Annotator GUI & SAM2 User Manual

Edited 03/10/25 by Madelyn Hair.
This workflow is intended to process videos from underwater cameras, create annotations that sample fish positions, and automatically mask and track fish of interest on every extracted frame. These mask results, when generated from synced stereo cameras, can be combined with the 3D habitat model and camera positions to triangulate fish positions in 3D space and recreate the fish trajectory within the habitat.

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 from the public GitHub Repository, Annotator_GUI. They can also be found in this SAM2 Google Drive folder.

Note for Project Leaders: the tools and instructions to sync paired videos and extract video information / SAM2 start frames are provided in this google drive folder.

Part 1- GUI Guide

Part 1 of this guide provides instructions on how to operate the LocalAnnotationsBitesGUI_0127.py and save annotations of fish positions and bites. 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.

Setting up for first-time GUI 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 LocalAnnotationBitesGUI.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

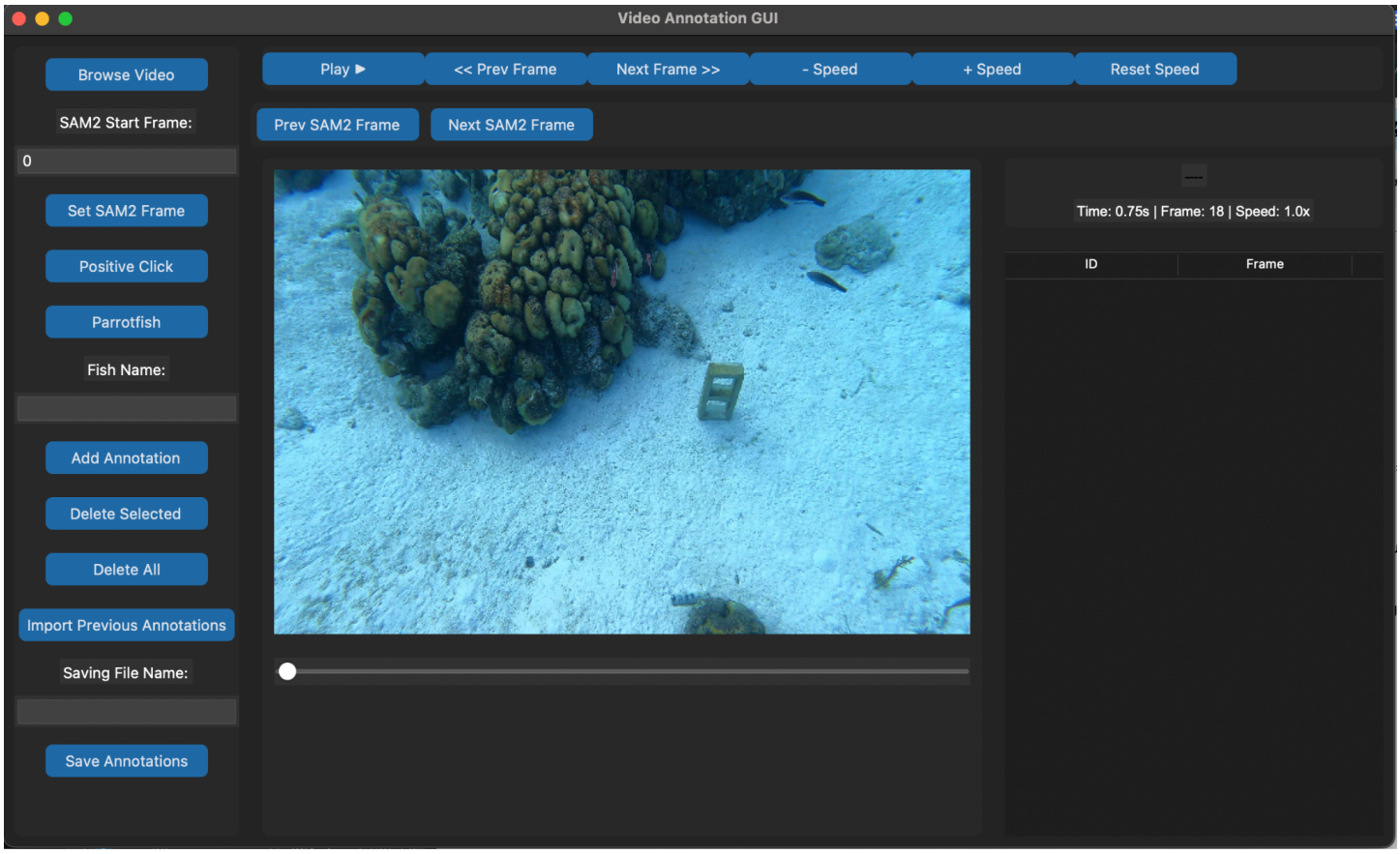
Note: The set-up can be the most challenging with different operating systems. Once all packages are installed, you should be able to run the GUI smoothly. If you are having difficulties installing packages, refer to google or contact your project lead/Madelyn for assistance.

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
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. 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 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. You should also add an entry and exit point for each individual you wish to track. With the correct parameters entered in fish label and fish family:
    a. Mark an entry with the keystroke 'e' on the first frame where the fish is visible.
    b. Mark an exit with the keystroke 'x' on the last frame where the fish is visible.
    Note: You can mark multiple entries and exits for a certain individual if it is not visible for an extended period of time, then re-enters the view.
12. 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.
13. Repeat this process for every fish of interest, **making sure to update your Fish Name** with each new fish.
14. 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.
15. 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

Creating SAM2 directories 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. CU Research Computing (CURC) has extensive documentation about their systems. Read and familiarize yourself with these pages: Logging In, Frequently Asked Questions, Data Transfer.
2. There are two main ways that we will access CURC: Through the Command Line, or through OnDemand. Make sure you know how to log-in to both, and choose your preferred method for data transfer.
3. Now, log-in to your CURC account through Command Line.
4. Type the following command to move into your scratch directory:
   cd /scratch/alpine/$USER
5. Use the following command to make a directory called SAM2 and move into it.

```
mkdir SAM2
cd SAM2
```

6. Use your preferred method for data transfer (either OnDemand or Terminal scp) to upload the environment.yaml file into your SAM2 directory

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 --export=NONE
```

OR

```
salloc --partition=atesting_a100 -N 1 -n 10 --gres=gpu:1 srun --pty $SHELL
```

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

3. Create a conda environment and install the SAM2 directory:

```
$ cd /scratch/alpine/$USER/SAM2
$ module load miniforge
$ mamba env create -f environment.yaml
$ cd /projects/$USER/software
$ git clone https://github.com/facebookresearch/sam2.git
$ cd sam2
$ SAM2_BUILD_ALLOW_ERRORS=0 python setup.py develop
$ cd checkpoints
$ ./download_ckpts.sh
$ cd ..
```

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 torch
```

>>> import numpy

If you get no error messages and no 'module not found' errors, then you can move on to the next step!

General 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.
- cd substitute/file/path is the command to change your current directory, substituting your desired file path after cd.
- ls will list all the files and folders within your current directory.
- pwd will print your current directory.

Part 3 - Extract Frames and Run SAM2

Extracting Frames

Before running the SAM2 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:
Substitute your own video_name (e.g., GH010367) in the code below, where applicable.

1. Log into your RC account through terminal or Putty (Host name: login.rc.colorado.edu) and navigate to your SAM2 directory:
   cd /scratch/alpine/$USER/SAM2
2. Create a subdirectory for your video and another to store your frames, substituting video_name for your video's name:
   mkdir video_name
   cd video_name
   mkdir frames
   cd ..
3. 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. Extract_Frames.sh
    b. Your original video (ex: GX079647.MP4)

4. Edit and run 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. If you are using OnDemand, you can edit a file by selecting the 3 dots next to its name, and choosing “Edit file”
    a. Change the email address at the top, in the last #SBATCH command, to your CU email address.
    b. Edit the line to input the SAM2 start_frame of your video (refer to your project’s instructions on where to find this; **it must be the same as used in the GUI**. The default for non-synced projects is 0).
    c. Confirm that the video_fps variable is set to the frame rate of your original video.
    d. Edit the line of ffmpeg code to input your video_name.MP4 at the beginning, replacing the sample video name.
    e. Save your changes and exit the script editor.
    f. On Command line, confirm that your working directory is SAM2/video_name
    g. Submit the script by typing: sbatch Extract_Frames.sh
    h. Wait until you get an email from Slurm that your Extract_Frames job is completed before moving to the next step for this video.

Run SAM2 Video Predictor:

1. Log-in to your RC account from Command Line.
2. cd into your scratch/alpine/$USER/SAM2/videoname directory. Type ls and press Enter to ensure that within this folder you have the following files (updated versions here):

    a. run_main.sh
    b. template_configs.yaml
    c. main.py
    d. utils.py
    e. plot_utils.py
    f. sam2_fish_segmeter.py
    g. create_video.py
    h. frames (i.e., a folder of extracted frames from your video)
    i. Your annotations for this video (i.e., initials_ObservationID_annotations.npy)

3. To run the scripts, you will first need to make some edits. To open the script editor for the configuration script, type:

    nano template_configs.yaml

    - Edit line 3 and 14 to replace maha7624 with your own CU identikey.
    - Confirm on line 7 this is the name of the folder containing frames from your video that were extracted in the previous step.
    - Confirm on line 28 this is 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 33 to be the SAM2_start frame of your video (if N/A keep as 0). **This must be the same number used in the GUI and in the extract_frames script**.

- Edit line 37 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 74 to input your initials and the observation ID in the name of your mask output file

Edits to specify GUI Version: Lines 41 & 45

- If your annotations were collecting using GUI_0127 or GUI_0226:
    - obj_id_name: 'fishLabel'
    - labels_name: 'clickType'
- If your annotations were collected using GUI_0310 or later:
    - obj_id_name: ‘ObjID’
    - labels_name: ‘ClickType’

4. All lines that need to be changed are marked with a # USER EDITS HERE. Go through and double-check that you have edited every line as needed, then exit and save your changes with the same file name.
5. Now, you should also make a small edit on the run_main.sh file:
   nano run_main.sh
    a. As the last #SBATCH command, there should be a –mail-user argument. Change the email address to your own CU email address.
    b. Exit the editor and save as the same run_main.sh name
6. Submit the SAM2 Video Predictor script as a scheduled job to run when the GPU resources are available:
   sbatch run_main.sh
7. 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
8. When your script has begun running, you will get an email from SLURM. When your script is done running (after about 10-20 min, depending on video length), you will get another email from SLURM stating that your job is done running.
9. When your job is completed, check the results by going to your video_name folder in OnDemand and downloading the output_video.mp4. This should show the predicted masks on top of your video at the reduced frame rate.
    a. Carefully watch the video and confirm that SAM2 masked the correct fish while they were in frame, and didn’t mask unnecessary fish.
    b. If your job failed or there was not an output_video created, send the associated .err or .out files to your project channel or Madelyn and we can help you troubleshoot.
10. If SAM2 is correct in its predictions and masks, congratulations!
    a. An Observation_ID_generated_masks.pkl file should have been created in your video directory. **Save this output** in a google drive folder according to your project’s directions.

11. If SAM2 is not completely correct, refer to Part 4 (below) for instructions on how to re-submit corrections to SAM2.

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. Import your previous location annotations for this video (initials_ObservationID_annotation.npy). You should now have both your original annotations, and the correction annotations, in the GUI.
7. Save your annotations as initials_ObservationID_correction. Confirm that an ObservationID_correction_annotations.npy file has been created in your working directory.
8. Exit the GUI.
9. Now, upload the observationID_combined_annotation.npy file to your video's scratch directory on CURC, using either onDemand or command line and scp.
    a. Template file path: /scratch/alpine/identikey/SAM2/video_name
10. Open up the template_configs.yaml in a script editor (either nano template_configs.yaml or through OnDemand).

   a. Edit line 37 and change the annotations_file name to “<observationID>_correction_annotations.npy”, the file containing both your original annotations and your correction annotations.
   b. Change line 83 to this : video_file: "./corrected_output_video.mp4"
   c. Leave all other configurations as they were from the first edits for this video.
   d. Save the script with the same name.
   e. These edits will ensure that you can run main.py again with the correction annotations without overwriting your previous video.
11. Now, open Terminal or Command Prompt, log-in to your RC account, navigate to your working directory, and run the video predictor again, using the corrected annotations:
    sbatch run_main.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 new correction points.