

# Creating a project

## Contents

- Starting SLEAP
- Opening a video
- Creating a Skeleton

## Starting SLEAP

If you haven't installed SLEAP yet, see [Installation](#) for instructions.

Once you have SLEAP installed, start by opening a terminal. If you installed via the recommended method, activate the environment with:

```
conda activate sleap
```

[Skip to main content](#)



### Hint

To open a terminal:

**Windows:** Open the *Start menu* and search for the *Anaconda Command Prompt* (if using Miniconda) or the *Command Prompt* if not.



### Note

On Windows, our personal preference is to use alternative terminal apps like [Cmder](#) or [Windows Terminal](#).

**Linux:** Launch a new terminal by pressing `Ctrl` + `Alt` + `T`.

**Mac:** Launch a new terminal by pressing `Cmd` + `Space` and searching for *Terminal*.

To launch the GUI, simply enter in the terminal:

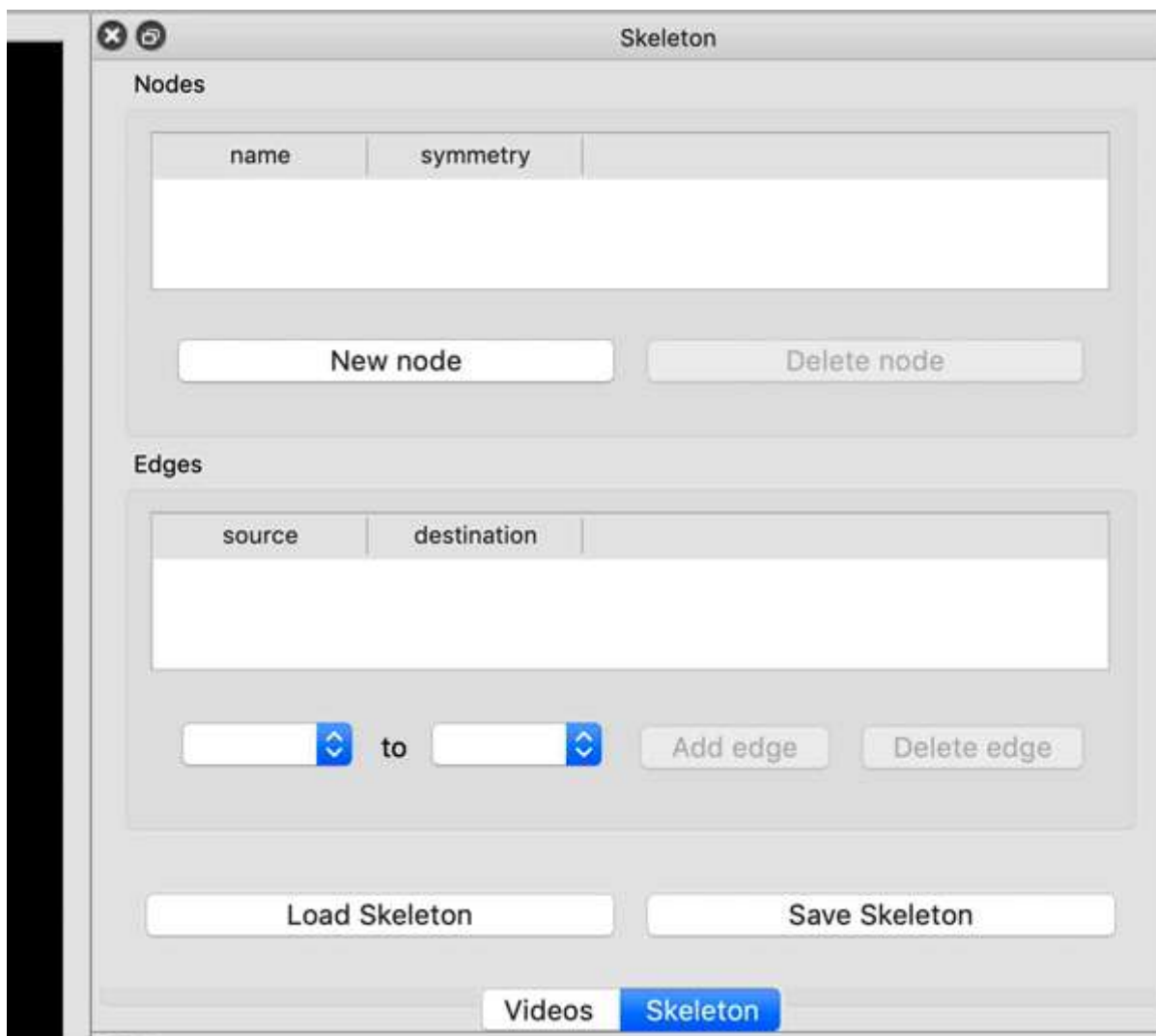
```
sleap-label
```

When you first start SLEAP you'll see a new, empty project.

## Opening a video

Add a video by clicking the “**Add Video**” button in the “**Videos**” panel on the right side of the main window, or by dragging-and-dropping your video file from its folder into the SLEAP GUI.

[Skip to main content](#)



The screenshot shows the 'Skeleton' window in the SLEAP application. The window has a title bar with standard macOS window controls. It is divided into two main sections: 'Nodes' and 'Edges'. The 'Nodes' section contains a table with columns 'name' and 'symmetry', followed by a 'New node' button and a disabled 'Delete node' button. The 'Edges' section contains a table with columns 'source' and 'destination', followed by a form with two dropdown menus and a 'to' label, and buttons for 'Add edge' and 'Delete edge'. At the bottom of the window are 'Load Skeleton' and 'Save Skeleton' buttons, and a tab bar with 'Videos' and 'Skeleton' tabs, where 'Skeleton' is currently selected.

name	symmetry
------	----------

New node Delete node

source	destination
--------	-------------

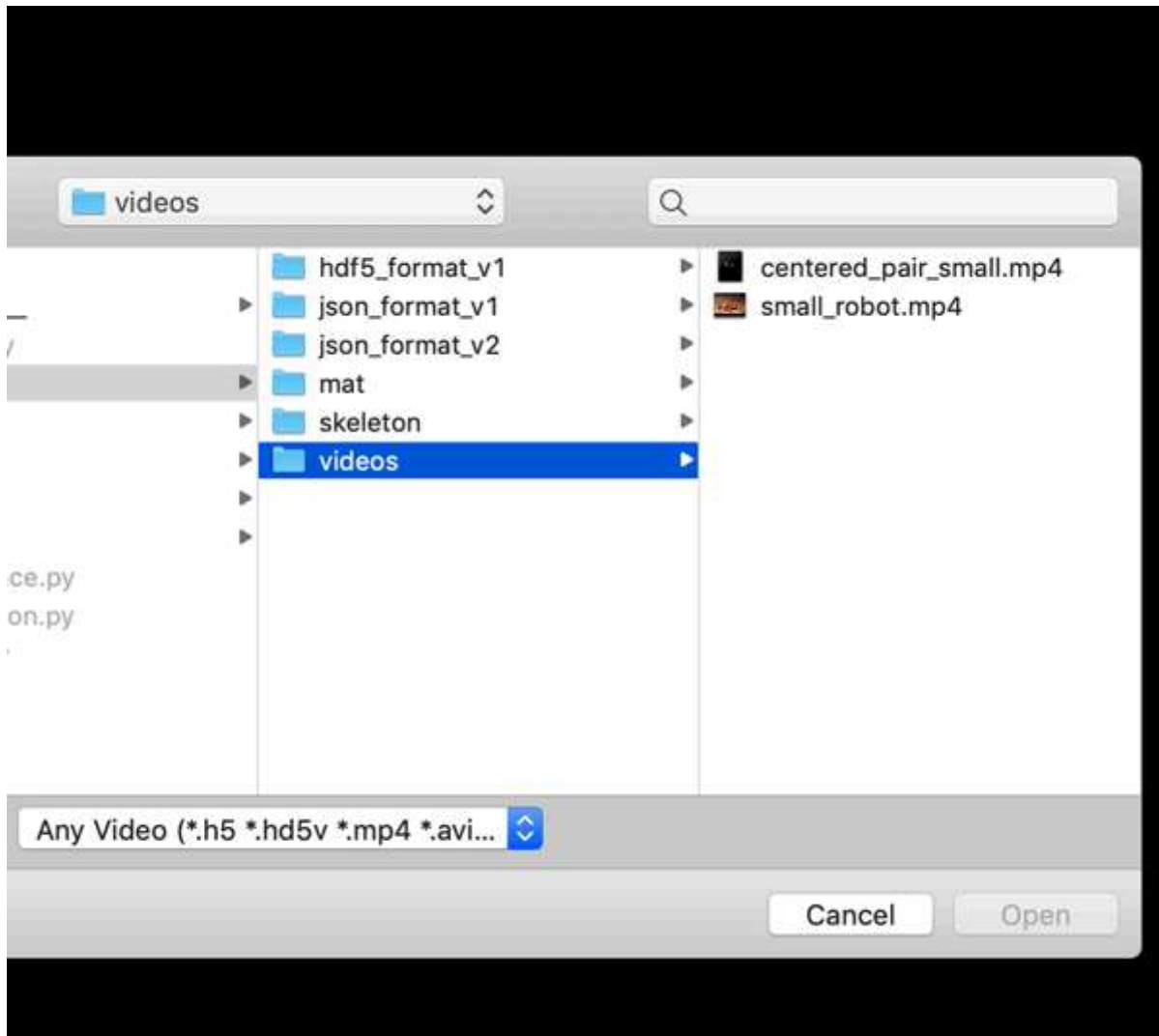
to Add edge Delete edge

Load Skeleton Save Skeleton

Videos Skeleton

You'll then be able to select one or more video files and click "**Open**". SLEAP currently supports mp4, avi, and h5 files. For mp4 and avi files, you'll be asked whether to import the video as grayscale. For h5 files, you'll be asked the dataset and whether the video is stored with channels first or last.

[Skip to main content](#)

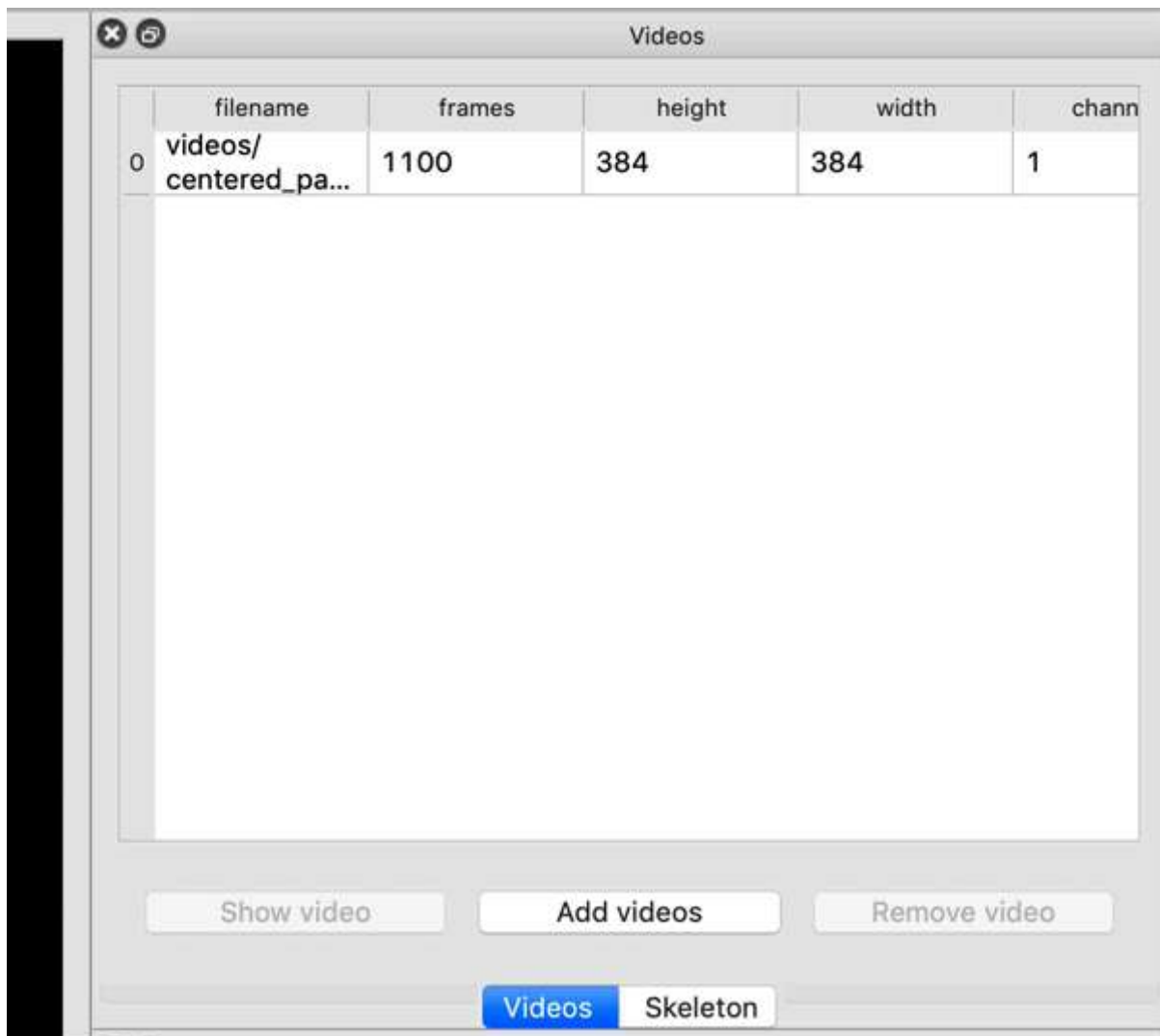


## Creating a Skeleton

Create a new **skeleton** using the “Skeleton” panel on the right side of the main window.

Use the “**New Node**” button to add a node (i.e., joint or body part). Double-click the node name to rename it (hit enter after you type the new name). Repeat until you have created all your nodes. You then need to connect the nodes with edges. Directly to the left of the “Add edge” button you’ll see two drop-down menus. Use these to select a pair of nodes, and then click “**Add Edge**”. Repeat until you’ve entered all the edges.

[Skip to main content](#)



Continue to Initial Labeling.

# Initial Labeling

## Contents

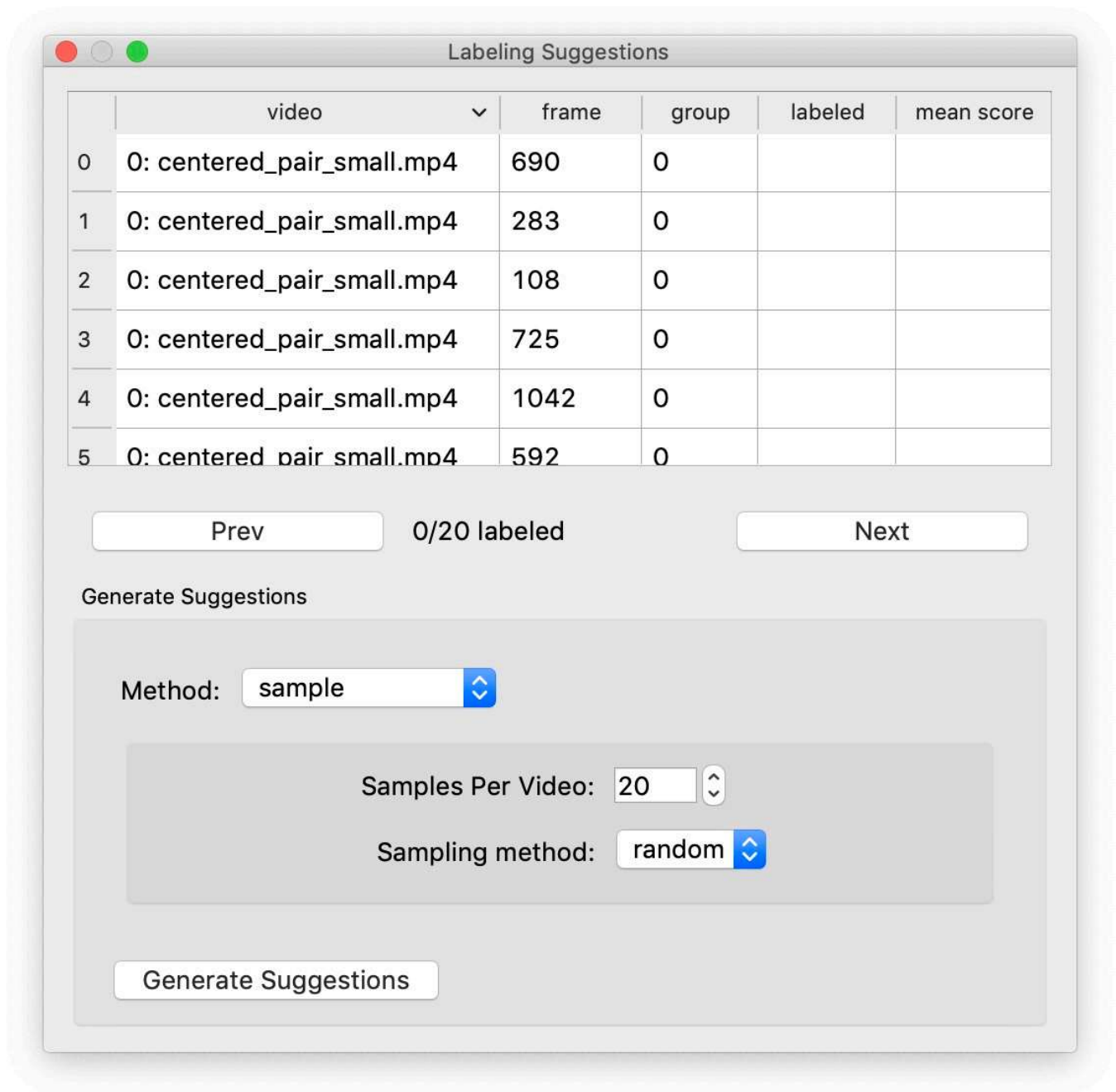
- Selecting frames to label
- Labeling the first frame
- Saving
- Labeling more frames

## Selecting frames to label

We start by assembling a candidate group of images to label. You can either pick your own frames or let the system suggest a set of frames using the “Labeling Suggestions” panel. SLEAP can give you random or evenly-spaced samples, or it can try to give you distinctive groups of frames by taking the image features into account.

For now, let’s just get 20 random frames. Choose “sample” as the method and “random” as the sampling method, then click “**Generate Suggestions**”.

[Skip to main content](#)



(See [Labeling Suggestions](#) for more information about the suggestion methods.)

You can then move between suggested frames by double-clicking the frame information in the table or by clicking the “Prev” or “Next” buttons.

## Labeling the first frame

Start by adding an **instance** of the skeleton to the current image by clicking the “**New Instance**” button in the Instances panel. The first instance will have its points located at random.

[Skip to main content](#)

Move the points to their appropriate positions by dragging with the mouse. Use the mouse scroll-wheel to **zoom**.

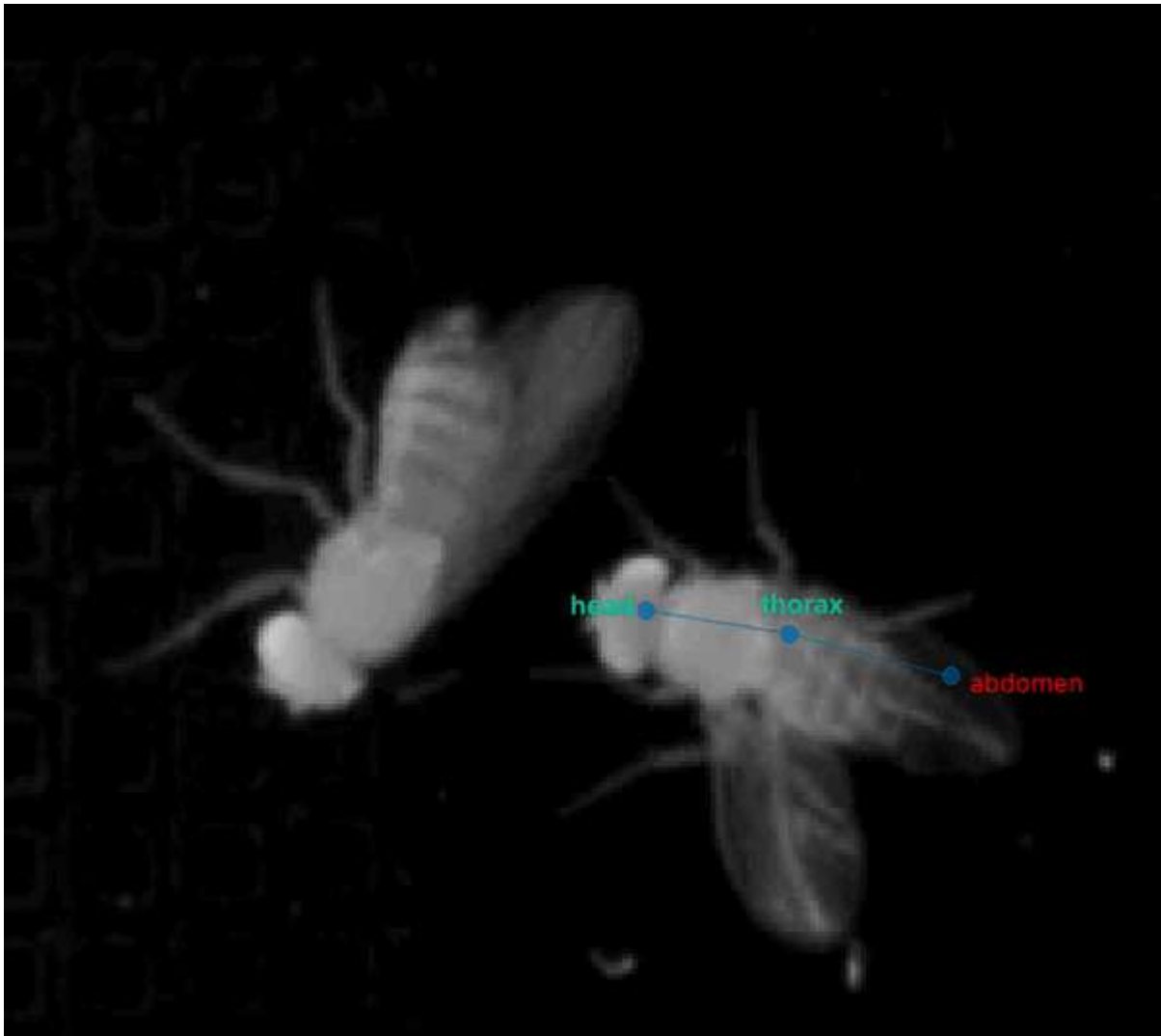


You can **move the entire instance** by holding down the Alt key (or Option on a Mac) which you then click on a node and drag the instance. You can **rotate the instance** by holding down the Alt key while you then click on a node and use the scroll-wheel.

For body parts that are not visible in the frame, right-click the node (or its name) to **toggle visibility**. The node will appear smaller to show that it's marked as "not visible". If you can determine where an occluded node would be in the image, you may label it as "visible" in order to train the model to predict the node even when it's occluded.

[Skip to main content](#)





When you label a frame, it's best if you can label **all** the instances of your animal in the frame. Otherwise, the models may learn to not identify things that look like the instances you didn't label.

## Saving

Since this is a new project, you'll need to select a location and name the first time you save. SLEAP will ask you to save before closing any project that has been changed to avoid losing any work. Note: There is not yet an "undo" feature built into SLEAP. If you want to make temporary changes to a project, use the "**Save As...**" command first to save a copy of your project.

## Labeling more frames

After labeling the first frame and saving the project, it's time to label more frames. Since you

[Skip to main content](#)

clicking **"Next"** under the list of suggested frames. You can also always pick a frame to label by using the seekbar under the video. Suggested and labeled frames will be shown as empty and filled blue bars, respectively.

When you add instances on subsequent frames, node positions will be copied from the instances in the prior labeled frame to increase labeling speed. Also try adding an instance by **right-clicking** on the location of the animal in the video. You'll see a pop-up menu with options for how we determine the initial node placement. Feel free to try the different options.

There's no need to be consistent about which animal you label with which instance for the case of multiple animals. For instance, suppose you have a male and a female. It's fine to label the male with the blue instance in some frames and the orange instance in others. Tracking (and track proofreading) is the final stage in the workflow and occurs after predicting body part locations.

Continue to [Training and Inference](#).

# Training and Inference

## Contents

- Training Options
- Start Training
- Inference

After labeling about 10 frames and saving the project you can train your first model and start getting initial predictions.

### Note

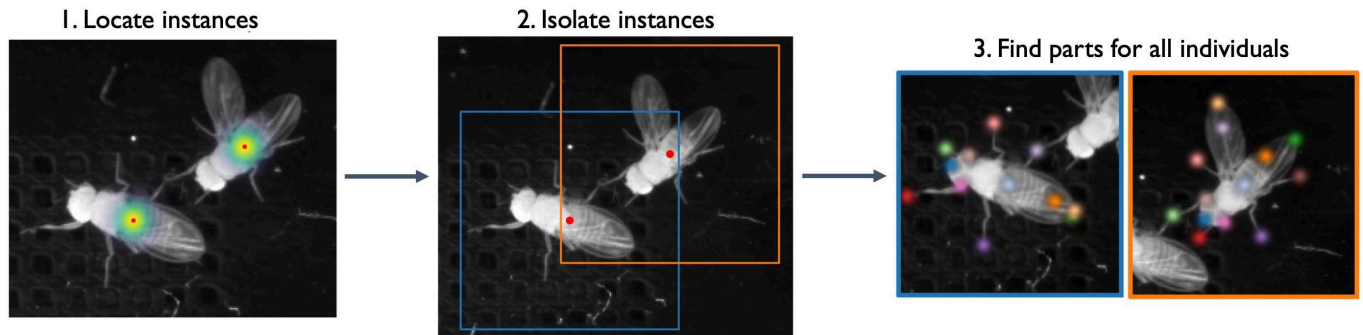
This tutorial assumes you have a GPU in your local machine and that TensorFlow is able to use your GPU. If you don't have a GPU or you're having trouble getting it to work, you can run training and inference in the cloud. See our [Run training and inference on Colab](#) guide! Or take a look at our other [Guides](#) about running SLEAP on remote machines.

## Training Options

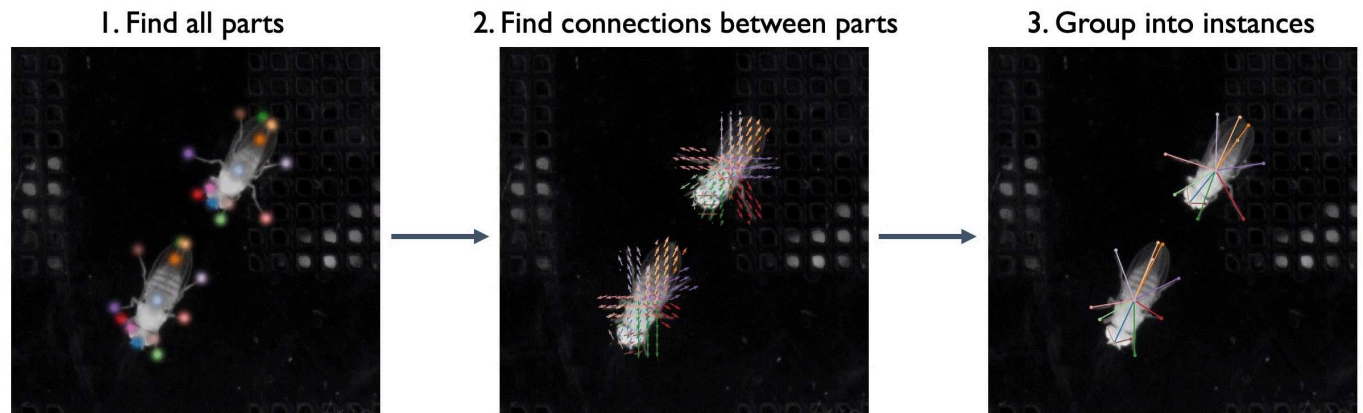
When you're ready to train you will have three choices for models: *single animal*, multi-animal **top-down**, or multi-animal **bottom-up**.

In **top-down** mode for multiple animals, a network first finds each animal and then a separate network estimates the pose of each found animal:

[Skip to main content](#)



In **bottom-up** mode, a network first finds all of the body parts in an image, and then another network groups them into animal instances using part affinity fields (Cao et al., 2017):



We find that top-down mode works better for some multi-animal datasets while bottom-up works better for others - to maximize the accuracy of predictions we recommend you to try both and see which one works best for your dataset.

In addition, SLEAP uses UNET as its default backbone, but you can choose other backbones (LEAP, resnet, etc.) from a drop down menu.

For more information about the types of models you can train, see [Configuring models](#).

## Start Training

To run training on your local machine, select **"Run Training..."** from the "Predict" menu. For this tutorial, let's use the default settings for training with the "top-down" pipeline and predict on 20 random frames.

[Skip to main content](#)

The screenshot shows a macOS-style window titled "Training Pipeline" with three tabs: "Training Pipeline" (selected), "Centroid Model Configuration", and "Centered Instance Model Configuration".

**Training Pipeline Type:** multi-animal top-down

**Multi-Animal Top-Down Pipeline:**  
This pipeline uses two models: a "centroid" model to locate and crop around each animal in the frame, and a "centered-instance confidence map" model for predicted node locations for each individual animal predicted by the centroid model.

**Sigma for Centroids:** 5.00

**Anchor Part:** thorax

**Sigma for Nodes:** 5.00

**Input Data Options**

**Convert Image To:** grayscale

**Output Options**

**Run Name Prefix:** topdown\_test

**Runs Folder:** models

**Tags:**

**Best Model:** ☒

**Latest Model:** ☐

**Final Model:** ☐

**Write Tensorboard Logs:** ☐

**Visualize Predictions During Training:** ☐

**Predict On:** random frames (20 total frames)

**View Training Image Inputs...**

**Buttons:** Save configuration files..., Cancel, Run

Recall that the "topdown" approach will train two models: one for locating each instance in the frame, and one for locating the parts for each of those instances. The models will be trained in that order.

When using the topdown approach, it's a good idea to choose an **anchor** part which has a

[Skip to main content](#)

option to “**Visualize Predictions During Training**” (although this will make training run a bit more slowly).

Once you hit the **Run** button, you should see a window which shows you a graph of training and validation loss for each model as it trains. Since the topdown approach trains two models, once you’re done training the centroid model the graph will reset to show you loss while training the centered instance model.

Just for this tutorial, let’s stop each training session after about 10 epochs. This should take a minute or two for each model (assuming you have a usable GPU!), and should be good enough to get some initial predictions.

## Inference

After each model is trained, inference will run and if everything is successful, you should get a dialog telling you how many frames got predictions. Frames with labels will be marked in the seekbar, so try clicking on the newly marked frames or use the “**Next Labeled Frame**” command in the “Go” menu to step through frames with labels.

Once you got the initial predictions from the trained models you enter into a human-in-the-loop training cycle, in which you receive predictions and use them to continue to label more examples. You will continue to iterate this process until you have achieved the desired pose labeling accuracy for your dataset. The goal at this stage is to get accurate pose predictions on individual frames. This means that the points for each body part are predicted in the correct locations and that the points are correctly grouped into distinct animal instances.

Continue to [Prediction-assisted labeling](#).

# Prediction-assisted labeling

## Contents

- Prediction-assisted labeling
- Reviewing and fixing predictions

*Prediction-assisted labeling* has two main goals. First, it speeds up the labeling process as it is faster to correct a predicted instance which is mostly correct than it is to add a new instance from scratch. Second, it provides feedback about where your model does well and where it does poorly, and this should give you a better idea of which frames will be most useful to label.

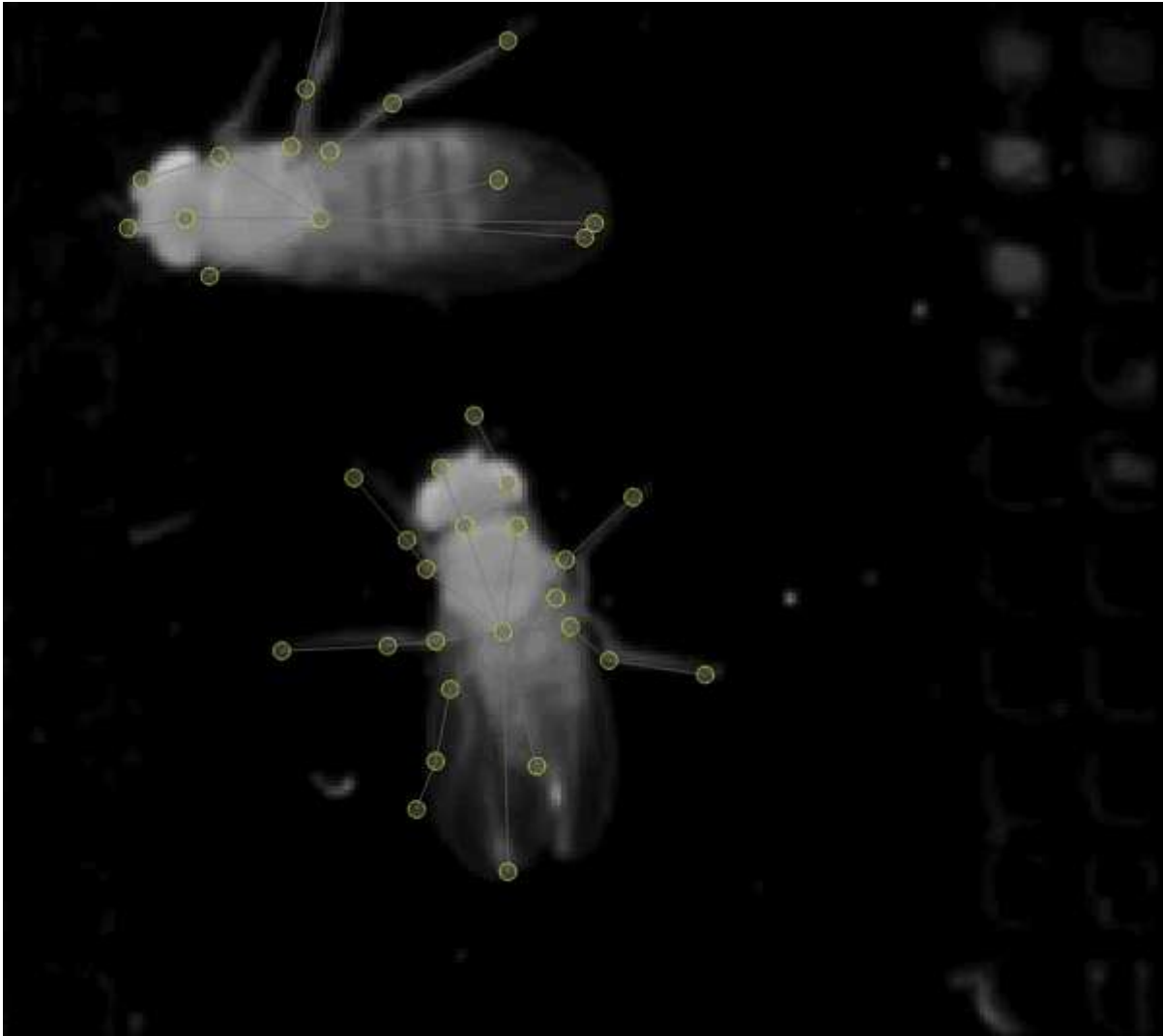
The GUI doesn't yet give you a way to monitor the progress during inference, although it will alert you if an error occurs during inference.

When inference finishes, you'll be told how many instances were predicted. Suggested frames with predicted instances will be marked in red on the seekbar.

After you've successfully trained models and predicted some instances, you'll get a message that inference has finished. Predictions will be marked with a thin black line on the seekbar, while frames that you manually labeled will be marked with a thicker black line. (For "suggested" frames, manually labeled frames will have a dark blue line and predicted frames will have a lighter blue.)

Predicted instances will *not* be used for future model training unless you correct the predictions in the GUI.

[Skip to main content](#)



Predicted instances in the frame are displayed in grey with yellow nodes. To edit a prediction, you'll need to replace it with an editable instance. **Double-click** the predicted instance and it will be converted into a regular instance.

#### Note

All node labels on the regular instance will be colored red. After moving nodes, the node labels will be colored green. This is just a visual indicator to keep track of which nodes have been moved from the original prediction.

You can now edit the instance as before. Once you've added and/or corrected more instances, you can repeat the process: train a new model, predict on more frames, correct those predictions, and so on. You'll want to regularly generate new frame suggestions, since active learning will return predictions for just these frames.

After you have accurate frame-by-frame prediction, you're ready to predict for entire video clips.

[Skip to main content](#)



across time (see [Tracking methods](#) for more details). SLEAP also includes a graphical proof-reading tool for quickly assessing the accuracy of tracking and correcting problems.

Continue to [Tracking instances across frames](#).

# Prediction-assisted labeling

## Contents

- Prediction-assisted labeling
- Reviewing and fixing predictions

*Prediction-assisted labeling* has two main goals. First, it speeds up the labeling process as it is faster to correct a predicted instance which is mostly correct than it is to add a new instance from scratch. Second, it provides feedback about where your model does well and where it does poorly, and this should give you a better idea of which frames will be most useful to label.

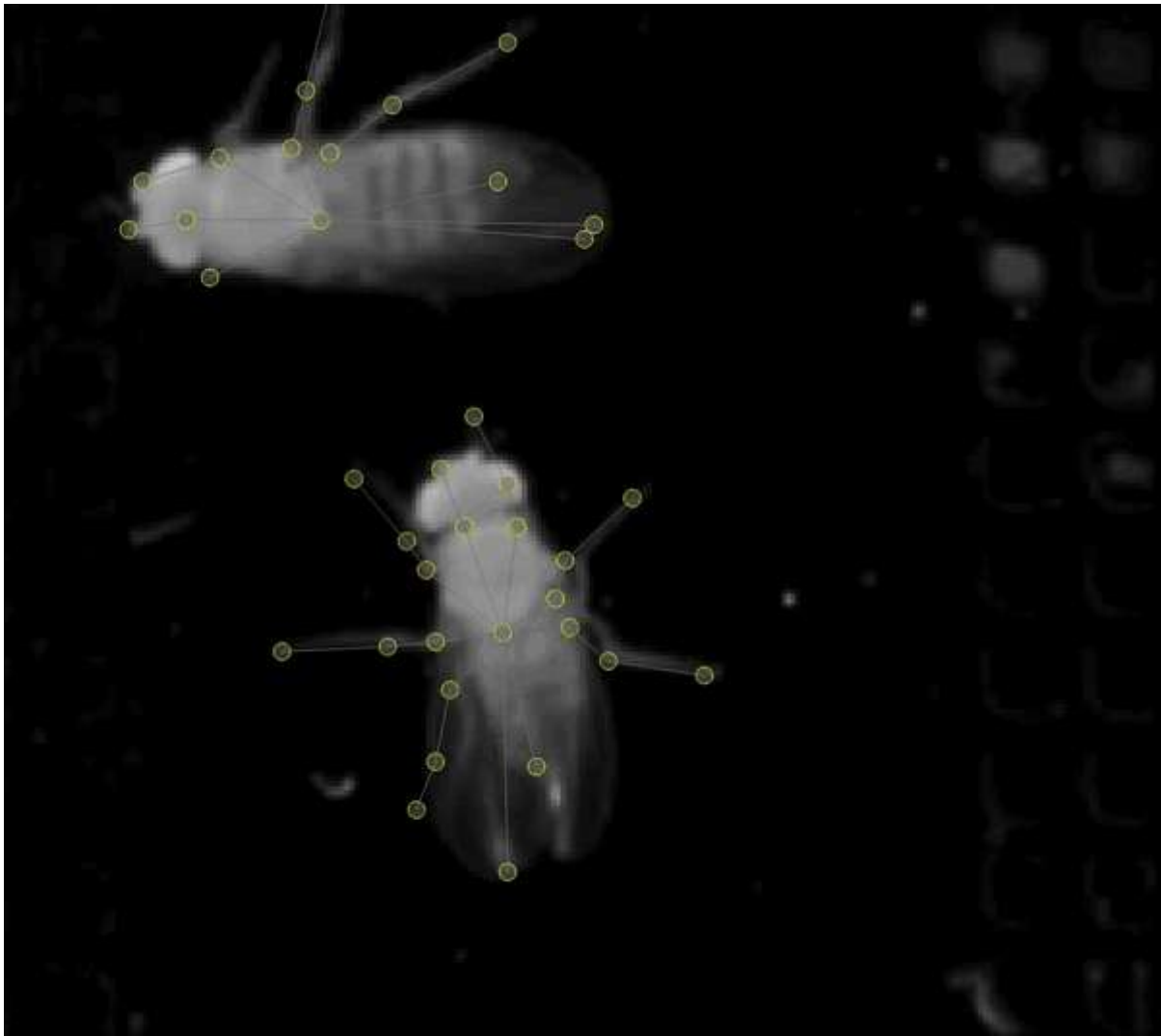
The GUI doesn't yet give you a way to monitor the progress during inference, although it will alert you if an error occurs during inference.

When inference finishes, you'll be told how many instances were predicted. Suggested frames with predicted instances will be marked in red on the seekbar.

After you've successfully trained models and predicted some instances, you'll get a message that inference has finished. Predictions will be marked with a thin black line on the seekbar, while frames that you manually labeled will be marked with a thicker black line. (For "suggested" frames, manually labeled frames will have a dark blue line and predicted frames will have a lighter blue.)

Predicted instances will *not* be used for future model training unless you correct the predictions in the GUI.

[Skip to main content](#)



Predicted instances in the frame are displayed in grey with yellow nodes. To edit a prediction, you'll need to replace it with an editable instance. **Double-click** the predicted instance and it will be converted into a regular instance.

#### Note

All node labels on the regular instance will be colored red. After moving nodes, the node labels will be colored green. This is just a visual indicator to keep track of which nodes have been moved from the original prediction.

You can now edit the instance as before. Once you've added and/or corrected more instances, you can repeat the process: train a new model, predict on more frames, correct those predictions, and so on. You'll want to regularly generate new frame suggestions, since active learning will return predictions for just these frames.

After you have accurate frame-by-frame prediction, you're ready to predict for entire video clips.

[Skip to main content](#)

across time (see [Tracking methods](#) for more details). SLEAP also includes a graphical proof-reading tool for quickly assessing the accuracy of tracking and correcting problems.

Continue to [Tracking instances across frames](#).

# Tracking instances across frames

## Contents

- Track proofreading

When you're satisfied with the predictions you're getting, you can use your models to predict on more frames by selecting "**Run Inference...**" from the "Predict" menu. This will use the most recently trained set of models.

The inference dialog is almost identical to the training dialog with a few key differences.

The inference dialog allows you to choose a method to use for tracking instance identities but also has a place to select the **tracker**.

[Skip to main content](#)

Tracker (cross-frame identity) Method: simple

**Pre-tracker data cleaning:**Target Number of Instances Per Frame: 1 ☒ No targetCull to Target Instance Count: ☐

Cull using IoU Threshold: 0.80

**Tracking:**

This tracker assigns track identities by matching instances from prior frames to instances on subsequent frames.

Similarity Method: iou

Matching Method: hungarian

Elapsed Frame Window: 5

**Kalman filter-based tracking:**

Uses the above tracking options to track instances for an initial number of frames, then initializes Kalman filters which are used to track instances on subsequent frames.

Enable Filters after Initial Frames: 10 ☒ Don't use filters

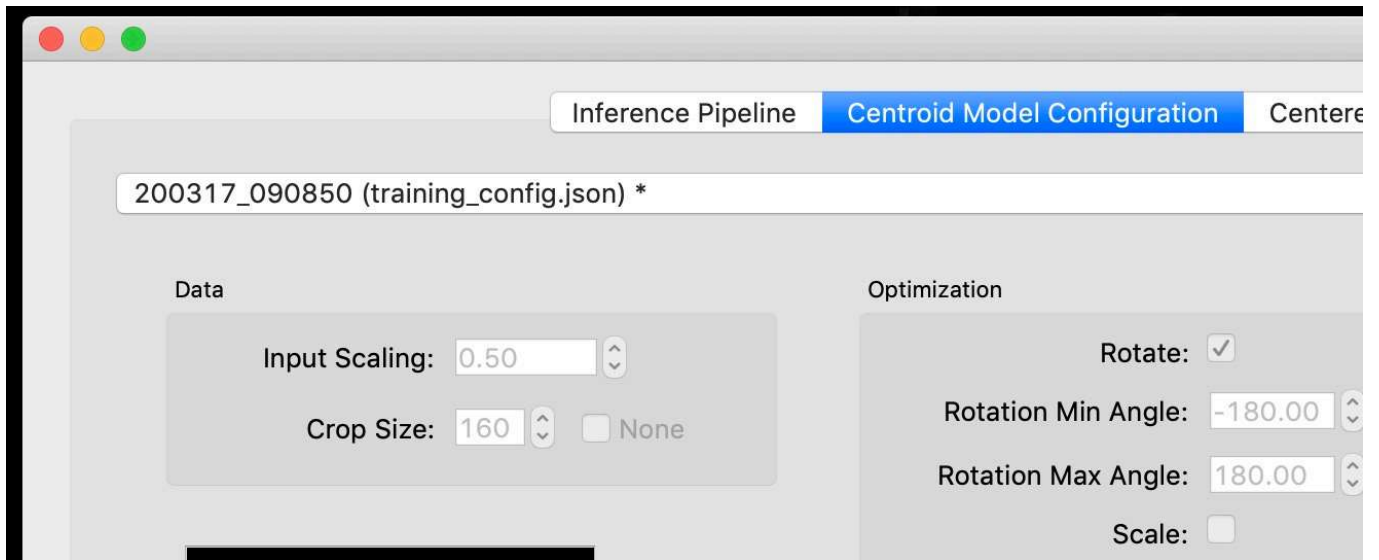
Nodes to use for Tracking: 0,1,2

**Post-tracker data cleaning:**Connect Single Track Breaks: ☐

See [Tracking methods](#) for more information about the tracker methods and options.

By default the inference dialog will use the most recently train model (or set of models), but if you want to choose another trained model, you can do this by using the dropdown menu on the tab for the relevant model type.

[Skip to main content](#)



## Track proofreading

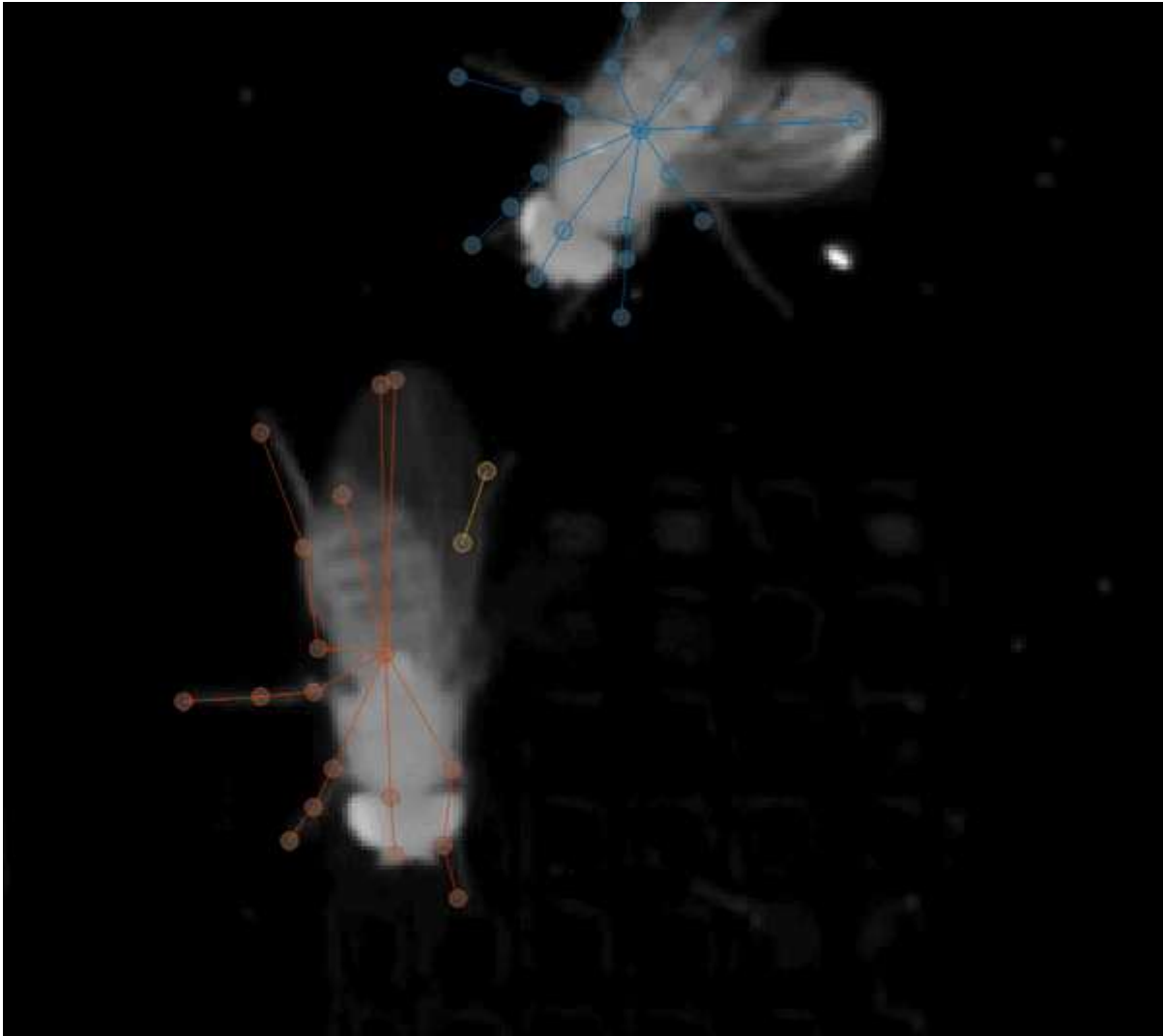
Once you have predicted tracks, you'll need to proofread these to ensure that the identities of instances across frames are correct. By default, predicted instances all appear in grey and yellow. Select "Color Predicted Instances" to show the tracks in color. (Note that colors in the frame match colors in the seekbar and colors in the "Instances" panel.) Click an instance to see it's track name. Double-click the track name in the "Instances" panel to change the name.

There are two main types of mistakes made by the tracking code: mistaken identities and lost identities.

**Mistaken Identities:** The code may misidentify which instance goes in which track, in which case you'll want to "swap" the identities.

You can swap the identities assigned to a pair of instances by selecting "Transpose Instance Tracks" in the "Labels" menu. If there are just two instances in the frame, it already knows what it do. If there are more, you'll have to click the two instances you want to swap.

[Skip to main content](#)



You can assign an instance to a different (or new) track from the “Set Instance Track” submenu in the “Labels” menu.

You can select instances by typing a number between 1 and 9, by clicking the instance in the frame, or by clicking the instance in the “Instances” panel (on the right side of your main window). When an instance is selected, you’ll see its track name. These track names can be edited by double-clicking the track name in the “Instances” panel.

When you assign an instance to a track, this change will also be applied to all *subsequent* frames. For instance, if you move an instance from track 3 to track 2, then any instance in track 3 in subsequent frames will also be moved to track 2. This lets you effectively “merge” tracks.

**Lost Identities:** The code may fail to identity an instance in one frame with any instances from previous frames. In this case, you’ll want to find the first frame in which the new track occurs and change the instance track to the track from previous frames. The “Next Track Spawn Frame” command in the “Labels” menu will take you to the next frame in which a new track is spawned

[Skip to main content](#)



For more tools and tips, see the [Tracking and proofreading](#) guide.

Once you're done proofreading, you'll want to [Export Data For Analysis](#).

# Export Data For Analysis

## Contents

- MATLAB
- Python

The easiest way to work with the data from SLEAP is to export an HDF5 file by choosing “**Export Analysis HDF5...**” in the “File” menu.

See [sleap.io.convert](#) for an explanation of the datasets inside the file.

## MATLAB

You can read this file in MATLAB like this:

```
occupancy_matrix = h5read('path/to/analysis.h5', '/track_occupancy')
tracks_matrix = h5read('path/to/analysis.h5', '/tracks')
```

See [here](#) for more information about working with HDF5 files in MATLAB.

## Python

To read the file in Python you'll first need to install the [h5py package](#). You can then read data from the file like this:

```
import h5py
with h5py.File('path/to/analysis.h5', 'r') as f:
    occupancy_matrix = f['track_occupancy'][:]
    tracks_matrix = f['tracks'][:]

print(occupancy_matrix.shape)
print(tracks_matrix.shape)
```

[Skip to main content](#)

**Note:** The datasets are stored column-major as expected by MATLAB. This means that if you're working with the file in Python you may want to first transpose the datasets so they match the shapes described in `sleap.io.convert`.