Final Project

DLCV Fall 2023

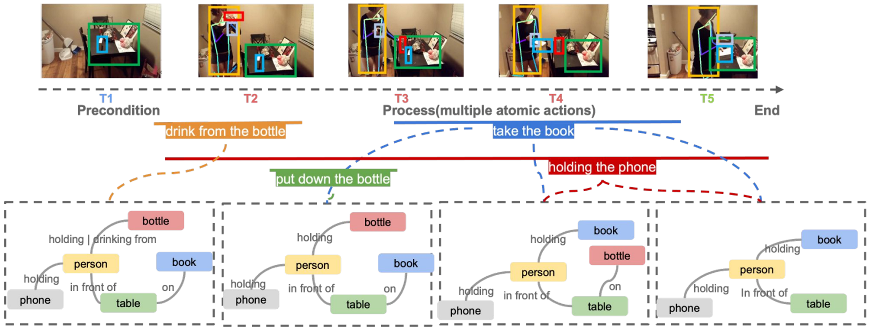
Update

- 11/30
  - Challenge videos are uploaded. [link1] [link2]
- 12/2
  - Team up results [link]
  - TA discussion available time slot [link]

Overview

- STAR:
  - Situated Reasoning (Video Question Answering)

- VQ2D:
  - Visual Queries 2D Localization

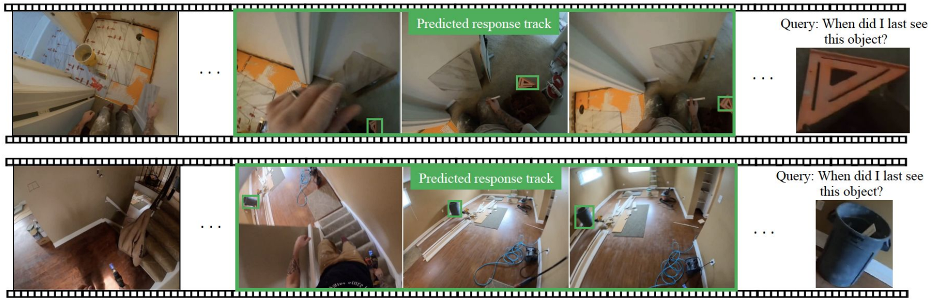


Interaction
Q: What did the person do with the bottle?
Q: [V]; S: [P,V, O]
a: Put down. b: Washed.
c: Took. d: Ate.

Sequence
Q: What object did the person take after the person put down the bottle?
Q: [O]; S: [T_R, P, V, O]
a: The book. b: The towel.
c: The phone. d: The table.

Prediction
Q: Which object would the person put down next?
Q: [O]; S: [T_R, P, V, O]
a: The towel b: The bottle
c: The table d: The book

Feasibility
Q: Which object is possible to be took when the person is in front of the table?
Q: [O]; S: [P, V, S_R, V, O]
a: The phone b: The bottle
c: The book d: The bag


. . .
Predicted response track
. . .
Query: When did I last see this object?
. . .
Predicted response track
. . .
Query: When did I last see this object?

Timeline & Deadlines (GMT+8)

Teaming-up Form Completion	2023/12/01 23:59
Announcement of teaming-up	2023/12/02
Choose when to meet with TAs	2023/12/05 23:59
Project Discussion with TAs	2023/12/13~12/15
Poster Submission	2023/12/25 11:59
CodaLab/Eval AI Submission	2023/12/28 07:59
On-Site **Presentation**	2023/12/28 13:00-17:00
GitHub Code Commit	2023/12/28 23:59

Outline

- General Rules
  - Teaming up
  - GitHub / CodaLab or Eval AI / Poster / Presentation
  - Grading
- Challenges
  - **Challenge 1** - Situated Reasoning in Real-World Videos (STAR Benchmark)
  - **Challenge 2** - Visual Queries 2D Localization (Ego4D Challenge)

Teaming Up and Challenge Selection

- Please fill in this form before **2023/12/01 23:59**
    - Each team should have **4** members
    - Team name
        - English letters (lowercase and uppercase) and numbers only; no spaces
        - **You must use the same team name for GitHub/CodaLab/Eval AI**
    - Team leader
        - Responsible for GitHub team creation and poster/code submission
- We will split the teams equally between the two challenges
    - Your topic choice will be determined by the order of form submissions

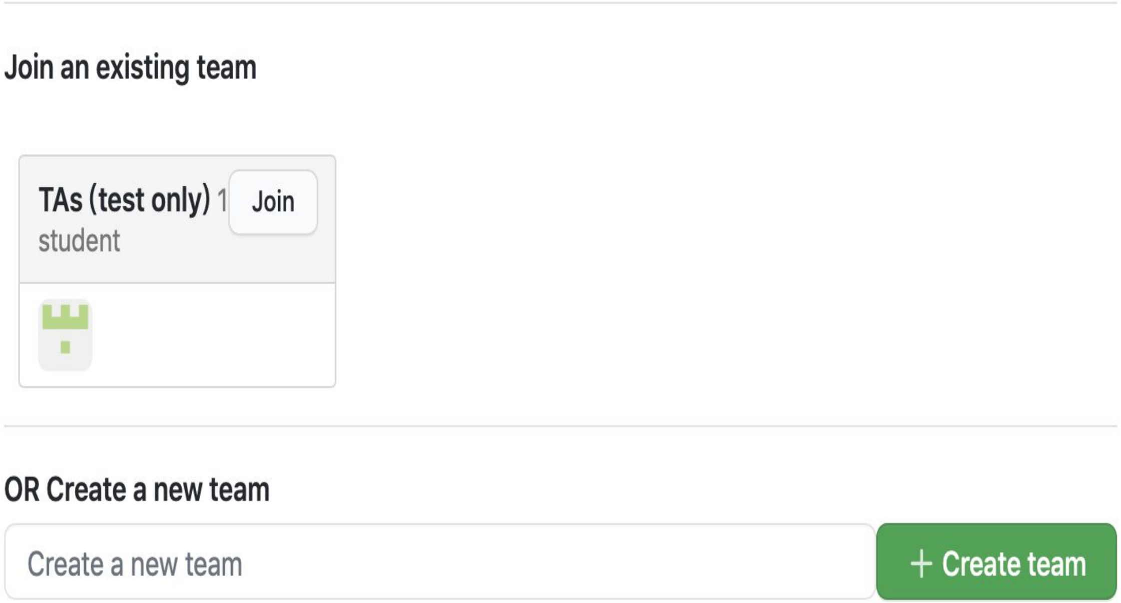
Discussion with TAs

- We will announce more details on 2023/12/02 after teaming-up done.
- Please choose when to meet with TAs before **2023/12/05 23:59**
- Each team will have **20 mins** to discuss.
- **Note that if you do not show on the meeting time, you will lose some points!**

GitHub

- Join the GitHub group assignments (for each challenge) with your **team name**
  - You must use the same team name for GitHub/CodaLab/Eval AI
  - **The team leader creates the team first, and the team members join afterwards**

If you are not the team leader ⟶



Join an existing team

TAs (test only) 1 student

If you are the team leader ⟶

OR Create a new team

Eval AI / CodaLab

- You need to participate in the CodaLab/ Eval AI challenge with your **team name**
    - Eval AI - for Challenge 1
    - CodaLab - for Challenge 2
- Maximum Daily Submissions: 5 times (for each team)
    - CodaLab and Eval AI - will be reset every day
- **Submission Deadline: 2023/12/28 07:59**

Poster for On-Site Presentation

- **PDF format of size A1 (Portrait, 84.1 cm x 59.4 cm)**
- TAs will print it out for your on-site presentation only if you submit it before the deadline.
- **Submission Deadline: 2023/12/25 11:59**
    - Submitted to the root directory of the team’s GitHub repository (format: **poster.pdf**)
    - You can leave some blank areas on your poster for further experiment results and fill them up right before the final presentation.
- If you do not submit your poster before the above deadline, you will need to print it out on your own.
- **Write your name and student ID on the poster!**

On-Site Presentation

- **Schedule: 2023/12/28 13:00-17:00**
- Location: 電二142
- 休息室: 電二144

13:00 - 13:20	**Challenge #1 (STAR) - Poster Readying**
13:20 - 14:40	**Challenge #1 - Presentation**
14:40 - 15:00	**Tea Break / Challenge #2 (VQ2D) - Poster Readying**
15:00 - 15:10	**Challenge #1 - Awarding Ceremony**
15:10 - 16:30	**Challenge #2 - Presentation**
16:30 - 16:50	**Tea Break**
16:50 - 17:00	**Challenge #2 - Awarding Ceremony**

On-Site Presentation

- **Poster Readying**
  - 13:00-13:20 for Challenge-1 and 14:40-15:00 for Challenge-2
  - Prepare your posters (i.e., pasting them onto to the boards) in the given time slots
- **Presentation**
  - Proceed team-by-team according to the **Team ID** for each challenge (Excel)
  - **Time Limit - 5 mins per team**
    - Each team will be given a maximum of 4 minutes for presentation
    - An additional 1 minute will be reserved for Q&A from the lecturer and the TAs
    - As we have a tight schedule, we will control your time strictly!
  - For each team, if no members show up for the final presentation, all team members will receive 0 points for this part (0 out of 25 points)

Code Submission

- **Code Submission Deadline: 2023/12/28 23:59**
- Submit all the training/testing code to your team’s Github repository
- Provide a detailed **README.md** file with example scripts for TAs to reproduce your results (including model training and inference)
- If TAs cannot reproduce your results, you will receive 0 points in the code part (unless minor errors)

Grading

- Model Performance - CodaLab / Eval AI
    - Baseline
    - Relative ranking
- Approach & Presentation
    - Discussion with TAs
    - Novelty and Technical Contributions
    - Completeness of Experiments
    - Poster & Oral Presentation
    - Bonus - Intra / Inter-Team Evaluation

Grading - Intra/Inter-Team Evaluation

- Intra-Team Evaluation
  - You must participate and work with your team member
  - We might adjust your final scores based on the evaluation
- Inter-Team Evaluation
  - The top 3 teams selected by (lecturer, guest, & TA) judges will receive cash prizes
  - The most-voted teams for each challenge will receive bonus points (or gifts)

Challenge 1 -STAR Benchmark

A Benchmark for Situated Reasoning in Real-World Videos

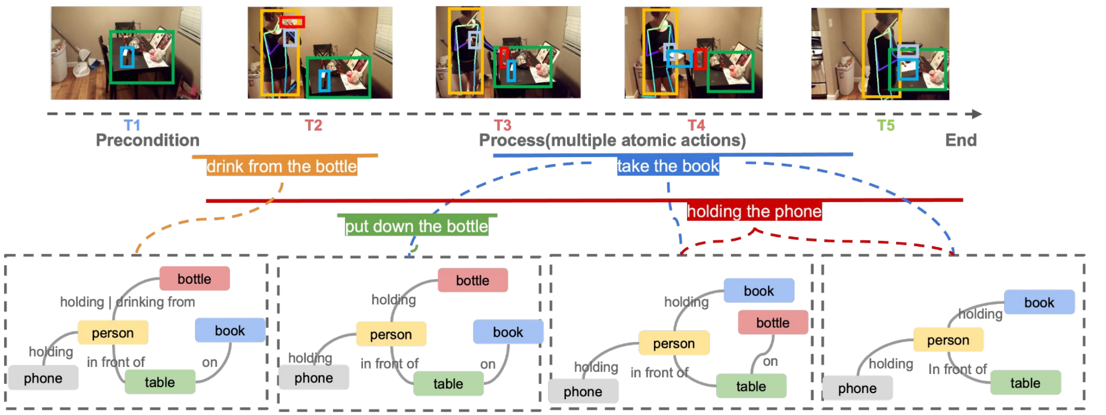
[Github classroom link]

[Eval.ai leaderboard link]

(Do not join them until we announce the final topics for your teams)

STAR Benchmark - Task (1/2)

- Multiple choice video questions answering
    - Input: Video + question + hypergraph(training only) + bounding box(training only)
    - output: Multiple choice answer.



Interaction
Q: What did the person do with the bottle?
Q: [V]; S: [P,V, O]
a: Put down. b: Washed.
c: Took. d: Ate.

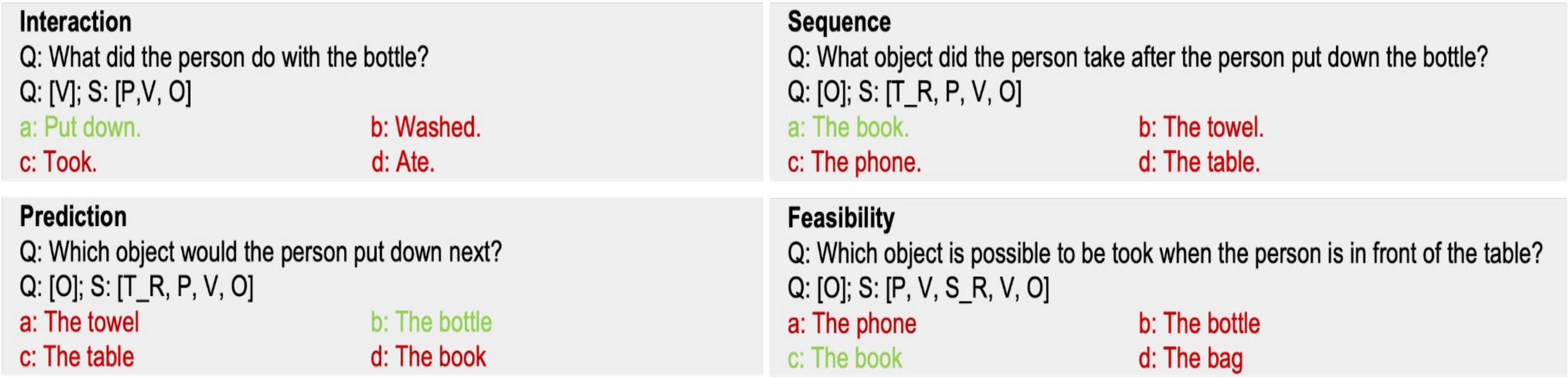
Sequence
Q: What object did the person take after the person put down the bottle?
Q: [O]; S: [T_R, P, V, O]
a: The book. b: The towel.
c: The phone. d: The table.

Prediction
Q: Which object would the person put down next?
Q: [O]; S: [T_R, P, V, O]
a: The towel b: The bottle
c: The table d: The book

Feasibility
Q: Which object is possible to be took when the person is in front of the table?
Q: [O]; S: [P, V, S_R, V, O]
a: The phone b: The bottle
c: The book d: The bag

STAR Benchmark - Task (2/2)

- Webpage link, Github link.
- 4 Question Types, example.
    - Interaction
    - Sequence
    - Prediction
    - Feasibility



Interaction
Q: What did the person do with the bottle?
Q: [V]; S: [P,V, O]
a: Put down. b: Washed.
c: Took. d: Ate.

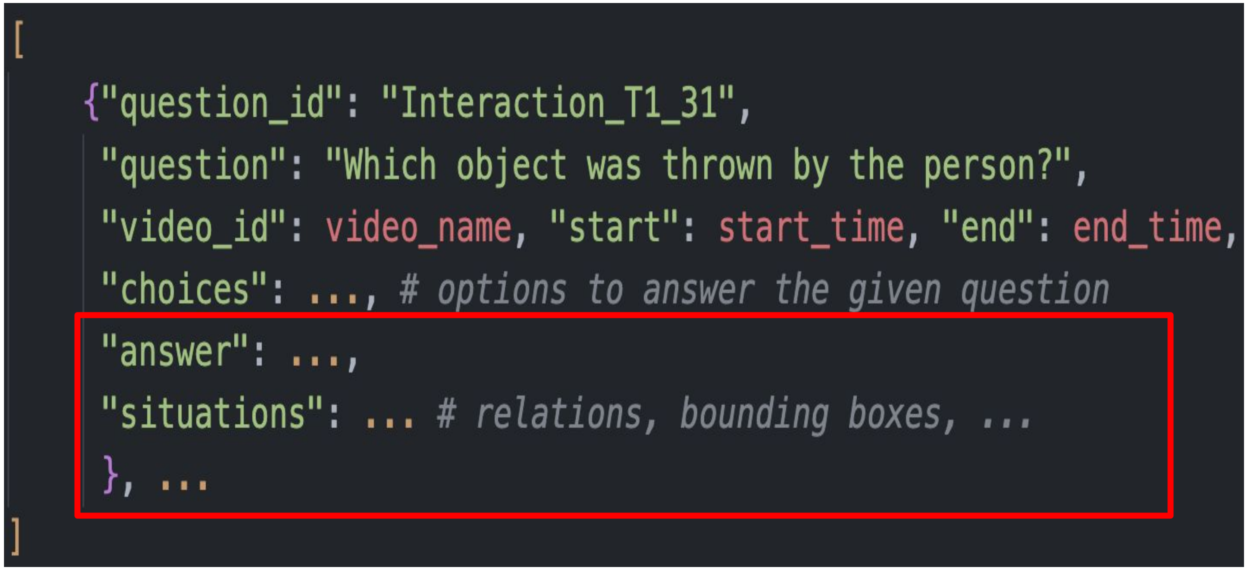
Sequence
Q: What object did the person take after the person put down the bottle?
Q: [O]; S: [T_R, P, V, O]
a: The book. b: The towel.
c: The phone. d: The table.

Prediction
Q: Which object would the person put down next?
Q: [O]; S: [T_R, P, V, O]
a: The towel b: The bottle
c: The table d: The book

Feasibility
Q: Which object is possible to be took when the person is in front of the table?
Q: [O]; S: [P, V, S_R, V, O]
a: The phone b: The bottle
c: The book d: The bag

STAR Benchmark - Dataset (1/4)

- **Data Format**
    - 9848 raw videos from Charades (scaled to 480p) ~16 G (~70G in frame-wise .png)
    - Train/val/test annotation files
    - Train: 45731/ val: 7098 / test: 7377

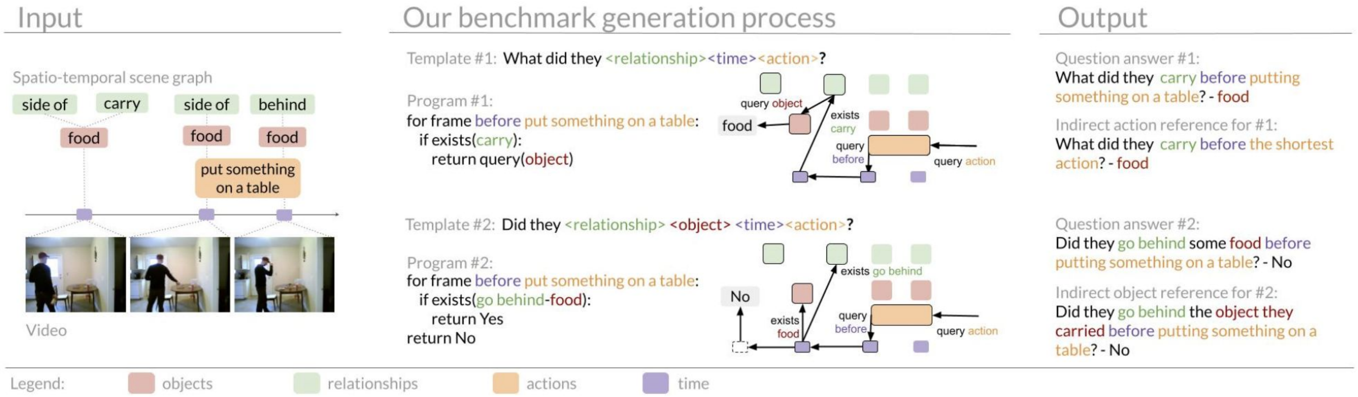


```
[
    {"question_id": "Interaction_T1_31",
     "question": "Which object was thrown by the person?",
     "video_id": video_name, "start": start_time, "end": end_time,
     "choices": ..., # options to answer the given question
     "answer": ...,
     "situations": ... # relations, bounding boxes, ...
     }, ...
]
```

Not given in testing data

STAR Benchmark - Dataset (2/4, optional)

- situations
  - situations: {'000206': {'rel_pairs': [['o000', 'o027'], ['o000', 'o019'], ['o000', 'o004'], ['o000', 'o004']], 'rel_labels': ['r002', 'r002', 'r002', 'r003'], 'actions': ['a004', 'a002', 'a056'], 'bbox': [[167.51, 317.98, 226.33, 331.49], [140.16, 268.79, 211.72, 323.48], [162.93, 298.17, 228.31, 331.25]], 'bbox_labels': ['o027', 'o019', 'o004']}}
- You can generate new questions base on situations (optional).
  - Question Templates (.csv), QA Programs (.csv)



Input

Our benchmark generation process

Output

Template #1: What did they <relationship><time><action>?

```
Program #1:
for frame before put something on a table:
  if exists(carry):
    return query(object)
```

Template #2: Did they <relationship> <object> <time><action>?

```
Program #2:
for frame before put something on a table:
  if exists(go behind-food):
    return Yes
return No
```

Question answer #1:
What did they carry before putting something on a table? - food

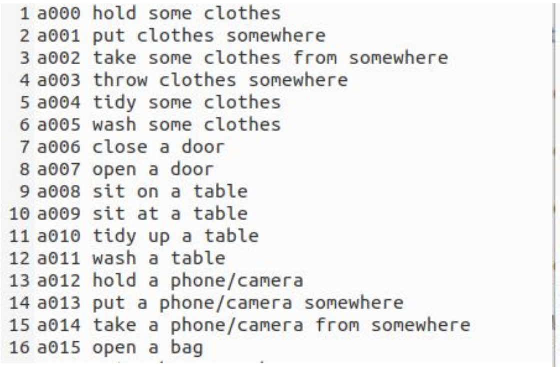
Indirect action reference for #1:
What did they carry before the shortest action? - food

Question answer #2:
Did they go behind some food before putting something on a table? - No

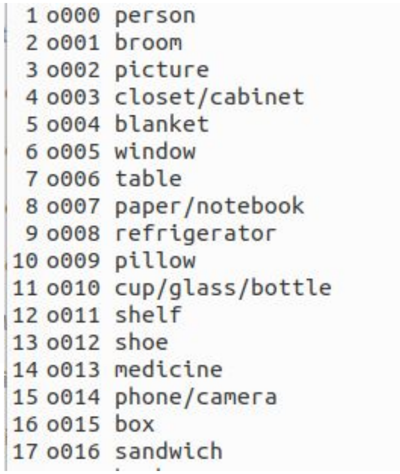
Indirect object reference for #2:
Did they go behind the object they carried before putting something on a table? - No

STAR Benchmark - Dataset (3/4, optional)

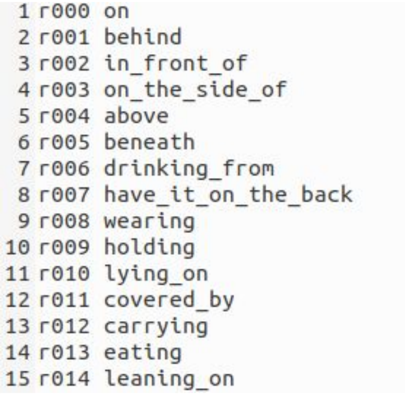
- Annotations
  - Classes Files (.zip)
  - Actions, objects, relationships, verbs



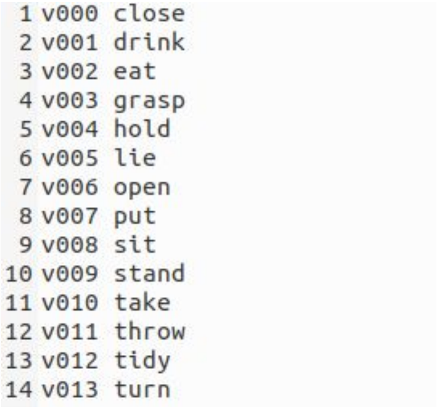
```
a000 hold some clothes
a001 put clothes somewhere
a002 take some clothes from somewhere
a003 throw clothes somewhere
a004 tidy some clothes
a005 wash some clothes
a006 close a door
a007 open a door
a008 sit on a table
a009 sit at a table
a010 tidy up a table
a011 wash a table
a012 hold a phone/camera
a013 put a phone/camera somewhere
a014 take a phone/camera from somewhere
a015 open a bag
```



```
o000 person
o001 broom
o002 picture
o003 closet/cabinet
o004 blanket
o005 window
o006 table
o007 paper/notebook
o008 refrigerator
o009 pillow
o010 cup/glass/bottle
o011 shelf
o012 shoe
o013 medicine
o014 phone/camera
o015 box
o016 sandwich
```



```
r000 on
r001 behind
r002 in_front_of
r003 on_the_side_of
r004 above
r005 beneath
r006 drinking_from
r007 have_it_on_the_back
r008 wearing
r009 holding
r010 lying_on
r011 covered_by
r012 carrying
r013 eating
r014 leaning_on
```



```
v000 close
v001 drink
v002 eat
v003 grasp
v004 hold
v005 lie
v006 open
v007 put
v008 sit
v009 stand
v010 take
v011 throw
v012 tidy
v013 turn
```

STAR Benchmark - Dataset (4/4, optional)

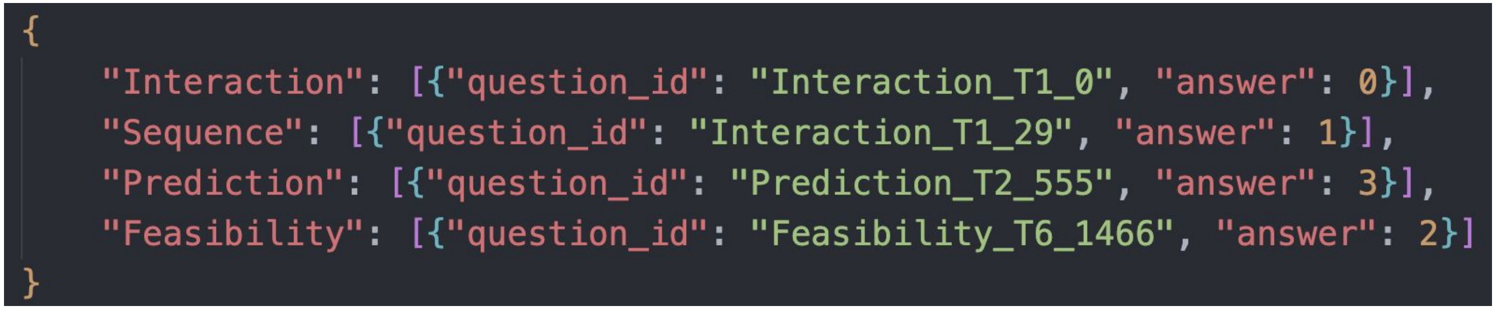
- Bounding boxes
  - Object Bounding Boxes (.pkl).
  - Human Bounding Boxes (.pkl).

STAR Benchmark - Evaluation (1/2)

- Metrics
    - Accuracies of 4 types of questions and overall accuracy.
        - Interaction
        - Sequence
        - Prediction
        - Feasibility

STAR Benchmark - Evaluation (1/2)

- Website
- Submission
  - 20 times per day / 150 times per month
  - Format: link
  - Example



```
{
    "Interaction": [{"question_id": "Interaction_T1_0", "answer": 0}],
    "Sequence": [{"question_id": "Interaction_T1_29", "answer": 1}],
    "Prediction": [{"question_id": "Prediction_T2_555", "answer": 3}],
    "Feasibility": [{"question_id": "Feasibility_T6_1466", "answer": 2}]
}
```

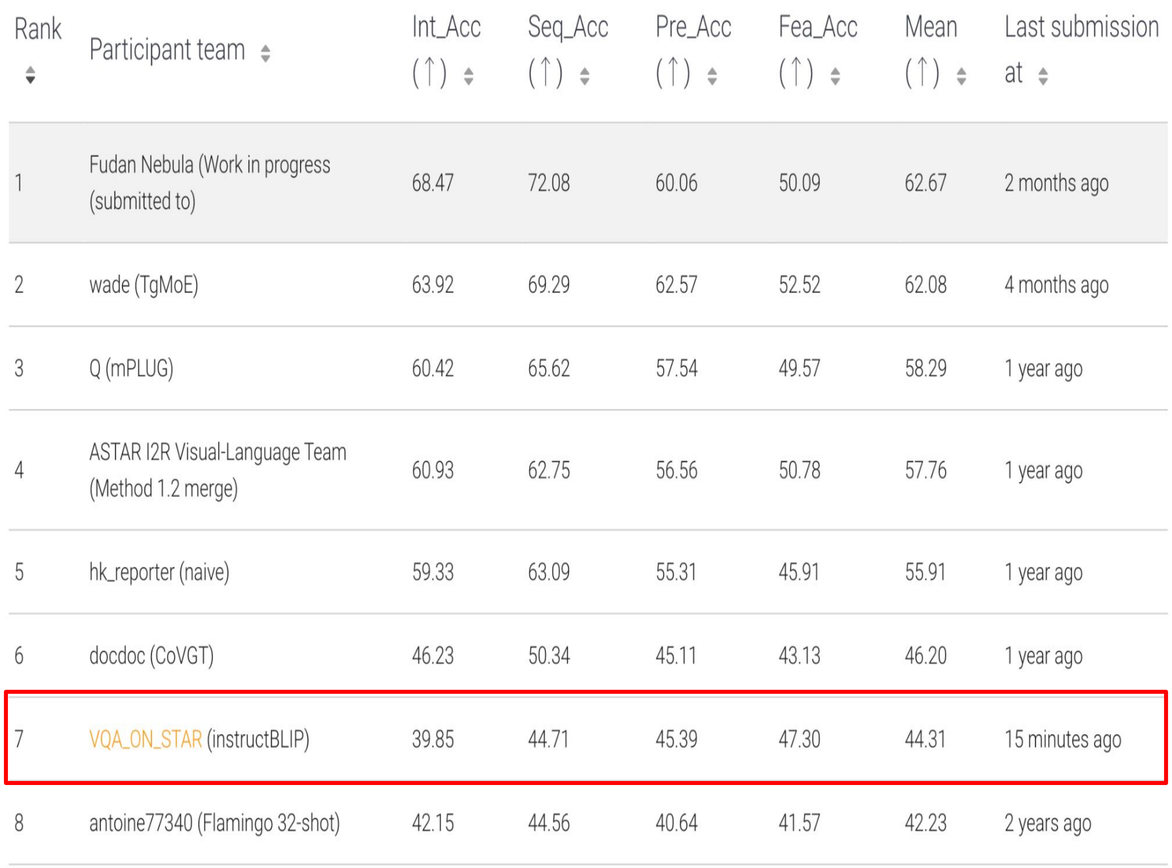
STAR Benchmark - Grading (1/2)

- **Final 34%** (Bonus up to **3%**)
  - **Model Performance - Eval.ai 9%**
    - Baseline **4%**
    - Relative ranking in class **5%**
  - **Approach & Presentation 25% + 3%**
    - Discussion with TAs **2%**
    - Novelty and technical contributions **10%**
    - Completeness of experiments **8%**
      (e.g., ablation study, visualization, etc.)
    - Poster & Oral Presentation **5%**
    - Bonus (intra / inter-team evaluation) up to **3%**

Points	Team Ranking
5	top 0% - 20%
4.5	top 20% - 40%
4	top 40% - 60%
3.5	top 60% - 80%
3	top 80% - 100%

STAR Benchmark - Grading (2/2)

- Baseline



Rank ⬍	Participant team ⬍	Int_Acc (↑) ⬍	Seq_Acc (↑) ⬍	Pre_Acc (↑) ⬍	Fea_Acc (↑) ⬍	Mean (↑) ⬍	Last submission at ⬍
1	Fudan Nebula (Work in progress (submitted to)	68.47	72.08	60.06	50.09	62.67	2 months ago
2	wade (TgMoE)	63.92	69.29	62.57	52.52	62.08	4 months ago
3	Q (mPLUG)	60.42	65.62	57.54	49.57	58.29	1 year ago
4	ASTAR I2R Visual-Language Team (Method 1.2 merge)	60.93	62.75	56.56	50.78	57.76	1 year ago
5	hk_reporter (naive)	59.33	63.09	55.31	45.91	55.91	1 year ago
6	docdoc (CoVGT)	46.23	50.34	45.11	43.13	46.20	1 year ago
7	VQA_ON_STAR (instructBLIP)	39.85	44.71	45.39	47.30	44.31	15 minutes ago
8	antoine77340 (Flamingo 32-shot)	42.15	44.56	40.64	41.57	42.23	2 years ago

STAR Benchmark - Rules

- Feel free to use any pretrained video/visual-language model.
- Your results need to be reproducible with your submitted code and models.
- Please use **python3** instead of python for your scripts.
- Any violation would result in 0 score for your final project.

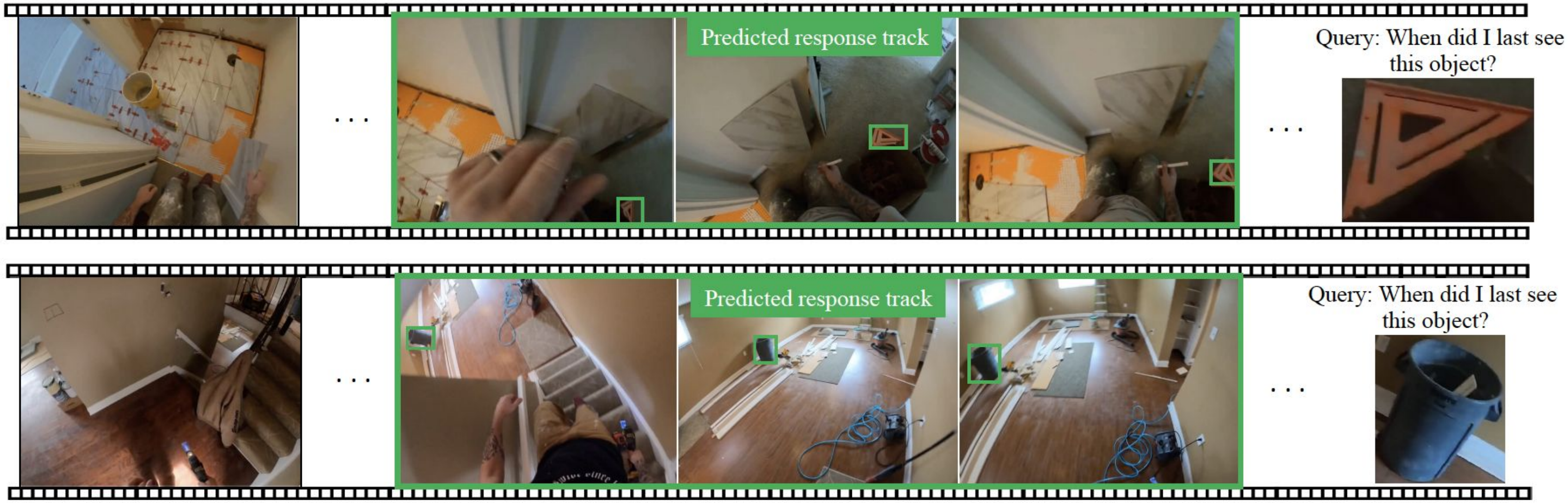
Challenge 2 - Visual Queries 2D Localization Task

(Ego4D Challenge)

Github classroom link
CodaLab leaderboard link
(Do not join them until we announce the final topics for your teams)

VQ2D

- Task
  - Given a **short clip** and an **image query**, predict the last time the object appeared in the clip
  - Predict the bounding box for each frame

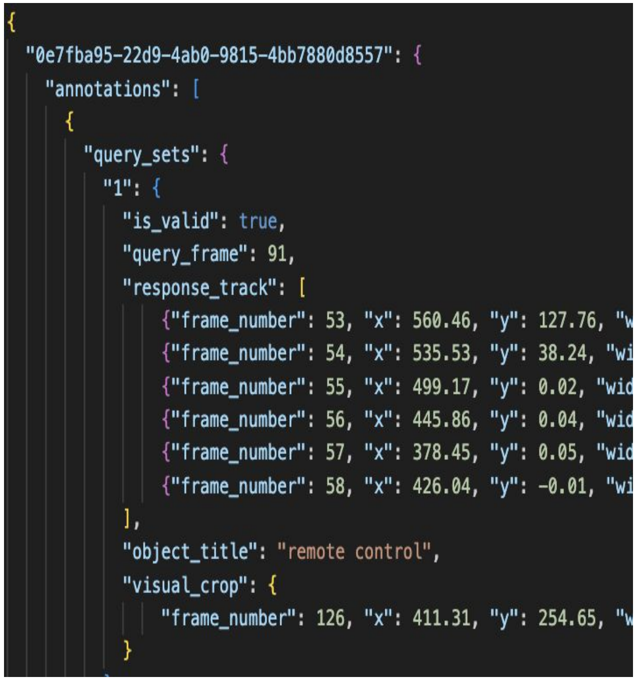

Predicted response track
Query: When did I last see this object?
Predicted response track
Query: When did I last see this object?

VQ2D

Query: When did I last see this object?

image query

- Definitions
  - **clip**: a short video
  - **image query**: an image cropped from the clip, defined by **visual_crop**
  - **response_track**: a consecutive window when the object last appeared; the ground truth
  - **query_frame**: **the timestep** when I ask “When did I last see object X?”
- Task description
  - Your task is to predict the **response_track**
  - Find the specific object in the image query
    - If there are multiple trash cans in the video, only the one in the image query counts



```
{
  "0e7fba95-22d9-4ab0-9815-4bb7880d8557": {
    "annotations": [
      {
        "query_sets": {
          "1": {
            "is_valid": true,
            "query_frame": 91,
            "response_track": [
                {"frame_number": 53, "x": 560.46, "y": 127.76, "w
                {"frame_number": 54, "x": 535.53, "y": 38.24, "wi
                {"frame_number": 55, "x": 499.17, "y": 0.02, "wid
                {"frame_number": 56, "x": 445.86, "y": 0.04, "wid
                {"frame_number": 57, "x": 378.45, "y": 0.05, "wid
                {"frame_number": 58, "x": 426.04, "y": -0.01, "wi
            ],
            "object_title": "remote control",
            "visual_crop": {
                "frame_number": 126, "x": 411.31, "y": 254.65, "w
            }
```

VQ2D - Dataset (1/4)

- **Data Format**
  - 1462 short clips (~30GB totally)
  - ~300 secs for each clip
  - Train/val/test annotation files
  - Train: 1171/ val: 146 / **test: 145**
  - You have to output **pred.json** based on **vq_test_unannotated.json**
  - Dataset link

```
VQ2D/
    vq_train.json
    vq_val.json
    vq_test_unannotated.json
    clips/
        {clip_uid}.mp4
        …
```

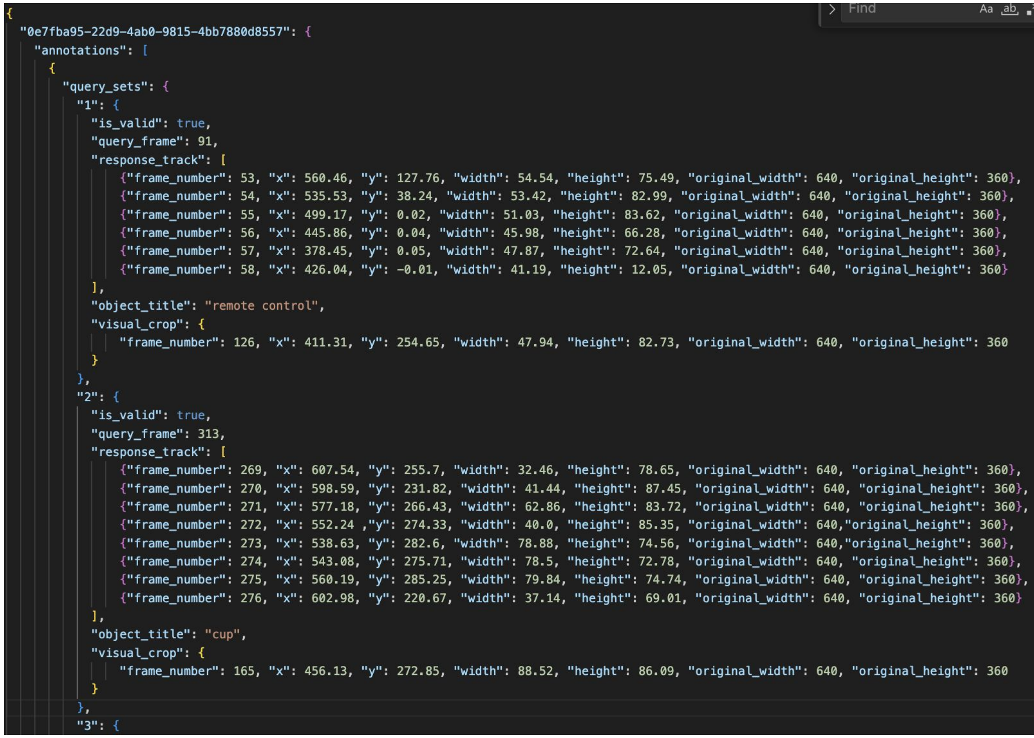
VQ2D - Dataset (2/4)

Annotation files

vq_train.json, vq_val.json

- <clip_uid> *(string)*
    - annotations *(array)*
        - query_sets
            - “1”
                - query_frame *(int)*
                - response_track
                    - frame_number *(int)*
                    - x
                    - y
                    - width
                    - height
                    - original_width
                    - original_height
                - visual_crop
                    - x, y, width, height, ...
            - “2” …
            - “3” …
        - annotation_uid *(string)*

response_track is the ground truth



```
{
  "0e7fba95-22d9-4ab0-9815-4bb7880d8557": {
    "annotations": [
      {
        "query_sets": {
          "1": {
            "is_valid": true,
            "query_frame": 91,
            "response_track": [
                {"frame_number": 53, "x": 560.46, "y": 127.76, "width": 54.54, "height": 75.49, "original_width": 640, "original_height": 360},
                {"frame_number": 54, "x": 535.53, "y": 38.24, "width": 53.42, "height": 82.99, "original_width": 640, "original_height": 360},
                {"frame_number": 55, "x": 499.17, "y": 0.02, "width": 51.03, "height": 83.62, "original_width": 640, "original_height": 360},
                {"frame_number": 56, "x": 445.86, "y": 0.04, "width": 45.98, "height": 66.28, "original_width": 640, "original_height": 360},
                {"frame_number": 57, "x": 378.45, "y": 0.05, "width": 47.87, "height": 72.64, "original_width": 640, "original_height": 360},
                {"frame_number": 58, "x": 426.04, "y": -0.01, "width": 41.19, "height": 12.05, "original_width": 640, "original_height": 360}
            ],
            "object_title": "remote control",
            "visual_crop": {
                "frame_number": 126, "x": 411.31, "y": 254.65, "width": 47.94, "height": 82.73, "original_width": 640, "original_height": 360
            }
          },
          "2": {
            "is_valid": true,
            "query_frame": 313,
            "response_track": [
                {"frame_number": 269, "x": 607.54, "y": 255.7, "width": 32.46, "height": 78.65, "original_width": 640, "original_height": 360},
                {"frame_number": 270, "x": 598.59, "y": 231.82, "width": 41.44, "height": 87.45, "original_width": 640, "original_height": 360},
                {"frame_number": 271, "x": 577.18, "y": 266.43, "width": 62.86, "height": 83.72, "original_width": 640, "original_height": 360},
                {"frame_number": 272, "x": 552.24 ,"y": 274.33, "width": 40.0, "height": 85.35, "original_width": 640,"original_height": 360},
                {"frame_number": 273, "x": 538.63, "y": 282.6, "width": 78.88, "height": 74.56, "original_width": 640,"original_height": 360},
                {"frame_number": 274, "x": 543.08, "y": 275.71, "width": 78.5, "height": 72.78, "original_width": 640, "original_height": 360},
                {"frame_number": 275, "x": 560.19, "y": 285.25, "width": 79.84, "height": 74.74, "original_width": 640, "original_height": 360},
                {"frame_number": 276, "x": 602.98, "y": 220.67, "width": 37.14, "height": 69.01, "original_width": 640, "original_height": 360}
            ],
            "object_title": "cup",
            "visual_crop": {
                "frame_number": 165, "x": 456.13, "y": 272.85, "width": 88.52, "height": 86.09, "original_width": 640, "original_height": 360
            }
          },
          "3": {
```

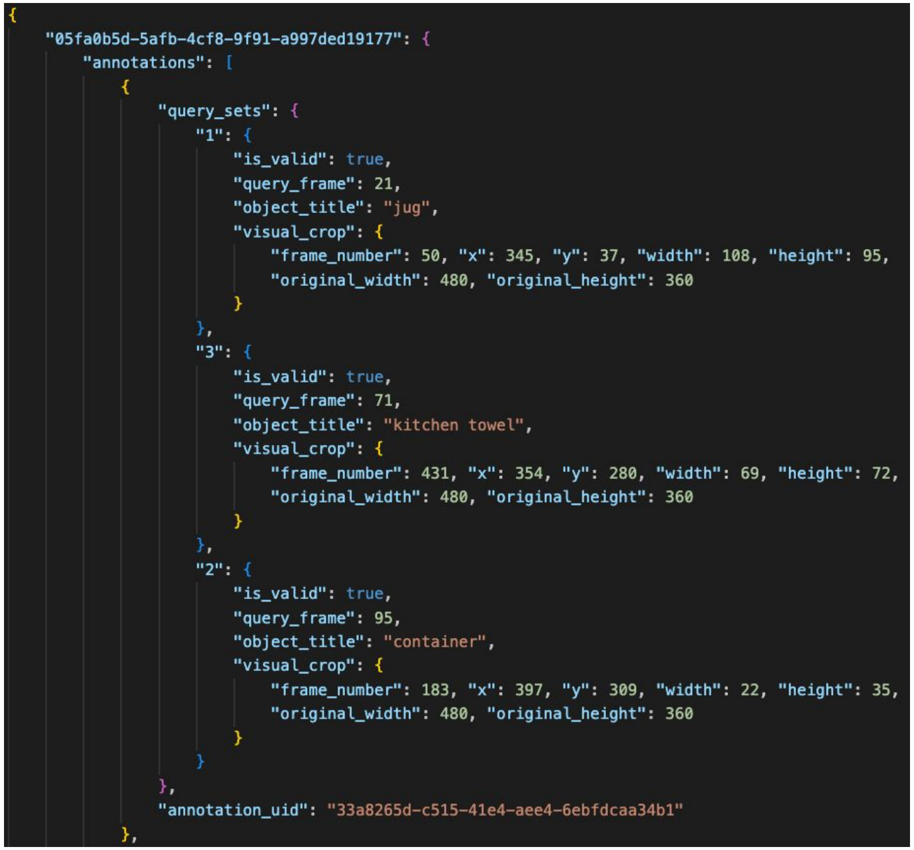
sample **vq_train.json**

VQ2D - Dataset (3/4)

Annotation files

vq_test_unannotated.json

- <clip_uid> *(string)*
    - annotations *(array)*
        - query_sets
            - “1”
                - query_frame *(int)*
                - visual_crop
                    - frame_number (int)
                    - x
                    - y
                    - width
                    - height
                    - original_width
                    - original_height
            - “2” …
            - “3” …
        - annotation_uid *(string)*



```
{
    "05fa0b5d-5afb-4cf8-9f91-a997ded19177": {
        "annotations": [
            {
                "query_sets": {
                    "1": {
                        "is_valid": true,
                        "query_frame": 21,
                        "object_title": "jug",
                        "visual_crop": {
                            "frame_number": 50, "x": 345, "y": 37, "width": 108, "height": 95,
                            "original_width": 480, "original_height": 360
                        }
                    },
                    "3": {
                        "is_valid": true,
                        "query_frame": 71,
                        "object_title": "kitchen towel",
                        "visual_crop": {
                            "frame_number": 431, "x": 354, "y": 280, "width": 69, "height": 72,
                            "original_width": 480, "original_height": 360
                        }
                    },
                    "2": {
                        "is_valid": true,
                        "query_frame": 95,
                        "object_title": "container",
                        "visual_crop": {
                            "frame_number": 183, "x": 397, "y": 309, "width": 22, "height": 35,
                            "original_width": 480, "original_height": 360
                        }
                    }
                },
                "annotation_uid": "33a8265d-c515-41e4-aee4-6ebfdcaa34b1"
            },
```

sample **vq_test_unannotated.json**

VQ2D - Dataset (4/4)

Output file

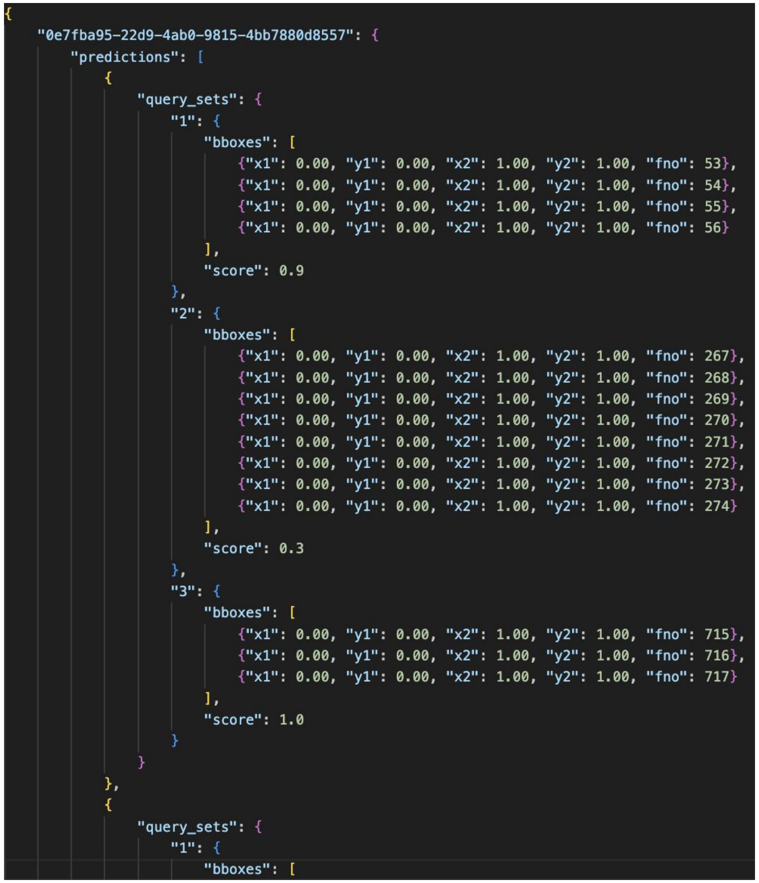
pred.json

- <clip_uid> *(string)*
    - predictions *(array)*
        - query_sets
            - “1”
                - bboxes
                    - x1
                    - y1
                    - x2
                    - y2
                    - fno
                - score *(float)*
            - “2” …
            - “3” …

x2 = x1 + predicted width

y2 = y1 + predicted height

fno: frame_number



```
{
    "0e7fba95-22d9-4ab0-9815-4bb7880d8557": {
        "predictions": [
            {
                "query_sets": {
                    "1": {
                        "bboxes": [
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 53},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 54},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 55},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 56}
                        ],
                        "score": 0.9
                    },
                    "2": {
                        "bboxes": [
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 267},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 268},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 269},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 270},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 271},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 272},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 273},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 274}
                        ],
                        "score": 0.3
                    },
                    "3": {
                        "bboxes": [
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 715},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 716},
                            {"x1": 0.00, "y1": 0.00, "x2": 1.00, "y2": 1.00, "fno": 717}
                        ],
                        "score": 1.0
                    }
                }
            },
            {
                "query_sets": {
                    "1": {
                        "bboxes": [
```

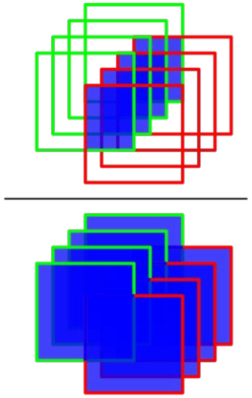
dummy example of **pred.json**

VQ2D - Evaluation (1/4)

- Spatio-temporal AP (stAP):

  mAP of object bounding tube (bounding boxes in contiguous frames) (with IoU threshold = 0.25)

$$\text{STT-IOU} = \frac{\text{volume of overlap}}{\text{volume of union}} =$$



VQ2D - Evaluation (2/4)

- Create an account and participate the competition (with your team name)
  - CodaLab Link
  - All members in a team should use the same account.

    (i.e. one team, one account)
  - Your team name should start with “English letter”

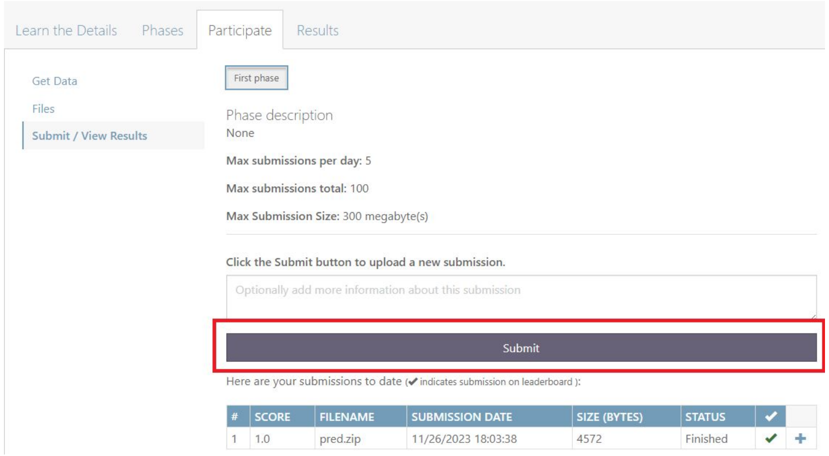
    (e.g. ❌1abc ⭕ a1bc)

VQ2D - Evaluation (3/4)

- Submission
  - 5 times per day / 100 in total
  - You should submit your **zip** file to the CodaLab competition
  - Please note that the **zip** file should **ONLY** contain **pred.json**
- Submission format
  - sample_submission.zip
    - pred.json
  - The **zip** file name is arbitrary, but the **json** file name should be **pred.json**

VQ2D - Evaluation (4/4)

- Submit your **zip** file in the “Submit/View Results” page.
- You can check your score and ranking in the “Results” page
- To upload the zip file to the “Submit/View Results” page:
  - Click the “Submit” button to choose your zip file
  - Wait for the evaluation
  - STATUS should go as “Submitting” → “Submitted” → “Running” → “Finished”
  - If it has been stuck at the "Submitted" status for a long time (more than 1~2 hours), you can e-mail TAs to re-run your submission.



Phase description
None

Max submissions per day: 5

Max submissions total: 100

Max Submission Size: 300 megabyte(s)

Click the Submit button to upload a new submission.

Optionally add more information about this submission

Here are your submissions to date (✓ indicates submission on leaderboard):

#	SCORE	FILENAME	SUBMISSION DATE	SIZE (BYTES)	STATUS	✓	
1	1.0	pred.zip	11/26/2023 18:03:38	4572	Finished	✓	+

VQ2D Benchmark - Grading (1/2)

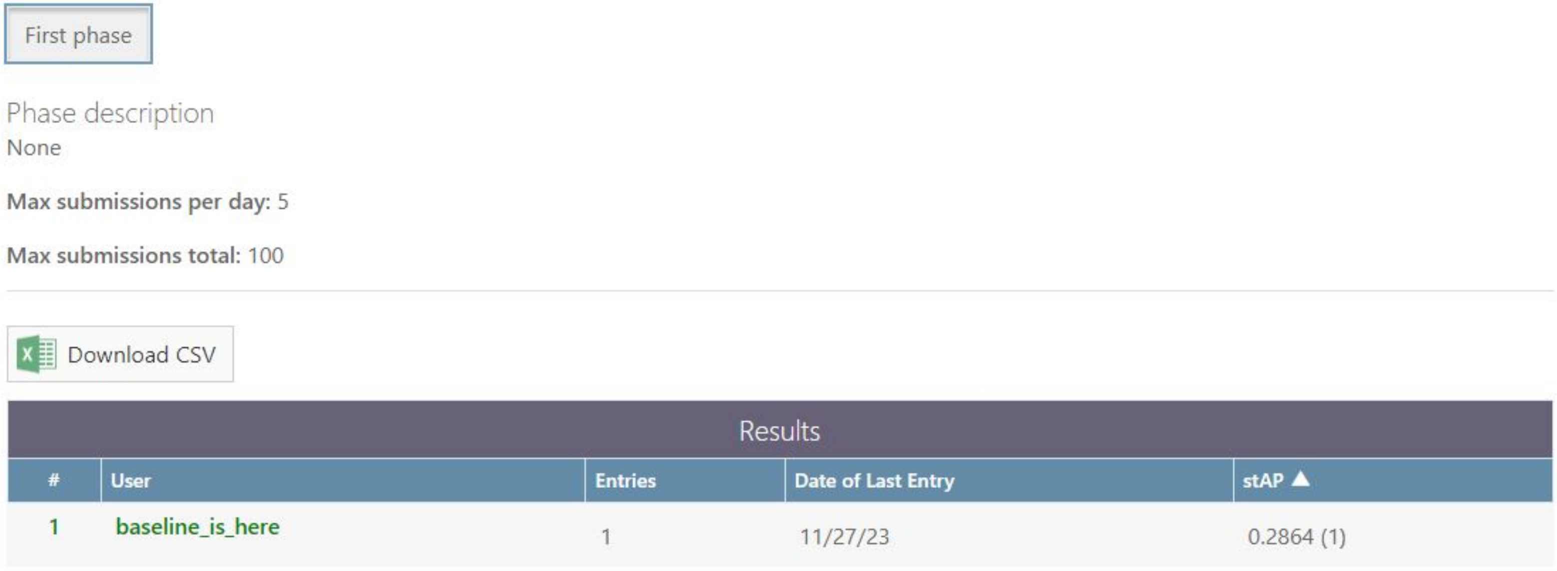
- **Final 34%** (Bonus up to **3%**)
  - **Model Performance - CodaLab 9%**
    - Baseline **4%**
    - Relative ranking in class **5%**
  - **Approach & Presentation 25% + 3%**
    - Discussion with TAs **2%**
    - Novelty and technical contributions **8%**
    - Completeness of experiments **10%**

      (e.g., ablation study, visualization, etc.)
    - Poster & Oral Presentation **5%**
    - Bonus (inter-team evaluation) up to **3%**

Points	**Team Ranking**
5	top 0% - 20%
4.5	top 20% - 40%
4	top 40% - 60%
3.5	top 60% - 80%
3	top 80% - 100%

VQ2D Benchmark - Grading (2/2)

- Baseline



Phase description
None

Max submissions per day: 5

Max submissions total: 100

Results

#	User	Entries	Date of Last Entry	stAP ▲
1	baseline_is_here	1	11/27/23	0.2864 (1)

VQ2D Benchmark - Rules

- Your results need to be reproducible with your submitted code and models.
- Please use **python3** instead of python for your scripts.
- Any violation would result in 0 score for your final project.