# Weekly Work Report 4/11/2025

**This Week:**

* Received video labeling results for 5 subjects from Marina.
* Developed a demo script for analyzing events and physiological signals.
* Prepared a list of things to discuss with Xuanchang for confirm.
* Prepared steps for “Analyzing and Predicting Aha! Moments” to discuss with Prof. Agam**.**

**Subject\_3, Session\_1**

*9\_8\_2022\_1\_34\_56\_PM\_****assessment****\_easy.json*

*🡪 Finished 4 puzzles*

*{"answerq1":0.0,"answerq2":0.125,"puzzle\_ID":68}*

*{"answerq1":0.0,"answerq2":0.0,"puzzle\_ID":28}*

*{"answerq1":0.125,"answerq2":0.0,"puzzle\_ID":7}*

*{"answerq1":0.5,"answerq2":0.5,"puzzle\_ID":20}*

*9\_8\_2022\_1\_34\_56\_PM\_****matrix****\_easy.json*

*🡪 Finished 4 puzzles, time start: 9/8/2022 1:34:58 PM, time end: 9/8/2022 1:40:56 PM*

*{"matrixWithNumber":"{\"0\":\"1\",\"1\":\"X\",\"2\":\"X\",\"3\":\"3\",\"4\":\"2\",\"5\":\"2\",\"6\":\"4\",\"7\":\"4\",\"8\":\"4\",\"9\":\"1\",\"10\":\"2\",\"11\":\"X\",\"12\":\"3\",\"13\":\"2\",\"14\":\"5\",\"15\":\"X\",\"16\":\"X\",\"17\":\"X\",\"18\":\"5\",\"19\":\"2\",\"20\":\"2\",\"21\":\"5\",\"22\":\"5\",\"23\":\"5\",\"24\":\"X\",\"25\":\"X\",\"26\":\"X\"}","time":"9/8/2022 1:34:58 PM"} (Start)*

*omit middle*

*---------NEW\_PUZZLE---------*

*omit middle*

*---------NEW\_PUZZLE---------*

*omit middle*

*---------NEW\_PUZZLE---------*

*omit middle*

*{"matrixWithNumber":"{\"0\":\"2\",\"1\":\"3\",\"2\":\"5\",\"3\":\"4\",\"4\":\"3\",\"5\":\"5\",\"6\":\"4\",\"7\":\"X\",\"8\":\"X\",\"9\":\"2\",\"10\":\"1\",\"11\":\"1\",\"12\":\"4\",\"13\":\"3\",\"14\":\"1\",\"15\":\"X\",\"16\":\"X\",\"17\":\"1\",\"18\":\"X\",\"19\":\"6\",\"20\":\"6\",\"21\":\"X\",\"22\":\"6\",\"23\":\"6\",\"24\":\"6\",\"25\":\"6\",\"26\":\"1\"}","time":"9/8/2022 1:40:56 PM"} (End)*

*9\_8\_2022\_1\_34\_56\_PM\_****player\_easy****.json*

*🡪 Finished 3 puzzles, time start: 9/8/2022 1:36:06 PM, time end: 9/8/2022 1:39:14 PM*

*---------NEW\_PUZZLE---------*

*{"number\_of\_completed\_puzzle":1,"puzzle\_ID":68,"puzzle\_lvl":"","score":72.0,"time":"9/8/2022 1:36:06 PM"} (Start)*

*{"number\_of\_completed\_puzzle":2,"puzzle\_ID":28,"puzzle\_lvl":"","score":155.0,"time":"9/8/2022 1:37:38 PM"}*

*---------NEW\_PUZZLE---------*

*---------NEW\_PUZZLE---------*

*{"number\_of\_completed\_puzzle":3,"puzzle\_ID":7,"puzzle\_lvl":"","score":227.0,"time":"9/8/2022 1:39:14 PM"} (End)*

*9\_8\_2022\_1\_34\_56\_PM\_****tracking\_easy****.json*

*🡪 Finished 4 puzzles, time start: 9/8/2022 1:34:58 PM, time end: 9/8/2022 1:40:56 PM*

*{"time":"9/8/2022 1:34:58 PM","headPosition":{"x":0.07545867562294007,"y":-0.010054973885416985,"z":0.004777736961841583},"matrix":"{\"0\":1,\"1\":0,\"2\":0,\"3\":1,\"4\":1,\"5\":1,\"6\":1,\"7\":1,\"8\":1,\"9\":1,\"10\":1,\"11\":0,\"12\":1,\"13\":1,\"14\":1,\"15\":0,\"16\":0,\"17\":0,\"18\":1,\"19\":1,\"20\":1,\"21\":1,\"22\":1,\"23\":1,\"24\":0,\"25\":0,\"26\":0}"} (Start)*

*omit middle*

*---------NEW\_PUZZLE---------*

*omit middle*

*---------NEW\_PUZZLE---------*

*omit middle*

*---------NEW\_PUZZLE---------*

*omit middle*

*---------NEW\_PUZZLE---------*

*omit middle*

*{"time":"9/8/2022 1:40:56 PM","headPosition":{"x":0.39421409368515017,"y":-0.028650907799601556,"z":1.4302128553390504},"matrix":"{\"0\":1,\"1\":1,\"2\":1,\"3\":1,\"4\":1,\"5\":1,\"6\":1,\"7\":0,\"8\":0,\"9\":1,\"10\":1,\"11\":1,\"12\":1,\"13\":1,\"14\":1,\"15\":0,\"16\":0,\"17\":1,\"18\":0,\"19\":1,\"20\":1,\"21\":0,\"22\":1,\"23\":1,\"24\":1,\"25\":1,\"26\":1}"} (End)*

***Matrix***

🡪 *Finished 4 puzzles, time start: 9/8/2022 1:34:58 PM, time end: 9/8/2022 1:40:56 PM*

***Tracking***

🡪 *Finished 4 puzzles, time start: 9/8/2022 1:34:58 PM, time end: 9/8/2022 1:40:56 PM*

***Player***

🡪 *Finished 3 puzzles, time start: 9/8/2022 1:36:06 PM, time end: 9/8/2022 1:39:14 PM*

***Video Label Result***

*A table with numbers and a number on it

AI-generated content may be incorrect.*

|  |  |
| --- | --- |
| Event Label | Timestamp (Real Time) |
| 0 | 9/8/22 13:35 |
| 1 | 9/8/22 13:35 |
| 2.1 | 9/8/22 13:35 |
| 4.1 | 9/8/22 13:35 |
| 0 | 9/8/22 13:35 |
| 1 | 9/8/22 13:35 |
| 2.1 | 9/8/22 13:36 |
| 0 | 9/8/22 13:37 |
| 1 | 9/8/22 13:37 |
| 2.1 | 9/8/22 13:37 |
| 4.1 | 9/8/22 13:37 |

***A screenshot of a computer

AI-generated content may be incorrect.***

**Q:** Matrix and Tracking were recorded almost simultaneously, as most of their start times are the same.

**Q:** Since you mentioned that the Timestamp (Real Time) represents the actual absolute time for Matrix, there’s no need for me to calculate the offset between the video-labeled event timestamp and Matrix’s timestamp.

*A screenshot of a computer

AI-generated content may be incorrect.****A screenshot of a computer

AI-generated content may be incorrect.***



**Q:**  Confirm the time synchronization between pupil data and event timestamps (Real Time).

1. Calculate the offset between Puzzle (Matrix) and Empatica: offset\_puzzle\_empatica (based on absolute time). 10s
2. Calculate the offset between Empatica and Pupil: difference\_empatica\_pupil (based on relative time).
3. Since both the video’s Unix timestamp and the pupil’s New Unix timestamp are in absolute time, I can use the video’s Unix timestamp—converted from the Real Time event timestamp—to align and analyze the pupil data.

timestamp\_float = 1617973123.123456

timestamp\_int = int(timestamp\_float)

print("Original float:", timestamp\_float)

print("Converted to int:", timestamp\_int)

<https://docs.pupil-labs.com/core/developer/#convert-pupil-time-to-system-time>

import datetime

Empatica Unix timestamp – Pupil’s timestamp = difference\_ Empatica Unix timestamp\_pupil

New\_real\_pupil’s timestamp = difference\_ Empatica Unix timestamp\_pupil + Pupil’s timestamp

start\_time\_system/ Empatica Unix timestamp = 1533197768.2805 # System Time at recording start

start\_time\_synced = 674439.5502 # Pupil Time at recording start

# Calculate the fixed offset between System and Pupil Time

offset = start\_time\_system - start\_time\_synced

# Choose a Pupil timestamp that you want to convert to System Time

# (this can be any or all timestamps of interest)

pupil\_timestamp = 674439.4695 # This is a random example of a Pupil timestamp

# Add the fixed offset to the timestamp(s) we wish to convert

pupiltime\_in\_systemtime = pupil\_timestamp + offset

# Using the datetime python module, we can convert timestamps

# stored as seconds represented by floating point values to a

# more readable datetime format.

pupil\_datetime = datetime.datetime.fromtimestamp(pupiltime\_in\_systemtime).strftime("%Y-%m-%d %H:%M:%S.%f")

print(pupil\_datetime)

# example output: '2018-08-02 15:16:08.199800'

# Hint: you can also copy and paste timestamps into various websites that convert them

# to the readable date time format!

**Steps for Analyzing and Predicting Aha! Moments: (Confirm with Prof. Agam)**

1. **Define Aha! Moments** – Identify when Aha! moments occur and determine the event sequence or patterns that typically precede them.
   1. All sequences should cover all scenarios to label Aha! as much as possible.
2. **Pupil Dilation Analysis** – Analyze pupil dilation for each atomic event and arrange all events according to the defined sequence.
3. **Identify Aha! Sequences** – Detect actual sequences that lead to Aha! moments based on physiological and event data.
4. **Label Signal Data** – Use the identified sequences to label the corresponding physiological signal data, find out the best window size.
5. **Modeling and Prediction** – Apply machine learning or deep learning models to predict Aha! moments.
6. **Interpret Results** – Explain the prediction outcomes and the patterns leading up to Aha! moments.