 2019

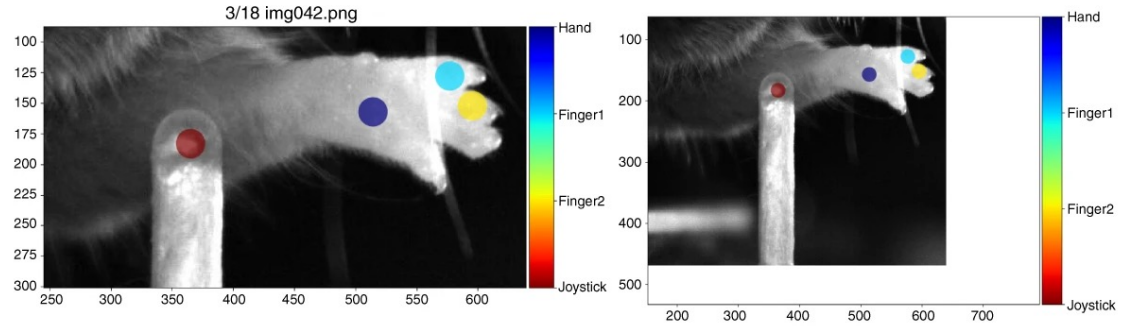
Label frames: deeplabcut.label_frames(..)

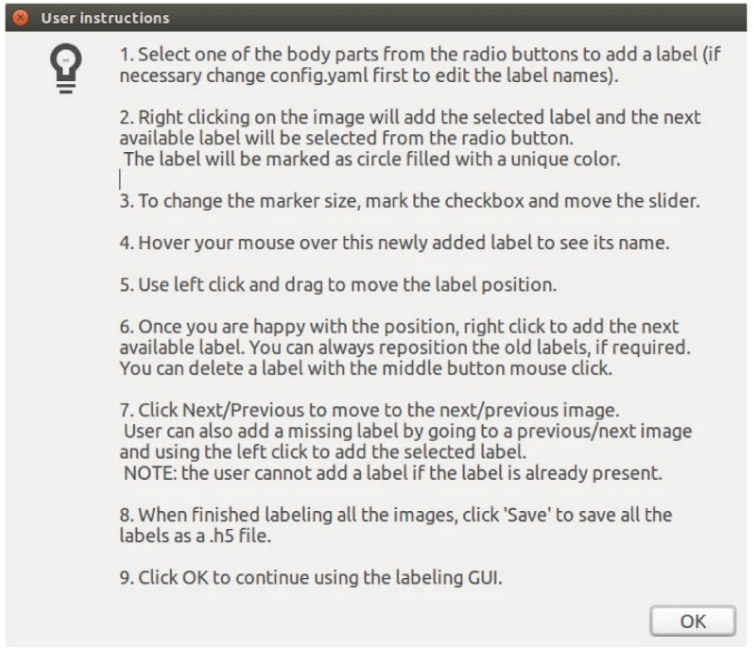
Label frames using the interactive GUI:

```
deeplabcut.label_frames(config_path)
```





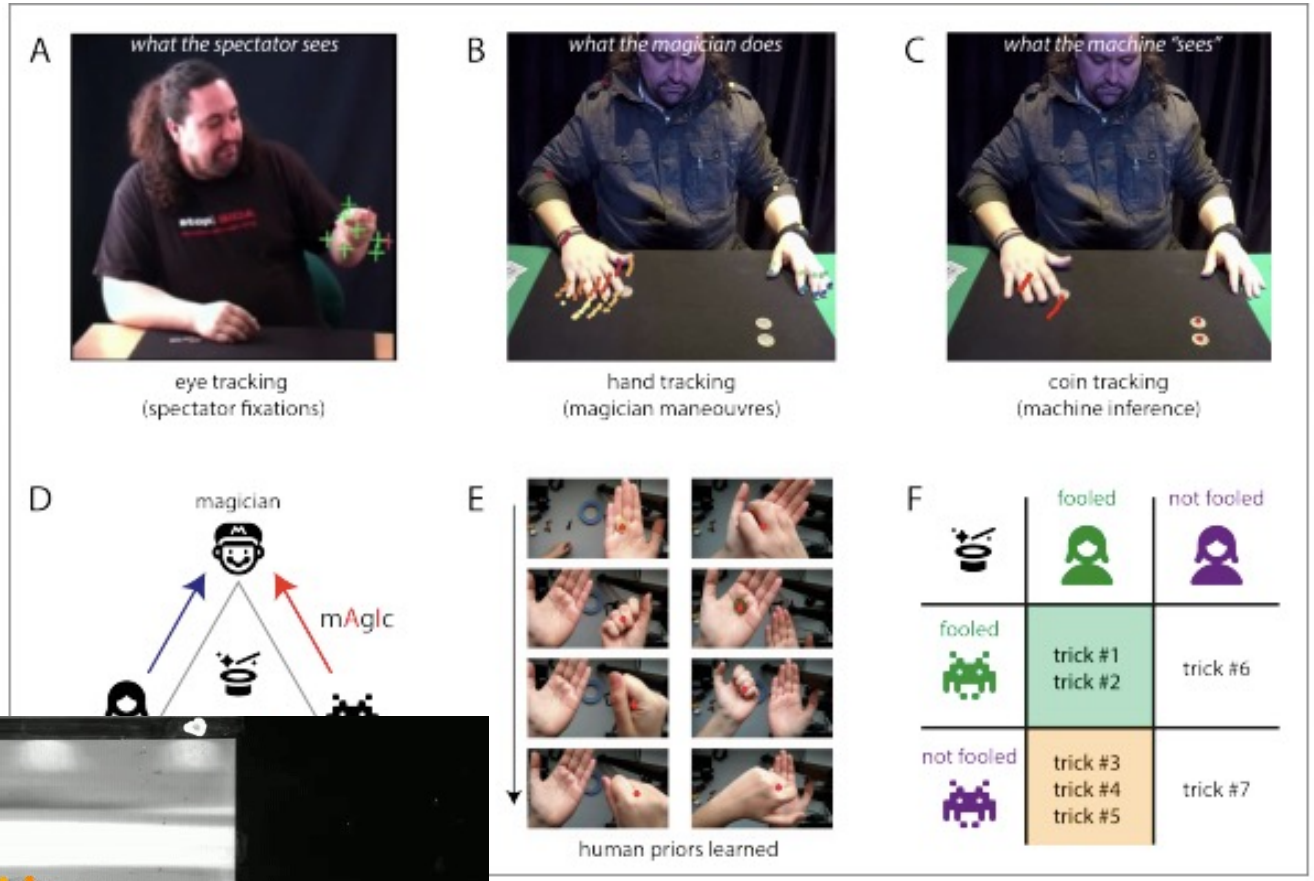


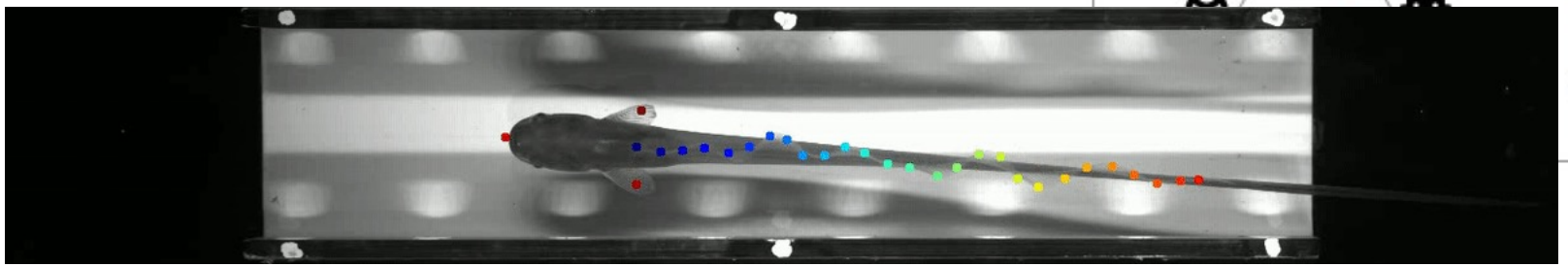


What bodyparts can be detected?

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!)

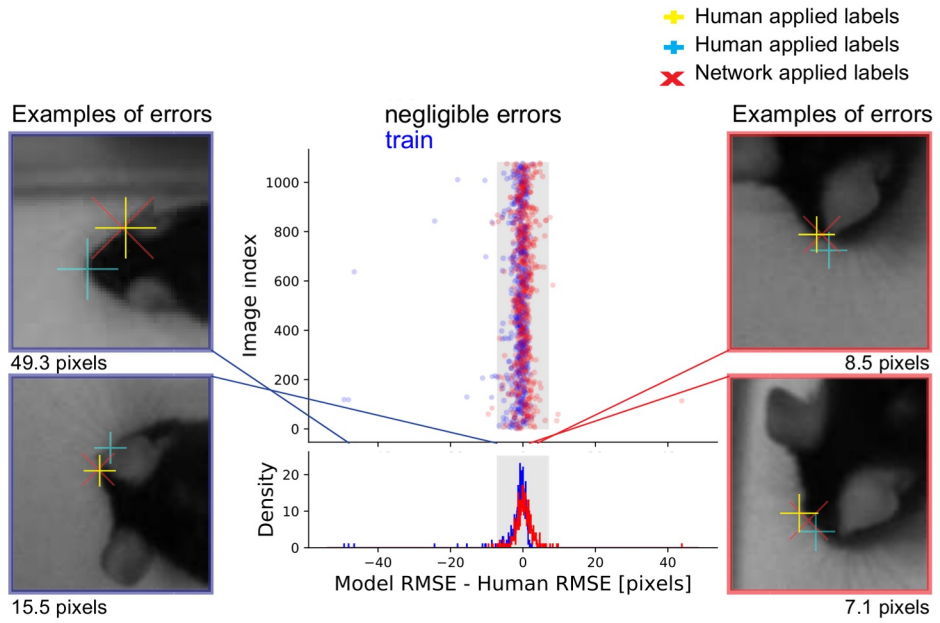




Playing magic tricks to deep neural networks untangles human deception
by Zaghi-Lara et al. arxiv 2019

Labeling considerations

- Label (feature locations) **consistently**
- Errors are bad! (especially when only a few images are labeled)
- Left vs. right body parts can be dist. if full animal visible (due to large rec. fields), but flipped labels are problematic
- You can check labels by plotting (do this before training!)
- You can correct wrong labels manually with the labeling GUI



Mathis et al., Nature Neuro 2018

What network & augmentation method?

Network architecture:

- Network backbones: MobileNets V2, *ResNets, DLC-Rnet* & EfficientNets,…
- Scale (level of deconvolution) -> decrease stride for closely interacting animals and and when you want high resolution
- Differences: **Inference & training speed, resolution**

Pre-training (Weights):

- *ImageNet pretrained (the generalist) -> see* Mathis et al. 2021 WACV
- MPII-pose pretrained
- **DeepLabCut model zoo**

Training method:

- Learning rate & optimizer (Adam/SGD, batch_size, …)
- Augmentation transformations (rotation, scaling, …)

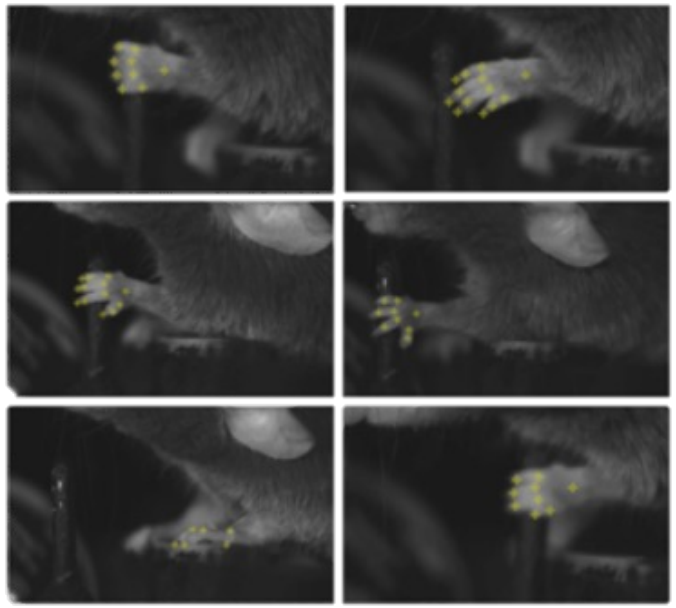
Online demo … let’s start training

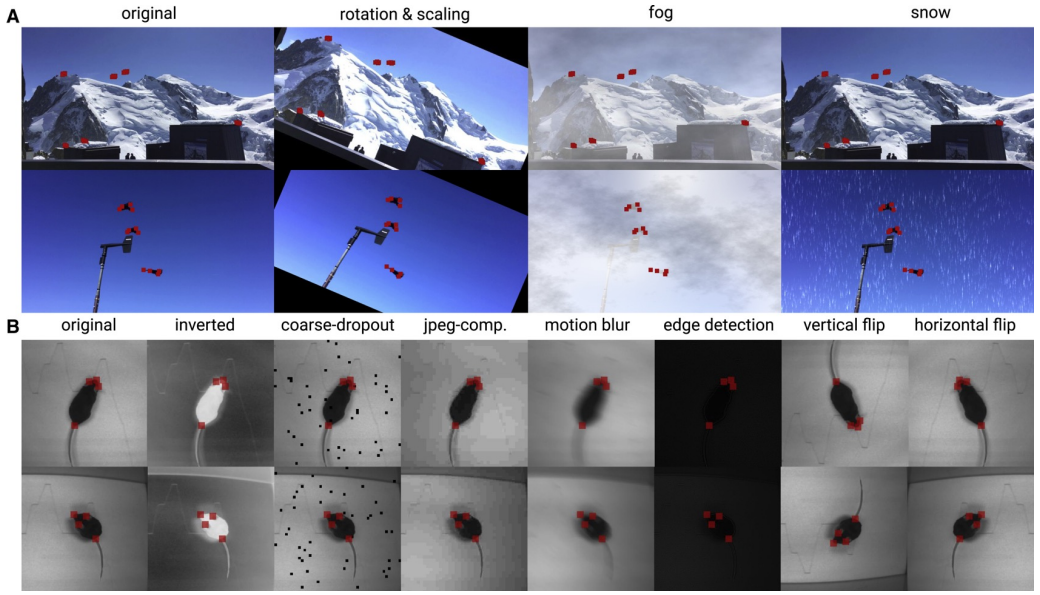
DEMO notebooks running on Google’s resources can be found here:
https://github.com/DeepLabCut/DeepLabCut/blob/master/examples/COLAB

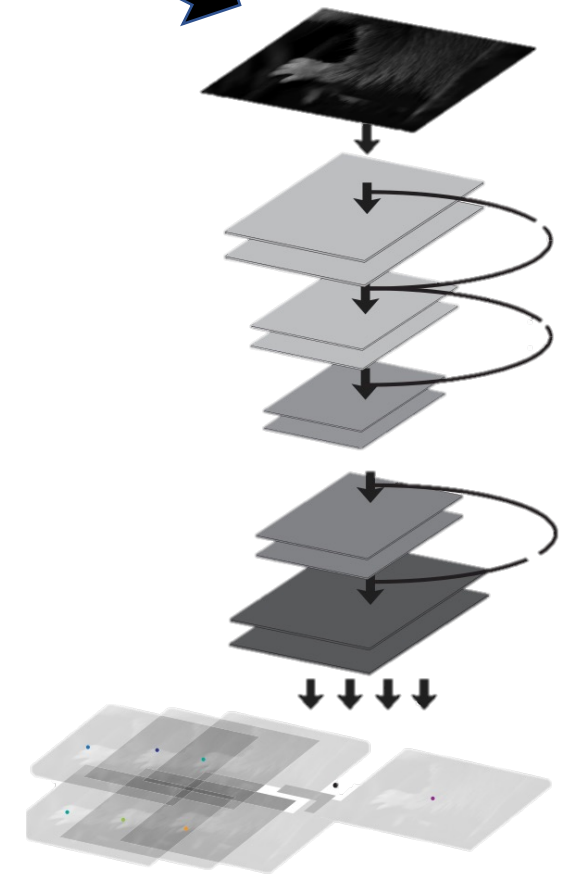
Contains examples for

- 3 mouse tracking with maDLC
- How to use a model from the model Zoo for inference
- How to go through all the steps of a DLC project
- And many more …

Augmentation

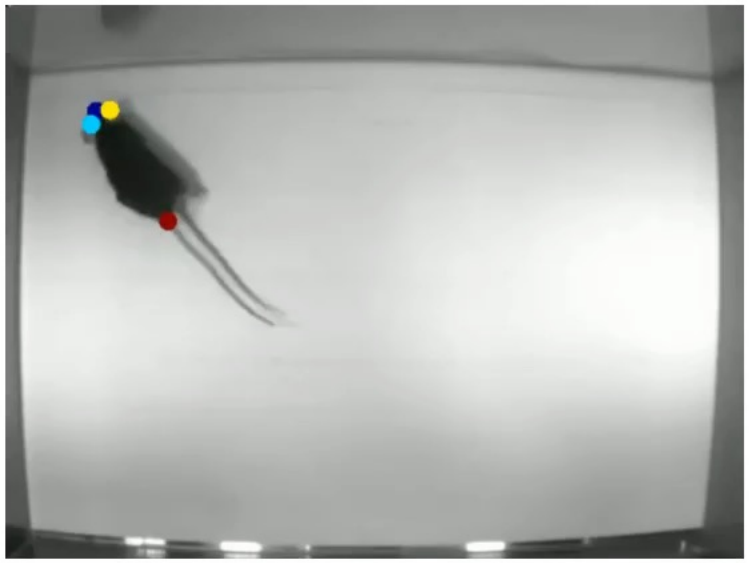




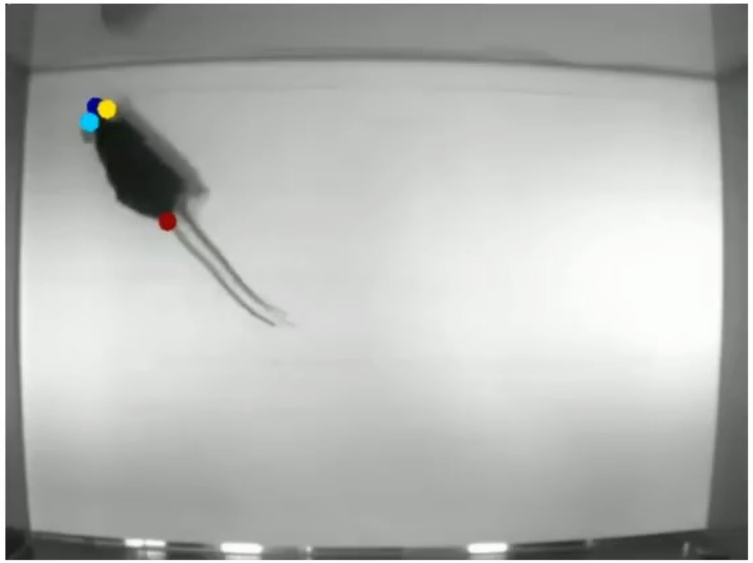


Mathis, Schneider,
Lauer & Mathis, 2020
Neuron

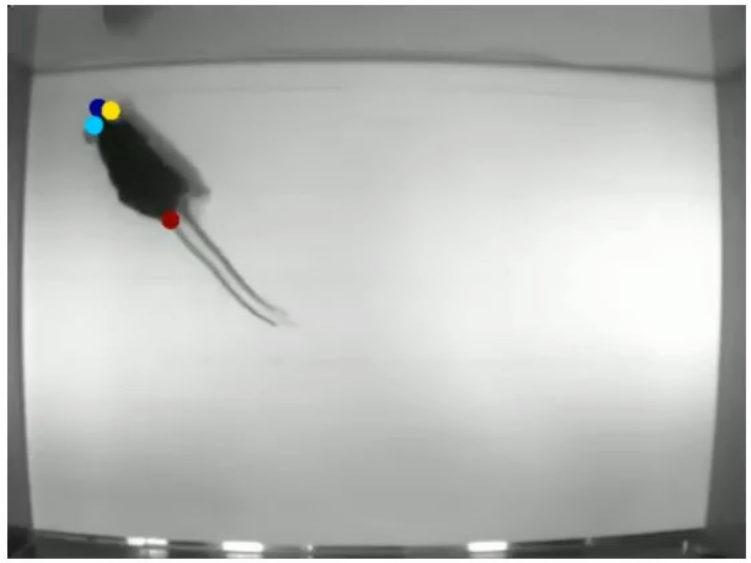
Data Augmentation: how to get the most out of your data!



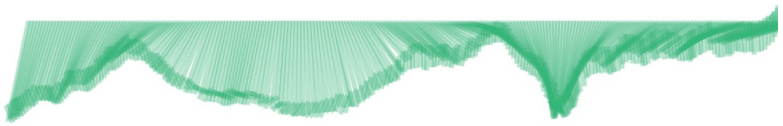
tensorpack

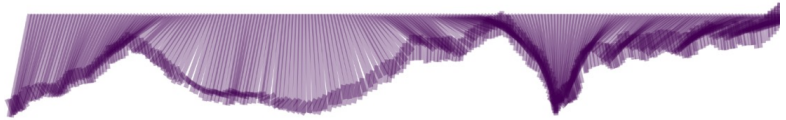


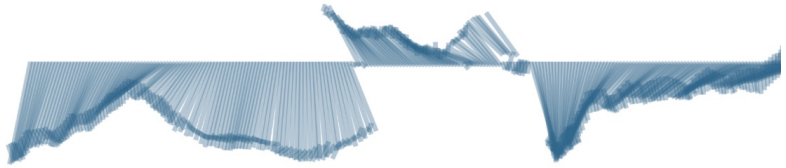
imgaug



scale-crop

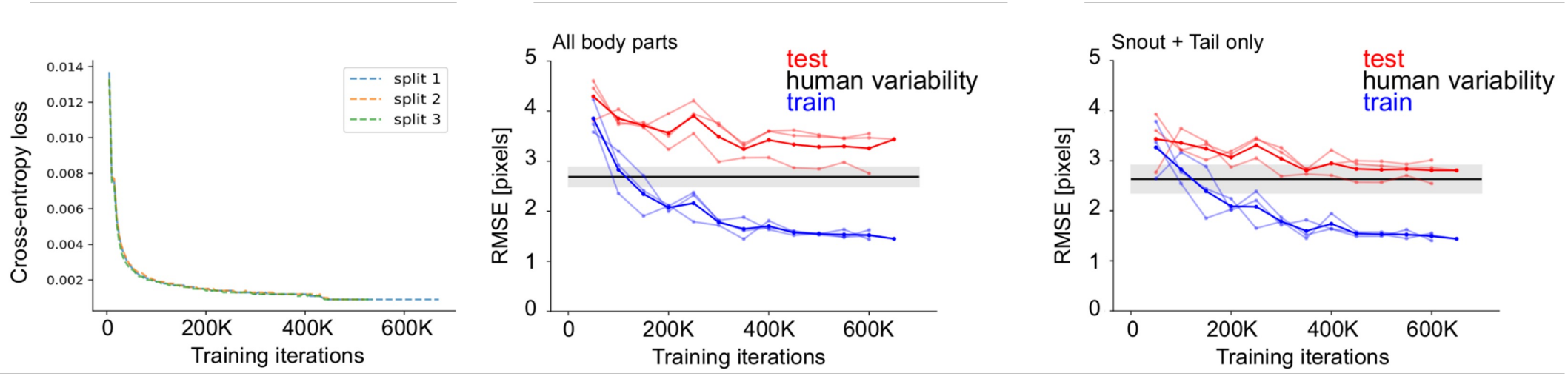






Mathis, Schneider, Lauer & Mathis 2020 Neuron

Evaluation: *Compare test RMSE and ideally your labeling error!*



Mathis et al. 2018, Nat Neuro

Evaluation => look at test images!

a Example of the terminal output

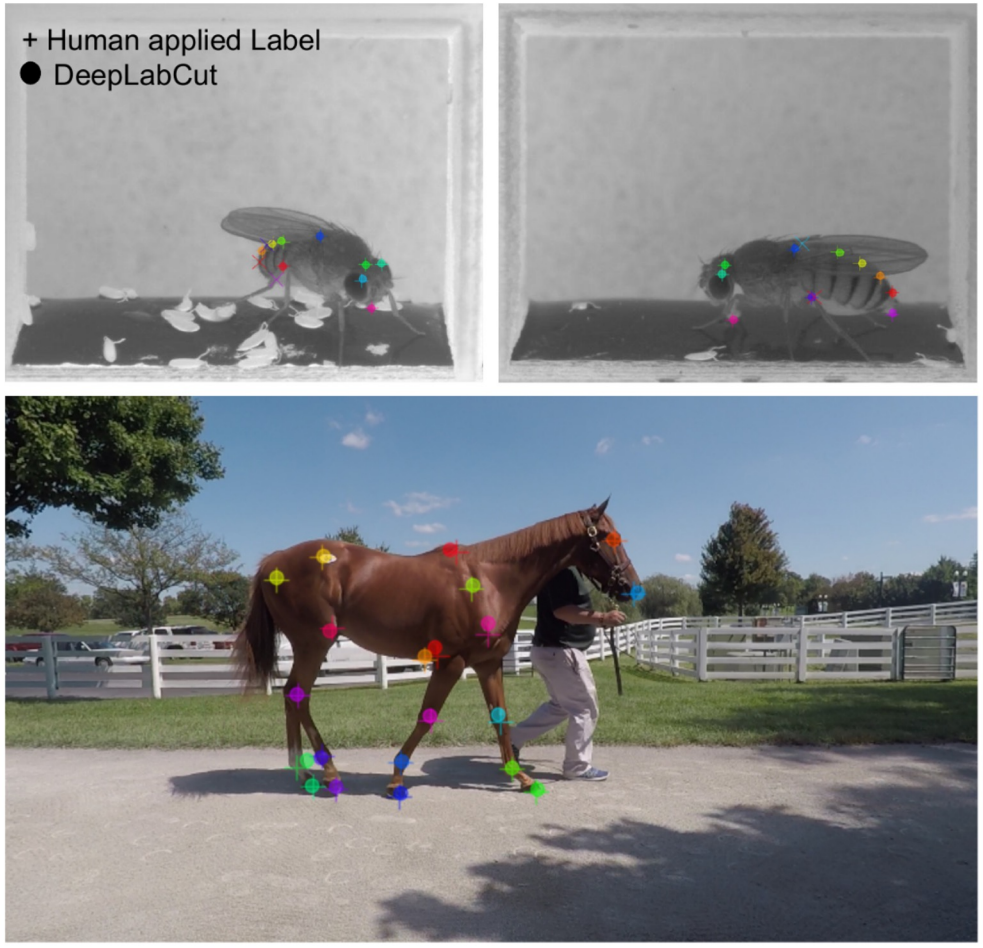
```
deeplabcut.evaluate_network('<path of the proj. config file>', shuffle = [1], plotting=True)
```

```
Assessing accuracy of shuffle #  1  with  99  % training fraction.
Found the following training snapshots:  [(200000, 0)]
You can choose among those for analyis of train/test performance.
Results for 200000 training iterations: 99 1 train error: 4.81 pixels. Test error: 7.02  pixels.
With pcutoff of 0.1  train error: 3.3 pixels. Test error: 5.2 pixels
```

b Examples of **training images** with Human and DeepLabCut labels

c Examples of **test images** with Human and DeepLabCut labels

d Examples of **test images** with Human and DeepLabCut labels



Nath* & Mathis* et al. Nature Protocols 2019

Table 1 | Summary of commands

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)`

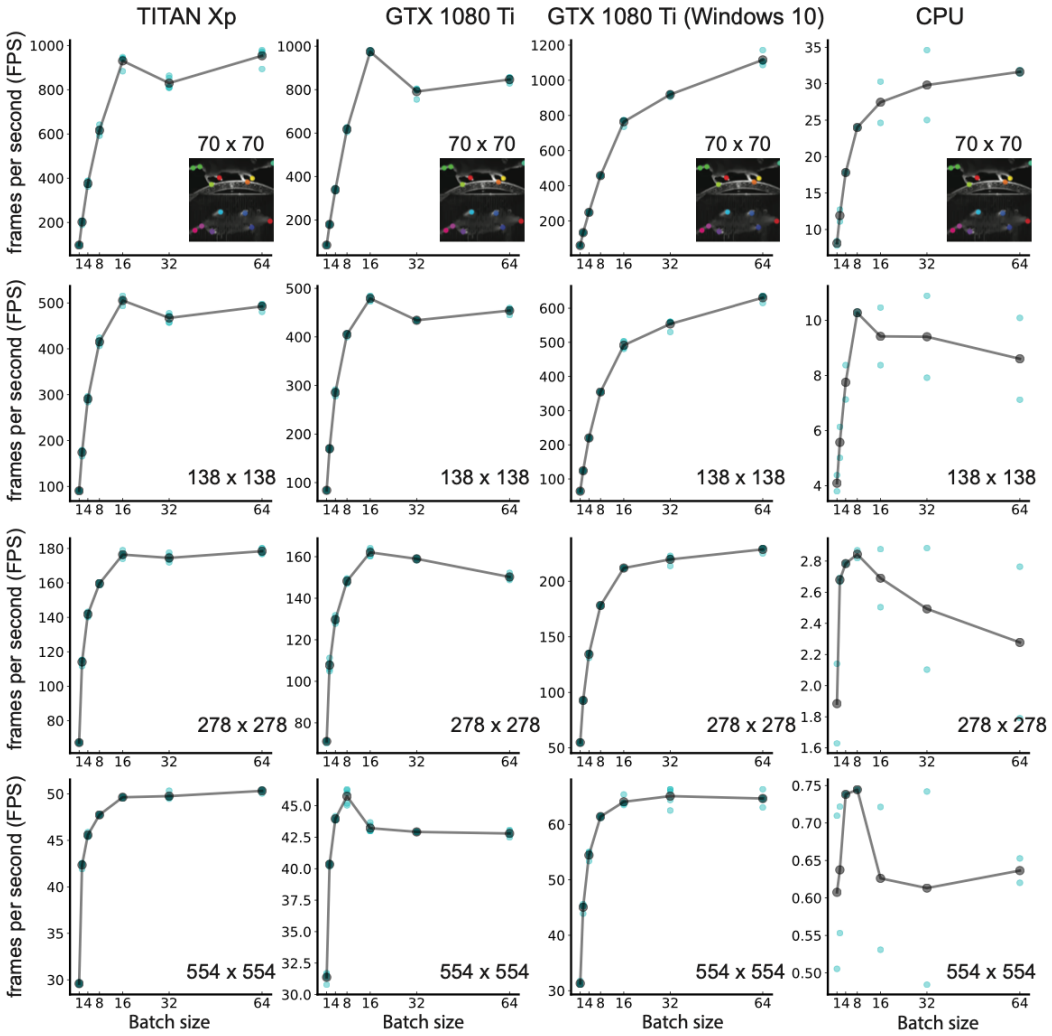
Video analysis

Videos for doing inference

need **not** be added to the config.yaml (only for labeling/active learning)!

Set *batch_size* in the config.yaml!

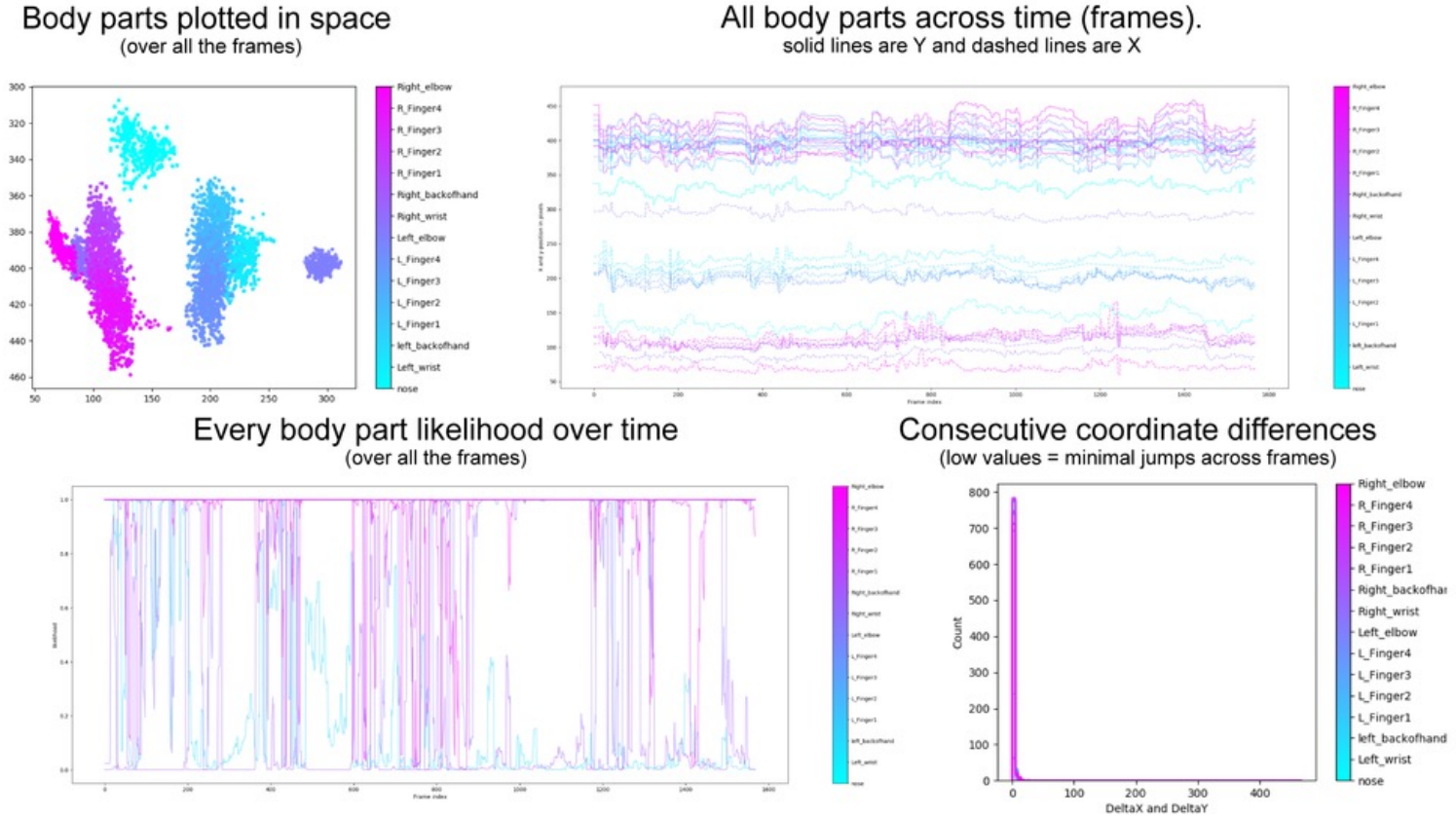
Also check out the real-time DLC paper for speed benchmarking:
https://elifesciences.org/articles/61909



Mathis & Warren 2018

Trajectory visualization

```
deeplabcut.plot_trajectories(config_path,['fullpath/analysis/project/videos/reachingvideo1.avi'])
```



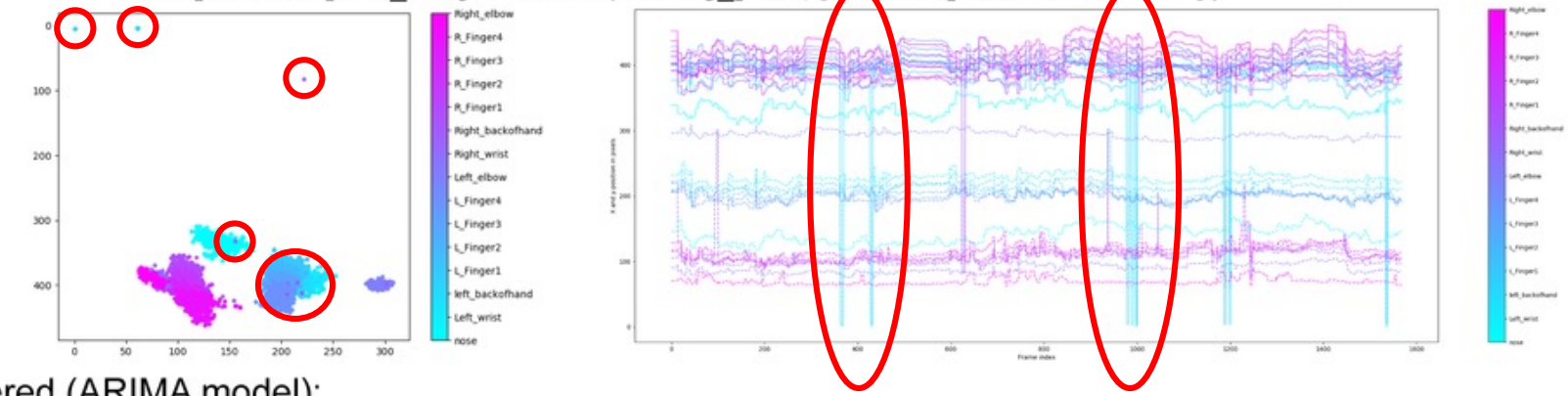
Nath* & Mathis* et al. Nature Protocols 2019

Trajectory filtering

```
deeplabcut.filterpredictions(config_path,['/videopath/mouse.avi'], p_bound=0.1, ARdegree=5, MAdegree=1)
```

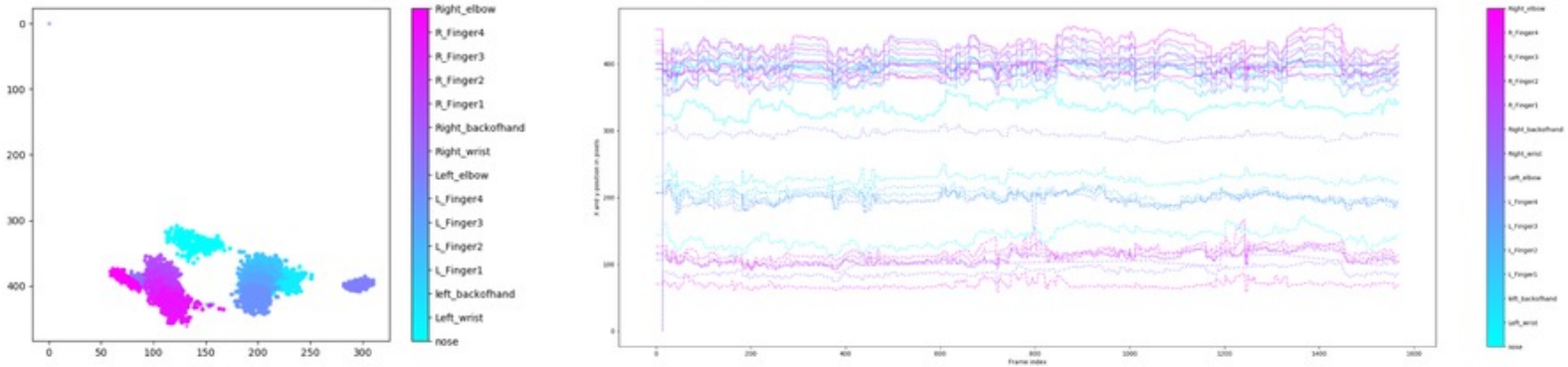
Unfiltered:

```
deeplabcut.plot_trajectories(config_path,['/videopath/mouse.avi'])
```

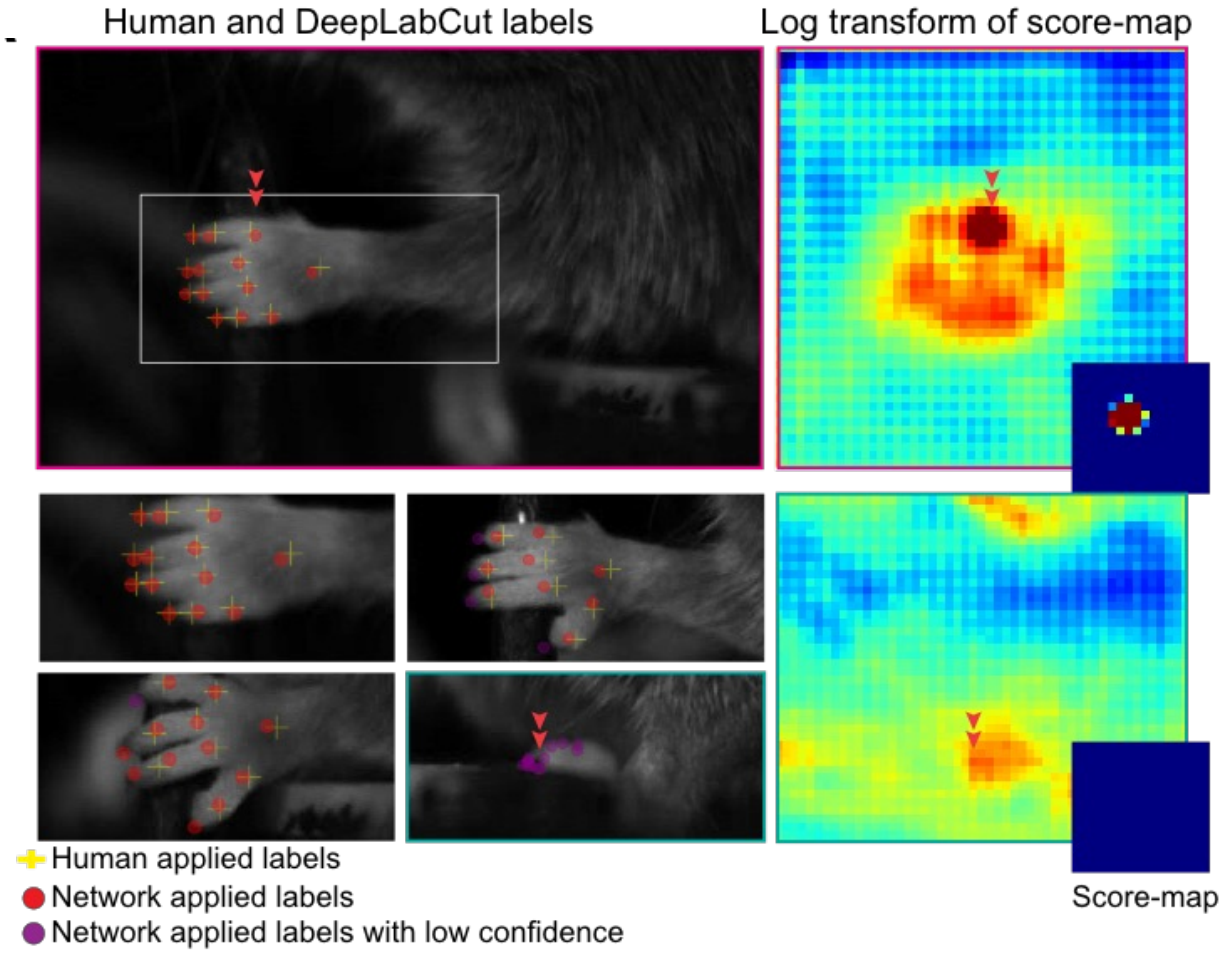


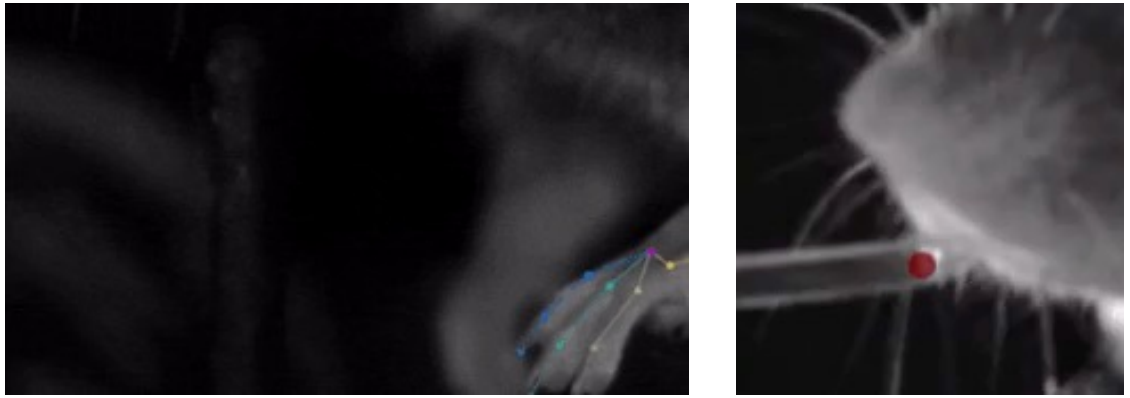
Filtered (ARIMA model):

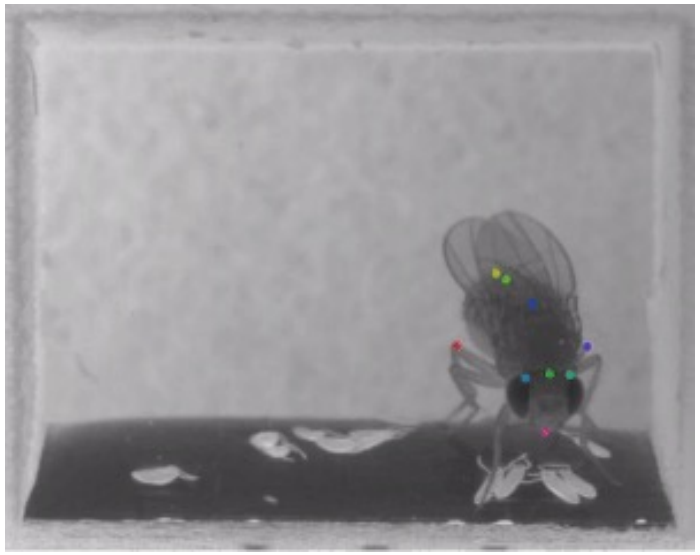
```
deeplabcut.plot_trajectories(config_path,['/videopath/mouse.avi'], filtered=True)
```



Score-maps provide network confidence readout

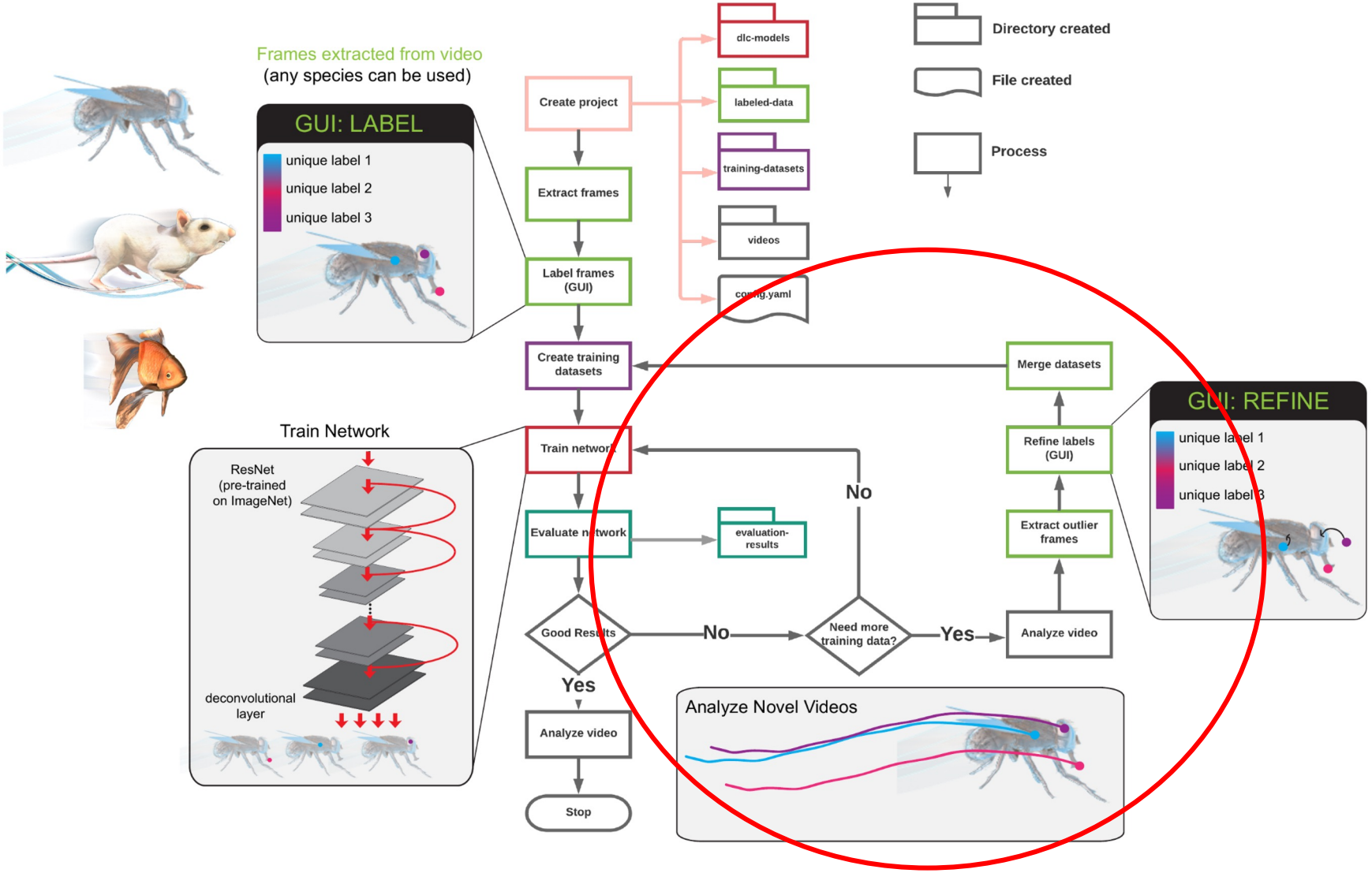






Mathis et al, Nature Neuroscience 2018

DeepLabCut 2.0 workflow



Nath* & Mathis* et al. Nature Protocols 2019

Active learning

- If there are errors in analyzed videos, you can correct those manually and re-train!
- Note you do not need to correct all errors (presumably some are fixed due to generalization)

Nath* & Mathis* et al. Nature Protocols 2019

Active learning

```
deeplabcut.extract_outlier_frames(config_path,['videofile_path'])
```

- Key arguments

```
outlieralgorithm: 'fitting', 'jump', or 'uncertain'
```

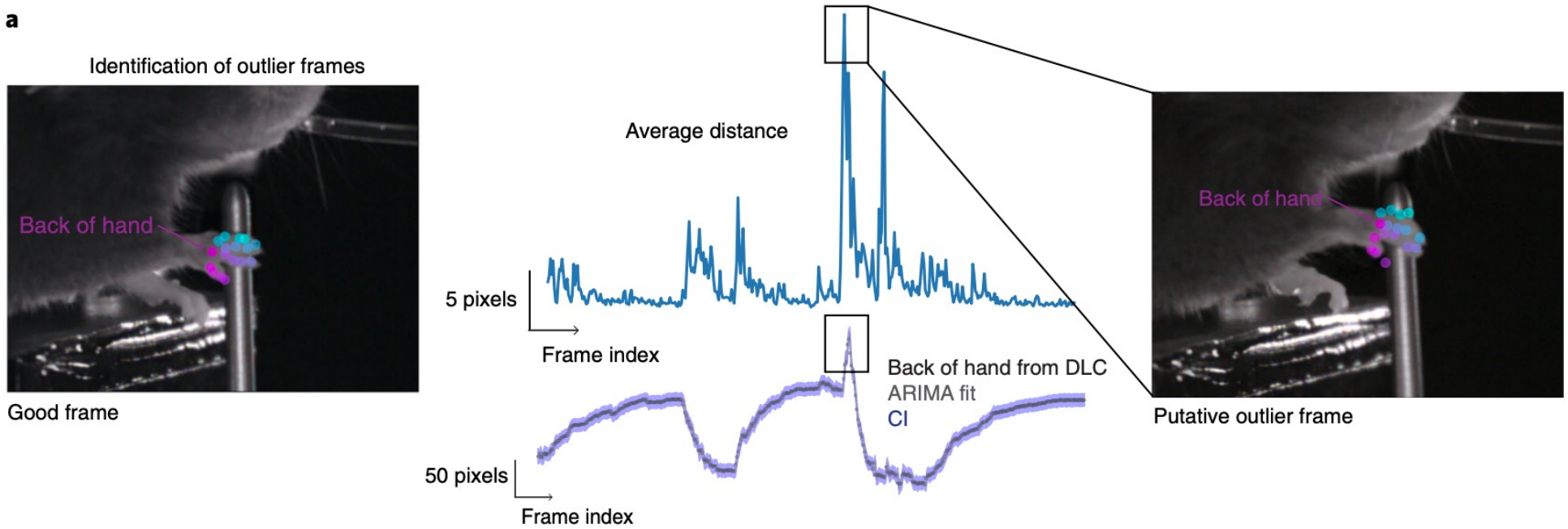
Select frames with jumps between consecutive frames > *epsilon* pixels

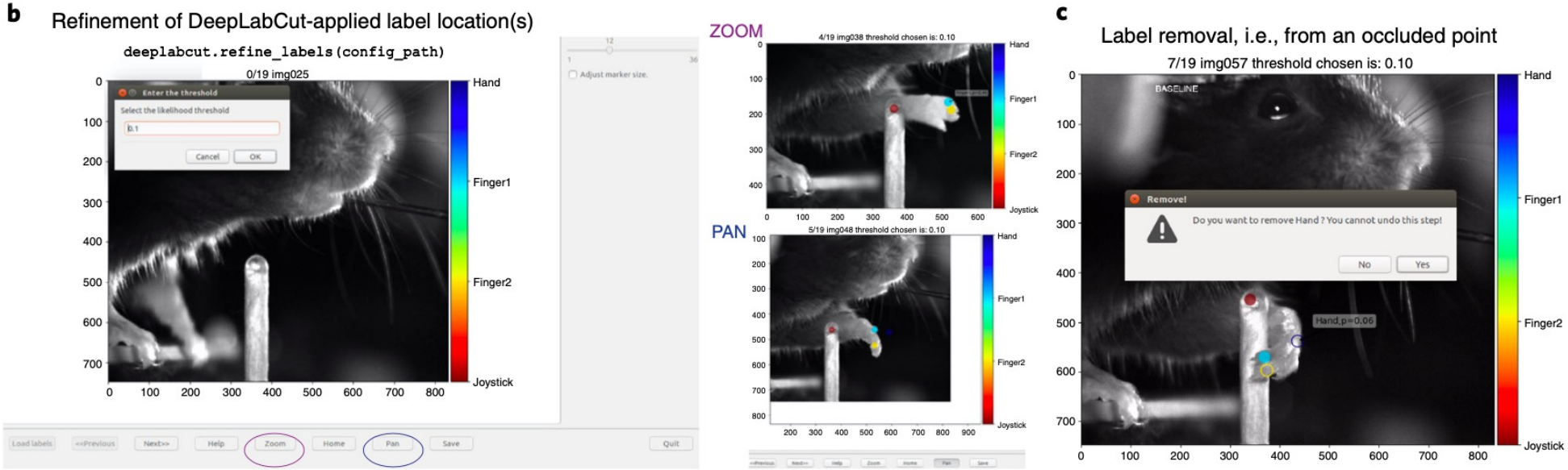
Select frames with confidences < *p_bound*

Select frames deviating from
a statistical model fit to the data

Random sampling	`extractionalgorithm='uniform'`
Clustering	`extractionalgorithm='k-means'`

Nath* & Mathis* et al. Nature Protocols 2019

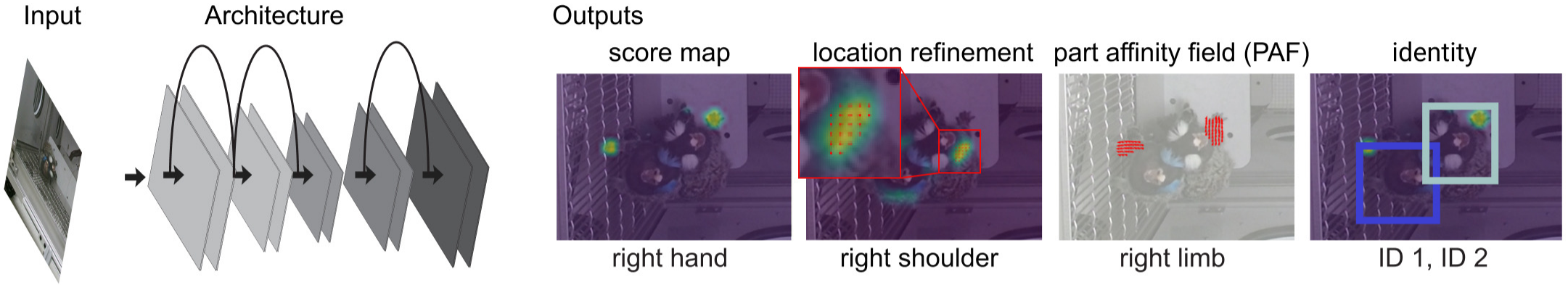

a
Identification of outlier frames
Back of hand
Good frame
Average distance
5 pixels
Frame index
Back of hand from DLC
ARIMA fit
CI
50 pixels
Frame index
Back of hand
Putative outlier frame

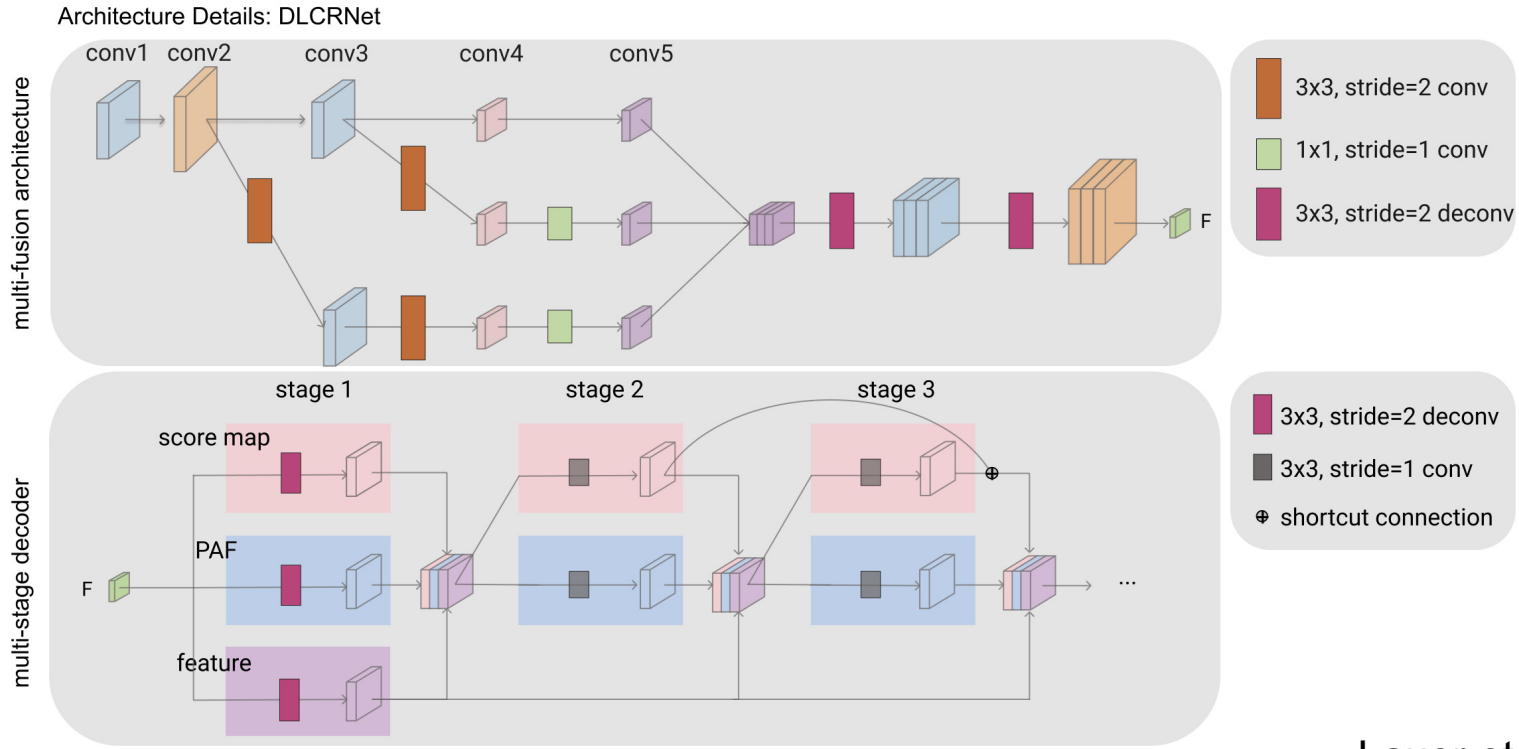

b
Refinement of DeepLabCut-applied label location(s)
deeplabcut.refine_labels(config_path)
0/19 img025
Enter the threshold
Select the likelihood threshold
Cancel
OK
Hand
Finger1
Finger2
Joystick
Adjust marker size.
Load labels
<<Previous
Next>>
Help
Zoom
Home
Pan
Save
Quit
ZOOM
4/19 img038 threshold chosen is: 0.10
PAN
5/19 img048 threshold chosen is: 0.10
c
Label removal, i.e., from an occluded point
7/19 img057 threshold chosen is: 0.10
BASELINE
Remove!
Do you want to remove Hand ? You cannot undo this step!
No
Yes
Hand.p=0.06

Multi-animal DLC

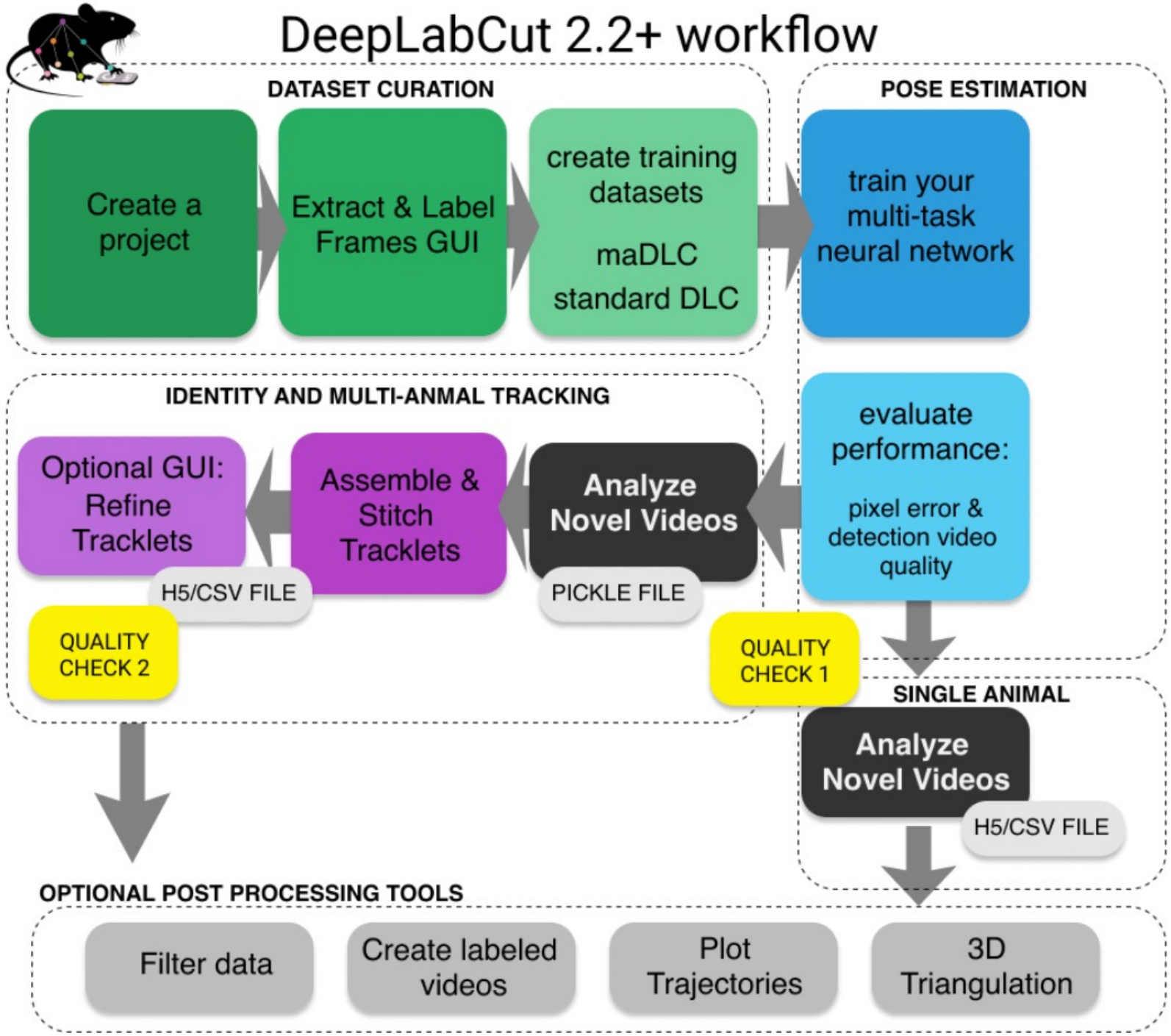
DEMO notebook for 3 mice:
https://github.com/DeepLabCut/DeepLabCut/blob/master/examples/COLAB

Multi-animal pose estimation & tracking with DeepLabCut





Lauer et al. Nat Methods 2022

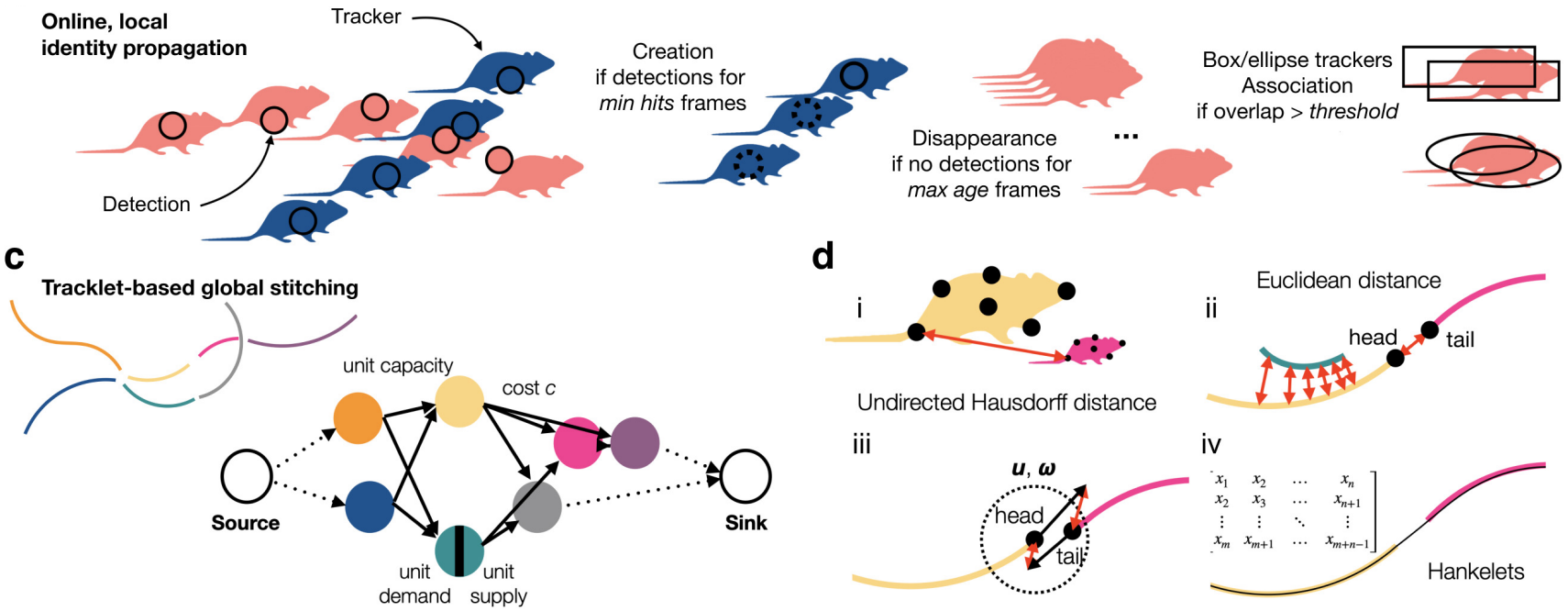


The workflow is available in the GUI or via Python.

Lauer et al. Nat Methods 2022

Tracking

Stitching



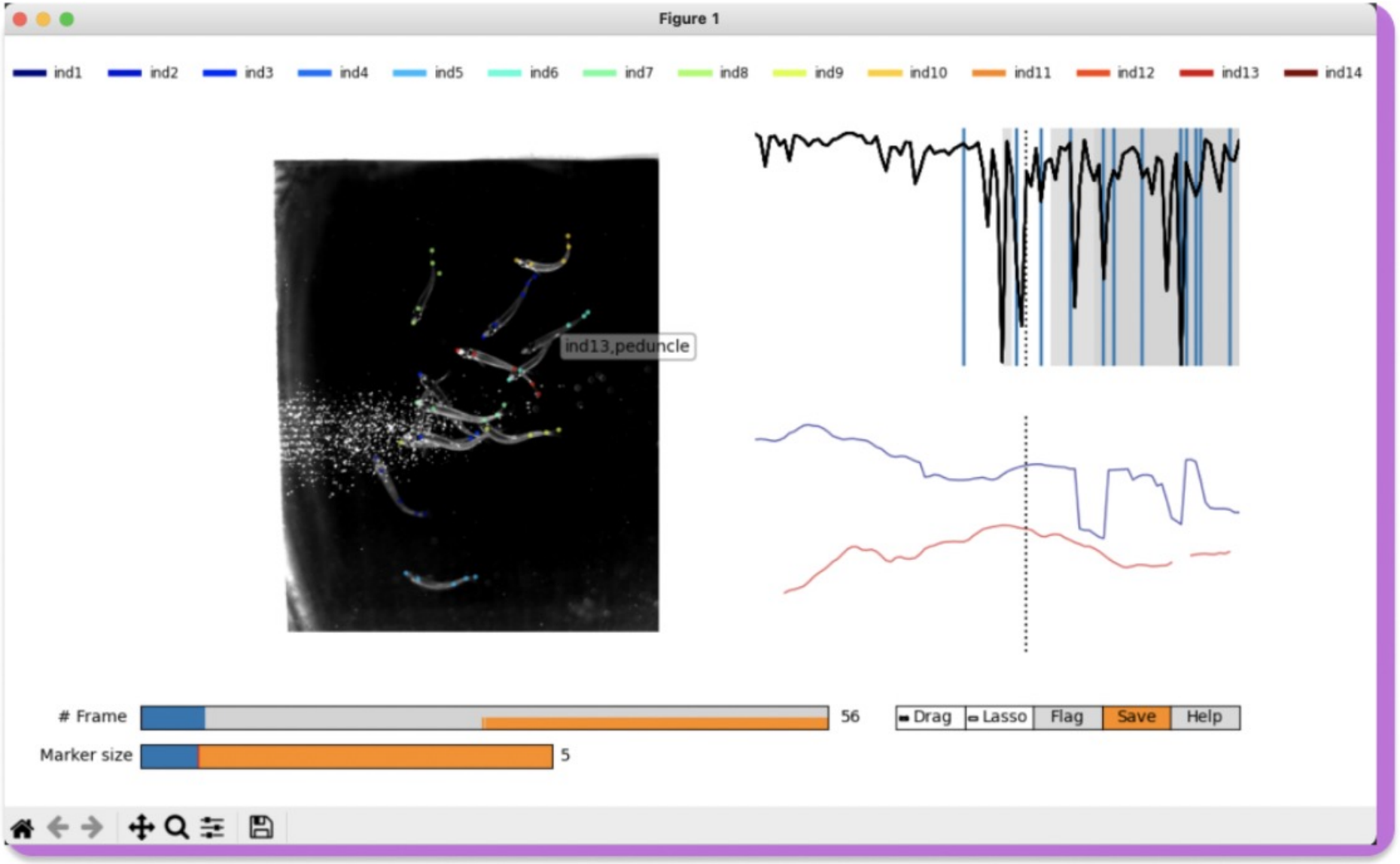
Corresponding commands:

```
dlc.analyze_videos( .. )

dlc.convert_detections2tracklets(…)

dlc.stitch_tracklets(…)
```

DEMO:

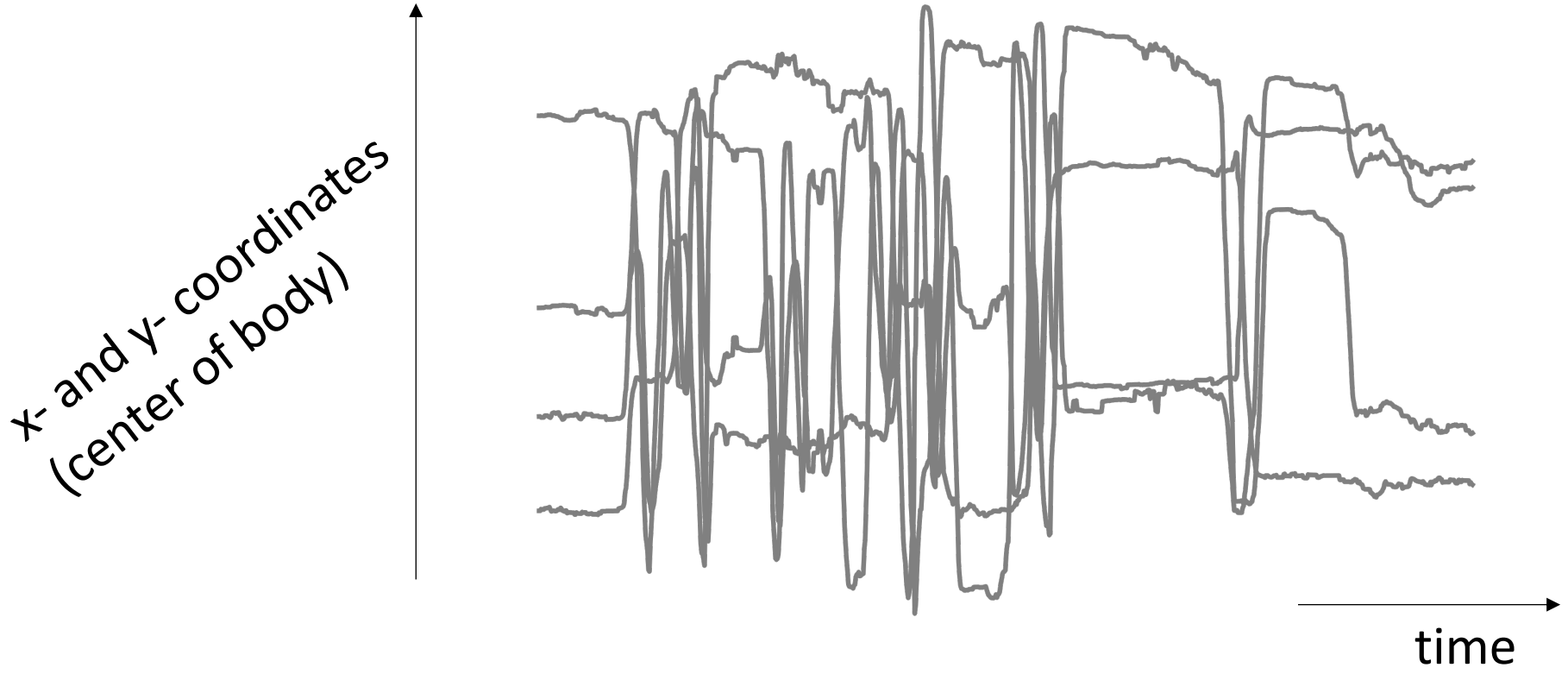
https://github.com/DeepLabCut/DeepLabCut/blob/master/examples/COLAB/COLAB_3

Refine tracklets GUI: deeplabcut.refine_tracklets()

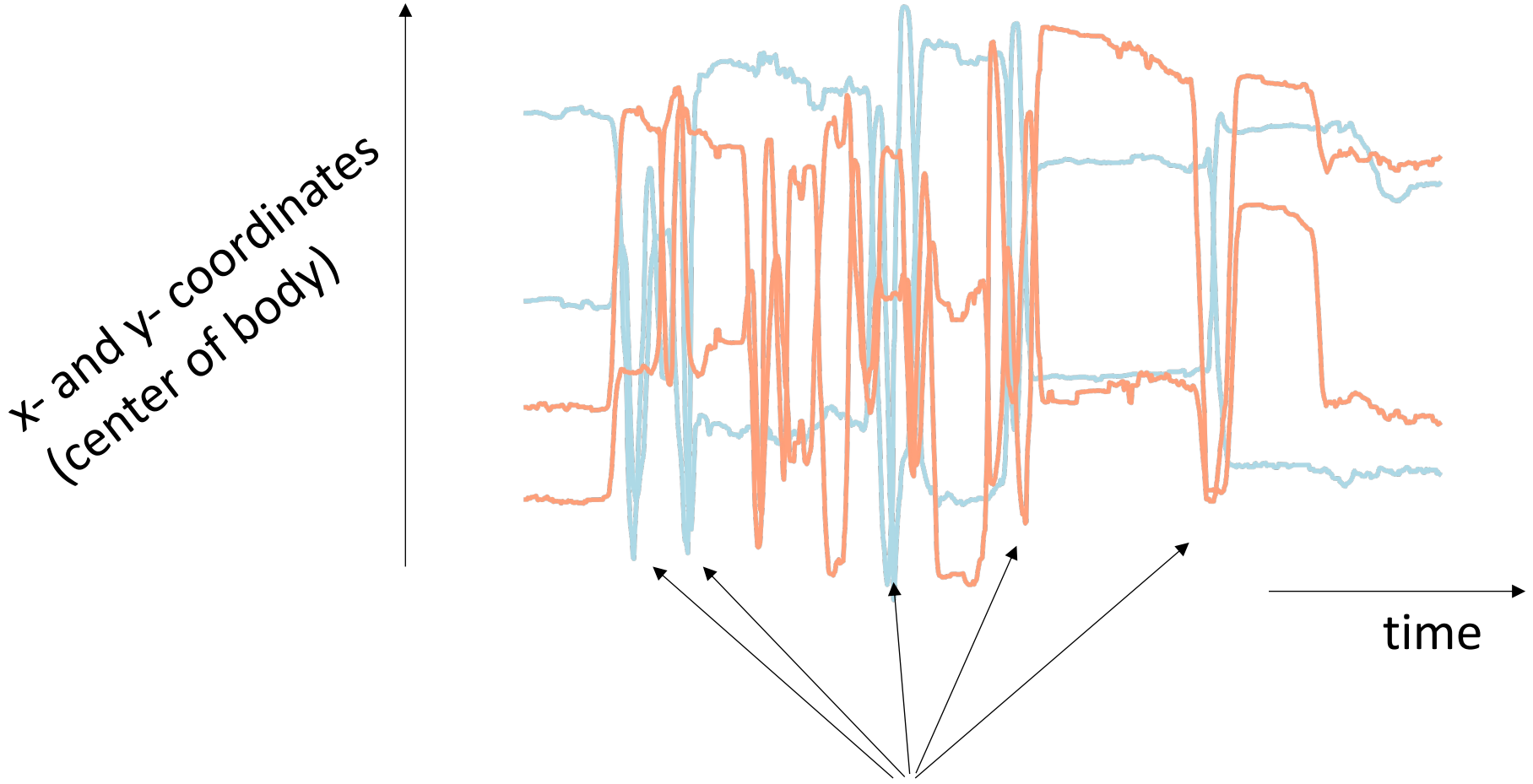


Lauer et al. Nat Methods 2022

Re-Identification is crucial for tracking in long-term scenarios

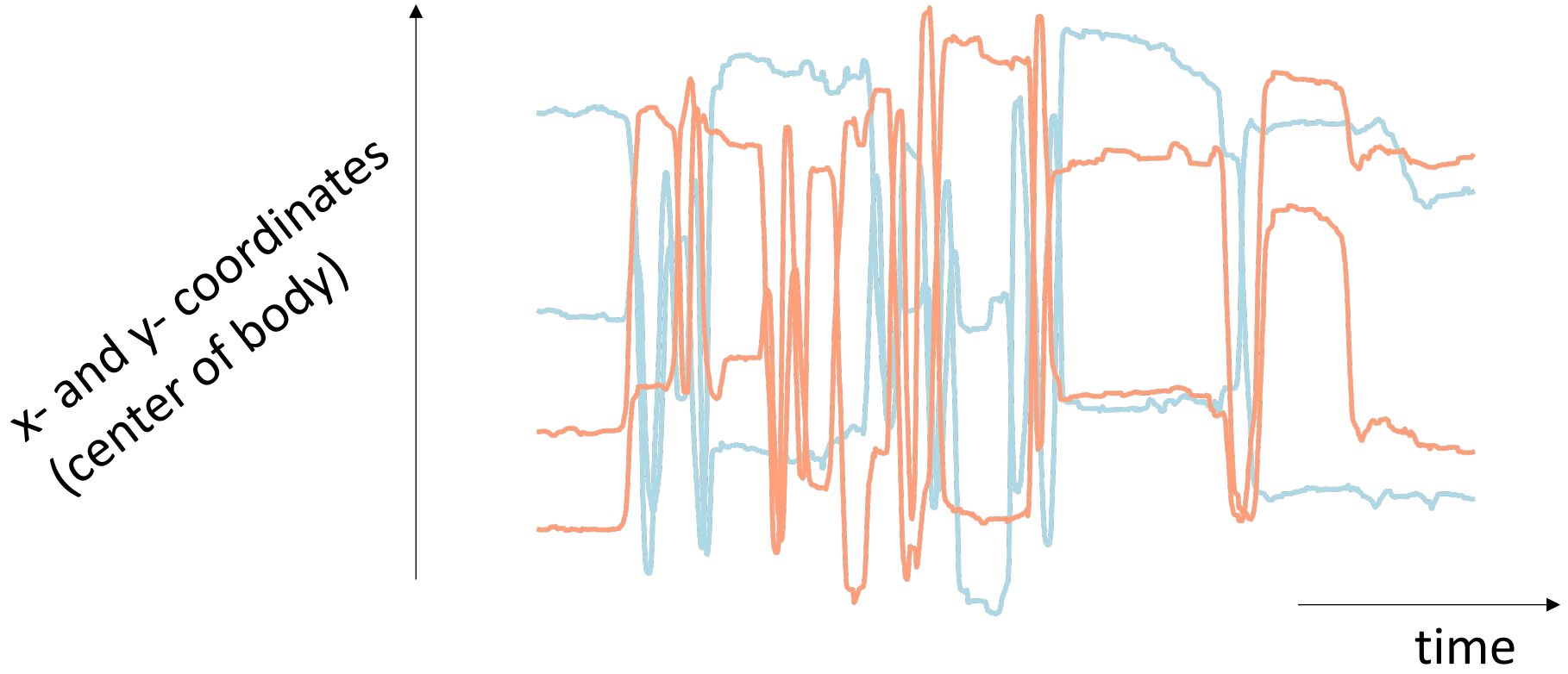


Ground truth data

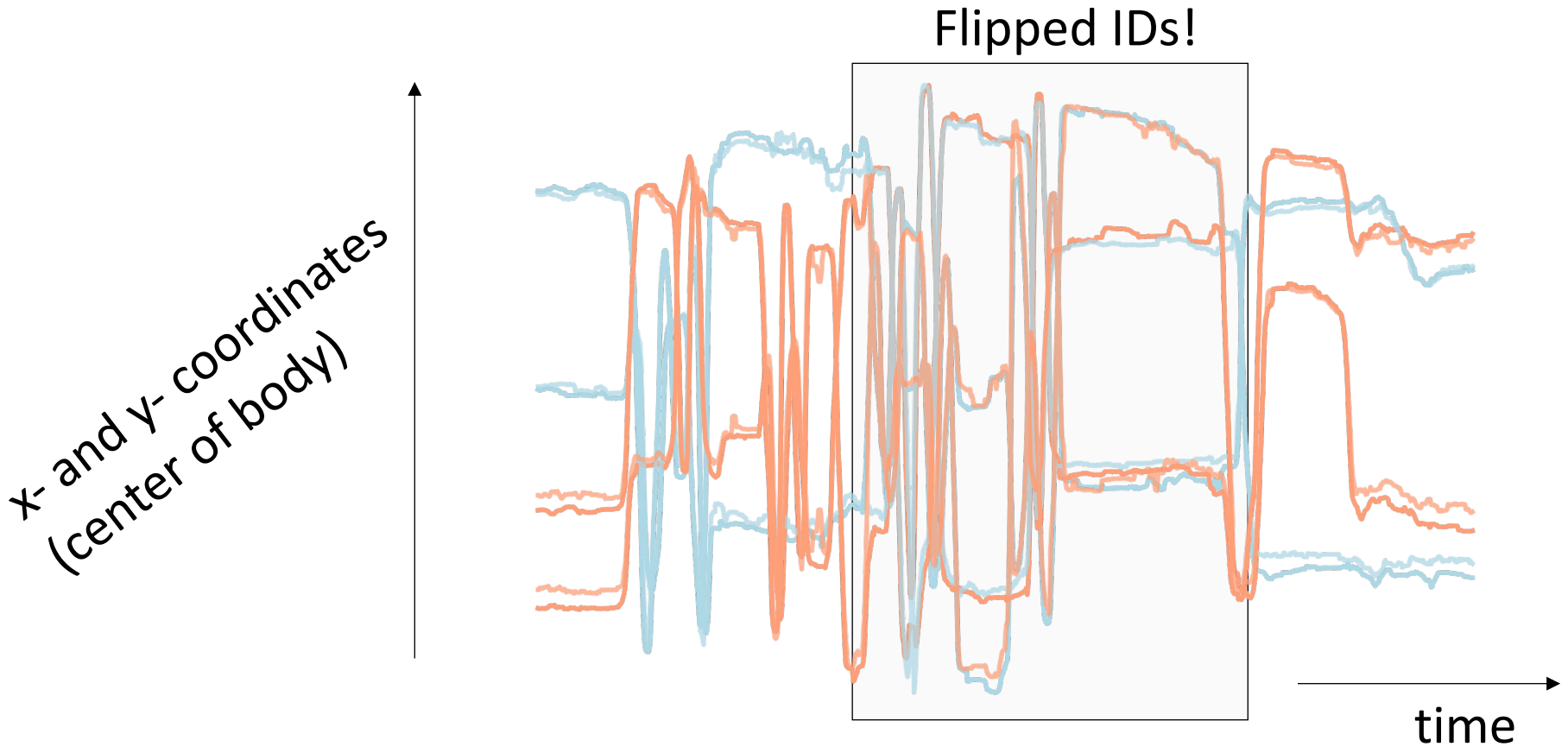


Example close interactions

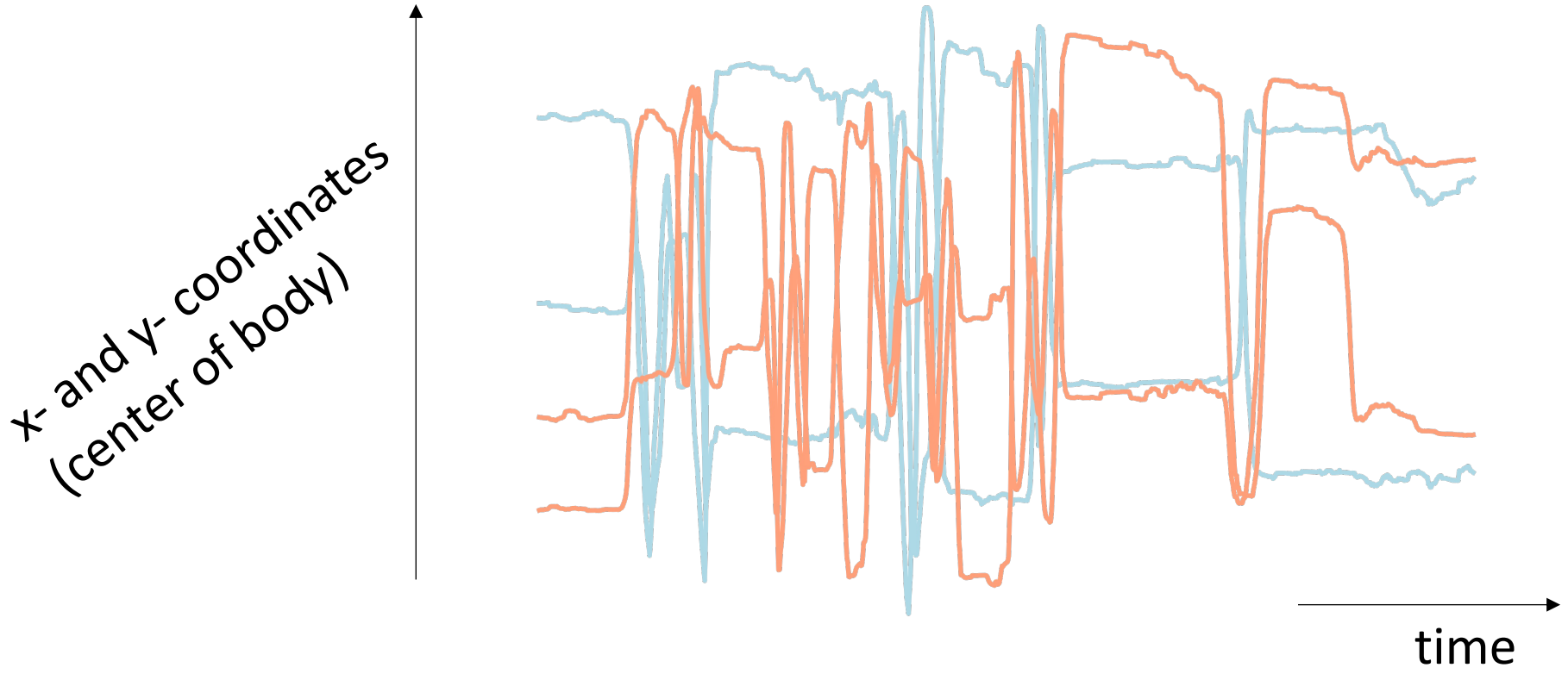
ID-based (temporally) *local* tracking



ID-based (temporally) *local* tracking

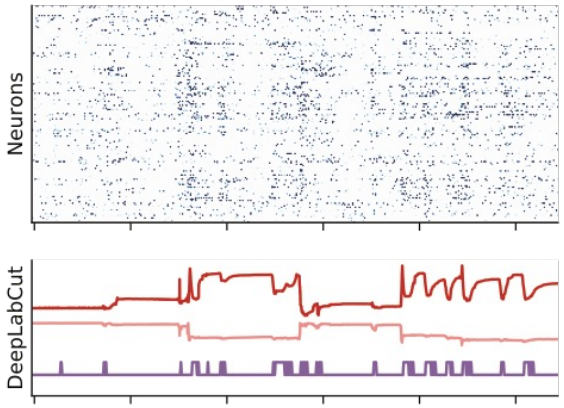


ReID based tracking (frame wise!)

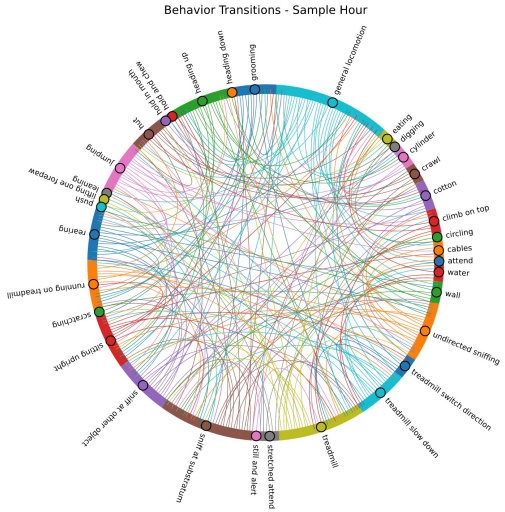


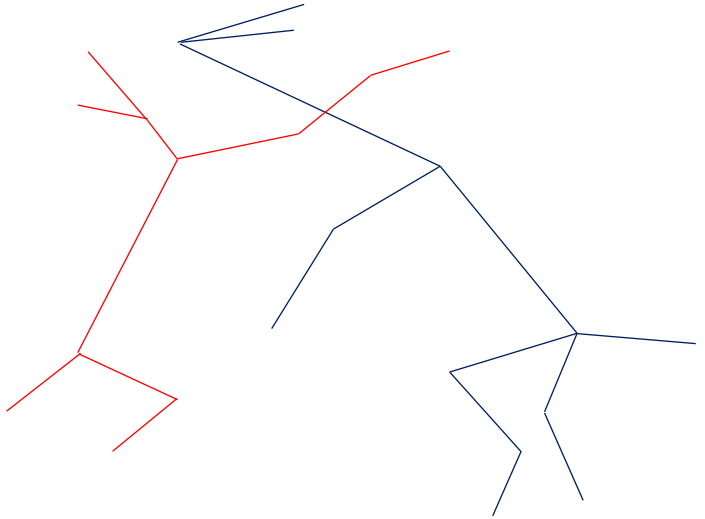
From pose to actions….

kinematics

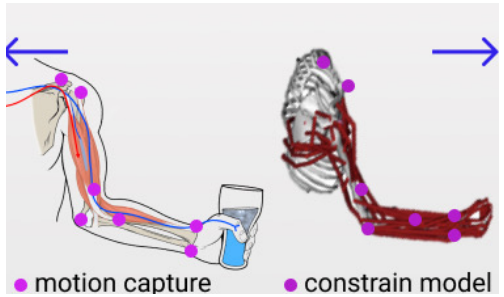


actions





biomechanics / modeling

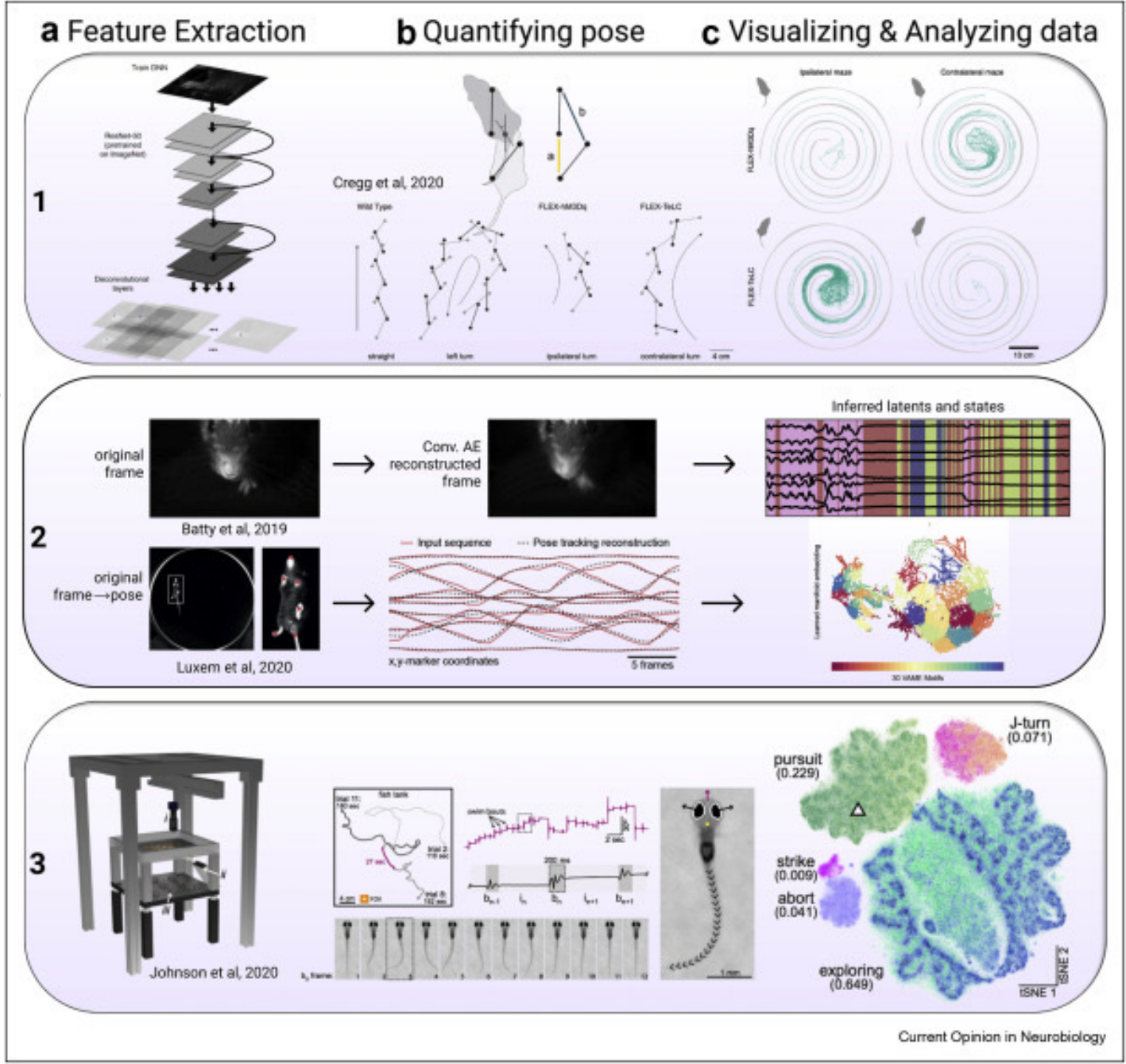


Primer

A Primer on Motion Capture with Deep Learning: Principles, Pitfalls, and Perspectives

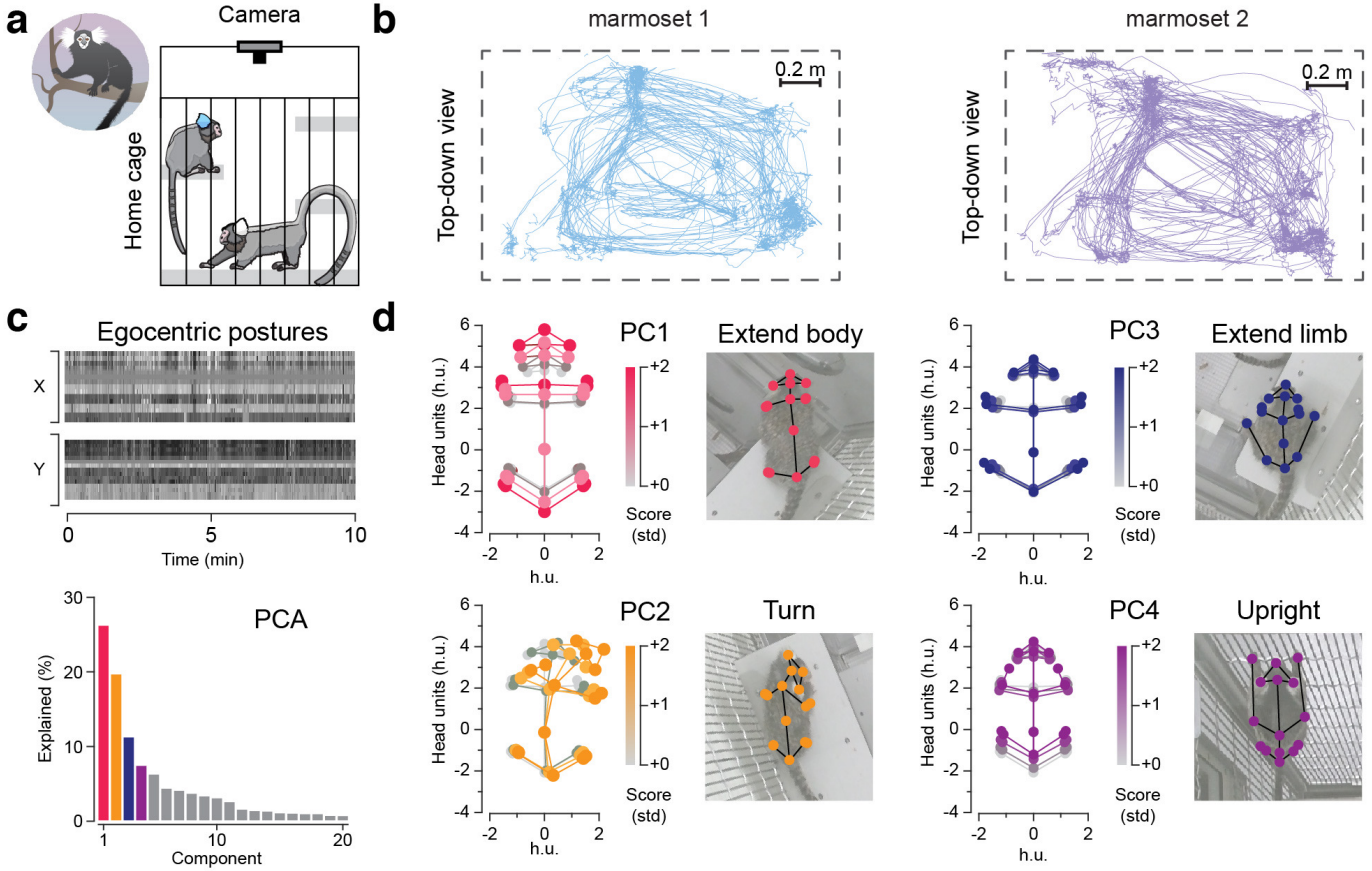
Alexander Mathis,[1,2,3,*] Steffen Schneider,[3,4] Jessy Lauer,[1,2,3] and Mackenzie Weygandt Mathis[1,2,3,*]
[1]Center for Neuroprosthetics, Center for Intelligent Systems, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
[2]Brain Mind Institute, School of Life Sciences, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
[3]The Rowland Institute at Harvard, Harvard University, Cambridge, MA, USA
[4]University of Tübingen and International Max Planck Research School for Intelligent Systems, Tübingen, Germany
*Correspondence: alexander.mathis@epfl.ch (A.M.), mackenzie.mathis@epfl.ch (M.W.M.)
https://doi.org/10.1016/j.neuron.2020.09.017

What to do with the pose estimation outputs?



Hausmann*, Marin-Vargas*, Mathis, Mathis
Curr Op. in Neuro 2021

Unsupervised Behavioral analysis based on DLC

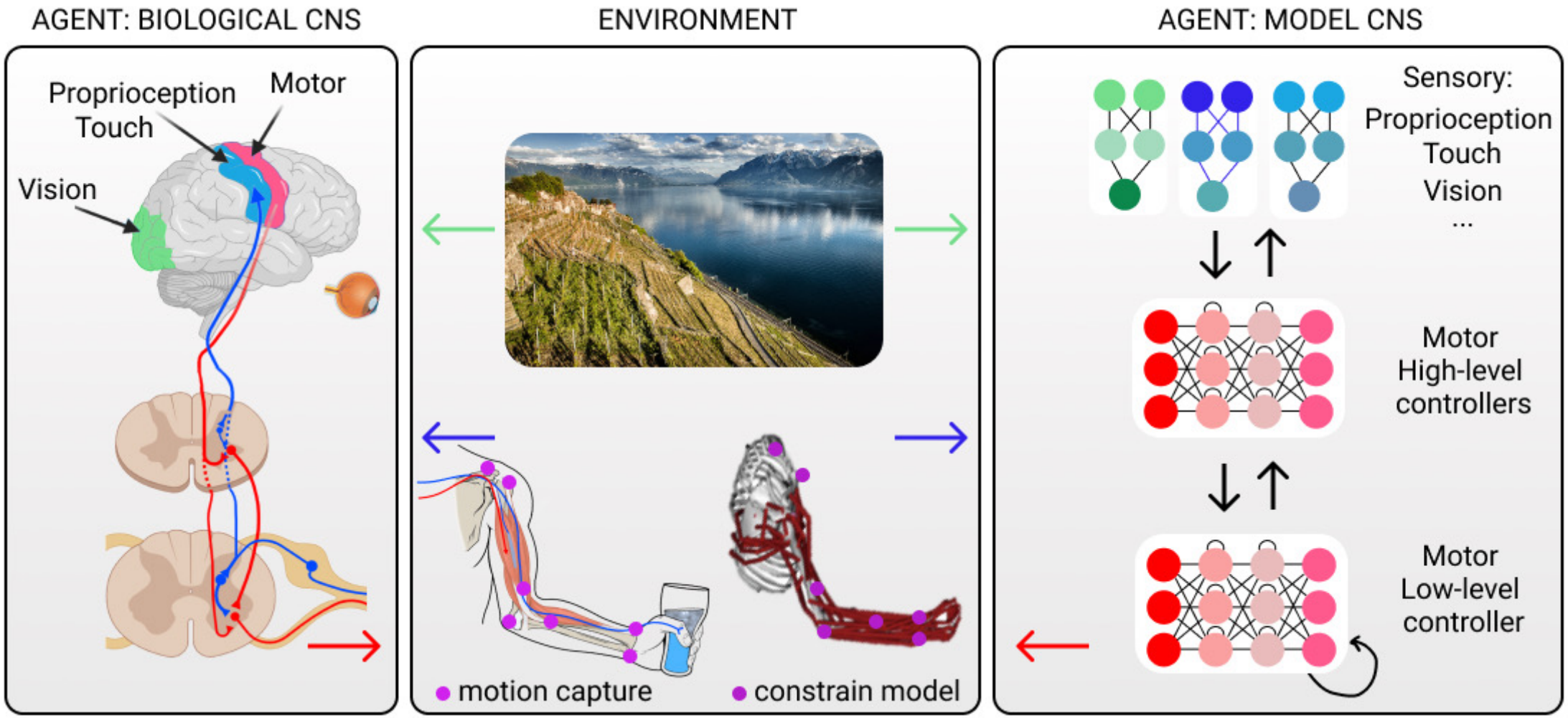


9h of continuous recording

With Dr. Will Menegas & Prof. Guoping Feng (MIT)

Lauer et al. BioRxiv 2021
Nature Methods 2022

Modeling of behavior with Deep Learning



Hausmann*, Marin-Vargas*, Mathis** & Mathis**, Curr Op. in Neuro. 2021

Box 2 | Parameters of interest in the network configuration file, *pose_cfg.yaml*

Please note, there are more parameters that typically never need to be adjusted; they can be found in the default *pose_cfg.yaml* file at https://github.com/AlexEMG/DeepLabCut/blob/master/deeplabcut/pose_cfg.yaml.

- `display_iters`: An integer value representing the period with which the loss is displayed (and stored in *log.csv*).
- `save_iters`: An integer value representing the period with which the checkpoints (weights of the network) are saved. Each snapshot has >90 MB, so not too many should be stored.
- `init_weights`: The weights used for training. Default: `<DeepLabCut_path>/Pose_Estimation_Tensorflow/pretrained/resnet_v1_50.ckpt`. For ResNet-50 or 101, -- this will be automatically created. The weights can also be changed to restart from a particular snapshot if training is interrupted, e.g., `<full path>-snapshot-5000` (with no file-type ending added). This would re-start training from the loaded weights (i.e., after 5,000 training iterations, the counter starts from 0).
- `multi_step`: These are the learning rates and number of training iterations to perform at the specified rate. If the users want to stop before 1 million, they can delete a row and/or change the last value to be the desired stop point.
- `max_input_size`: All images larger with size width × height > `max_input_size*max_input_size` are not used in training. The default is 1500 to prevent crashing with an out-of-memory exception for very large images. This will depend on your GPU memory capacity. However, we suggest reducing the pixel size as much as possible; see Mathis and Warren[27].

The following parameters allow one to change the resolution:

- `global_scale`: All images in the dataset will be rescaled by the following scaling factor to be processed by the convolutional neural network. You can select the optimal scale by cross-validation (see discussion in Mathis et al.[12]). Default is 0.8.
- `pos_dist_thresh`: All locations within this distance threshold (measured in pixels) are considered positive training samples for detection (see discussion in Mathis et al.[12]). Default is 17.

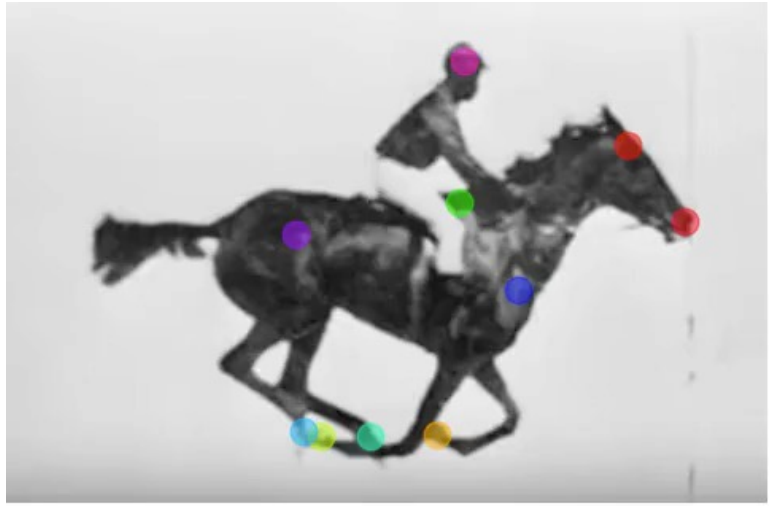
The following parameters modulate the data augmentation. During training, each image will be randomly rescaled within the range `[scale_jitter_lo, scale_jitter_up]` to augment training:

- `scale_jitter_lo`: 0.5 (default).
- `scale_jitter_up`: 1.5 (default).
- `mirror`: If the training dataset is symmetric around the vertical axis, this Boolean variable allows random augmentation. Default is `False`.
- `cropping`: Allows automatic cropping of images during training. Default is `True`.
- `cropratio`: Fraction of training samples that are cropped. Default is 40%.
- `minsize`, `leftwidth`, `rightwidth`, `bottomheight`, `topheight`: These define dimensions and limits for auto-cropping.

Nath* & Mathis* et al. Nature Protocols 2019

Questions?

Please get in touch!



alexander.mathis@epfl.ch

Open Source Code: https://github.com/DeepLabCut/
Tutorials, Data, Papers etc.: https://deeplabcut.org
User forum: https://forum.image.sc/tag/deeplabcut