LabGym Practical "How To" Guide

Carrie Ferrario, Bing Ye and Yujia Hu

Last Updated: 10/10/2024

The below introduction is meant for use with LabGym version 2.5. It begins with a simplified description of what LabGym can do, followed by an overview of the steps for using LabGym and then gives a detailed description and guidelines for LabGym's use. We do our best to define terminology as it is introduced, and provide a list of terms and their definitions at the end of this document. How to install LabGym on your computer, example video datasets, an additional user guide with the technical details of LabGym and updated versions of this user guide can be found here: https://github.com/umyelab/LabGym

Part 1: A brief introduction to LabGym

LabGym identifies and quantifies user-defined behaviors. In general terms, you "show" LabGym video examples of a given behavior and from those examples it learns to identify that behavior. If you as an observer can tell the difference between two behaviors, then LabGym should be able to as well. It does not require any physical markings on your animals or on key aspects of the environment during initial video recording/acquisition. It will automatically generate a labeled video showing which behavior it thinks your animal is doing and how confident it is that the behavior is occurring (Fig 1). It will also automatically provide quantitative measures for each behavior. This includes, number of occurrences, latency, duration, and velocity, among others. LabGym automatically gives you a summary table with quantitative measures for each behavior of interest for each video and frame-by-frame quantitative information for each behavior of interest for each video (.csv or .xls files). It also automatically performs statistical analysis directly from the data files it generates. Moreover, LabGym automatically generates several visualizations of the analyzed data including color-coded raster plots showing when each behavior occurred across time for each animal (Fig 2). These outputs can be tailored to specific windows of time within your analyzed videos.

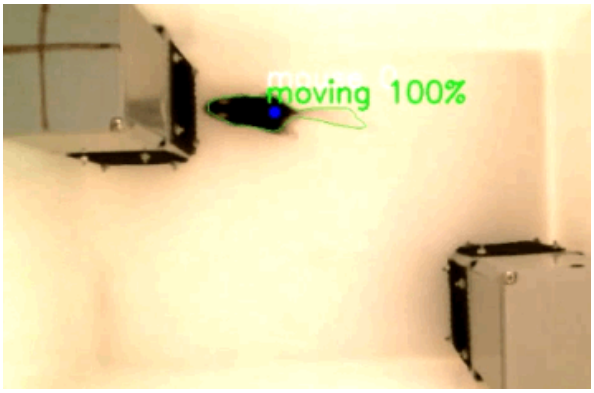


Fig 1: In the example at the left, when LabGym thinks the mouse is 'moving' (however the user defined that behavior), it outlines the mouse in a color that is unique to that behavior, and reports how confident is it that that specific behavior is occurring (in the example, it's 100% confident, and the mouse was in fact moving across the environment). This confidence is

continuously updated in each frame of the video. You can watch this video here:

https://raw.githubusercontent.com/umyelab/LabGym/master/Examples/Categorizer_animal_object_2.gif

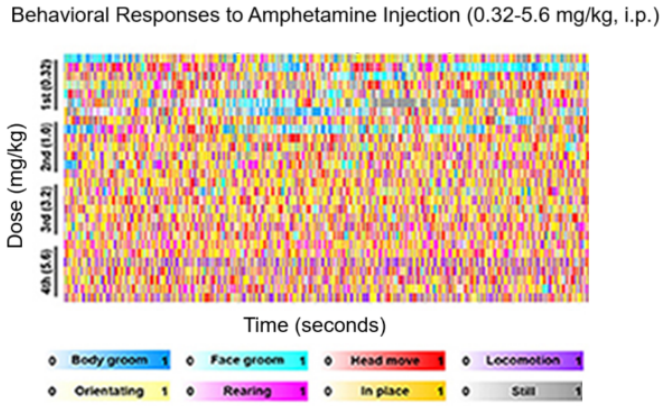


Fig 2: For each dose, one row corresponds to one animal. LabGym color codes each behavior as it occurs across time. Color intensity indicates LabGym's confidence in its categorization of the behavior (1=100%). This data is also automatically given in table form and applied to video outputs (adapted from Hu et al., 2023, PMID: 37056376).

LabGym can identify individual animals and their interactions with the environment, including other animals. To identify a behavior, LabGym detects your animal(s), distinguishes them from the background, and (if relevant to your studies) identifies when your animal is interacting with other things in the environment. These other things can be discrete locations, physical objects, or other animals. In cases where multiple animals are interacting, LabGym maintains the unique identities of each animal. It provides quantitative data for each animal in each video analyzed as described in Fig 2 above and in detail below (see Section IV). In Part 5 within this section we also show some examples of what LabGym can do (Fig 4 still images; video examples here:

https://drive.google.com/file/d/1-4Cy4yBxAaQSU3bH0kT1tuIYp7PbYa73/view?usp=sharing).

LabGym can be applied across datasets and laboratories: Video datasets and files needed (i.e., trained models) for animal(s) detection and behavioral quantification that have been established by one individual can be shared and used by others. We have created the "LabGym Zoo", a database of trained models and example videos in rats, mice, flies and monkeys that can be used by others working on the same behaviors and species. These can be found here:
https://github.com/umyelab/LabGym/blob/master/LabGym_Zoo.md

Part 2: Before you begin using LabGym

Before you start using LabGym with your own video data, you need to watch your videos and determine what behaviors you want to measure (i.e., operationally define each behavior) and pick a few videos to use to teach (i.e., “train”) LabGym to identify your behaviors of interest (see also below ‘Categorizing Behaviors’). For each behavior you want to identify (i.e., categorize), you will need ~100 distinct examples spread across 3-5 videos. If a given behavior is very regimented or stereotyped (i.e., largely looks the same across individuals) then you can use fewer examples and videos. If a given behavior is varied (like social behavior, which can look

different within and across individuals) then you will need more examples and videos. The bottom line is that for each behavior of interest you need to 'show' the computer examples that capture the full range of expression for that behavior (as you are defining it). It is best to define your behaviors of interest in your own mind first and choose training videos that show those behaviors before you begin to set up LabGym for your analysis.

One thing to keep in mind: When LabGym does the final analysis of a video it 'asks'- of the behaviors I was trained to identify, which is the animal doing right now? It will categorize what it sees into one of the behaviors you asked it to identify (and tell you how confident it is in that categorization). If it can't sufficiently differentiate the behaviors it's been asked to categorize it will report 'NA' for that period of time (see Section IV). So, let's say you want to measure time drinking from a sipper and walking around an enclosure. But your animal is also likely to spend time sitting still/sleeping, jumping, or digging. It is best to train LabGym to identify these behaviors as well; this will result in higher accuracy. It is easy to update and revise what you teach LabGym. So, we advise starting off with your behaviors of interest, and a few additional behaviors that either occur often or that you think might be easily confused with your behavior of interest, and then refine from there. Another strategy is to have an "extraneous" behavior type which can be a catch all for things your animal(s) may do often, but you don't really want to measure. If you try this approach, you should not combine behaviors that look very different from each other, as this can also result in poor training.

Importantly, in the final analysis step (Section IV), you will set a threshold for the Categorizer to report that the two most likely categories are too similar in probability to make a determination. In this case, the Categorizer assignment will read as 'NA' (no category) on the visualizations and in the data file.

Once you have decided what behaviors to you want LabGym to "categorize" and you've selected a few videos to work from, you are ready to begin using LabGym on your videos.

Part 3: Using LabGym

In the next sections we give a description and overview of the processes that you will complete in LabGym, what you can use them for, and provide some guidance about how to choose which processes to use given your analysis goals (Parts 3 & 4). This is followed by some examples from our own work (Part 5) and a detailed step-by-step procedures and guidelines for using LabGym (Part 6). You can combine LabGym's features in sophisticated ways; examples below are meant as a starting point.

We do our best to define terminology as it is introduced and provide a list of terms and their definitions at the end of this document (see also FAQ section). We will begin by learning a little

bit about what LabGym is doing when it's analyzing videos and how to think about your data in relation to setting up LabGym (i.e., teaching or "training" it) vs using LabGym (i.e., "analyzing" videos). Ultimately, LabGym will identify animal(s) and objects/places in the environment (if needed), categorizing behaviors based on the examples you gave it, and quantify those behaviors. To do this you must teach it to detect your animal(s), detect objects/places (if needed), and to identify distinct behaviors.

Part 3.1: Detecting animal(s)

As mentioned above, to identify a given behavior, LabGym must distinguish your animal(s) from the background. This can either be done by using a simple "Background Subtraction" approach (for cases where the background/illumination in videos is static, or where you want to analyze interactive behaviors but don't need to maintain the identity of individual animals), or by "Training a Detector" (for cases where the background/illumination in videos changes across time, or when you want to maintain the identities of individual animals, e.g., for social interactions). *Note*, *training a Detector is necessary if you want to quantify interactions between animals and maintain their unique identities (e.g., social behaviors).*

When should I use Background Subtraction vs a "Detector" to detect my animal(s)?

If the background is static and the illumination is stable across a given video, the background subtraction method is likely to work well. However, if the contrast between your animal and the background is low, or your background varies a lot within one video, then training a Detector may work better than simple background subtraction to identify your animal. It is not hard to try out these different approaches and we will go through examples of each approach below (see also Fig 4 below).

Part 3.2: Identifying behaviors

We also need to teach LabGym to identify (i.e., categorize) behaviors. This is done through a separate process called "Categorizer Training" that is described in detail below. Essentially, you will 'show' LabGym examples of each behavior (through the 'generate behavior examples' process). This will produce a "model" that LabGym will then apply to your videos in the analysis step.

Part 3.3: Identifying behaviors that rely on the 'background'

If a behavior relies on the animal interacting with an object (including other animals) or place in the environment or "background" of your videos (e.g., sniffing a specific object, pressing a lever, spending time in a specific place) or is conditional on the presence of something in the environment (e.g., cue light on), then you will need LabGym to know about objects in the environment when you "Train the Categorizer". This can be achieved in two ways: one where you digitally mark where the object is using LabGym's 'Draw Markers' function within the

"Preprocessing" module (Part 4.6 and Section I below), and another where you train LabGym to identify the object (this is done using the "training a Detector" process, essentially the Detector will detect your objects in addition to detecting your animal).

The Draw Markers function is good for objects in a fixed position in the environment and that cannot be easily distinguished from the background (e.g. a specific corner of an enclosure, two identical-looking arms of a Y maze). Importantly, you only need to draw markers on the first frame of one video; LabGym will then apply those markings to all frames of all videos you select. The Markers will essentially become part of a behavior of interest when you train the Categorizer (i.e., it will learn that walking in this arm of the maze is one type of behavior and walking in the other arm is a second type of behavior). The "training a Detector" approach is useful for objects that move (e.g., other animals, objects an animal can pick up) or that are stationary and can be easily distinguished from the background environment (e.g., a novel object, recessed food cup, or distinctly colored place within the environment).

In general, the method you choose to detect *your animal* (Part 3.1) does not affect how you train a "Categorizer". That is, whether you use the background subtraction or training a detector approach to identify your *animal(s),* you can still choose to include (or not) the background information when you train a Categorizer (depending on whether the background information is useful for identifying a behavior of interest). It can affect some of the choices you make when completing Categorizer training, we go through this in detail in Section III.

Part 3.4: Trained Categorizers and Detectors can work independently

Because the Categorizer and Detector are trained separately, you can also revise them independently (though the steps for training both are located within the "Training Module" of LabGym). You can also use an existing trained categorizer to analyze behavior in a new environment (if the behavior of interest is independent of the testing environment). This also means you can apply Categorizers trained by others to your own videos, or even modify an existing Categorizer (by adding behavior examples or behavior categories and re-training the Categorizer) to suit your needs (see also LabGym Zoo).

You can mix and match trained Categorizers and Detectors to analyze your videos. For example, if you have a trained Categorizer with good accuracy and want to use it to analyze some new videos, but the trained Detector doesn't work well for the new videos, you can train a new Detector for new videos but use the old Categorizer. And vice versa. However, the Detector is less generalizable than the Categorizer, so sometimes you need to train new Detectors for new videos. *Note you can also train a Detector and run the "Analyze Behavior" function in LabGym "Analysis Module" to simply track and quantify the movement of your animal(s).*

Training a Detector and using a Detector for analysis are the most computationally demanding processes (i.e., requires the most 'computing power') in LabGym. They can be completed using

reasonably powerful computers without a GPU (Graphics Processing Unit, i.e., graphics card/unit like those used for online gaming), but will be completed more quickly on computers equipped with a GPU. You can reduce the processing time required by adjusting your videos using the Preprocessing Module in LabGym (Part 4.6 and Section I). Some guidelines regarding hardware are given at the end of this document.

Part 3.5: Selecting videos to train LabGym

You will select a subset of your videos to use to train the Detector (if needed) and Categorizer. Most often you can use the same videos to train the Detector and Categorizer, though this is not required. The videos can be a subset of the videos from your experimental dataset, or they can be a different set of videos entirely; as long as they are representative of the testing environment and the behaviors you want to quantify. Importantly, for training the Detector, you want to use videos that show the full range of lighting and animal visibility. Similarly, to train the Categorizer you want to choose examples that show the full range of each behavior of interest. Importantly, to get high accuracy categorization, you should also include examples of other behaviors that occur often (or are similar but distinct from your behavior of interest) even if you do not 'need' to quantify them. For simplicity, you can try grouping all behaviors that are not of interest into one big category, like 'extraneous' or 'other'. However, we do not advise grouping behaviors that look very different into the same category, as this is likely to reduce the overall accuracy of the Categorizer (e.g., if you want to quantify running on a wheel, but not running/walking around the cage, rearing or grooming, it'd be best to train a Categorizer with categories of: wheel running, walking/running, rearing, grooming).

For many behaviors, LabGym can generalize across different environments and camera angles (especially for behaviors that occur independently of the testing environment or objects in the environment). It is best practice to use one set of videos to train LabGym and a different set of videos to test LabGym's accuracy (these can be a subset of all your experimental videos or a different set of similar videos). We will go through how to test the accuracy of LabGym using both approaches below in the "Test Detector" and "Testing the Categorizer" sections.

Part 3.6: Important note; how to think about using LabGym

One final note about how to think about using LabGym before we move on. We say it is *categorizing* behavior, rather than identifying behavior, because fundamentally when LabGym analyzes a video it 'asks'- of the behaviors I was trained to identify, which of them is the animal(s) doing right now? It does this frame-by-frame and categorizes the behavior that an animal performs in each frame into one of the behaviors you defined. Thankfully, it also reports how confident it is that each frame belongs in the category it assigned (Fig 1), and allows you to set a threshold for reporting that it can not make a reasonable distinction (Section IV). So, for high accuracy, it's ideal to teach LabGym about the most often occurring behaviors in your

videos, even if you are not particularly interested in quantifying some of those behaviors (see also Section III B for details about Categorization and how that influences the way you train LabGym). We will come back to this idea in several sections below.

Importantly, in the final analysis step (Section IV), you will set a threshold for the Categorizer to report that the two most likely categories are too similar in probability to make a determination. In this case, the Categorizer assignment will read as 'NA' (no category) on the visualizations and in the data file.

Part 4: Overview of processes in LabGym

Part 4.1: Automated detection of the animals(s) [and objects in the environment if needed; e.g., left corner, food cup, cue light, lever].

This involves generating static example images from your videos, labeling the animal(s) in those images, and then using the labeled images to train a LabGym "Detector" to identify your animal(s) in the environment. You accomplish this in the "Training Module" of LabGym; the labeling processes uses an external resource called "RoboFlow", which makes labeling (i.e., annotating) the images very fast and easy. If you want LabGym to identify an object in the environment on its own (like a cue light on, nesting material the animal can move, or a stationary object that is distinct from the background), you also do this using RoboFlow. You then use the labeled static images to train a Detector so LabGym can automatically detect the animal(s) [and objects in the video]. This whole process is called "Training a Detector". Not all analyses will require you to train a Detector (see Part 3.1 above and Part 6 below).

Part 4.2: Automated categorization of behaviors of interest

This involves generating and sorting examples of each behavior from your videos, and then using these examples to train LabGym to detect your behaviors of interest. This process is called "Training the Categorizer" and is included in the "Training Module". Once you generate behavior examples, you will sort them either within LabGym using automated keystroke functions, or outside of LabGym using a 'drag and drop' approach. We go through each in detail below (Section III).

If you want to measure behaviors based on interactions with specific fixed position(s) in space within the environment (e.g., a food bowl, specific corner, cue light on, lever…), then you have three options to choose from which we describe below. *Note, if using one of these options, they must be implemented **before** generating any example images to train the Categorizer*:

1) Define places or objects in the "Preprocessing Module" using the "Draw Markers" button (see #5 below).
2) Train a Detector to recognize these places (this will be done at the same time as you

train the Detector to detect your animal(s).

3) This approach combines 1 & 2. Use the 'Draw Markers' function to digitally define places/objects and then train a Detector to recognize the markers you placed (essentially the markers allow the computer to 'see' what you see).

Option 2 is most useful when the illumination in your videos is unstable but the object/place is easy to distinguish from other environmental features, while option 3 is most useful when the illumination in your videos is unstable and the object/place you want to detect are not easy for the computer to recognize from the video information alone (e.g., the object/place looks very similar to the background, or you want to distinguish several similar-looking objects).

You should choose one of the approaches above before starting the Categorizer training steps below (Section II). There is more than one way to accomplish the same goal within LabGym, so the examples here and in the detailed procedure below are not exhaustive but are meant to help you learn how to think about using machine-learning based approaches to enhance your analysis. See also examples of when to use different approaches within LabGym (Part 5 and Fig 4).

Part 4.3: Testing the accuracy of your Detector and Categorizer

The testing modules within LabGym will help you evaluate how accurately it is detecting your animal(s) and their environment, and how accurately LabGym is identifying your behaviors of interest. In the details below, we provide guidelines for refining detection or categorization. This is often an iterative process, and the degree of accuracy required is ultimately up to the user. For example, 50% accuracy (a coin flip) is unacceptable, but 80-90% accuracy will likely have enough sensitivity to detect experimental effects. Additionally, performance will not vary across individual videos or datasets; once you have trained a Detector and/or Categorizer in your conditions, it will work the same way across all videos. The Detector should only need to be retrained if you make substantial changes to the testing environment, animal subject, or video conditions. In general, Categorizers generalize better across conditions than Detectors and should only need to be retrained if you change the way you define a behavior. Even in these cases, you can easily modify an existing behavior example dataset (e.g., by adding more categories or expanding the examples given) to retrain a Categorizer.

Part 4.4: Analyze videos

When you move to the Analysis module, it will ask you to specify the method to detect animals or objects (you already made this choice when you trained the Categorizer, here you are just applying the choices). As described above, there are two methods through which LabGym distinguishes the animal(s) from its environment (i.e., detection). The first is based on a simple background subtraction method. This is used when a) your behaviors of interest occur in an environment (i.e., background) that does not change across time, b) the illumination/lighting in

your video is stable, c) you only want to make measures from single animals (i.e., one animal in an enclosure), or d) you want to measure behavior (either non-social or social) from multiple animals but you don't need to maintain the identity consistency all the time (e.g., Example C below). The other method is based on a machine-learning approach (i.e., using a Detector) and is used when a) your behaviors of interest occur in an environment that changes across time (e.g., animals building a nest), b) illumination/lighting in your video varies (e.g., due to recording conditions, or deliberate changes in illumination) or c) you want to measure behaviors from multiple animals within the same video and want to maintain their unique identities. Within the LabGym Analysis module you will choose which method to use. In the procedure below we will explain each method in detail separately.

Part 4.5: Modifying your videos

LabGym contains a "Preprocessing Module" where you can modify your videos in several ways including: cropping the frame, trimming the video length, enhancing video contrast (Fig 3, middle panel), brightness, reducing frame rate or frame size. These modifications are optional. Videos are often large files. Reducing the frame rate or frame size will reduce the size of your file, which will also reduce the amount of processing power you need to work with the files. Most videos are 15 or 30 frames per second (fps) and in general, you do not need a high frame rate to get high quality training or accurate analysis from LabGym. *Note, in the analysis module, LabGym will analyze every single frame of your video, but you do not need to use every single frame during the initial training of the Categorizer or Detector.*

The frame size must be the same for all videos to be analyzed and used to train the Detector and Categorizer. You can batch process all your videos through the Preprocessor, or you can process only those that are needed to train the Categorizer and Detector either within Preprocessing module or within the training modules (e.g., if you have not yet collected all your experimental video data but have enough to start training LabGym). In the detailed procedure below, we provide some additional guidelines and rules of thumb for making decisions related to frame rate and frame re-sizing.

Important notes about altering your videos:

i) If you use the Preprocessing Module to reduce frame size or frame rate of your videos, you must make the same adjustment to all videos within one experiment (i.e., any data that will be directly compared), including videos used to train the Categorizer and Detector.

ii) If you use the Preprocessing Module to reduce frame rate you should not adjust it again in any other module (e.g., Training Detector). Any additional changes you make within the other modules will be additive with frame rate changes made during preprocessing.

Part 4.6: Marking your videos with the Draw Markers function

You can use the “Draw Markers Function” in the Preprocessing Module to define fixed positions in space within your videos that are relevant to a behavior of interest (Fig 3, rightmost panel). You need to do this before training a Categorizer, if you want to use this feature. Importantly, you only need to draw markers on the first frame of one video; LabGym will then apply those markings to all frames of all videos you select (See example in Section I below, and information in Parts 3.3 and 4.2 above). *Note, when you use the Draw Markers function, LabGym will automatically open the first frame of your video and you will make the marks on that frame. Therefore, you must choose a video where the objects you want to mark are visible to you in the first frame (or you can trim the video appropriately).*

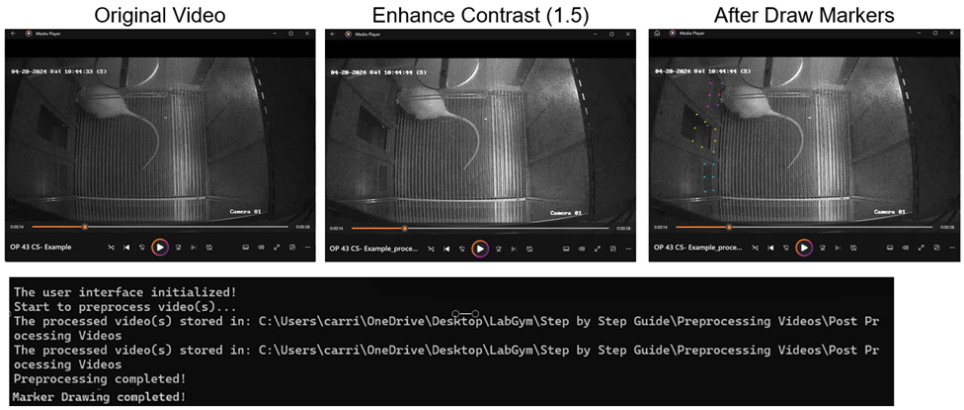


Figure 3: Example still image before (left) and after (middle) enhancing the contrast (1.5X), and after using the Draw Markers feature (right most panel) in the Preprocessing module. These changes will produce new video files for each video that will be stored in a folder you designate. The storage location and when the process is complete will be shown in the command/terminal window (bottom panel).

Part 5: Some examples of how we used LabGym for different analysis goals

Below we describe how we used LabGym to quantify different types of behaviors in different settings (videos corresponding to the still frames below can be found here: https://github.com/umyelab/LabGym There is more than one way to achieve the same analysis goal using LabGym; examples below are not exhaustive. *Note, the modes for Categorizer Training mentioned in the captions below are explained in detail in Section III A, step 3.*

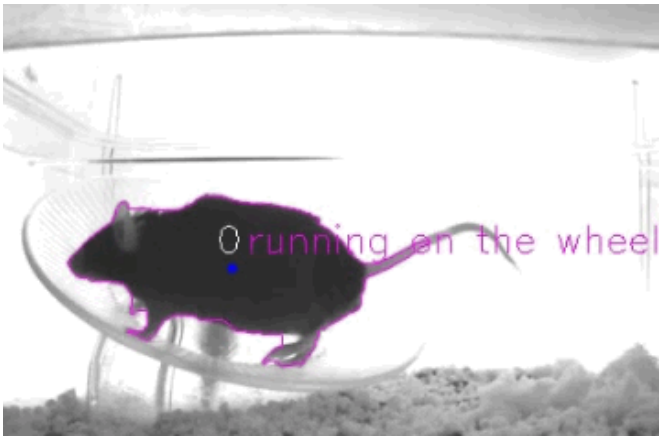


Fig 4A) To detect the mouse running on a

wheel in its home cage, we used simple background subtraction and trained a Categorizer using the "non-interactive" mode. The background is stable, and there is good contrast between the mouse and the background, so the simple subtraction approach worked well here to detect the mouse. Wheel running is very similar across animals, so relatively few examples were needed (3 videos, ~ 50 instances of running).

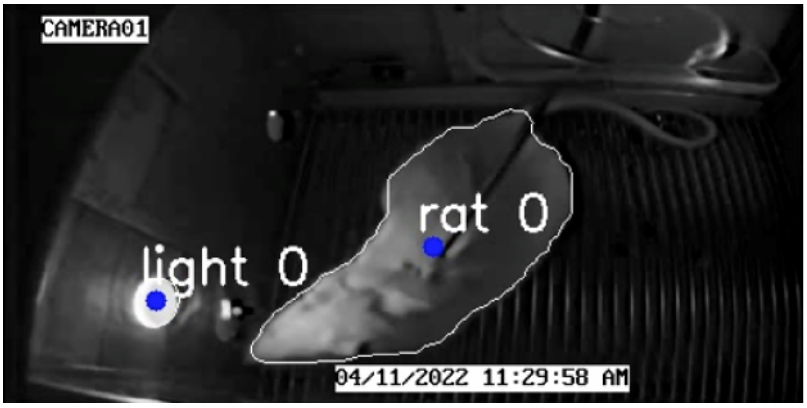
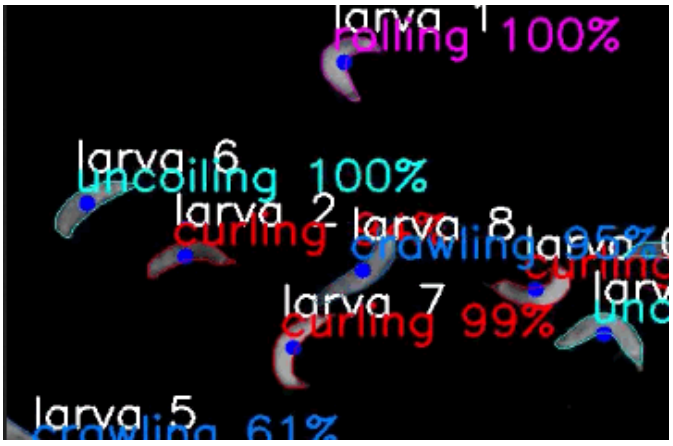


Fig 4B) For cases like the one shown to the left, background subtraction approaches did not work well. The contrast between the rat and the background is not great (especially in corners), and the cue light (lower left corner) casts shadows that change illumination over time when it goes on and off; so even if there were enough contrast, the lighting conditions are too varied to across time for the background subtraction approach to work well. Therefore, we needed to train a Detector to distinguish the rat from the background. This worked well. For this study, we cared about the animal's behavior in relation to when the cue-light was on, so we also trained the Detector to detect the light when it was on. It took 3 video examples to accurately train the Detector. We then used the "interactive-advanced" mode when training the Categorizer.

Fig 4C) Here we wanted to detect behaviors in multiple animals (fly larvae) at the same time. We did not want to measure any interactions between individual animals or their environment, but we wanted LabGym to keep track of the unique identify of each individual larva over time. So, we trained a Detector to recognize 'larva' and during the annotation in Roboflow we made sure to segment two (or more) larvae whenever they overlapped or collided in an image. This enables the Detector to identify unique individuals, even when they collide (this requires many annotated images of colliding animals, at least 100). We then trained a Categorizer to recognize our behaviors of interest (curling, crawling, rolling) using the "non-interactive" mode. It took about 3 videos to train the Categorizer.



D) Here we wanted to measure the rat's interactions with fixed objects (a recessed food trough, and two lever positions one on either side of the trough) in the environment (i.e., background). To do this, we used the "Draw Markers" approach to define where the food trough (red dots) and lever positions (blue and green) are in our background before training a Categorizer. We chose this approach because the two levers are similar in appearance and can 'come and go' in different videos (in some videos there are no levers present, but we still want to measure how the rat interacts with the spot where the levers usually appear). To detect the animal itself, a background subtraction approach could work, but there are a lot of shadows and variations in light, so we found that training a Detector worked better. We used the "non-interactive" mode when we trained a Categorizer and include background in behavior examples so that LabGym can "see" these markers distinguish when the rat is near them or not. In this case, when we sorted behavior examples into categories, we had a category for each behavior we wanted to measure in relation to each object (e.g., sniffing the 'lever' marked in blue), and a contrasting category for when the rat is not interacting with each object (e.g., sniffing the air above the blue mark; the goal is to 'show' it sniffing we define as not directed at the blue mark. These examples do not have to be exhaustive).

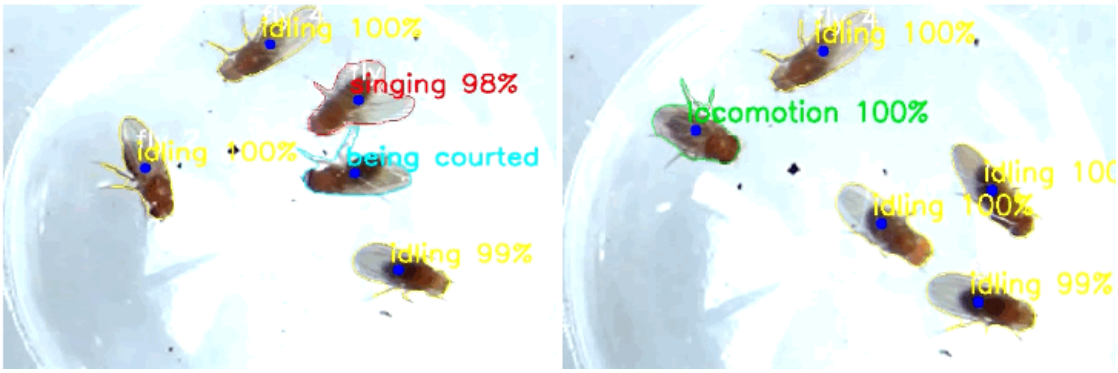


Fig 4E) Above we wanted to measure social interactions between unique individuals (fruit flies). Even though the conditions would allow for the background subtraction approach to detect the animals, in order to keep the identities of individuals unique, we must train a Detector and cannot use a background subtraction approach. For the specific study, we trained a Detector for each set of flies within one dish to maintain identities; this is actually faster and easier than it sounds. We used the "interactive-advanced" mode when we trained a Categorizer.

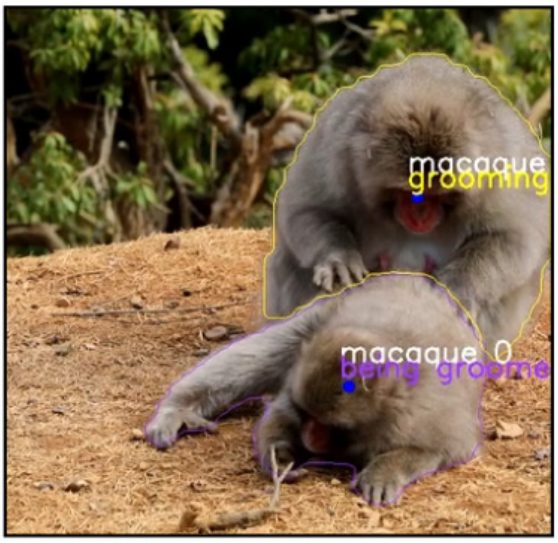


Fig 4F) At the right, we wanted to quantify social interactions between two monkeys in a field. There is extensive body contact between the two animals, and we wanted to know which animal was being groomed and which was doing the grooming. So, we trained a Detector to keep the identity of each individual unique and used the "interactive-advanced" mode when we trained a Categorizer. Background subtraction would not work in this case.

Part 5.1: Some additional descriptions and suggested approaches to get you started

If you simply want to quantify behaviors of a single animal and the background is not relevant to your behaviors of interest, then you can use the background subtraction approach to detect your animal and train a Categorizer without using the backgrounds to generate behavior examples. In this case, you would skip 'Training a Detector" because you don't need one. If the simple background subtraction approach doesn't work well for your conditions (this is sometimes the case when there is low contrast between your animal and the background, or when the lighting varies a lot within a video), you can instead train a Detector to detect your animal(s). You can still train the Categorizer and choose not to use the background information when you generate your behavior examples (i.e., if behaviors are not related to the background environment).

If you want to quantify behaviors of a single animal, and you also want to measure some of these behaviors in relation to a fixed object or place in the testing environment, you can first use the 'Draw Marker' feature to digitally mark those objects/places, before training the Categorizer. Importantly, you only need to draw markers on the first frame of one video; LabGym will then apply those markings to all frames of all videos you select. In this case, you can still use the simple background subtraction approach in LabGym to detect your animal. When you train the Categorizer, you will include the background information in your behavior examples, and you will make a distinct behavior category for the behavior that is contingent upon the marker

position.

If you want to quantify behavior of your animal interacting with other animals, or with objects in the environment that can move, then you will want to train a Detector to automatically identify your animal(s), and these objects. You will then use this Detector when you "Generate Behavior Examples" that are needed to train the Categorizer, and again to analyze videos in the "Analysis Module".

Part 6: Detailed procedures and step-by-step instructions

Below we provide step-by-step instructions with images and some guidelines for training a Detector and the Categorizer, followed by how to conduct Analysis and work with the outputs generated by LabGym. These sections assume you've read Part 3 and 4 and understand the terminology and organization of LabGym described in those sections.

Important:

1) Be sure to write down the frame size you use if/when you choose to resize your video(s) in the processes below and use the same frame size across all your videos. If you adjusted the frame rate in the Preprocessing module, any adjustments made below will be additive (don't do that).
2) You will generate a lot of folders and files; so, do your best to develop naming conventions and strategies to keep them organized (see example at end of doc). Importantly, whenever LabGym names files for you (or creates new files from existing ones) it keeps the file's original name and adds a descriptor to the end. It never overwrites your original files.
3) It is often best to start by working with a small subset of your videos (3-5 videos) using the simplest approach possible, and then adjust as you see how LabGym is tracking animals/objects and categorizing behaviors.

If a) your behaviors of interest occur in an environment (i.e., background) that does not change across time, b) the illumination/lighting in your video is stable, c) you only want to make measures from single animals (i.e., one animal in an enclosure), or d) you want to measure behavior (non-social or social) from multiple animals but you don't need to accurately track their unique identities over time. Then you can use the 'background subtraction' approach to detect your animals, and you do not need to train a Detector. You will only need to train a Categorizer. In this case, when you generate behavior examples to train the Categorizer and when you use the Analysis Module to analyze behaviors, you will choose the background subtraction **approach** to detect your animal(s).

If you want to use the Draw Markers Function (see Part 4.6 above and Section I below) to define positions in space or use a Detector to detect your animal(s) or objects (see Part 3.3 above and Section II below), then you must complete these processes before Training a Categorizer (Part 4.2 above and Section III below).

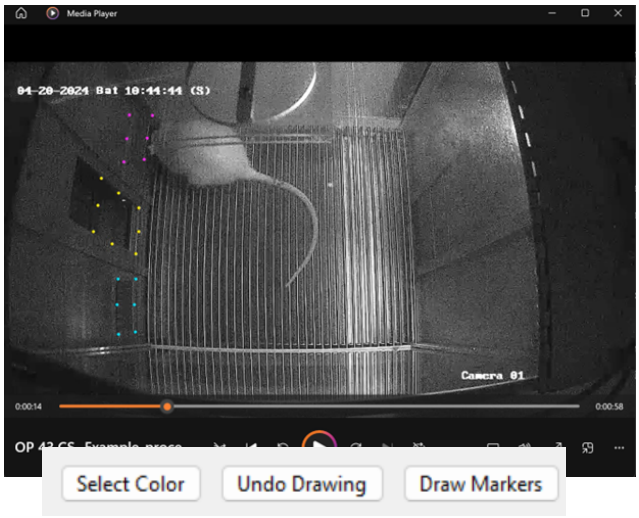
Training a Detector is used when a) your behaviors of interest occur in an environment that changes across time (e.g., animals building a nest), b) illumination/lighting in your video varies (e.g., due to recording conditions, or deliberate changes in illumination) or c) you want to measure behavior from multiple animals within the same video and want to maintain their unique identities

Section I: Using the Draw Markers Function to define fixed positions in space within your video

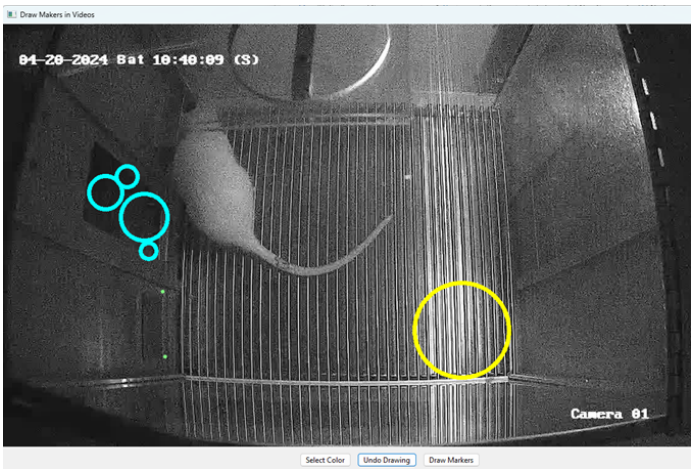
This approach is one of two ways you can teach LabGym about objects within the animal's environment. It uses "Draw Markers" within the "Preprocessing Module" of LabGym. This works well when objects/places are important for identifying a behavior of interest and are in the same position across your videos. You must Draw Markers before you teach LabGym to categorize your behaviors of interest (Section III: Training the Categorizer). If the 'background' is irrelevant to your behaviors of interest, or if the objects you want to detect can be easily distinguished from other environmental features, you do not need to use this function (see also Part 4.2 above).

How will LabGym use these Marks to categorize my behaviors? When you train the Categorizer, you will 'show' it examples of interactions with the marked object (via the "Sort Behavior Examples" process Section III, part B). To LabGym the object essentially becomes part of this behavioral category. This interaction is not based on proximity or pixels. It is based on how you define interactions with the mark when you train the Categorizer. For behaviors like 'sniffing' an object, the animal can sniff things/places that are not the object. So when you train the Categorizer you will include two categories: sniffing the object and sniffing elsewhere. This 'elsewhere' category does not have to include all possible things the animal could sniff. The goal is to show it scenarios where even a person would have to pause and ask, "Is that really sniffing the object?" (e.g., you likely would not consider sniffing the air above the object, with the nose not directed towards the object as a true interaction- so that would go in the 'elsewhere' example; we expand on this below).

In Draw Markers you will use colored circles to mark place(s) in your video. Each color should represent one object/place. There is no limit on the number of objects/places. Importantly, you only need to draw markers on the first frame of one video; LabGym will then apply those markings to all frames of all videos you select. You need the same markers on all of the videos you want to analyze for a given behavior. We recommend you start by marking your training videos and see how well it works before marking the rest. *Note, that LabGym will open the first frame of your video and you must make the marks on that frame. Therefore, you should choose a video where the objects you want to mark are visible to you in the first frame (or you can trim the video appropriately).*



Section I.I: The Categorizer uses computer vision to 'see' these markers of different colors and assess how the rat interacts with these color-marked objects/places. For example, a Categorizer can distinguish between a rat sniffing the yellow marker from a rat sniffing the blue marker, or a rat sniffing the yellow marker from a rat sniffing the air above the yellow marker. The more pixels each circle occupies, the easier it will be for the computer to distinguish those positions. You can draw several circles of the same color to make them thicker. You can also drag to expand the size of your circle; a single click will be a single point.

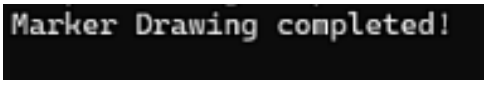


When LabGym uses these markers to categorize behaviors, it analyzes how the animal interacts with these markers; it is not using distance from a position in space, rather the markers become part of the environment and that defines the behavior itself (i.e., sniffing in the food trough is distinct from sniffing directed at the yellow marker and sniffing directed at the air above the yellow marker).

To use this function:

1) Start LabGym
2) Click on the "Preprocessing Module" button

3) In the next window Click the “Draw Markers” button
4) In the final window Click the first button to “select the videos (you want) to mark”. These should be the videos you chose ahead of time to use for training. If you resize one video, you must resize all of them
5) Click the second button to point LabGym to the folder where you want these new marked videos to be stored. *Note, this process will result in new video files that you will need to use to “train the Categorizer” (see section III below), so be sure to make note of where you stored them.*
6) Click “Start to Draw Markers”. This will open the first video file you selected in step 4. Use the buttons at the bottom center of the window to select a color for your first object/place. Drag and drop to place a circle (or several circles) on your object/position. Each color corresponds to one object. When you are done drawing all your objects, click “Draw Markers”. LabGym will automatically place these markings on all frames of each video you selected. It will report when the process is complete in the command window.

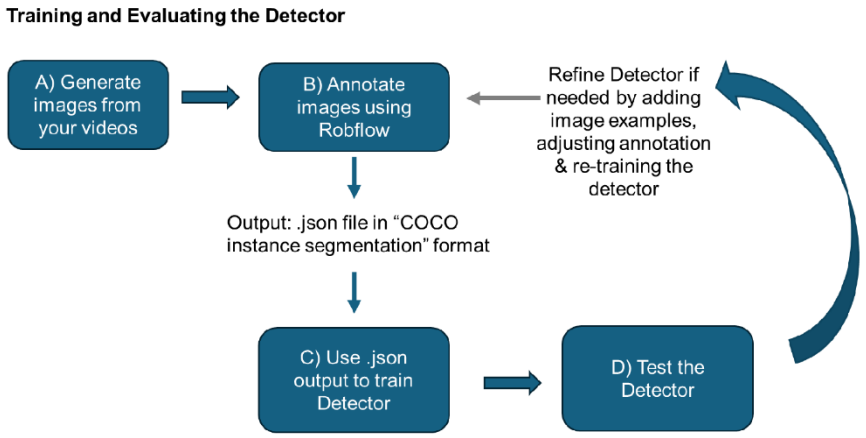


7) When complete, there will be new video files (.AVI) that now contain the colored markers as you placed them in the folder you designated in step 5. The files will have their original name followed by “_marked”. **You will use these new video files to “Train the Categorizer” (see Section III below).**

Section II: Training and evaluating a Detector (this section has four parts, A-D).

See Part 3.1, 3.3 and Fig 4 to help you decide if you need to train a Detector.

First, we will ask LabGym to generate static images (i.e., frames) from some of your videos; these are called “image examples” (.jpeg files). Then, we will label (i.e., annotate) those images in Roboflow; this will result in new output files (.json file in “COCO instance segmentation” format) that we will use to train the Detector. One exciting feature of LabGym is that you can use this Detector in combination with any categorizer; that is, you could decide to measure different behaviors later, but still use the same trained detector to identify your animal(s).



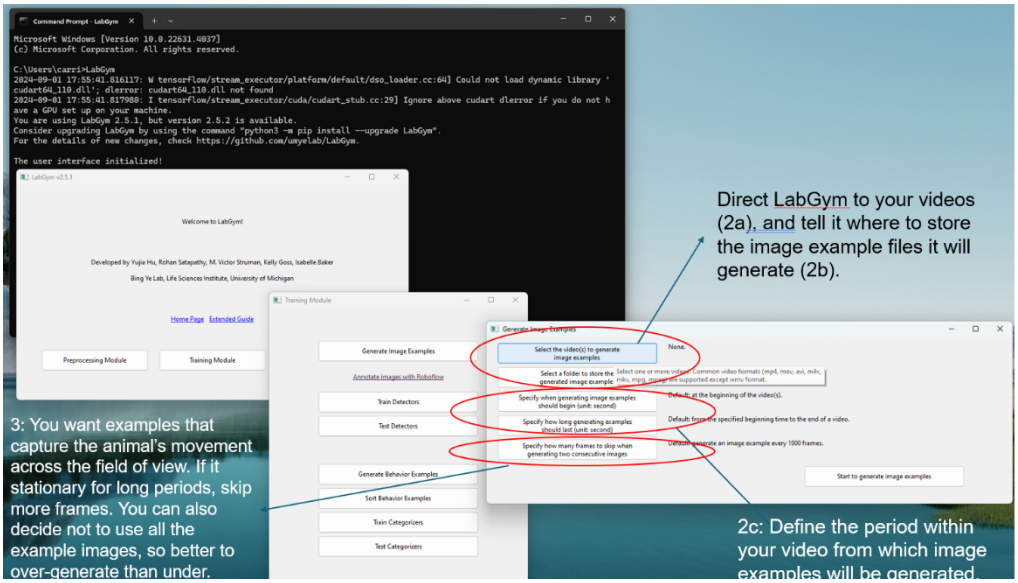
Section II A: Generating image examples (this section has 4 steps).

Note, you can start by using the default settings in steps below; if that generates too many image examples, you don't have to use all of them.

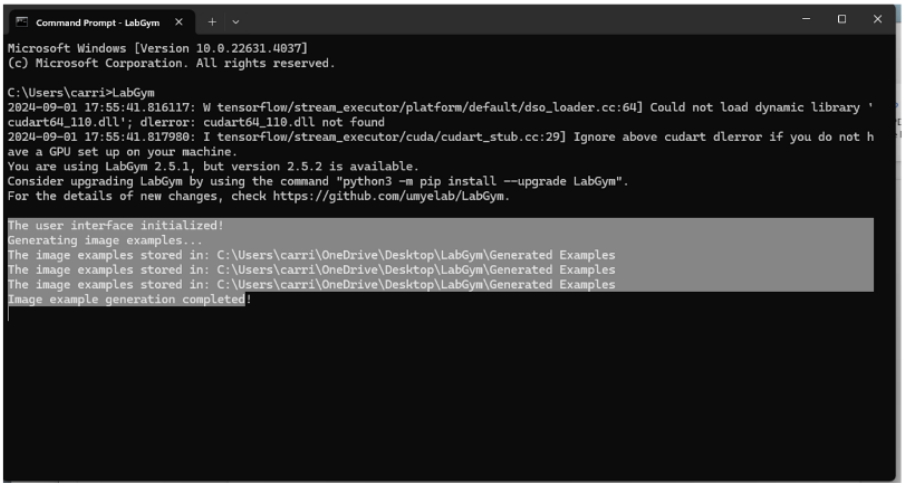
1) Put the videos you selected to use for training the Detector into an easy to find folder on your computer.
2) Start LabGym and click on "Training Module"; this will open a second window. Within this window click "Generate Image Examples" and a third window will open. Note, you can start with the default settings (noted at the right next to each button; we give some guidelines for modifying the settings below).
    a. Click "Select the video(s) to generate image examples" to direct LabGym to the folder containing your training videos. After you select your videos you'll see a popup window that asks if you want to re-size your video frames. Select no for now (see Part 4.5 for more information and guidelines about this).
    b. Click "Select the folder to store the generated image examples" to tell LabGym where you want to store the image example files that it will generate in this process. You will use these files later.
    c. You can use the next two buttons to define the period within your video from which image examples will be generated. The same window of time will be used for all the example videos. Note that LabGym asks for the absolute time in your videos (in sec) at which it should start generating images ("Specify when generating image examples should begin"), and the total duration from that point in time (in sec) through which it should generate examples ("Specify how long generating examples should last). *Example: You have a 3:00 min video (180 seconds). You want to generate images from 0:30 through 3:00, you would set the start at 30 seconds and the duration as 150 seconds (180-30=150).*
3) The last button within the Generate Image Examples window is called, "Specify how

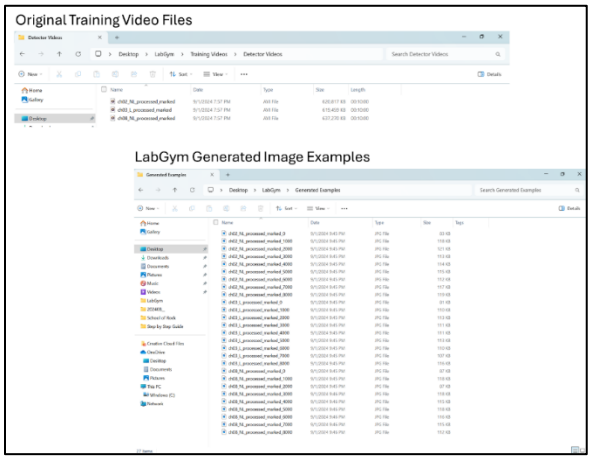
many frames to skip when generating two consecutive images". Remember, your goal here is to "show" LabGym examples that will help it differentiate your animal from the 'background'. This module will generate those examples. You want your example to show the full range of variation in your videos (e.g., animals in different locations in a frame, in different postures, frames with different illuminations) so that the trained Detector can learn what is signal (your animal) and what is just 'noise' (e.g., the variation in illumination). A dataset containing many of the same images (e.g., a sleeping animal, or all the same exact illumination) will reduce the effectiveness of training the computer to distinguish the animal from the background. You want the generated examples to be as varied as possible within the variance of your videos (e.g., showing the animal in many different positions/lighting/contrast).

So, how should you choose the number of frames to skip? If within your video your animal is stationary for long periods of time, or changes in lighting happen rarely, then you can skip a lot of frames (because frames that are identical do not provide any new information). Alternatively, if your animal is very active across the field of view or illumination changes quickly, then you can skip fewer frames. If you are unsure, just use the default setting. You don't have to use all the images that are generated to train the detector.



4) Click "Start to generate image examples" and answer "Yes" in the next pop-up screen. The command/terminal window will tell you that the process has been started, and will show when it is finished:





When the image example generation is complete you will find a new set of .jpeg files in the folder you designated in step 2b above. Note that LabGym automatically names the new files using the naming convention of the parent video.

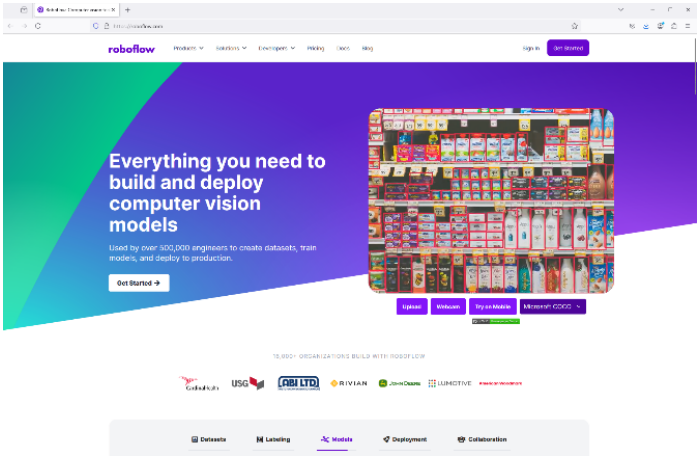
Section II B: Annotate example images (this section has 13 steps)

Next, we will upload the image examples we created to Roboflow and use Roboflow to identify the animal(s) within our example images (i.e., annotate them). This is surprisingly simple, largely because of advances in computer vision (you click on the center of the animal's body and the computer completes an outline of the animal; you can drag and drop points on the outline to adjust them as needed). Note, LabGym does not use point tracking for its detection. You can use other resources to annotate the image examples, but we find that Roboflow works well.

You can use Roboflow for free or get additional features by paying for a monthly subscription. The free option has all the capabilities needed for using LabGym, but the images you upload will be publicly available. The paid version will keep images private.

Roboflow has lots of capabilities and options to explore and try. Below we give you the simplest approach to get the job done. You can go back to Roboflow at any time to add images and re-do or refine any of the steps below.

1) Within the “Training Module” window click the link “Annotate Images with Roboflow” (or go to https://roboflow.com in any web browser). This will open a webpage in your computer’s default browser that will look something like this:

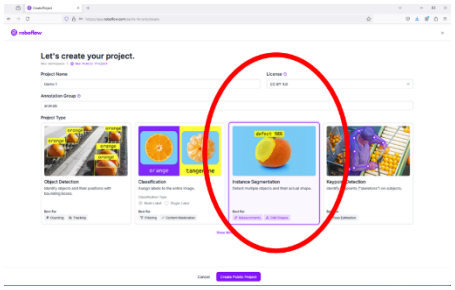


2) Click “get started” to set up an account or create a new account.

3) Choose “Public Plan” and name your “workspace”; any name that is acceptable for public viewing and that you can remember is fine. I called this “demo”.

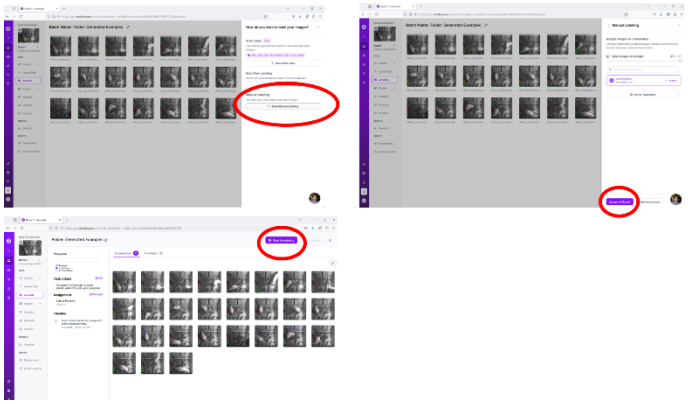
4) Click “Create Workspace”; you can also add collaborators (via email) or select continue without collaborators. This “workspace” will be accessible under your login/account going forward.

5) The next screen is titled, “Let’s create your project”. For Project Type select “Instance Segmentation” then click “Create Public Project” at the bottom of the screen.

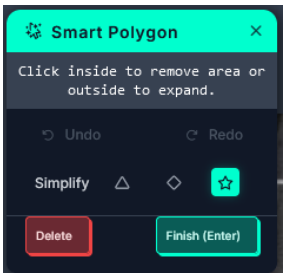


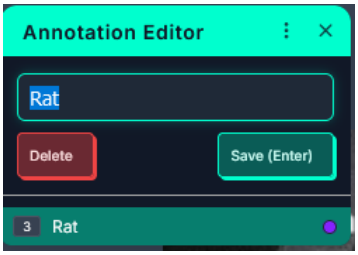
6) In the next window click on “Upload Data” in the menu at the left of the screen. You want to upload the image examples you generated above (these will be .jpeg files). Click “save and continue” at the top right of the window.

7) At the far right click “start manual labeling”. It will give you the option to “assign images” to different ‘teammates’ or assign them all to one person. This feature allows multiple people to work with different images at the same time. Here we will just assign them all to ourselves- so click “Assign to myself” at the bottom right of the screen.



8) Click “start annotating”. This will open a window showing the first of your example images. As you slide over the image, the computer will automatically detect objects and color them (see images on the next page). Click on the center of your animal and the program will draw an outline around it. If it does a decent job, click “Finish (Enter)” in the “Smart Polygon” window. This will transform the outline into an outline with white boxes. You can drag these boxes to adjust the outline. The outline does not need to be very precise. The small pop-up window will now say “Annotation Editor” at the top. When you are satisfied with the outline you will define this label (e.g., “rat”) and click “Save (Enter)”. You can then go on to either define other features in this image (e.g., another rat if you’re going to measure social behaviors or multiple animals in a single video, or an object in the environment like bedding material, or an illuminated cue light), or you can move to the next image by clicking the arrow at the top middle of the screen. See also screen shots below.



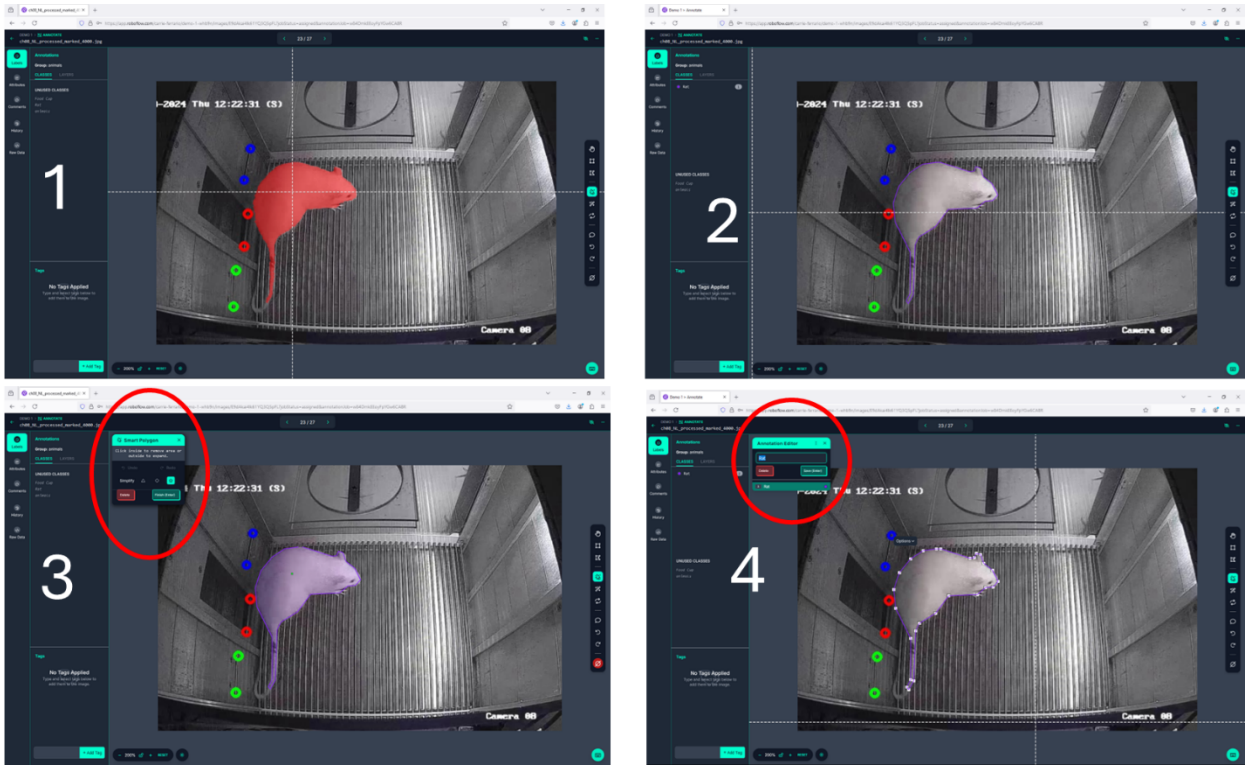


Note, you can “undo” in any window, go back to a previous image, or click on other parts of the object and Roboflow will add that part to the outline.

<u>If you have more than one animal in each video:</u> You need to click on each individual animal, and make sure each outline only contains one animal; these can all be under the same label (e.g., Rat).

<u>Annotating Objects in the environment:</u> If you want to measure behavior in relation to objects in the environment (e.g., something the animal can interact with), then you can also annotate these objects using the process above (e.g., for a cue light, you would just ‘draw’ outlines around the cue light when it is on). In cases like these, when you train a Detector, it will detect your animal(s) and these objects.

Note: If you used the Draw Markers function within LabGym (Section I above) to define a location, these markers will be visible in your example images (e.g., the colored dots in the images below showing where a food trough [red] and retractable lever positions [blue, green] are). If you want to train a Detector to detect these Marks, then you should also annotate the marks in Roboflow (one color = one object). If you do not want a Detector to detect these marks, you can ignore them here (you will instead use the marks by includig backgrounds when you train the Categorizer).

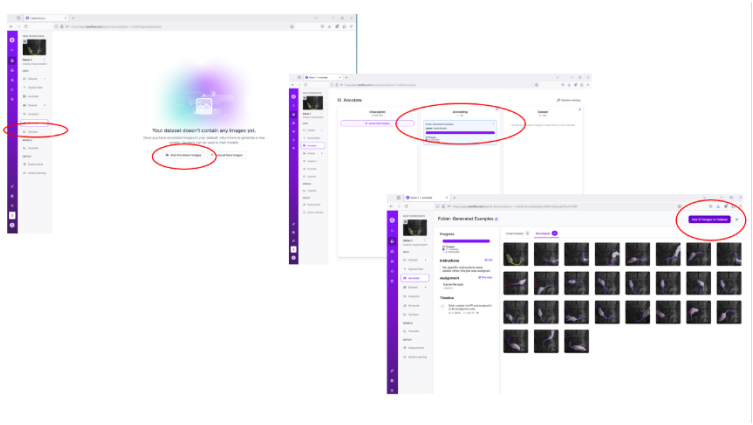


How many images do I need to annotate?

The number of annotated images you need to successfully train a detector varies, but to identify a single animal, typically ~50 annotated images are needed. The key is that you want to annotate images that show the range of poses and positions of your animal/object that capture the expected range in your dataset. For datasets that involve multiple animals or objects, more annotated images are needed (~150-300). In this case, the majority of your examples should include images where two or more animals are in close body contact (or body/object contact). In this case, you only need to include a few examples (~5) where the different individuals are well separated. The more body contact examples the Detector “sees” during training, the higher accuracy of the detection will be and the lower chance of identity switching.

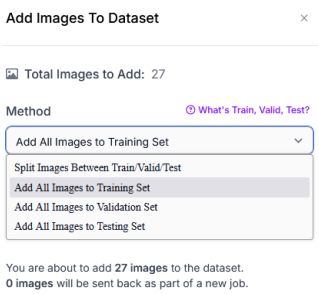
Thankfully, you can increase the number of annotated images simply by flipping and reversing the images after you’ve annotated them; to a computer a mirror image is a new example. Roboflow will do this automatically for us; we’ll go through this in the next steps.

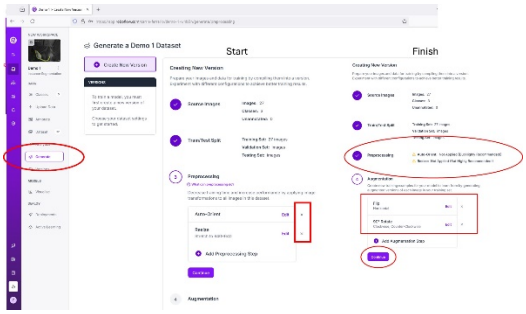
9) When you are finished annotating your images, click the (tiny) ‘back’ arrow in the upper left corner (just under your browser’s back arrow). At the far left, click “Versions”; and in the center click “add Annotated Images”. Next select your annotated images by clicking on “Annotating” (this should appear in the center of the window), then click “Add # images to Dataset” in the upper right.

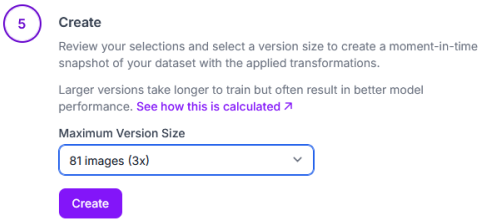


10) In the dropdown menu at the right select “add all images to training set”

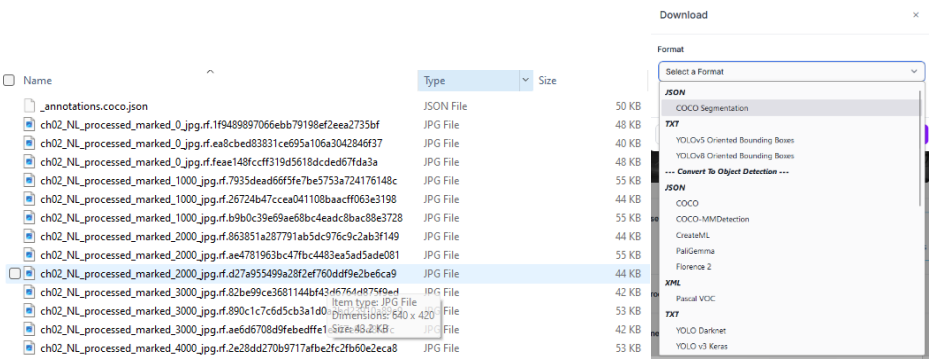
11) Now click on “Generate” in the list menu at the left. Roboflow will give you options for processing your images. You **do not** want to use this feature. Remove “auto-orient” and “stretch” by clicking the x next to these options. Next click on “Augmentation” and click “Add Augmentation step”. This is where you can increase the number of annotated images by choosing “flip horizontal” and/or “90° Rotate”. This will create augmented versions of your annotated images, essentially doubling or tripling the number of examples you have. Importantly, an augmentation will only be useful for training the Detector if this new version of the image might occur in your actual videos (e.g., vertical flipping is not likely to be useful if your video was taken from a side view, since animal is not going to be upside down in your real videos). Click “Continue”, then click “Create”







12) When it finishes, click “Download dataset” at the top right and select COCO Segmentation under the JSON heading, and choose download zip to computer. When you open this new folder, you will see one JSON file, and all the annotated images (originals, and augmented).



You can go back to Roboflow at any time to continue working, add images, change your annotations, add a new object etc.

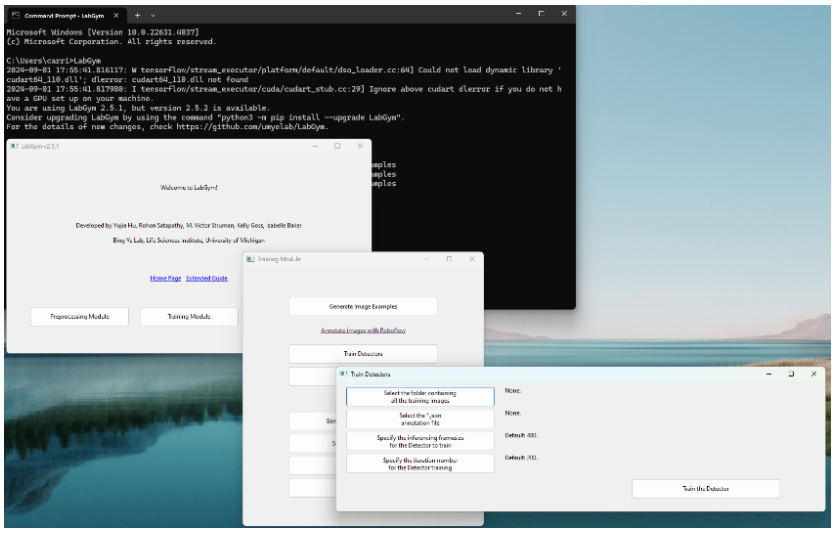
13) Now that we have our annotated images, we can use them and the .json file to train the Detector.

Section II C: Train the Detector (this section has 3 primary steps)

Now that you've generated and annotated your examples (Section II A and B), you are ready to "Train the Detector". LabGym will use your annotated images and a machine-learning based approach to learn to distinguish your animal(s) and objects from the background. This is computationally demanding, so the process will be slower if you're using a computer without a GPU (40 min or more using one with a powerful CPU only) than if you're using a computer with a GPU (<10 min), but it can be done on either type of computer. This will result in a trained Detector (a new folder with a set of files needed to use the Detector will be created). LabGym will ask you to name the Detector and the folder will have the same name. The trained Detector folder will be stored on your computer in the same location where LabGym is installed (e.g., C:\Users\19122\.pyenv\pyenv-win\versions\3.10.9\Lib\site-packages\LabGym\models *Note, this folder may be hidden by Windows*).

The trained Detector will automatically appear in LabGym GUI when it's needed, and you will use the Detector in several different ways: 1. To detect your animal(s)/objects when you "Generate behavior examples" and select 'the method to detecting your animals' in "Training a Categorizer" (Section III). 2. When you "Analyze behaviors" in videos and select 'the method to detecting your animals' in the final Analysis step (Section IV; Analysis and Quantitative Measures). 3. When you want to select a trained Detector to test or delete (in the "Test Detectors" Module; Section II, D).

1) Open LabGym, click on “Training Module” and in the next window click “Train Detectors”. This will open up a new window where you will tell LabGym where to find the files we just created using Roboflow and where you can specify some parameters.



a. “Select the folder containing all the training images”: i.e., select the folder containing the images you annotated in Roboflow and the augmented images.

b. “Select the .json annotation file”: Here you must go to the folder we downloaded from Roboflow and select just the .json file.

c. “Specify the inferencing frame size for the detector training”: This setting determines the balance between how fast LabGym can finish training the Detector and how accurately the detector will perform. The higher this number, the more accurate and slow the detector training will be. A golden rule is, if you are using a GPU, simply set this number equal to the frame size of your videos. In previous sections, we asked you to use a consistent frame size across all your videos. So for example, if your frame is square (e.g., 640 X 640) use 640 as the ‘inferencing frame size’ of the Detector. If your frame is not square (e.g., 960 X 540), use the larger number (960 in this example) as the ‘inferencing frame size’ of the Detector. If you are not using a GPU, set it to 320 regardless of your frame size.

d. “Specify the iteration number for the Detector training”: This setting determines the number of training iterations that will be performed (more iterations will generally yield higher accuracy, but will take longer). A reasonable range for this setting is anywhere from 500-10,000. If you are using a GPU, then this process will be relatively fast, so you can set a high number of iterations (8,000 or even 10,000). If you are not using a GPU, you should start with a small number of iterations (e.g., 500). For studies where the goal is to examine interactions between multiple animals while also keeping track of the unique identity of each animal, we’ve found that using a very large number of iterations (>=8,000) reduces identity errors (e.g., switching identities of two individuals).

You will be able to evaluate the accuracy of your detector in the “Test detector” module (Section II D); so you can always increase the iteration size if the Detector performs poorly. Ways to improve Detector accuracy are also discussed in that section.

2) Click “Train Detector” and “Yes” to start the training process. It will also ask you to name this

Detector (it's a folder that stores all the files related to the Detector); you will be asked to refer to this folder when you Test the Detector and Analyze Behaviors in the final process. You will see lines of code appear in the command window as LabGym begins the steps of training the detector (do not worry if the "train Detectors" window shows *'not responding'*; this is normal; you can only run one process with LabGym at a time). Training the detector can take some time (particularly on a computer with a CPU); so go get a coffee, you deserve it!

3) LabGym will tell you when it finishes training the Detector in the command/terminal window like this:

```
[09/02 11:07:12 d2.checkpoint.detection_checkpoint]: [DetectionCheckpointer] Loading from C:\Users\carri\AppData\Local\P
rograms\Python\Python310\Lib\site-packages\LabGym\detectors\test2\model_final.pth ...
Detector training completed!
```

You can now close the "Train Detectors" window. LabGym will automatically add the trained Detector to the LabGym GUI (it will have the name you gave it above) and will offer it to you as an option whenever its needed.

Section II D: Testing the Detector (this section has 5 steps)

This module will tell you the mean average precision (mAP from 0-100%) of the Detector you trained. It is not expected to reach 100% (perfect) but should be above 60%. It will also generate labeled images showing you what it was able to detect in each example image (see images below). Testing your Detector is necessary; the Detector must work reasonably well in order for the analysis to be accurate. The simplest thing to do is test the detector using the same example images you used to train it. This approach is a useful initial step (you should get out what you put in; if not, you probably need to refine how you annotated the videos or maybe add more image examples).

For publication purposes the "gold standard" in the field is to test the Detector on new annotated images (i.e., those not used in training). To accomplish this, you should repeat the steps in Section II A and B above (Generate and Annotate image examples) on images that you did not use to train the Detector. Be sure to add these to a different folder when you upload them to Roboflow. You can then use this second group of images and the accompanying JSON file to test the Detector you trained. In either case (testing the detector using the same image examples or new ones), the steps and process within the "Test Detectors" module is the same, and is pretty fast.

Note: Another approach to evaluate a trained Detector is to run the Detector on some videos in the Analysis Module (Section IV below) without specifying a Categorizer. LabGym will output annotated videos with detected animals/objects tracked and annotated. This will give you confidence in what LabGym is detecting, but will not provide the publication quality metrics that theTesting the Detector function provides.

So, let's get to the steps of "Testing the Detector". I am going to use the same example images

and .json file that I used to train the Detector. This is a good first pass, since that should work well, and if it doesn't, I likely need to add more example images or adjust how I'm annotating them.

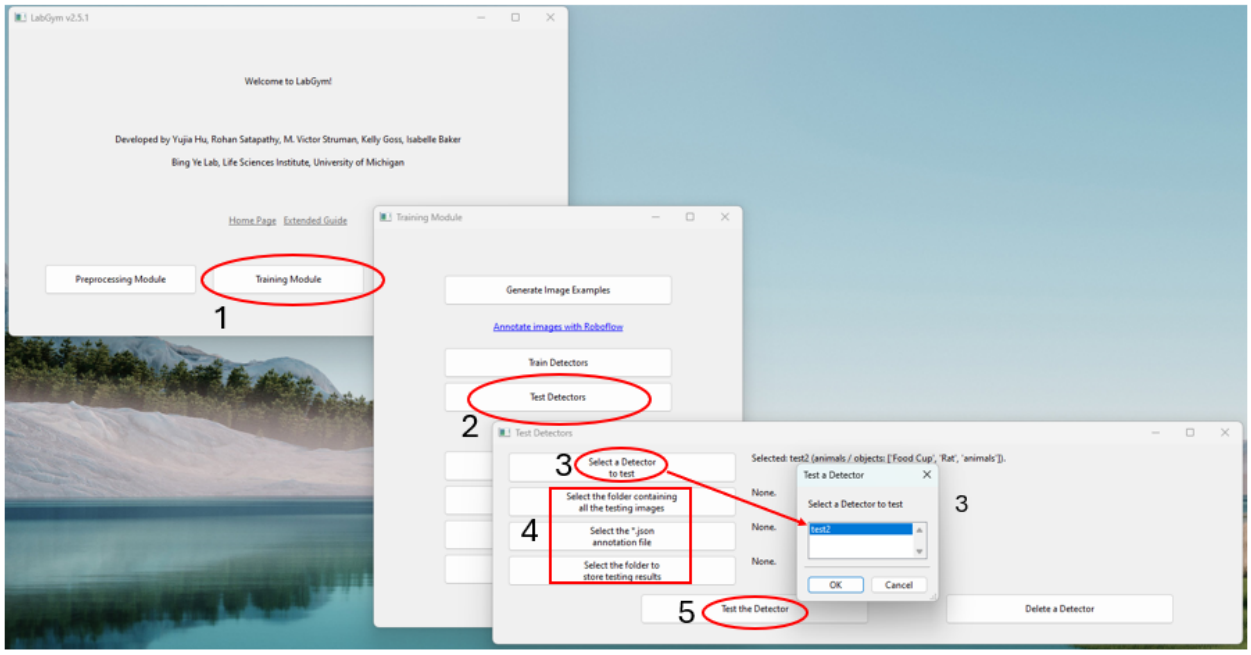
1) In the main LabGym window, Click the "Training Module" button.

2) In the next window click the "Test Detectors" button.

3) Click the first button: "Select a Detector to test". All Detectors within your LabGym program will be listed; choose the one you want to test.

4) Click though the next three buttons in sequence and select the appropriate folder or file. "Select the folder containing all the testing images" refers to the folder containing the images you annotated in Roboflow and the augmented images. "Select the .json annotation file": Here you must go to the folder you downloaded from Roboflow and select just the .json file. Click the Last button to tell LabGym where to store the results.

5) Click the "Test Detectors" button. LabGym will show you that it's started the process and



when it finishes the process in the black command/terminal window.

LabGym gives you the mean average precision (mAP from 0-100%) in the second to last line of the command window (just above the line reading "Detector testing completed!"; see below) and labeled images showing you what it was able to detect in each example image (Fig 5). These images will be in the folder you designated above for results in step 4. These images are useful for visualizing what LabGym is detecting and what it may be having trouble with (the mAP averages over all the types of things you annotated, so it can be low if some things are being poorly detected while others are being detected accurately). It is not expected to reach 100% (perfect) but should be above 60%.

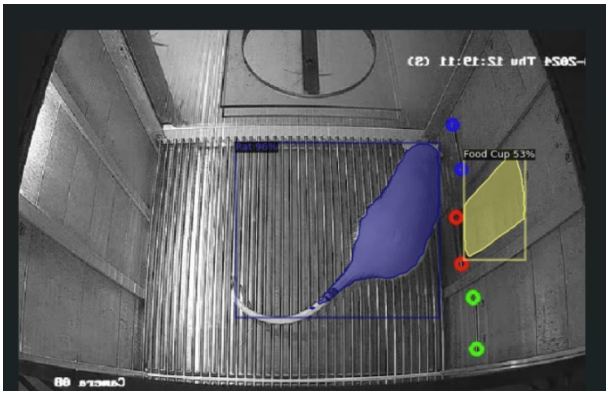


Fig 5: Example Image resulting from Testing the Detector. Animals/objects that it has identified are colored in. The small colored dotes in this example (blue, red, and green) are Markers I placed using the Draw Markers Function (section I) and will not be included in your images if you did not use that feature.

Did my Detector pass the test?

The detector I tested did not do very well; only 36.7% mAP, yikes! (Fig 6 below). This is bad, especially because I tested the example images I used to train the Detector. This suggests something is not as I expected—and indeed that is the case! When LabGym starts the test, it tells me some key information in the command/terminal window; this includes telling me what my Detector was trained to detect (red circle in Fig 6 below after "total categories of animals in the Detector"). I wanted the Detector to detect the rat and the food trough (i.e., food cup), but it shows that this Detector has Food Cup, Rat and a third category- Animals. I started this example from an existing Roboflow workspace and did not annotate anything for "Animals". This explains the poor performance. So, I should delete this detector (by clicking the "Delete a Detector" button within the "Test Detectors" window), it's not usable. I can then return to Roboflow (described in Section II B above), remove the unnecessary category from my Roboflow, workspace and repeat steps 9-13 in Section II B above. Then I can repeat Section II part C ("Train the Detector") to make a new trained Detector.

Similarly, you can improve your Detector by generating and annotating additional image examples by going back through the steps in Section II parts B and C above.

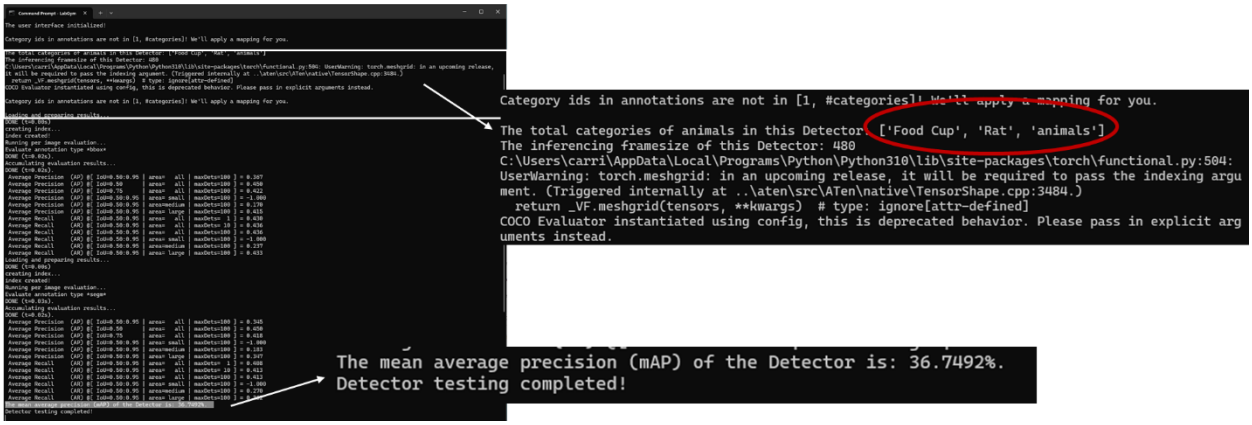
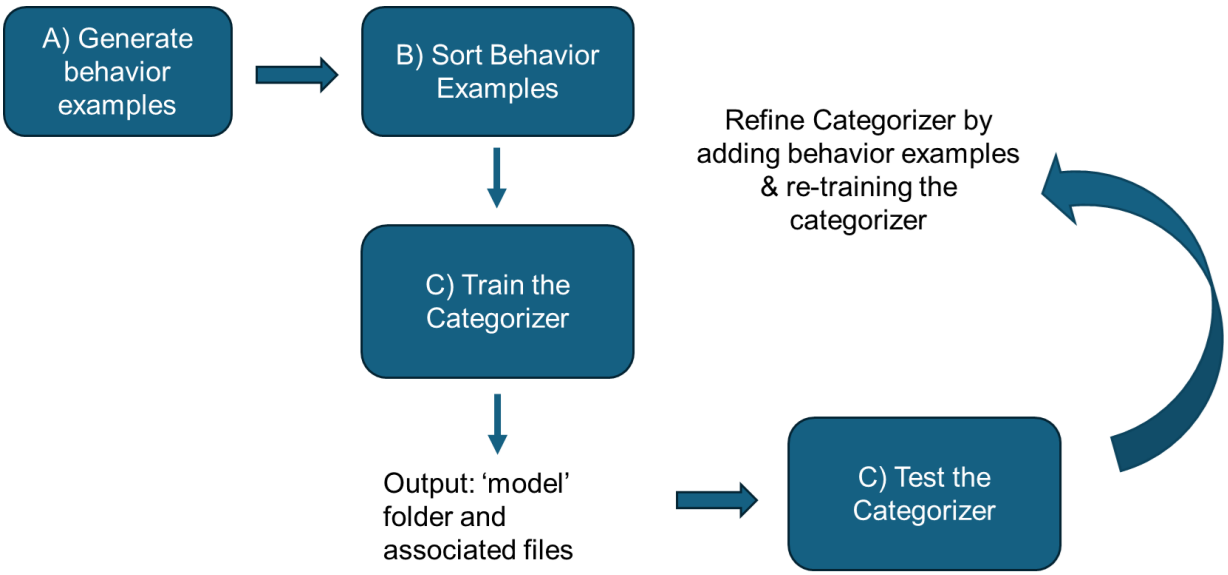


Fig 6: Output after completing 'Test Detectors' module. This detector failed; the mAP is 36.7%, this is well below chance. The red circle shows me what the Detector was trained to detect (rat,

food trough, and animal). I did not annotate anything for 'animal', so it makes sense that the detector did not work well.

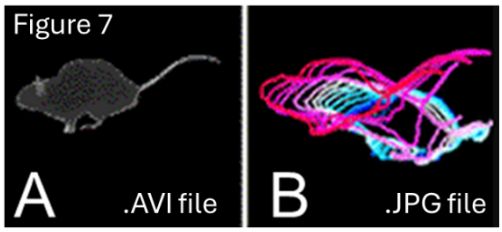
Section III: Training and Evaluating a Categorizer (This section has 4 parts, A-D).

In this section you will give LabGym the information it needs to learn to identify your behaviors of interest. This involves generating and sorting examples of each behavior from your videos, and then using these examples to train LabGym to identify your behaviors of interest. Typically, you'll use the same videos that you used to train the Detector above, though you can use different videos with the same features (i.e., lighting, visibility, enclosure). You want to pick videos that best show the behavior(s) of interest, and that capture the full range of how each behavior can be displayed (see also Part 2: Before you begin using LabGym). If you want to use the Marker function to mark objects/positions in space, or a trained Detector to detect objects/places, then you must complete those processes before training a Categorizer (see also Part 3.3 and 4.2 above for how to decide when to use Markers or a trained Detector to recognize objects/places that are relevant to your behaviors of interest).



Within the steps below there is a fair bit of terminology and settings to choose. HAVE NO FEAR! We will walk you through terminology and selections. Once you understand the terms, completing the steps below is fairly quick. The defaults provide a strong starting point, and you can easily try out just a few videos initially to see what the outputs look like.

Section III A: Generating 'behavior examples' of each behavior of interest from your videos



Through the process detailed below LabGym will agnostically generate behavior examples from your videos based on some parameters you provide. Essentially it will pick out snippets from your video and generate a “pattern image” that corresponds to each frame of the video snippet (i.e., an “animation”). This will result in many ‘pairs’ of files (Fig 7); an animation (.AVI file, Fig 7A) and a corresponding “pattern image” file (.jpg, Fig 7B). These pairs of files are called “behavior examples”. In Section III B you will sort these examples according to your behaviors of interest; you do not need to use all the behavior examples and can discard ones that are not useful.

*Note: **DO NOT** change the file names for the generated behavior examples; they contain the key information that is needed in the final step to Train the Categorizer (Section III C).*

Note: One categorizer must be trained using behavior examples generated with the same parameters (however, the method of animal/object detection can differ), although most of the paramter information is included in the file names we still recommend writing down the values you use for each setting, and the choices you make for each parameter (see also step #14); you will need this information to add additional behavior examples.

1) Put the videos you selected to use for training the Categorizer into an easy to find folder on your computer. *Note, if you used the Draw Markers Function (Section I above) you should use those videos in this process*

2) Start LabGym and Click the “Training Module” button.

3) In the next window click the “Generate Behavior Examples” button. This will open a new window with several buttons. You will click on each button in turn and be given some additional options in small pop-up windows. As a starting point you can just accept the defaults, and follow the prompts as you click through, though we do provide some guidance for how to think about each setting below.

4) Click on the “Specify the mode of behavior examples to generate” button. You have four options, each followed by a description (see also Fig 4 captions*)*:

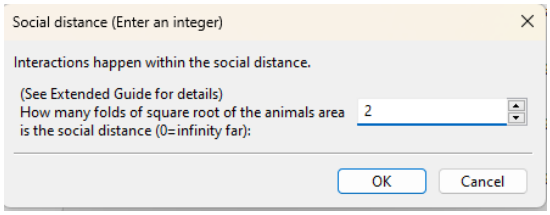
   a. “Non-interactive”: choose this option if you want to measure behavior in only one animal within the field of view or if you want to measure behavior in multiple animals, but you do not need to keep their unique identities. Despite the name, behaviors here can include interactions between animals and objects/places in the background (regardless of your method of animal detection, i.e., background subtraction or a trained Detector, or object detection, i.e., a trained detector or Draw Markers).

b. "Interactive basic": choose this option if you want to analyze behavior of one animal where behaviors of interest rely on the animal interacting with another animal or objects/places in the background. This mode will not keep track of the individual identity of the animals (e.g., it can measure one animal licking another animal, but it won't distinguish which animal is licking vs being licked).

c. "Interactive advanced": choose this option if you want to analyze behaviors of one or more animals interacting with each other and/or objects/places in the background. You must choose this option if you are going to measure directed interactions (e.g., animal 1 doing something to animal 2).

d. "static image": this option is not for use with videos; use this to examine non-interactive behavior in static images (i.e., digital pictures). We do not address this function in this guide.

If you want to measure several different behaviors that fall into different categories above, choose the setting that applies to the most complex behavior (in terms of interactions between the animal(s) and the background).

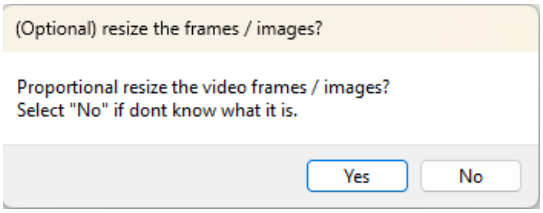


If you select the "Interactive advanced" option above, a new window will appear asking you to define the "social distance". Despite the name, this refers to the threshold for the distance between any one animal *or object* and any other animal *or object* in one frame in which an interaction could happen. LabGym will use this distance to generate appropriate behavior examples, for instance, excluding animals/objects within a frame that are too far apart when it generates behavior examples. It's important for the behavior examples to start at a point where there is physical separation between the animal and things it can interact with, so that you can then give LabGym good examples of animals moving to interact with other animals/objects.

Setting this number to 2 is generally a good starting point (2= other animals/objects are twice as far away as the animal's body size); click "OK".

Note, if you set it to 0 every animal/object in a frame will always be included in every behavior example. You don't have to use every example LabGym generates, but using all the frames will result in a lot of files and will take longer to generate.

5) Click the next button “Select the video(s)/image(s) to generate behavior examples”. Here you will select the videos you want LabGym to make example behaviors from. These should be the files you chose for training the Categorizer. Note, you must select all the videos you want to include, not a folder. *(Aside: this window states that you can select “images”; this is for using the “static image” mode mentions in #4 above; we are covering that here).*



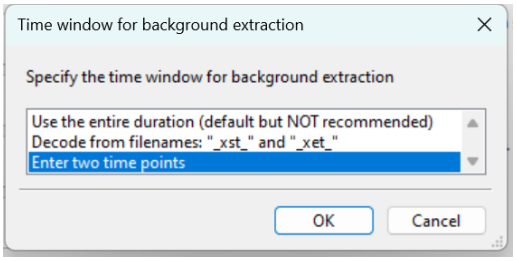
Once you select the videos, a pop-up window will appear asking if you want to “resize the frames/images” *(we are working with videos, so it’s just asking if you want to change the frame size [height and width] of your video)*. If you resized the frame when you trained a Detector, you must resize it again here in the same way. Since we said “no” when we trained a Detector in Section II above, I will say “no” here (see Part 4.5 and Section II C, step # 1C for guidance on changing frame size).

Note: Reducing the frame size can make the processing speed faster, but if the frame size is too small the frame will become blury, the accuracy might decrease.

6) Click on the next button, “Select a folder to store the generated behavior examples”. *Be sure you know where this is on your computer; you will need it later.*

7) Click on the next button, “Specify the method to detect animals or objects”. The options available here will be automatically provided by LabGym, based on the “mode” you selected in step 4. If in step 4 you select “non-interactive” or “interactive basic” then you can choose either option (subtract background or use trained Detectors). If you selected interactive-advanced, you will only have one option: use trained Detectors. If you want to use a Detector to detect objects or places in the environment, you must choose ‘use trained Detectors’ here.

<u>If you chose “Subtract background”:</u>



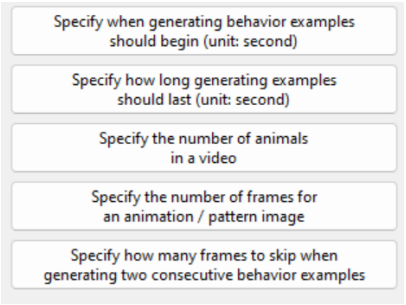
There will be a series of pop-up windows that ask questions about your videos; these will largely be self-explanatory. The last pop-up will ask you to “specify the time window for background extraction” and provides three options. The option “Enter two time points” is the simplest (you provide a start and stop time that will be used for all the videos selected in step 5). You do not need a long period of time (typically 10-60 seconds is sufficient), but you want to choose a period of time during which the animal(s) moves around the environment. Shorter time windows will result in faster processing speed. This will result in image files for each video showing the background with the animal removed (i.e., extracted backgrounds). You can use these ‘extracted background’ files in the final analysis step (Section

IV).

<u>If you chose “use trained Detectors”:</u>

There will be a series of pop-up windows that ask questions about your videos; these will largely be self-explanatory. In the “Animal/Object kind” pop-up you will specify what you want the Detector to detect; you should check all that apply to the behaviors you want to Categorize.

The remaining buttons are also largely self-explanatory, and default settings should provide a good starting point. However, we will make a few clarifications and recommendations here.



8) “Specify when generating behavior examples should begin” and “Specify how long generating examples should last” is asking what period of time (in seconds) in our videos you want LabGym to generate behavior examples from (i.e., start/stop). This is useful if you have a set of videos where you know one behavior happens a lot from time A to B, but not time C to D.
9) “Specify the number of animals in a video”: If you are using the background subtraction method or a Detector trained only to identify animal(s), then this is asking how many animals are in one video. If you chose ‘use trained Detector’ in previous option, and if your Detector is trained to detect multiple types objects, then you need to specify the numbers of objects in *each category* (e.g, a Detector trained to identify two animals and one lever: aniamls=2, lever=1). LabGym will automatically retrieve the list of what the Detector was trained to detect from the appropriate source file.
10) “Specify the number of frames for an animation/pattern image” The animations and their partner pattern images (i.e., behavior examples) will span a duration (in number of frames; must be an integer) defined by you. This should approximate the duration of a single behavior episode (i.e., how long a given behavior would need to last for you to recognize and consider it to be that behavior). This duration needs to be the same across all the behavior examples that are used to train one Categorizer. If the duration of different behavior episodes is different, then use the longest one. Stated another way, this is asking how long a single instance of a given behavior should last (e.g., rearing up onto hind paws may last ~ 1-2 seconds. If my frame rate is 15 frames/sec, then I’d want to set this number to 15 or 30. Likely 15 is sufficient).

Note: You can always try different settings to generate example behaviors and see which settings can best capture your behaviors of interest. This will become clearer once you learn how the behavior examples are sorted below (Section III B). It’s very easy to add more behavior examples, re-sort them, combine or add behavioral categories, or train a new Categorizer. So don’t overthink it, just try something that seems reasonable to you and adjust based on the

Categorizer testing (see #14 for information about what aspects must be the same in order to combine behavior examples generated at different times).

11) "Specify how many frames to skip when generating two consecutive behavior examples": If two consecutively generated examples are too close in time (e.g., one generated at the 10th frame and the other at 12th frame), and above you set the duration to 10 frames, these examples will have 8 overlapping frames. Thus, they would be too similar to each other to provide 'new' information and will impair the training efficiency when you use these examples to train a Categorizer. Generating too many similar examples will also make sorting the examples laborious. Therefore, you should skip some frames between each example generation. A practical recommendation is to set this number equal to the duration you set for a behavior episode.

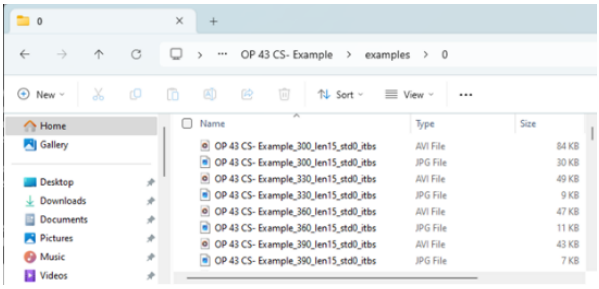
12) Click the "start to generate behavior examples". This will trigger a few final pop-ups

    a. The first will ask if you want to "include background in animations"? If the background is relevant to at least one behavior of interest, click "Yes", otherwise click "No". You must click "Yes" here if you want to use "markers" that you drew to define a behavior. If you are using a trained Detector to detect the "markers" you do not need to include the background here.

    b. 'Including body parts?': Choose 'Yes' if the motion pattern of individual body parts (e.g., limbs or nose) are critical to identifying one of your behaviors. If you choose 'Yes', you will be asked to choose a standard deviation (STD) value (between 0-255) for "motionless pixels". The larger this value is, the less detail of the animal's body parts will be shown in pattern images. Showing too many unnecessary details of 'body parts' (e.g., the texture of the fur on its limbs), can sometimes impair the accuracy of the Categorizer trained on such examples. One approach is to generate behavior examples with different STD values (e.g., 0, 100, 200) and see which value gives you the amount of detail necessary for you to distinguish different behavior categories by eye. Another practical recommendation is to set this number to 50 and adjust from there.

13) Last, LabGym will ask “Start Behavior Examples” Yes/No. If you’re ready, click “Yes”. LabGym will show its progress and when it is finished in the command/terminal window. When it’s finished you will find one folder for each of your videos, each folder will contain pairs of animations and pattern images, like those shown in Fig 7 above. They will be named the same as the parent video file, plus additional critical information about the settings you used to generate them.

```
The user interface initialized!
Preparation started...
2024-09-06 18:36:05.890952
Video fps: 44
The original video framesize: 720 X 1280
The resized video framesize: 270 X 480
Extracting the static background...
Background extraction completed!
Estimating the animal size...
2024-09-06 18:36:30.535562
Estimation completed!
Single animal size: 3959.56704545457
Preparation completed!
Generating behavior examples...
2024-09-06 18:36:38.100122
Behavior example generation completed!
Preparation started...
2024-09-06 18:37:00.830128
Video fps: 45
The original video framesize: 720 X 1280
The resized video framesize: 270 X 480
Extracting the static background...
Background extraction completed!
Estimating the animal size...
2024-09-06 18:37:24.964482
Estimation completed!
Single animal size: 4931.038888888889
Preparation completed!
Generating behavior examples...
2024-09-06 18:37:32.867927
Behavior example generation completed!
```

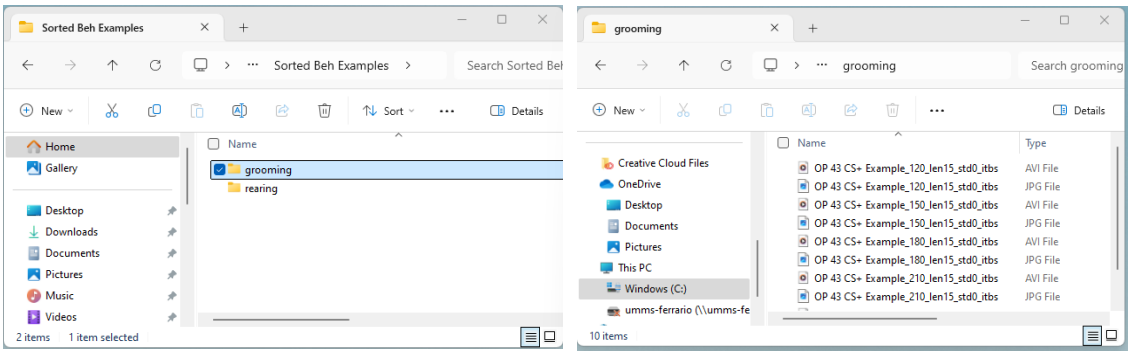


DO NOT *change the file names of these generated behavior examples; they contain the key information that is needed in the final step to Train the Categorizer (Section III C). The ability to ‘Sort Behavior Examples’ using keystrokes will also be based on these file names.*

You can generate behavior examples using different detection methods (e.g., some using a trained Detector and others using background subtraction—as long as both work well in detecting the animals). However, the following parameters must be set to the same values/choices if you generate behavior examples at different times to train the same Categorizer: 1) The duration to generate behavior examples 2) whether or not to include background in ‘animations’ 3) whether or not to include body parts in ‘pattern images’ and if body parts are included, the ‘STD’ value 4) the ‘behavior mode’, and if using ‘interactive advanced’ mode, the ‘social distance’ value.

<u>If you are using “Draw Markers” to mark objects/places within your background without using a Detector to identify the marks,</u> the process above will be exactly the same *but* in step 12a you would choose ‘include background in animations’ during the generate behavior examples process. The colored markers need to be visible in the animation so they can be used to categorize behavior (e.g., sniffing the blue marks). In this way LabGym can “see” the markers in the behavior examples. This is needed because the marker will essentially become part of this specific behavioral category (e.g., sniffing the blue mark). If a given behavior can happen at other places (e.g., sniffing the air and not the blue mark), then you will also need to include this as separate category (see also Section I).

If you are using a Detector to identify any objects/places in the background that are relevant to a behavior of interest the process above will be exactly the same, but in step 7 you would “specify the method to detect animals or objects” as “use trained Detectors”. This will result in a pop-up window (“Animal/Object kind”) where you will select the relevant objects.



Section III B: Sort behavior examples

Now that you’ve generated behavior example pairs (animation .avi and pattern image .jpg above), these pairs of files need to be sorted into different folders according to the behavior type you want to categorize. You can do this using a drag and drop approach on your computer or within LabGym using user defined keystrokes (in the Training Module, Sort Behavior Examples button). Below we go through the steps of using each approach.

Sorting behavior examples is easy (and, call me crazy, but kinda fun!). This process is completed the same way regardless of the mode you used above to generate the examples. The goal here is to “show” LabGym examples of each unique behavior of interest (Fig 8). If you can identify a behavior and distinguish it from another behavior by watching the animation or by looking at the pattern image, then LabGym should be able to categorize that behavior. You do not need to use all the example behavior pairs.

The accuracy of the trained Categorizer will depend on the number of behavior example pairs you choose for each behavior, and the diversity of the example pairs you choose (i.e., the better they show the full range of variation for displaying that behavior). For general scenarios, 100~200 pairs of behavior examples for each behavior type are sufficient to train a highly accurate Categorizer.

If a given behavior is very regimented or stereotyped (i.e., largely looks the same across individuals) then you can use fewer examples and videos. If a given behavior is varied (like social behavior, which can look different within and across individuals) then you will need more examples. We recommend starting with a relatively small number of examples, and seeing how the Categorizer performs. You can easily add more examples to the folders at any time.

One last important note about categorization that affects how you choose to organize and sort the behavior examples: A trained Categorizer is going to ‘scan’ through each of your videos and

'ask' which of the behaviors that I was trained to identify is the animal doing now? If you trained it to detect behaviors A and B, but your animal is doing behavior C (i.e., a behavior you did not give it examples of), then it will forcibly categorize behavior C into Category A or B (in this case, the confidence it reports will likely be very low; so you will be able to easily tell that this has occurred). To avoid this potential issue, you want to create behavior types that span the range of behaviors your animal(s) is likely to do (e.g., grooming, rearing, moving) even if you don't necessarily want to quantify that behavior; you can ignore that data/output. For example, we've found that it can be useful to include "still or in place" as a behavior type separate from "sniffing" (where a rodent tends to be 'in place, but its nose is moving'). Another strategy is to have an "extraneous" behavior type which can be a catch all for things your animal(s) may do often, but you don't really want to measure. If you try this approach, you should not combine behaviors that look very different from each other, as this can also result in poor training.

If a behavior type that you have not given examples of happens rarely, then you may not need to include it, since it will not affect the categorization accuracy very much. *Note that the performance of a Categorizer will also be the same across all of your videos.* You can also remove any behavioral categorization that falls below a certain % confidence (defined by you) in the "Analysis Module" (or by sorting the final data in .xls or .csv files; see Section IV below).

Importantly, in the final analysis step (Section IV), you will set a threshold for the Categorizer to report that the two most likely categories are too similar in probability to make a determination. In this case, the Categorizer assignment will read as 'NA' (no category) on the visualizations and in the data file.

We've found that LabGym is actually rather robust; so, we recommend starting with behaviors that are most obvious and interesting to you. If you find that the Categorizer is making mistakes, you can always add a behavior type, refine or expand the example behavior pairs.

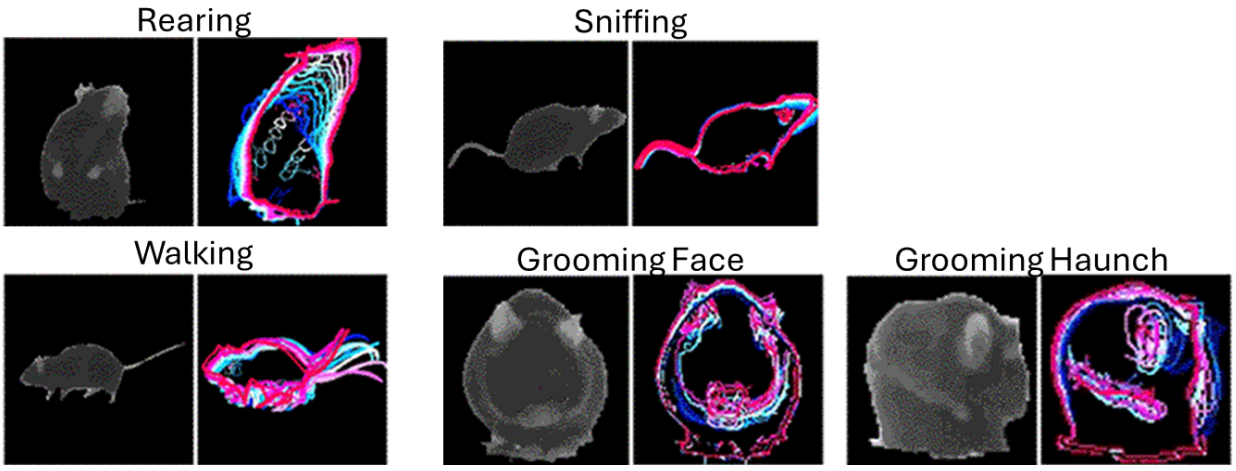


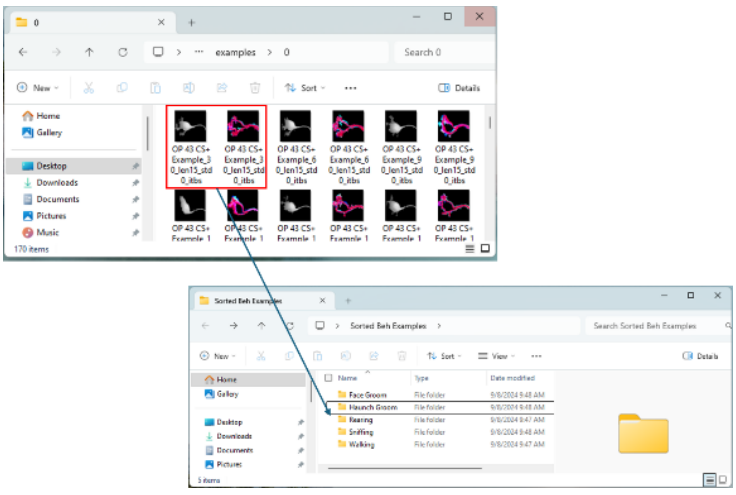
Fig 8: The corresponding behavior category is written above each example behavior pair. The behavior examples within the same behavior category should share similar features or patterns

that can be identified by watching the animation or looking at the pattern image with your naked eyes. Likewise, the behavior examples belonging to different behavior categories should possess different features or patterns that you can identify by watching them. If you can identify the shared features or patterns of one behavior type and distinguish the behavior types by watching the animation or looking at the pattern image, then LabGym should be able to learn to categorize these behaviors. From the examples above we could decide to have one behavior type for grooming that includes the mouse grooming its face and its haunches, or we can create two behavior types, one for the mouse grooming its face and a second one for grooming its haunches, depending on our analysis goals.

The 'drag and drop' sorting method (Fig 9): Open one of the folders with your unsorted behavior examples. Click on an animation to watch it or view the pattern image. If it shows a discrete behavior you're interested in, drag both files into the appropriately labeled folder (you must create the folders; use short, simple names you can remember). This process happens outside of the LabGym program. *Be sure to move both files of the pair; you will get an error in the training step if one of the files in a pair is missing.*

Fig 9: Drag and drop pairs of example behavior files (animation and pattern) into new folders according to the behavior shown in that pair (each folder corresponds to one behavior, i.e., category). You do not need to use all the pairs; each pair can only be assigned to one behavior type. You can mix file pairs from different videos and animals into the behavior folder, or keep separate behavior type folders for examples from each video; LabGym will compile them for you in the "train Categorizer step").

The 'shortcut keys' sorting method: This is done within LabGym; it will open each example behavior animation and pattern image side-by-side, in turn (you can go forward or back using preset shortcut keys in the table below). You define a keystroke to correspond to each behavior type. LabGym will then move those example behavior pair files to the corresponding folder (it will give the folder the same name you assign to the keystroke). The folder organization at the end of sorting with this method will be the same as the screen shots above.



a. Start LabGym and click the "Training Module" button. In the next window, Click the "Sort Behavior Examples" button. Use the first two buttons to tell LabGym where the unsorted behavior examples are located (*note, you must select the sub-*

folder within the "examples" folder that corresponds to examples made from one video), and where you want to store the sorted behavior examples (use a new folder, not within the unsorted folder).

b. Click on the third button ("Enter a single character shortcut key and the corresponding behavior name").

c. Define a single keyboard key to correspond to each behavior type (LabGym will automatically create folders and give each folder the same name you assign to the keystroke; you can change it later if you like). You don't need to write down the key names because you can use different keys to sort more behavior examples into the same behavior folders (i.e., categories) next time, just make sure the behavior names, which are the folder names, are the same. For our example, I am going to define these keystrokes: r-rearing,g-grooming (*note there are no spaces*).

d. Click the "Sort behavior examples" button. The first example pair will open, use "p" and "o" to move back and forward in through the pairs. If you don't want to use an example, move to the next one ('o'). If it shows a behavior of interest, hit the corresponding key [e.g., "r" if it shows "rearing"]. LabGym will move this behavior pair to the correct folder immediately.

Preset Shortcut	Action
o	previous
p	next
q	quit
u	undo

e. When you are done sorting the examples close the "sort behavior example" window (you can stop at any time by hitting 'q'; you do not need to go through every example). When you finish, you'll see the new folders named by each behavior type; the examples you chose will be within them and will no longer be in the original "examples" folder made when you generated the examples (Section III A). There will be a set of behavior folders for each video from which you generated example behaviors. LabGym will automatically compile them into one data set in the "Train Categorizers" step below (Section III C below).

Section III C: Train Categorizers

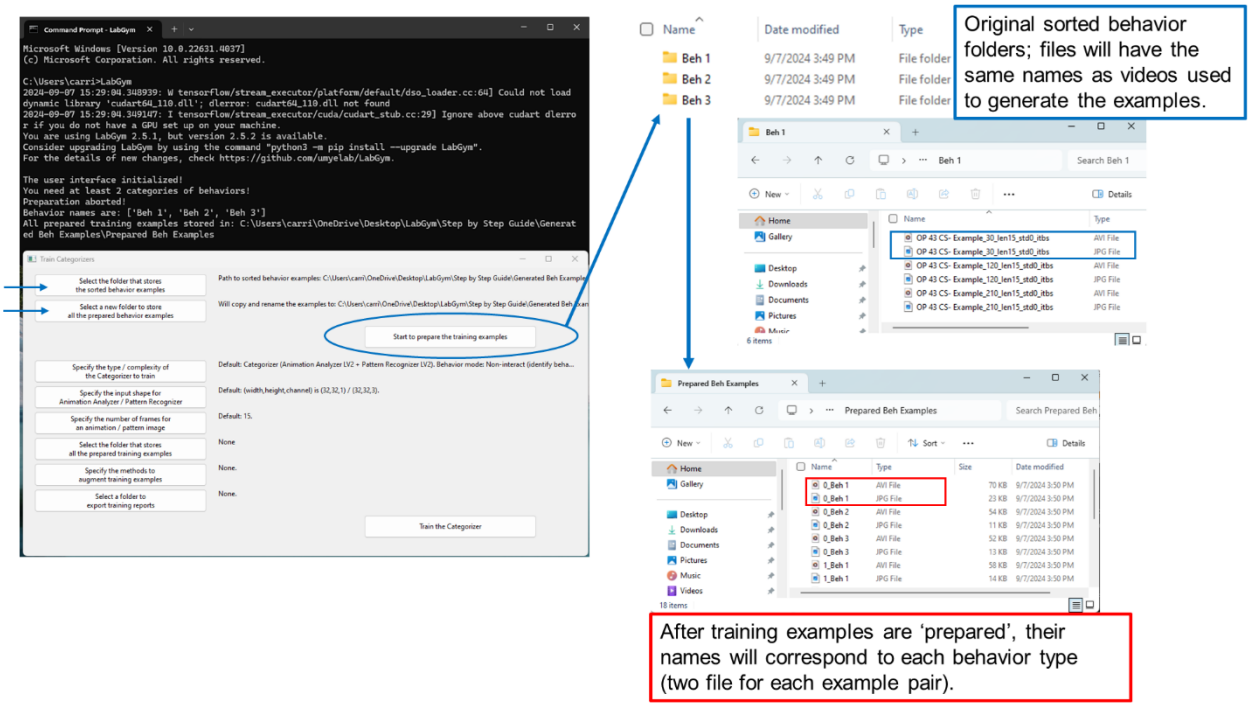
1) First, LabGym will take the sorted behavior examples you generated in the section above (Section III B) and compile them so that there is only one folder with all the behavior types. This is called 'prepare the behavior examples'; these will be used to train the Categorizer.

a. Start LabGym and click the "Training Module" button. In the next window, click the "Train Categorizers" button. This will open a new window.

b. Click the first button to tell LabGym where the folder that contains all the sub-folders for each sorted behavior type is and click on the second button to tell LabGym where to store the new "prepared" behavior training examples. Then

click “Start to prepare the training examples”.

LabGym will tell you when this process is complete and where the files are stored in the command/terminal window and you will see a new folder called “prepared behavior examples



that contains the reorganized “prepared behavior examples” (red boxes below).

Now that the behavior examples are prepared, you will click through the remaining buttons in the “Train Categorizers” window. There are several different options and settings. Below we give some guidelines for making your choices; you can start with the default settings and train a well-functioning Categorizer. However, it does take some trial and error to find the best settings for your behavioral data. The trained Categorizer (also called a ‘model’) will be stored in the Categorizer list in *LabGym* and ready to use as soon as it is made (it will automatically appear within the LabGym GUI).

2) Click the “Specify the type/complexity of the Categorizer to train” button.

2a) The first pop-up will ask you to “specify the mode of behavior for the Categorizer to identify”. This mode should match the choice you made when you “generated the original behavior examples” (Section III A, step 4 above) and follows the same guidelines as given in that section. We have repeated the information here for convenience:

i. “Non-interactive”: choose this option if you want to measure behavior in only one animal within the field of view or if you want to measure behavior in multiple animals, but you do not need to keep their unique identities. Despite the name, behaviors here can include interactions between animals and objects/places in the background (regardless of your method of animal detection, i.e., background

subtraction or a trained Detector, or object detection, i.e., a trained detector or Draw Markers).

ii. "Interactive basic": choose this option if you want to analyze behavior of one animal where behaviors of interest rely on the animal interacting with another animal or objects/places in the background. This mode will not keep track of the individual identity of the animals (e.g., it can measure one animal licking another animal, but it won't distinguish which animal is licking vs being licked).

iii. "Interactive advanced": choose this option if you want to analyze behaviors of one or more animals interacting with each other and/or objects/places in the background. You must choose this option if you are going to measure directed interactions (e.g., animal 1 doing something to animal 2).

iv. "static image": this option is not for use with videos; use this to examine non-interactive behavior in static images (i.e., digital pictures). We do not address this function in this guide.

As above, if you select the "Interactive advanced" option, a new window will appear asking you to define the "social distance". As was the case for generating behavior examples this refers to the distance between any one animal and any other animal *or object* in a frame. If you generated behavior examples in 'interactive advanced' mode, then you already set the 'social distance' value and should use the same value here (you can find that information in the file names of the generated behavior examples: the number after 'scdt' is the social distance value, '…..scdt2…' the value is '2', and you need to enter '2' here.

The next set of pop-ups and windows refer to the properties of the neural network that will be used to generate the trained Categorizer. More complete, scientific explanations of these neural networks can be found in (https://doi.org/10.1016/j.crmeth.2023.100415). Below we give some practical guidelines.

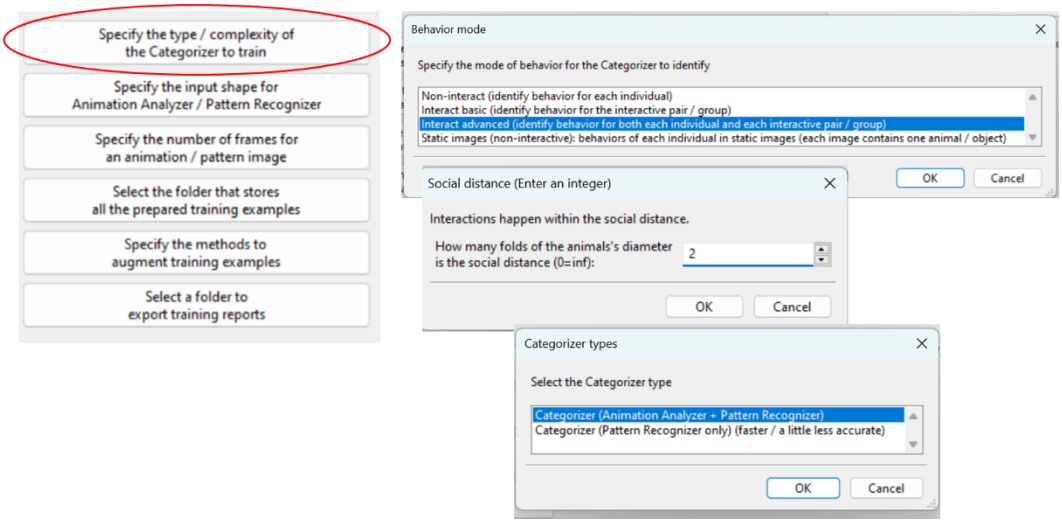
2b) The next pop-up asks you to "Select the Categorizer Type".

i. Categorizer (Pattern Recognizer only) will only use the pattern image from the behavior example pair to train the Categorizer. This method can work well and will be completed relatively quickly (does not require a lot of computational power).

ii. Categorizer (Animation Analyzer + Pattern Recognizer) has the potential produce the highest accuracy but needs relatively more processing power and thus will take longer to complete.

If the processing speed is critical to you, choose the Categorizer with only Pattern Recognizer. If you want to maximize the analysis accuracy and care less about the analysis speed, choose the

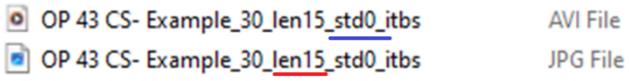
Categorizer with both Animation Analyzer and Pattern Recognizer. *Note, if you choose option i LabGym will only ask you to provide settings for the Pattern Recognizer in the next set of pop-up windows.*



2c) The next pop-ups (Animation Analyzer level and Pattern Recognizer level) ask you to set the "Complexity" (of the neural network that will actually do the 'learning') from 1-7, with 1 being the simplest. You should start in the 1-3 range and increase it from there depending on the accuracy of the trained Categorizer (From Testing the Categorizer in the next section). You can use the same number for both the animation and the pattern level.

If the Animation Analyzer is gray scale (animal/object color is behavior irrelevant), you can make the complexity level of Pattern Recognizer a little higher than that of Animation Analyzer. If you used the 'Draw Markers' function to use colors to mark objects/places, you cannot use a gray-scale Animation Analyzer, you must choose 'no' when it asks you whether the Animation Analyzer is gray scale.

3) Click the next button, "Specify the input shape for Animation Analyzer / Pattern Recognizer". The number here refers to the 'width' of the network architecture. Start by setting this to 16 or 32 (a relatively small size). If many details within each frame are relevant to your behavior (e.g., using whiskers to touch an object) use a large size like 64, 96 or even 128. Do not set this number below 8.



You will likely need to adjust the settings above to improve the performance of your Categorizer.

4) Click the "Specify the number of frames for animation/pattern image" button. This must match the duration (in frames) of the behavior examples that you set in the "Generating behavior examples" function (Section III A). This information is within the file name of the behavior examples you generated above: "_len*XX*_" *XX* is the number you need to enter here (*XX*=15 for the files shown above; red underlined text).

5) Click the “Select the folder that stores all the prepared training examples”. Here it will ask if the examples include the background; if any do, say yes. It will also ask for the “STD for motionless pixels”. This number should match the STD used to generate the behavior examples and is also found in the name of the behavior example file (“std”; for the files shown above this value is 0; blue underlined text).

6) Click the “Specify the methods to augment the training examples” button. This feature is conceptually similar to the augmentation used when generating images to train a Detector; the computer can flip and rotate the animation and pattern images, essentially increasing the information it can has to work with to train a Categorizer. The augmentations here all refer to altering the both the ‘animations’ and ‘pattern images’.

The default augmentation will use '*rotation*', '*flipping*', '*brightening*' and '*dimming*'. Additional choices include '*shearing*', '*rescaling*' and '*deletion*'. We recommend starting with the default setting and adjusting from there.

'*rotation*': will rotate clockwise; '*flipping*' will make a mirror image; '*brightening*' an '*dimming*' will increase and decrease the brightness, respectively; '*shearing*' will introduce distortion; '*rescaling*' will change the width:height ratio; '*deletion*' will delete one or two frames in the animations (this mimics a scenario in which animals are not detected in one or two frames during analysis).

> 6.1) The next pop-up (Augment validation data) will ask if you also want LabGym to perform the augmentations you selected on the ‘validation data’. During training, the training examples will be divided into a ‘training’ set and a ‘validation’ set. The model learns from the training set and tests its accuracy on the validation set after each training iteration to see whether the model improves or not. Choosing ‘Also augment the validation data’ will increase the amount and diversity in the ‘validation’ set and hence increase the statistical power when testing the model’s accuracy after each training iteration. So we recommend that you always select ‘yes’ when being asked whether to augment validation data, unless you have many behavior examples (> 1,0000 pairs of sorted examples before augmentation), or the memory of your computer is low (< 16GB RAM and/or 50 GB free disk space).

7) Click the “Select a folder to export training reports” and select “Yes” if you want to export the training reports as a sheet. Otherwise, the training reports will be just printed in your command/terminal window.

8) Click the “Train the Categorizer” button. It will ask you to name the Categorizer and then it will start the trainning process. LabGym will update you about the process and let you know when it’s finished in the command/terminal window.

Section III D: Test Categorizer

Here you can use the folder containing the “prepared” behavior examples (i.e., those compiled by LabGym after you completed your sorting; Section III C step 1) to test the Categorizer. A well-trained Categorizer should accurately identify the behaviors *You* identified when looking at the animations and patterns. If not, you should adjust the settings in the Categorizer Training.

LabGym will provide quantitative measures of precision, recall, and f1 score for each behavior category, as well as the overall accuracy at the end of testing. Precision refers to how precise the Categorizer is in categorizing behavior. The recall refers to how sensitive the Categorizer is. The f1 score is the weighted average of precision and sensitivity. These metrics range from 0 to 1, with 1 being the best. Generally speaking, a value above 0.6 is considered successful training (i.e., the Categorizer can differentiate each beavhior with a fixed error rate).

$$precision_i = \frac{true\ positives_i}{true\ positives_i + false\ positives_i}$$

$$recall_i = \frac{true\ positives_i}{true\ positives_i + false\ negatives_i}$$

$$f1\ score_i = \frac{2 \times (precision_i \times recall_i)}{precision_i + recall_i}$$

$$overall\ accuracy = \frac{total\ correct\ predictions}{total\ predictions}$$

If your Categorizer is not performing well you can: 1. Check that behavior examples were sorted into the correct categories, 2. Revise the sorting of the behavior examples (e.g., dividing one category into two, combining similar categories, or creating additional behavior categories) 3. Increase the number of behavior examples within a categoriy 4. Try different settings of the Categorizer (complexity level / input size). Complex models might not always be better so we suggest you try simpler models first. In general, we’ve found that the performance of the categorizer is most affected by the number of examples within each behavior category, and being creative about which behaviors we ask LabGym to categorize (e.g., teaching it a few behaviors we may not be specifically interested in, but that can still help it identify our behavior of interest).

Section IV: Analysis and Quantitative Measures

Now that you have trained and tested a Categorizer (and a Detector if needed) you are ready to analyze your experimental videos! LabGym gives you a lot of options below within the analysis module. In most cases, you can use the simplest option and you can decline to use some of the features.

“Analyze Behaviors” (Section IV A): detects and track your animals, categorizes your behaviors of interest, and provides quantitative measures of these behaviors that are organized into easy to work with frame-by-frame and summary data for each video (.xls, .csv; see screen shots below). And automatically generate visualizations of the results. “Mine Results”: seamlessly performs parametric and non-parametric statistical comparisons on quantified behavioral data

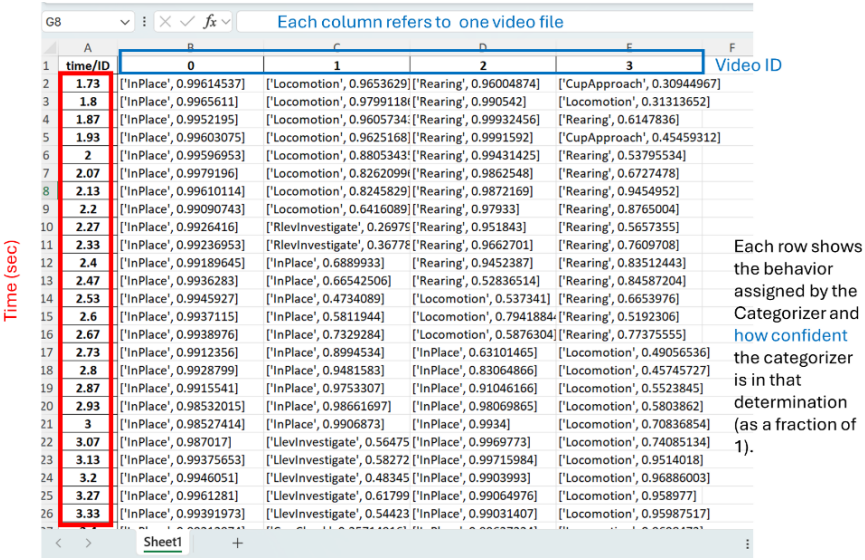
(Section IV B). Finally, “Generate Behavior Plot”: Customizes raster plots of quantified behaviors (Fig 2 above; Section IV C).

LabGym will batch process your videos, and you can customize the quantitative measures that it provides (see Section IV C below). We will walk you through the process for the “Analyze Video” function below. When the video analysis is complete, LabGym will automatically generate a raster plot for all behavior events for all videos (Fig. 2), an annotated video copy for each video, and several spreadsheets storing quantified results for each selected behavior parameter.

Food Cup Behavior Summary

Video ID	ID/parameter	ID/par	count	duration
0	0	0	36	6.55
1	1	0	149	28.46
2	2	0	142	31.47
3	3	0	97	23.35

Count=number of instances of the behavior
Duration = how long (in sec) the behavior lasted across all occurrences



Note: The frame rate of your experimental videos must match that used to train the Categorizer you want to use in the Analysis step. You can modify the frame rate using the Preprocessing Module (See Part 4.5 above).

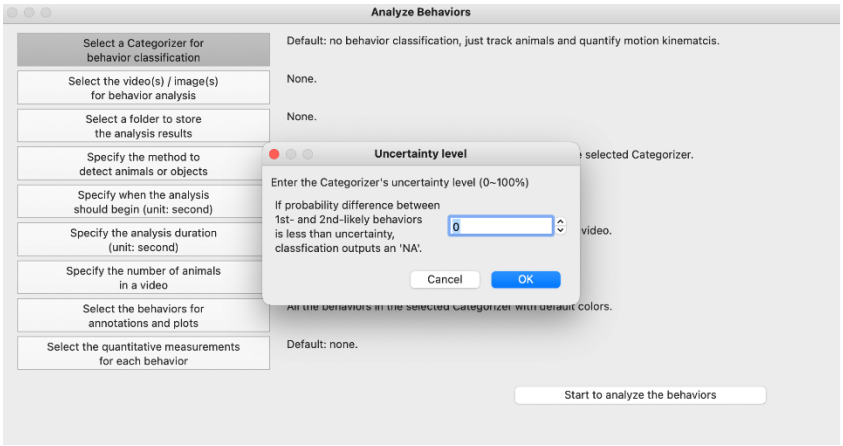
Section IV A: Analyze Behaviors

The concepts that you learned above apply to the analysis settings below. There are a number of options and settings that you can choose from. In general, you should use settings and option choices that are the same as those used during initial training of the Categorizer and/or Detector. Many of the choices are self-explanatory. However, we will make a few clarifications and recommendations below.

1) Click the ‘Select a Categorizer for behavior classification’ button. Choose a trained Categorizer to classify the behaviors in your videos. *Note, you can also let LabGym just track the animals and calculate their motion parameters and body kinematics without using a*

Categorizer. If you choose this option, you will need to specify a time window for calculating the motion parameters and body kinematics.

1a) A pop-up window will ask you to specify an “Uncertainty level” for the Categorizer to use. As described above (Section III B), the Categorizer will always assign behaviors to one of the behavior types you trained it to identify. Here you will set the threshold for the Categorizer to report that the two most likely categories are too similar in probability to make a determination. This number is the difference between the two most likely categories it assigns, as a percent. In this case, the Categorizer assignment will read as ‘NA’ (no category) on the visualizations and in the data file. For example, let’s say the Categorizer determines the probabilities of behaviors A, B, and C are 60%, 10%, and 30%, respectively. If the uncertainty level here is set to be 31, the behavior classification will be ‘NA’, because the uncertainty level exceeds the difference between probability of the most likely behavior (A, 60%) and the second most-likely behavior (C, 30%), which is 30%. Setting an uncertainty level can reduce false positives in behavior classification, since ambiguous classifications will not be included in a behavioral category (and will be reported as ‘NA’). If you want the behavior classification to be more precise, set the uncertainty level to a higher percentage (e.g., 50%). Setting this to 0% would allow for very ambiguous classification. For example if the probability of behavior A is 49.999% and behavior B is 50.0001% with an uncertainty level of 0, it would categorize the behavior as behavior B. This would be bad because the Categorizer would basically be functioning at the level of chance. Finally, the number of behaviors you ask LabGym to identify will influence the predicted difference between potential behavioral categories (If you are quantifying a large number of behaviors, you would want to set this number on the lower side). We recommend starting with 5-10% and refining from there. Importantly, LabGym will annotate the ‘NA’ designation in your videos, so you will be able to watch the analyzed videos and make your own determinations about what the animal is doing during the that period of time and adjust the uncertainty level accordingly.



2) Click the “Select the video(s) / image(s) for behavior analysis” button. *Note, here we are only considering videos; you can use LabGym for static images, but we are not discussing that here.*

2a) The first pop-up window will ask if you want to change the frame size [“(Optional) resize the frames / images?’]. If you already adjusted the frame size of the videos you want to analyze in the Preprocessing section do not do it again here. If you trained your Categorizer

and/or Detector using images from videos that were re-sized, the frame size of the videos you want to analyze must be the same.

3) Click the ‘Specify the method to detect animals or objects’ button. There will be two options: ‘Subtract background’ and ‘Use trained Detectors’. The logic here is the same as it was when you determined the method of animal or object detection to use in the Training modules above.

<u>If you select ‘Subtract background’ you will be asked the following:</u>

'Animal brighter than background', 'Animal darker than background', or 'Hard to tell'. Select the most appropriate option for your videos.

A pop-up window will ask ‘load existing background?', this is optional. If you used the ‘subtract background’ approach above when you generated behavior examples (Section III A, step 7), LabGym automatically extracted backgrounds (as image files) for each video you processed. Therefore, this pop-up option can be used to save the step of background extraction here if the background for the videos you want to analyze have already been extracted (just select that folder in the GUI). If not, skip this step and conduct the background subtraction as part of this analysis.

You will also be asked if “the illumination in videos is unstable”. Select yes/no as is appropriate for your videos.

Finally, you will be asked to specify a time window for background extraction (whether or not you loaded an existing background). An appropriate time window for background extraction should be a period during which the animal(s) moves around the frame (typically 10~60 seconds is sufficient). This time window should be as short as possible to increasing processing speed. There are 3 options to specify this time window:

i) 'Use the entire duration of a video' (not recommended if processing speed is critical.
ii) 'Decode from filenames: _xst_ and _xet_': This option is useful if you are analyzing multiple videos *and the time window for background extraction is different for each video*. Here ‘xst’ indicates the extraction start time and ‘xet’ indicates the ‘extraction end time’, where ‘t’ is an integer of time in seconds. ‘xst’ and ‘xet’ need to be added to the file name of each video. For example, if the original video is named Rat1_day1.avi, to set the time window for background extraction from the 25th second to the 47th second, rename the video file to Rat1_day1_xs25_xe47.avi
iii) 'Enter two time points' to define the window for background extraction. This should be used if only one video is selected, or if multiple videos share the same time window for background extraction.

If you select ‘Use trained Detectors’ you will be asked the following:

(Button) ‘Specify when the analysis should begin (unit: second)’

(Pop-up window) ‘Illumination shifts?’. Here you will specify whether there are sudden bright-to-dark or dark-to-bright illumination transitions. If you choose ‘Yes’, there will be 3 options to specify the beginning time: 'Automatic (for light on and off)', 'Decode from filenames: _bt_', and 'Enter a time point'. If you choose ‘No’, only the latter two options will be available.

i) 'Automatic (for light on and off)' uses automated dection of dramatic changes in illumination to identify when to begin the video ‘analysis’ (e.g., if the animal is in a completley dark enclosure and then the overhead lights come on, LabGym can detect this light-on as the point in time to start analysis).

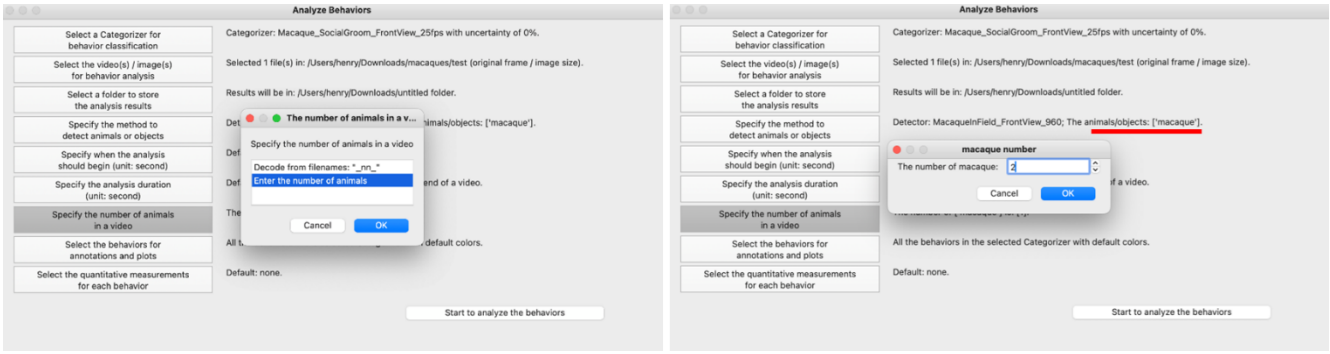
ii) 'Decode from filenames: _bt_' can be used if multiple videos are selected and the beginning time for analysis of each video is different. You need to add ‘_bt_’ to the file name where ‘t’ corresponds to the beginning time in sections. For example, if a video name is Rat1_day1.avi, to set the beginning time to 12.35 seconds, rename the video file to Rat1_day1_b12.35.avi

iii) 'Enter a time point' can be used to define when within the video the analysis begins. This should be used if only one video is selected, or if multiple videos share the same beginning time for analysis.

(Button) ‘Specify the number of animals in a video’. This is asking the same thing as “Specify the number of animals in a video” in Section III A step 9 and follows the same rules given there. There are two options: ‘Decode from filenames: _nn_' and 'Enter the number of animals'.

i) 'Decode from filenames: _nn_' can be used if multiple videos are selected and the number of animals in each video is different. The code tag ‘_nn_’ can be added to the file names, where the second ‘n’ indicates the number of animals (must be an integer). For example, if a video name is “Day1.avi”, to set the number of animals to 8, rename the video to “Day1_n8.avi”. If you are using a Detector to detect animals and objects, you can add an ‘_nn_’ code for each item. For example, let’s say my Detector is trained to detect the following: “rat”, “food cup”, “lever” and in a given video there is one rat, two food cups and three levers, I would name my file: “Day1_n1_n2_n3.avi”. Note, that the order of each n defined here must be consistent with the order of each object in the trained Detector (when you select a Detector, the animal / object name will display in order).

ii) 'Enter the number of animals' should be used if only one video is selected, or the batch of video to analyze have the same number of animals/objects.



(Button) ‘Select the behaviors for annotations and plots’; self-explanatory.

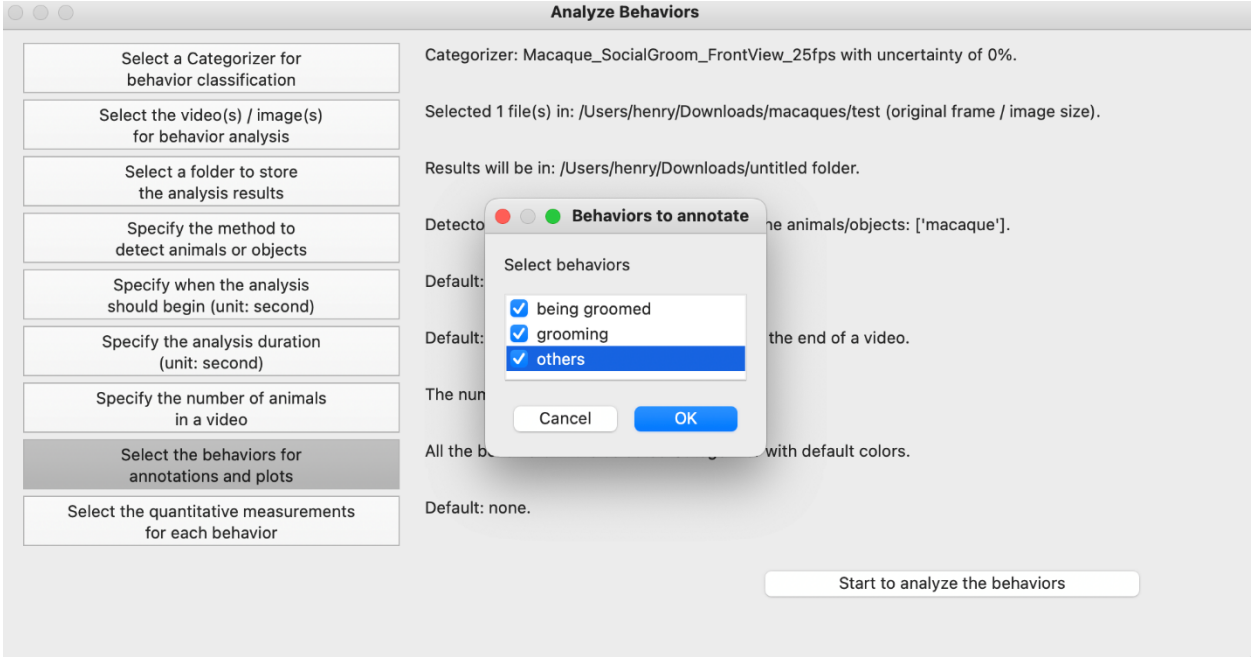
(Pop-up window) ‘Specify individual-specific behaviors?' (optional)

If the selected Categorizer is for “Interactive advanced” mode, you can select individual specific behaviors to initiate the “behavior-guided” identity correction. For example, let’s say you have two indiviuals in your video (indvidual “0” and individual “1”), and only individual “0” can show behavior A. If you specify here that behavior A only occurs in invidual “0”, then when LabGym detects behavior A it will default to calling this individual “0”. This can help LabGym to correctly maintain individual identities when two animals are interaction. This is useful if for example you’re studying mating behavior in a pair of animals and only one animal in the pair will ever show mounting behavior.

(Pop-up window) 'Specify colors for behaviors?' (optional)

Specify a color to represent a behavior category in the annotated videos and the raster plot for behavior events. If choose not to specify the colors, LabGym will use the default colors to represent the behaviors (Fig 4 colored outlines). In the annotated videos, the % confidence of behavior categorization will be shown; in the raster plot, the color intensity indicates the value of % confidence, from 0% of the color intensity (clear) indicating 0% of the confidence, to 100% of the color intensity indicating 100% of the confidence (Fig 2 above).

(Pop-up window) 'Legend in video?' This is asking if you want to show the legend of behavior names in the annotated videos.



(Button) 'Select the quantitative measurements for each behavior'. There are 13 quantitative measurements (parameters) for each behavior that you can choose to use, as appropriate to your behaviors of interest. Details of how each measure is calculated can be found in our publications ([1] [2])

The count is the summary of the behavioral frequencies, which is the occurrence number of a behavior within the entire duration of analysis. Consecutive single occurrences (at a single frame) of the same behavior are considered as one count.

The latency is the summary of how soon a behavior starts, which is the time starting from the beginning of the analysis to the time point that the behavior occurs for the first time.

The duration is the summary of how persistent a behavior is, which is the total time of a behavior within the entire duration of analysis.

The speed is the summary of how fast the animal moves when performing a behavior, which is the total distance traveled (can be back and forth) (d) (between the two centers of mass of the animal) during the time window (t_w) for categorizing the behavior divided by t_w.

The velocity is the summary of how efficient the animal's movement is when performing a behavior, which is the shortest distance between the start and the end positions (dt) (between the two centers of mass of the animal) divided by the time (t) that such displacement takes place.

The acceleration / velocity reduction is the summary of how fast the animal's velocity changes while performing a behavior, which is the difference between maximum velocity

(v_{max}) and minimum velocity (v_{min}) divided by the time (t_v) that such velocity change takes place.

The distance is the total distance traveled of the animal by performing a behavior within the entire duration of analysis.

The intensity (area) / intensity (length) is the summary of how intense a behavior is, which is the accumulated proportional changes of the animal body area (a) / length (l) between frames divided by the time window for categorizing the behaviors (t_w) when performing a behavior.

The magnitude (area) / magnitude (length) is the summary of the motion magnitude, which is the maximum proportional change in animal body area (a) or length (l) when performing a behavior.

The vigor (area) / vigor (length) is the summary of how vigorous a behavior is, which is the magnitude (area) / magnitude (length) divided by the time (t_a or t_l) that such a change takes place.

(Pop-up window) 'Normalize the distances?': this is asking if you want distance measures to be given in pixels or pixels normalized by the size of your animal.

If your choose ‘No’, the unit of measure for all distance related measurements will be given in ‘pixels’ or pixel related. If choose ‘Yes’, all the distances (calculated in pixels) will be normalized to (i.e., divided by) the size of a single animal (also calculated in pixels). In this scenario, all distance related measurements will be normalized measurements (e.g., normalized speed) and will not have a unit. This normalization is useful because you will not need to worry about the ratio of pixel to actual size (e.g., length) across different videos with different recoding methods or distances. As long as the size of animals used in these videos are consistent, the normalized measures will be comparable. The ratio of pixels to actual size (length) is not easy to determine and is subject to change easily (e.g., when the zoom-in level changes or distance of the camera to the subject changes). With the option of normalizing distances to the size of a single animal, you can compare the analysis results across different recordings or experimental sessions without worrying about the changes in the ratio of pixel to actual size.

Drum roll please….Click your final button…. ‘start to analyze behavior’

When you click ‘start to analyze behavior’ and while its running, you’ll see the progress update in the command/terminal window as it completes each step and some information about your video files (blue text). An example of this text is below. When it finishes, it will also tell you where the analyzed files and videos are located (you defined this in the analysis set up, but if you forget, it will be in the command/terminal window; green text below).

Preparation started...

2024-08-15 13:04:18.105329

The total categories of animals / objects in this Detector: ['mag_entry', 'magazine', 'rat']

The animals / objects of interest in this Detector: ['magazine', 'rat']

The inferencing framesize of this Detector: 480

Video fps: 25

The original video framesize: 720 X 1280

The resized video framesize: 270 X 480

Preparation completed!

Acquiring information in each frame...

2024-08-15 13:04:18.895792

1000 frames processed...

2024-08-15 13:06:11.364059

2000 frames processed...

2024-08-15 13:08:12.447904

3000 frames processed...

2024-08-15 13:10:19.772655

4000 frames processed...

2024-08-15 13:12:32.631226

5000 frames processed...

2024-08-15 13:14:43.645994

6000 frames processed...

2024-08-15 13:17:01.487826

7000 frames processed...

2024-08-15 13:19:23.713683

The area of magazine is: 2173.438244820688.

The area of rat is: 4637.587047707117.

Information acquisition completed!

Crafting data...

2024-08-15 13:20:35.004910

Data crafting completed!

Categorizing behaviors...

2024-08-15 13:20:35.020532

235/235 [==============================] - 2s 7ms/step

235/235 [==============================] - 1s 6ms/step

Behavioral categorization completed!

Annotating video...

2024-08-15 13:20:45.881105

Video annotation completed!

Quantifying behaviors...

2024-08-15 13:21:24.480983

Behavioral quantification Completed!

Exporting results...

2024-08-15 13:21:24.496601

All results exported in: C:\Users\19122\Desktop\LabGym related\Lauren\ch16_Test Box 16_OR 26_CS+

Exporting the raster plot for this analysis batch...

2024-08-15 13:21:27.355780

The raster plot stored in: C:\Users\19122\Desktop\LabGym related\Lauren

Analysis completed!

Section IV B: Mine Results

This module performs parametric and non-parametric statistical analysis of behavioral measures that you select, between groups that you select. It will automatically display detailed data for statistically significant differences. The statistical tests available are the same as those within GraphPad Prism 9.

These statistical analyses include: The Shapiro test to assess the normality of the data distribution. For normally distributed data LabGym can conduct unpaired or unpaired t-tests between two groups and one way ANOVAs for three or more groups with either Tukey (comparing each pair) or Dunnett's (comparing all groups against the control group) post hoc comparisons. For data that is not normally distributed, LabGym can conduct Mann Whitney U, Kruskal Wallis, Wilcoxon or Friedman tests with Dunn’s post hoc comparisons. The selections of the tests are consistent with those in GraphPad Prism 9.

Section IV C: Generate Behavior Plot

Generate a raster plot for all behavior events of every individual in ‘all_events.xlsx’ file, which is output by LabGym analysis. This function can be used to re-generate the raster plot if the original raster plot generated by LabGym does not meet the users’ need (e.g., you want to eliminate some categories from the visualization, or you have selected a subset of frames to include).

USEFUL INFORMATION & DEFINITIONS

Modifying Videos:

1) The frame size must be the same for videos to be analyzed and those used to train the Detector and Categorizer. You can reduce frame size by batch processing all your videos in the preprocessing module, or by batch processing the videos you selected to use to train LabGym

within the training modules (e.g., if you have not yet collected all your experimental video data, but have enough to start training LabGym).

2) If you use the preprocessing module to reduce frame size or frame rate of your videos, you must make the same adjustment to all videos within one experiment (i.e., any data that will be directly compared), including videos used to train the Categorizer and Detector.

3) If you use the preprocessing module to reduce frame rate you should not adjust this again in any other module (e.g., Training Detector). Any additional changes you make will be additive.

Definitions:

Static background: lacking in movement, action, or change (e.g., illumination, content, shape)

GUI: Graphical user interface. Basically, any window with buttons you can click on or menus to choose from is a GUI.

GPU: Graphics processing unit in your computer (and other devices) that helps conduct graphics-related work including videos (often used for high speed gaming).

CPU: Central Processing unit is the component of a computer that conducts all computing processes (interpreting and executing instructions from software programs, performing calculations, and controlling the flow of data within the system). The CPU, also referred to as the “central” or “main” processor.

FPS: Frames per second; the rate of video capture.

Background: Any aspect within the video that is not your animal(s). Often ‘background’ refers to information in the video that is not relevant to detecting your animal or behaviors.

Detector Training: Teaching LabGym to identify animal(s) and any object or place in the environment it needs to “know” about to identify a behavior of interest. Objects and places can be static (e.g., location of a novel object), or dynamic (e.g., an item the animal can pick up and manipulate). Through the detector training process, you are “teaching” a neural network to generate a model that is then applied to your videos to detect animals, objects or places.

Detector (i.e., a trained Detector): is a deep learning model that can detect and segment objects that are defined by you. It is essentially a model generated by a neural network that is applied to your videos to detect animals, objects, or places.

Categorizer Training: Teaching LabGym what each behavior looks like; this results in a “model” that you apply to Categorize behavior (i.e., identify). Through the Categorizer training process you are “teaching” a neural network to generate a model that is then applied to your

videos to detect and then categorize behaviors. The “model” is the “categorizer”

Annotating Images: A process through which you label animal(s), objects, or places in still frames taken from your videos. This is done using an external resource called “RoboFlow”, which makes labeling (i.e., annotating) very fast.

Image examples: Generated as part of “Detector Training” and used to train the detector to identify your animal. These are essentially still frames generated by LabGym from your videos.

Behavior examples: Generated as part of “Categorizer Training” and used to train the Categorizer to identify discrete behaviors. Are comprise of ‘pairs’ of files (Fig 7); an animation (.AVI file, Fig 7A) and a corresponding “pattern image” file (.jpg, Fig 7B). These files are generated by LabGym from your videos.

Total Loss/Val (validation) Loss: metrics that measure the overall error rate when training a Detector (you can monitor the “total loss”) or a Categorizer (you can monitor the “val_loss”). You can monitor these metrics during the process of training. Ideally, the ‘total loss’ of a Detector training goes below 0.2 when the training approaches to the end; the ‘val_loss’ of a Categorizer goes below 0.5 when the training approaches to the end.

Consistency testing: Analyzing the same videos you used to train the Categorizer. You should be able to get out what you put in, if not, then you likely need to add examples to your training set.

Generalizability: Analyzing different videos than you used to train the Categorizer. You should be able to apply your Categorizer to videos with the same experimental set up and features.

Mean average precision (mAP; from 0-100%): Is a metric that is used to determine how well a trained Detector is performing (>60% is reasonable, 100% is likely impossible for most behaviors). The mAP averages over all the types of things you annotated, so it can be low if some things are being poorly detected while others are being detected accurately. For this reason, it helps to view the images generated by Testing the Detector.