

Documentation on how to Set-up the & experimenting with the HanoiWorld experiment

Pre-requisite:

- Know how to obtain the CometML API
- Know how to run the kaggle notebook

Materials:

- The Kaggle Notebook (public - the HanoiWorld train on 5000-Steps)
[HanoiWorld-Roundabout](#)
[HanoiWorld-Merge](#)
[HanoiWorld-Highway](#)
- The model JEPA-Model pre-trained checkpoint link
[The JEPA-Encoder Checkpoint Model Checkpoint - We only consider jepa1_final.pt as the encoder being used](#)
- The HanoiWorld Model Checkpoint: (note usually if you load & open the notebook the model shall come with the notebook as the embedded before)
 - For the highway-v0 use case: <https://www.kaggle.com/models/davestran/jepa-1-highway-env/PyTorch/default/1>
 - For the roundabout-v0 use case: <https://www.kaggle.com/models/davestran/jepa-1-roundabout-5k/PyTorch/default/1>
 - For the merge-v0 use case: <https://www.kaggle.com/models/davestran/jepa-merge-5k/PyTorch/default/1>

How to run the & reproduce the experiment

→ These experiments are identical on all merge, highway, and roundabout

⇒ In this instruction, we shall introduce how you can run the experiment based on our checkpoint

- Step 1: Navigate toward the kaggle notebook with the corresponding environment - which explicitly named on the notebook link (in the 1th bulletpoint of the Material Section)

Daves Tran · 4d ago · 23 views

1

Share

Edit

JEPA-1-RSSM-Roundabout

Copied from Daves Tran (+4, -4)

Notebook

Input

Output

Logs

Comments (0)

Settings

In [1]:

```

!pip install torchcodec
!pip install torch-geometric
!pip install -q highway-env gymnasium

```

Collecting torchcodec
 Downloading torchcodec-0.9.1-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (11 kB)
 Downloading torchcodec-0.9.1-cp312-cp312-manylinux_2_28_x86_64.whl (2.1 MB)
 ----- 2.1/2.1 MB 23.7 MB/s eta 0:00:00
Installing collected packages: torchcodec
Successfully installed torchcodec-0.9.1
Collecting torch-geometric
 Downloading torch_geometric-2.7.0-py3-none-any.whl.metadata (63 kB)
 ----- 63.7/63.7 KB 2.3 MB/s eta 0:00:00
Requirement already satisfied: aiohttp in /usr/local/lib/python3.12/dist-packages (from torch-geometric) (3.13.2)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch-geometric) (2025.10.0)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.12/dist-packages (from torch-geometric) (3.1.6)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from torch-geometric) (2.0.2)
Requirement already satisfied: psutil>=5.8.0 in /usr/local/lib/python3.12/dist-packages (from torch-geometric) (5.9.0)

Version 7 of 8

Runtime

1h 33m 47s · GPU P100

Input

MODELS

jepackpt_5k · default · V1

Tags

GPU

Language

Python

Table of Contents

RSSM

Utility for RSSM

Frozen Encoder

Config

- You shall expect to see the Notebook that you currently chosen which named as follow: JEPA-1-RSSM-[Environment name]
- Step 2: Take the latest version by click on the button (Version x of y)
 - With y is the latest version & x is the current success running version of the original author been run
 - After that - click -edit (the black-button with pen icon on the top-right of the browser for enter)

JEPA-1-RSSM-Roundabout

Draft saved

File Edit View Run Settings Add-ons Help

+ -

Run All

Code

Draft Session off (run a cell to start)

↑

↓

🗑

+ Code

+ Markdown

```

!pip install torchcodec
!pip install torch-geometric
!pip install -q highway-env gymnasium

```

```

%%bash
cd /kaggle/working
rm -rf WorldModel-Self-Driving-Car
git clone -b debug_branch https://github.com/CS-3331-Fundamental-of-AI/WorldModel-Self-Driving-Car.git

```

```

cd WorldModel-Self-Driving-Car

```

```

!pip install -r requirements.txt

```

```

!pip install ruamel.yaml==0.18.16 ruamel.yaml.clib==0.2.15
!pip install comet_ml

```

```

%%bash
cd /kaggle/working/WorldModel-Self-Driving-Car
cat << 'EOF' > ./env
WORKSPACE=dtj-tran

```

Notebook

Input

+ Add Input

Upload

MODELS

jepackpt_5k · default · V1

JEPA-1-Roundabout-5k · default · V1

Output

/kaggle/working

Table of contents

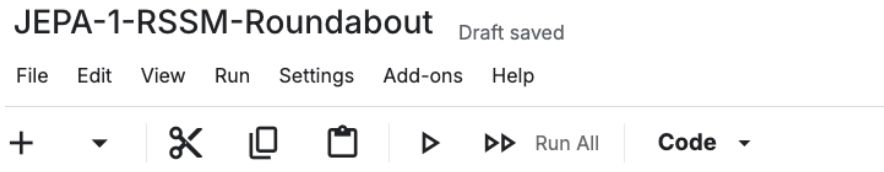
No sections detected

Add markdown headers to add a section

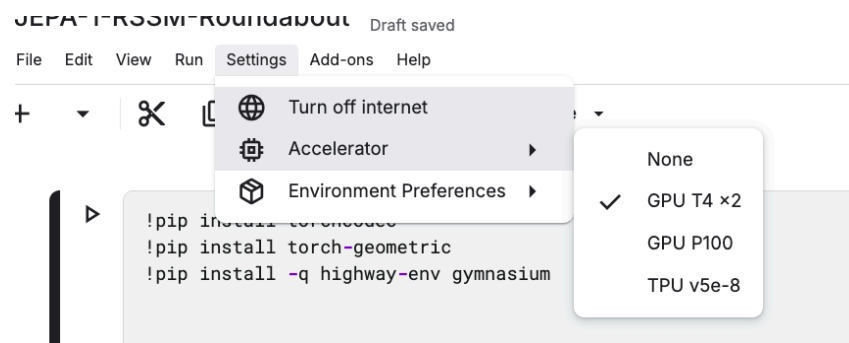
Session options

Schedule a notebook to run

- You shall be expect to see this result - a jupyter notebook show you been successfully access to the working enviroment
- Step 3:
 - On the top-bar include the notebook name choose **Setting**



- Pick Accelerator (GPU selecting)



- In order to use the Kaggle virtual GPU - you essentially need the already verified account by using phone number *(for this problem please check the setting section of your kaggle account console)*
- Step 4: To activate the experiment click the switch-on button



- you expect to observe this banner on your working notebook & the spot shall change to Green when the Accelerator & Dataset downloading are ready to be work with
- Step 5: Check the **Table of contents** of the resource

Table of contents

RSSM

Utility for RSSM

Frozen Encoder

Config

HanoiWorldJepa1

HanoiAgent

Dataset preparation

Training

Re-load the CKPT from prior work

TRAIN

Evaluate

Demonstration

- From top to bottom is the section including the Setting-up , RSSM code-base & Encoder, Config setup, Agent & World Model logic, Training & Evaluation section
- For working Properly - you shall only need to activate all-block from the start to the section **Re-load the CKPT from prior work** for updating the code & pre-loaded function
 - For demonstration of our model - You additionally activate the "Demonstration" section
 - For the Re-trained - Activate the "TRAIN" section additionally
 - For the Evaluate (without no video recording) - use Evaluate section
- Step 4: Running the Demonstration Code blocks
 - Pre-requisite - You have to run All the method block in the Training finish & success without failure

Demonstration

```
[32]: # you shall run this block - it's the wrapper for recording the interaction between environment & agent
import sys, os
from pathlib import Path
import gymnasium as gym
from gymnasium.wrappers import RecordVideo

ROOT = Path("/kaggle/working/WorldModel-Self-Driving-Car")
sys.path.append(str(ROOT))

from HanoiWorld.envs.highway_base import HighwayEnv

class RecordableHighwayEnv(HighwayEnv):
    def __init__(self, *args, render_mode="rgb_array", **kwargs):
        super().__init__(*args, render_mode=render_mode, **kwargs)
        self.render_mode = render_mode # expose public attribute

    def render(self):
        return self.env.render()

VIDEO_DIR = "eval_videos"
os.makedirs(VIDEO_DIR, exist_ok=True)

env = RecordableHighwayEnv(
    name=env_name,
    render_mode="rgb_array", # REQUIRED
    obs_type="image",
    action_type="discrete",
    offscreen_rendering=True, # Kaggle-safe
)

env = RecordVideo(
    env,
    video_folder=VIDEO_DIR,
    episode_trigger=lambda ep: True, # record every episode
    name_prefix=f"({env.name}).eval",
)
```

MODELS

- Jepa_ckpt_L5k - default - V1
- JEPa-T-Roundabout-5k - default - V1

Output

• /kaggle/working

Table of contents

Now It's the time for activating the rest of

- RSSM
- Utility for RSSM
- Frozen Encoder
- Config
- HanoiWorldJepa1
- HanoiAgent
- Dataset preparation
- Training
- Re-load the CKPT from prior work
- TRAIN
- Evaluate
- Demonstration

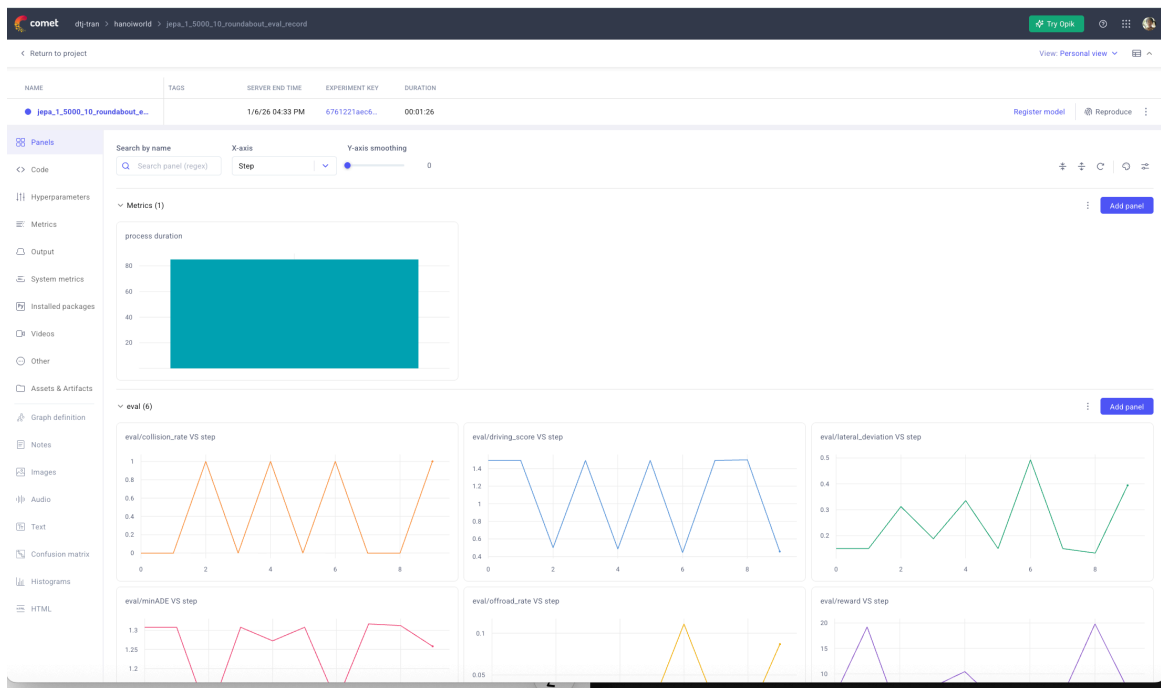
Session options

• Schedule a notebook to run

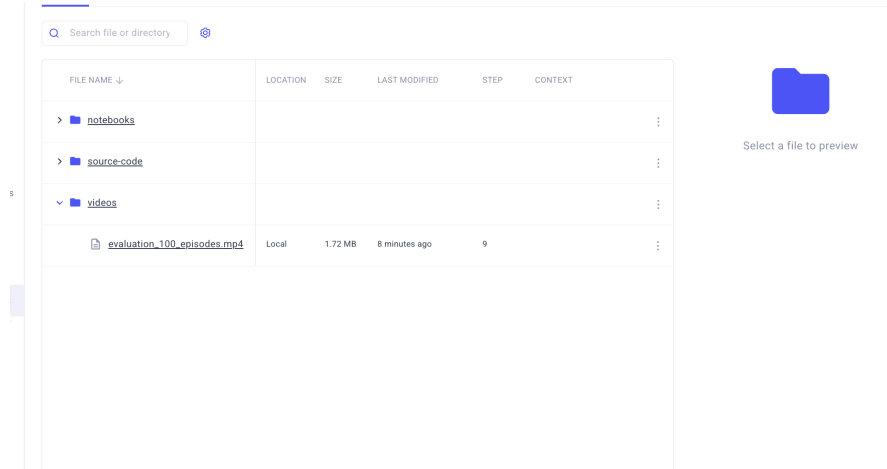
- Step 5:
 - Check on the Comet-ML by navigate in that yeild as below

```
COMET WARNING: To get all data logged automatically, import comet_ml before the following modules: sklearn, keras, torch, tensorflow, tensorboard.
COMET WARNING: As you are running in a Jupyter environment, you will need to call 'experiment.end()' when finished to ensure all metrics and code are logged before exiting.
COMET INFO: Experiment is live on comet.com https://www.comet.com/dtj-tran/hanoiworld/6761221aec6d4a38aa65bcee65ec5629
```

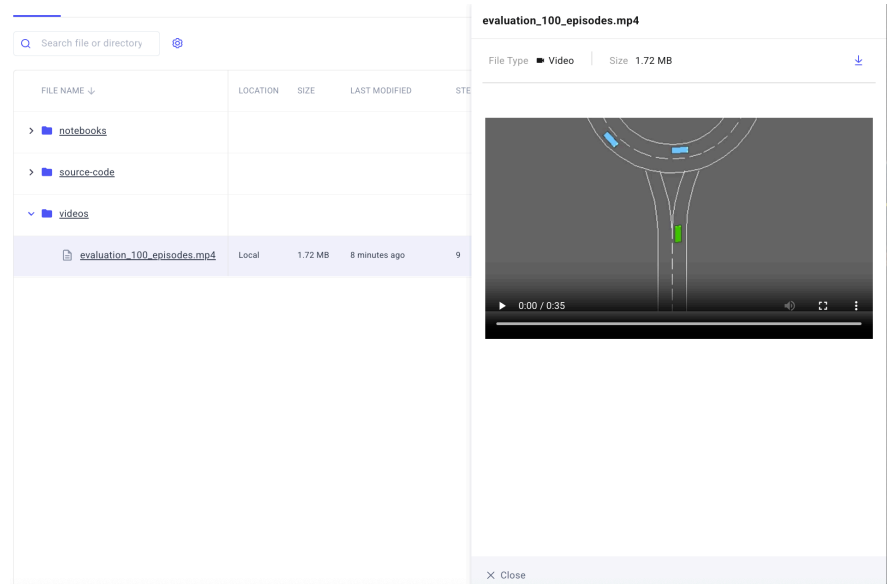
⇒ The https link is the Place where you shall got & observe the experiment & collect the result as the below figure



- Step 6: In the Section on the side-columns pick "Asset & Artifacts"
 - Choose **video** folder



- Click on the .mp4 which is the evaluation video for you to debug & what the model performance



- You shall be expected the work like this

Hope you guys work effectively with this instruction!