

GUI - User Manual

Edited 01/09/25 by Madelyn Hair

Blue text is to indicate code to be entered in the terminal/Command Prompt. After each line of code, press Enter.

[All updated scripts](#) can be found and downloaded in a google drive (Request access), along with a folder for the Project Leaders to use containing the tools and instructions to [sync paired videos](#) and extract video information.

Part 1- GUI Guide

Part 1 of this guide provides instructions on how to operate the LocalAnnotationsBitesGUI_mmdd.py and correctly submit annotations. This part simply collects annotations of clicked points on the video frames, does not run SAM2 and thus does not require GPU access, and can be done on a local machine.

First-time Users:

1. If you haven't already, create a Gil_Lab folder on your computer, and within that create a folder for your project (e.g., Damselfish or Stationary_Array), then create a subdirectory within that for videos you wish to annotate.
 - a. Note: Having spaces within folder names can cause issues when using the command line. Use "_" instead of spaces to make your coding life easier.
2. Download the most-recent LocalAnnotationBites.py script from [here](#) and save within your Gil_Lab project folder.
3. Download [Python](#) (Version 3.12.7) and follow the instructions to install on your computer. Make sure you tick the box, 'add to path' at the beginning of the installation.
 - a. Confirm that you have installed Python by opening Terminal (Mac) or Command Prompt (Windows) and typing:
`python -V`
Or
`python3 -V`
 - b. You should see a message return "Python 3.12.7"
 - c. If only python -V or python3 -V return the above message, you must use 'python' or 'python3' respectively every time you write python in the following steps.
4. Install pip:
 - a. Open Terminal or Command Prompt and copy the following lines of code:
`curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py`
`python3 get-pip.py`
 - b. Confirm that you have pip installed by opening Terminal or Command Prompt and typing:

`pip -V`

or

`pip3 -V`

- c. You should see a message return something like “pip 24.2 from /opt/anaconda3/lib/python3.12/site-packages/pip (python 3.12)”
 - d. If pip and/or python is not correctly installed, contact your project leads or Madelyn for assistance before continuing.
 - e. As above, if only pip -V or pip3 -V return the above message, you must use ‘pip’ or ‘pip3’ respectively every time you write pip in the following steps.
5. Install requirements (Packages): Run the following line of code in Terminal or Command Prompt:

`pip install opencv-contrib-python tk customtkinter numpy pillow pandas`

Or

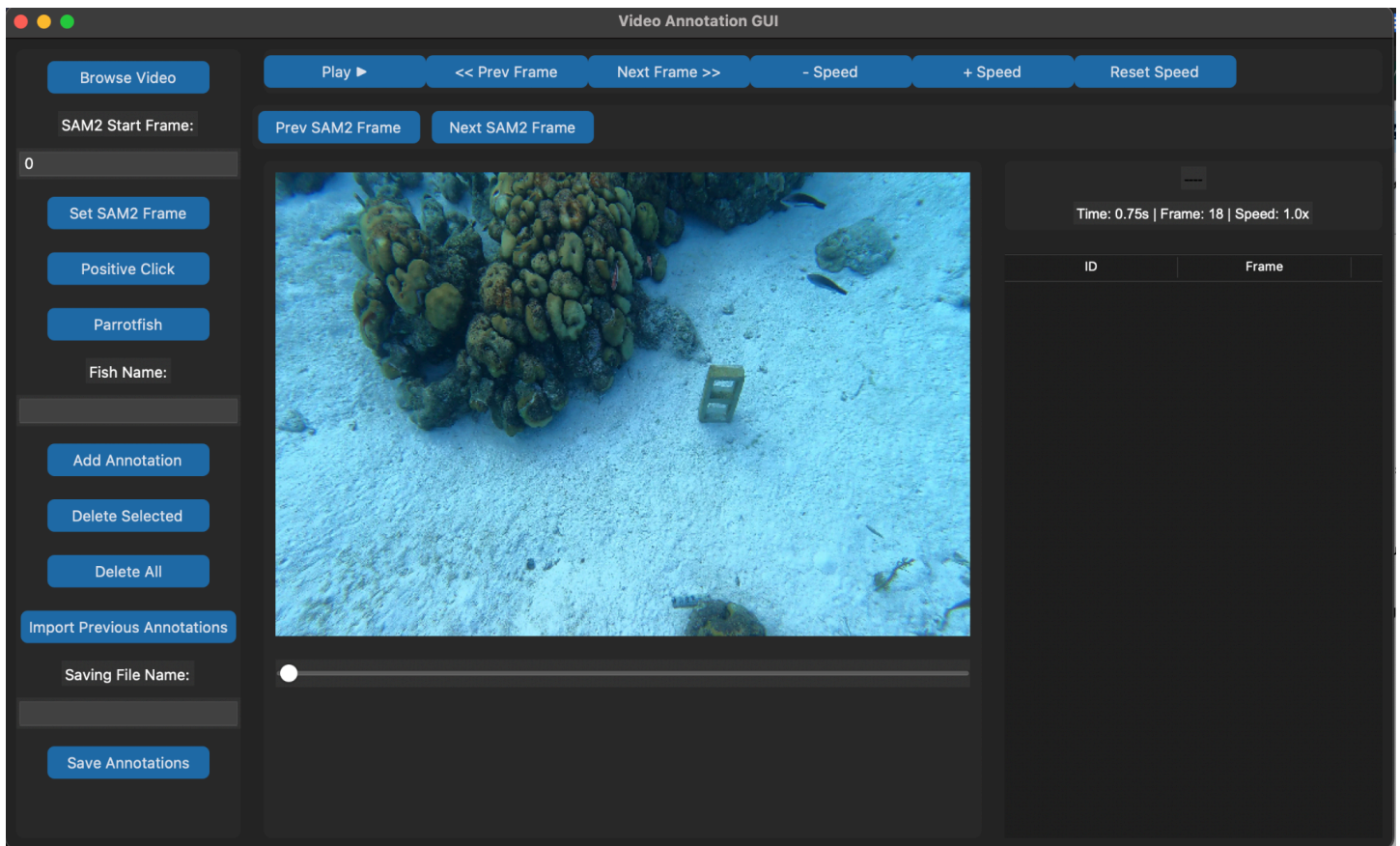
`pip3 install opencv-contrib-python tk customtkinter numpy pillow pandas`

Note: The set-up can be the most challenging with different operating systems. Once all packages are installed, you should be able to run the GUI smoothly. If you are having difficulties installing packages, refer to google or contact your project lead/Madelyn for assistance.

Using the GUI:

1. Download the video you wish to annotate and within the videos folder of your project folder.
2. To run the Local Annotations GUI, open terminal and enter your project’s working directory by typing (substituting your local computer’s path to the working directory where the video and scripts are saved)
`cd path/to/Gil_Lab/ProjectName`
`python LocalAnnotationBitesGUI_0127.py` (or latest file name)
3. After a few seconds or up to a minute, the GUI should open in a new Python window on your computer.
4. Operating the GUI is very straightforward. First, Click “Browse Video” and navigate to the video you wish to annotate. Double check that this is the correct video according to your project’s instructions!
 - a. Note: Loading a video can take a long time, depending on the size. Python will also say “Application not responding”. This is normal, be patient. Once the video loads python should operate smoothly.
5. The “Start SAM2 Frame” and “Set SAM2 Frame” options are designed for SAM2 Tracking at a lower frame rate (3fps) than the original video. If SAM2 tracking is an aspect of your project, refer to your project’s specific instructions on how to find the Start SAM2 frame. Instructions on how to calculate the SAM2 start frame are [here](#). Otherwise, leave as the default (0).
6. Press Play and Pause to move through the video as you’d like to find frames to annotate. Use the Prev Frame and Next Frame buttons to move through one frame at a time. You can also use the slider below the video player to move forward and backward in time. You can adjust the playback speed with the - Speed and + Speed buttons, or reset it to 1. Once you find your desired frame to annotate, pause the video.

7. There are three Click types that you can toggle between:
 - a. Positive Click: To mark points where your fish is located on the frame.
 - b. Negative Click: To mark points where your fish is NOT on the frame. Use this to select points just outside of the fish body so the model can learn the boundaries of your fish and mask it accurately.
 - c. Bite: Use this to mark when and where your fish is taking a bite from the substrate (i.e., it's mouth is nearly touching the substrate).
8. There are four Fish Family options for our purposes: Parrotfish, Surgeonfish, Damselfish, or Other.
9. Fish Name should be a unique integer label (e.g., 1, 2, 3, etc.) as your fish label. It's best practice to collect all three types of clicks for fish 1 before moving on to fish 2.
10. When your click-type is toggled to the correct option and your fish name is entered correctly, click on the point in the video that you wish to annotate, then click Add Annotation.
 - a. You can also press Enter to add an annotation.
 - b. You should observe a new annotation appear in the table on the right.
 - c. Review and confirm the click information (click type, fish label, and coordinates).
 - d. If you make a mistake, you can select that annotation and choose "Delete Selected"
 - i. You can delete multiple selected annotations, but only about 5-10 at a time.
11. Follow your project's specifications about how many clicks to make. For a well-defined fish individual, you will typically only need a single positive click on ~5 frames for SAM2 to produce the most accurate track, **particularly targeting moments when the fish has re-entered the view or is about to leave the view**. You will likely need to watch the video carefully and record a bite click for every bite that is observed by your fish of interest.
 - a. Note: Only make positive and negative clicks on SAM2 frames. Use the arrow keys to quickly advance to the next SAM2 frame.
12. Repeat this process for every fish of interest, **making sure to update your Fish Name** with each new fish.
13. To save, enter your Observation ID in the File Name textbox, then press Save Annotations. You can do this repeatedly throughout the video to save your progress.
14. When you have completed your annotations, click "Save Annotations" again. Confirm that a ObservationID_bites.csv and ObservationID_annotations.csv has been created in your working directory, then exit the GUI.



Part 2 - Alpine Connection and Set-up

Installing SAM2.1 on Alpine (first-time users)

Note: These instructions are specific to using CU Boulder's HPC called Alpine. If you are not a CU student or faculty member, you will need to adapt these instructions to your own institution's HPC or another GPU machine.

1. From a login node, start an interactive session on a GPU node:

```
sinteractive --partition=atesting_a100 -N 1 -n 10 --gres=gpu:1
```

...once the job starts, you'll be on a GPU testing partition, which will facilitate installation of this environment, which requires CUDA.

2. Create a conda environment with a recent version of pytorch, as well as jupyter and matlab:

```

$ module load miniforge
$ conda create -y -n sam2.1_pytorch241_cuda124 -c conda-forge python=3.12
$ conda activate sam2.1_pytorch241_cuda124
$ pip3 install --no-cache-dir torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cu124
$ pip3 install --no-cache-dir ipykernel matplotlib==3.7.0
$ cd /projects/$USER/software
$ git clone https://github.com/facebookresearch/sam2.git
$ cd sam2
$ pip3 install --no-cache-dir -e .
$ cd checkpoints
$ ./download_ckpts.sh
$ cd ..

```

Next, you'll need to install some packages into your environment:

```

$ python3 -m pip install opencv-contrib-python
$ python3 -m pip install customtkinter
$ python3 -m pip install tkvideoplayer==2.3
$ conda install -c conda-forge "numpy<2"
(for now, ignore) $ conda install -c conda-forge numpy=1.16.4

```

now test the environment to make sure packages load (just load python from command line rather than start jupyter notebook)

```

$ python3
Python 3.12.7 | packaged by conda-forge | (main, Oct 4 2024, 16:05:46) [GCC 13.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from sam2.build_sam import build_sam2
>>> from sam2.sam2_image_predictor import SAM2ImagePredictor
>>> import customtkinter

```

If you get no error messages and no 'module not found' errors, you're good to go!

Command Line Tips

To exit python and return to the "shell", press control d (for MacOS) or control z then Enter (for Windows)

To exit your computing node and return to a login node, type `squeue -u $USER`

This will show you your active and pending job(s). Copy the JobID that is printed from your running interactive job, then type `scancel JobID` (e.g., `scancel 11325241`) - this will close your connection to the computing node and return you to your login node.

Part 3 - Extract Frames and Run SAM2

Extracting Frames

Before running the VideoPredictor script, you will first need to create a folder containing .JPG files for each frame in your video. To do so, simply follow the instructions below.

Madelyn's step-by-step instructions:

Substitute your own identikey (e.g., maha7624) and your own video_name (e.g., GH010367) in the code, where applicable.

1. Log into your RC account through terminal or Putty (Host name: login.rc.colorado.edu) and navigate to your scratch directory:
`cd /scratch/alpine/identikey`
2. If it does not already exist, create a folder for SAM2: `mkdir SAM2`
3. Enter that directory and create a subdirectory for your video and another to store your frames:
`cd SAM2`
`mkdir video_name`
`cd video_name`
`mkdir videoname_frames`
`cd ..`
4. Upload the following materials (found [here](#)) to your **video_name folder** (NOT frames folder) using [OnDemand](#) or `scp` from terminal (scp and other file transfer instructions [here](#)):
 - a. VideoPredictorV10.4.py (latest version)
 - b. run_vidpred.sh
 - c. Your original video (ex: GX079647.mp4)
 - d. Your annotations for this video (ex: _annotations.npy)
 - e. Upload the Extract_Frames.sh file into your video_name folder.
5. For Extract_Frames.sh. (If you are working in command line and not OnDemand, you can edit scripts in command line by typing `nano Extract_Frames.sh`)
 - a. Change the email address at the top to yours.
 - b. edit the line to input of your start_frame (for synced video projects only, should be 0 as default).
 - c. Edit the line of ffmpeg code to input your video_name at the beginning and end of the line
 - d. Save your changes and exit the script editor.
 - e. Run the script by typing: `sbatch Extract_Frames.sh`

Run SAM2 Video Predictor on a virtual machine with GPU access:

1. `cd` into your scratch/alpine/userID/SAM2/videoname directory. Type `ls` and press Enter to ensure that within this folder you have the VideoPredictorV10.4.py, the annotations.npy, the run_vidpred.sh, and a folder containing the frames from your video.
2. To run the VideoPredictorV10.4.py script, you will first need to make some edits:
 - i. Edit the script to reference your frames directory:

[nano VideoPredictorV10.4.py](#)

- Edit line 20 to the name of your annotations.npy file
 - EX: "EH_051524_site2_east_A_Left_GX137102_annotations.npy" (change this to the actual name!)
- Edit line 28 to be the frame rate of your original video (find this information from the video metadata. For windows, this is right click on the video file, properties, details, scroll down to frame rate)
- Edit line 29 to be the SAM2_start frame of your video (if N/A write 0).
- Edit line 33 to the frames directory (where your frames were saved from the step above).
 - EX: "./GX137102_frames"

2. Exit and save your changes with the same file name.

3. Now, you should also make some edits on the run_vidpred.sh file:

[nano run_vidpred.sh](#)

- a. As the last #SBATCH command, there should be a --mail-user argument. Change the email address to be your own CU email address.
- b. Confirm that the script referenced by python3 after #Run VideoPredictor script is the updated script version that you are using.
- c. Exit the editor and save as the same run_vidpred.sh name

4. Submit the SAM2 Video Predictor script as a scheduled job to run when the GPU resources are available:

[sbatch run_vidpred.sh](#)

5. You should get a message like:

sbatch: Note: Users are limited to their jobs collectively using up to 1/2 (2/3) of the total GPUs in the "aa100" ("ami100") partition, and additional jobs will remain queued. This partition is heavily used and it is not uncommon for wait times to exceed 24 hours during peak periods. Please be aware that "squeue" shows the temporal order that jobs were scheduled, and is not an accurate indicator of the position of your job(s) in the queue. The wait time for any job is a function of the recent usage by the user and their institution, the specified size and duration of the job, the qos, and the age of the job.

Submitted batch job 11091198

6. When your script has begun running, you will get an email from SLURM. When your script is done running (after about 20-30 min, depending on video length), you will get another email.
7. When your job is completed, check the results by going to your video_name folder in OnDemand and downloading the output.mp4 video. This will show the predicted masks on top of your video at the reduced frame rate.
8. If SAM2 is correct, congratulations! If not, stay tuned. Instructions on how to re-submit corrections to SAM2 are in development.

Part 4 - Correcting SAM2 predictions

While SAM2 performs very well for focal follows and only one tracked object that is frequently in frame, SAM2's accuracy decreases significantly for stationary videos where there are multiple fish to be tracked, and many of those individuals will leave the field of view and not return. SAM2 will frequently try to re-identify those individuals from other fish, drifting debris, etc. If you watch your output_video.mp4 from the SAM2 tracking and notice mistakes in the track (positive or negative errors, see below), then you should follow the instructions below.

False Positive error: SAM2 has masked an object that should not be masked.

False Negative error: SAM2 has not masked an object that should be masked.

1. Open the GUI and load your original video. If applicable, change the SAM2 Start Frame.
2. Open the output_video from SAM2's results. Position the two video players next to each other.
3. Begin watching the output video slowly, using the arrows to move frame by frame. Pause when you come to an erroneous frame.
4. Use the second number in the output video frame title to determine the relevant frame on the GUI. Advance to that frame on the GUI. Determine the type of error present,
 - a. False Positive error: Change your fish label to the ID that erroneously appeared. Change your click type to "Negative". Click on the object that was mistakenly masked and add multiple (~3) negative annotations on and around that object.
 - b. False Negative error: Change your fish label to the ID that should have appeared- You can typically move forward or backward a few frames in the video to see what fish label you had used for that fish. Change your click type to "Positive". Click on the fish in the center of its body and add positive annotations on that fish. Then change your click type to "Negative" and add 2-3 negative clicks on the substrate surrounding the fish.
5. When you have added annotations to correct SAM2's error, continue watching the output_video slowly to find any remaining errors. Repeat steps 3 and 4 until you have completed the video and addressed all errors.
6. Save your annotations as ObservationID_correction. Confirm that an ObservationID_correction_annotations.npy file has been created in your working directory.
7. Next, you will append your correction annotations to your original annotations using the following lines of code in terminal/command prompt. Replace with your path/to/working/directory and Observation ID file names.

```
cd path/to/working/directory
python
import numpy as np
np.save("observationID_combined_annotations.npy",
np.append(np.load("observationID_annotations.npy", allow_pickle=True),
np.load("observationID_correction_annotations.npy", allow_pickle=True)))
```
8. Now, upload the observationID_combined_annotation.npy file to your video's scratch directory (your working directory in CURC).
9. Open up the [VideoCorrectorV1.py](#) script in a script editor.
 - a. Change your annotations_file to "observationID_combined_annotations.npy", the file containing your previous annotations and your correction annotations.
 - b. Adjust the fps value and SAM2 start_frame as necessary.
 - c. Save the script and upload it to your working directory in CURC.
10. Open up the [run_vidcorr.sh](#) script in a script editor.
 - a. Change the #SBATCH --mail user=maha7624@colorado.edu to your own CU email address.
 - b. Save the script and upload it to your working directory in CURC.

11. Now, open Terminal or Command Prompt, log-in to your RC account, navigate to your working directory, and run the video corrector:
`sbatch run_vidcorr.sh`
12. When the job has completed, check your corrected_output_video.MP4 for lingering errors. If necessary, repeat steps 1-11, making sure to always include all of your previous annotations with your correction points.