Visual Language Maps (VLMaps) for Robot Navigation

Project Overview

This notebook presents a **reimplementation of the Visual Language Maps (VLMaps) framework** for language-conditioned robot navigation, as introduced by Huang et al. (ICRA 2023). The objective of this work is to construct a unified spatial representation that fuses **geometric structure** with **open-vocabulary semantic understanding**, enabling robots to interpret and execute complex natural language navigation commands.

Unlike classical SLAM-based navigation systems that operate purely on geometry, VLMaps encode **visual-language embeddings directly into a spatial map**, allowing robots to ground free-form language queries such as *"go to the chair next to the table"* or *"navigate between the sofa and the cabinet"*.

This implementation follows the methodology described in the original paper and evaluates its effectiveness on simulated indoor environments using RGB-D data.

Motivation

Robotic navigation in unstructured environments requires both:

- **spatial reasoning** (where am I? where are obstacles?), and
- **semantic understanding** (what objects exist and how are they described in language).

Traditional navigation pipelines excel at geometry but lack semantic flexibility, while modern vision–language models excel at semantic grounding but lack persistent spatial memory. VLMaps address this gap by introducing a **persistent, spatially grounded, open-vocabulary map** that unifies both capabilities.

This enables:

- zero-shot generalization to unseen objects,
- interpretation of relational language instructions,
- and adaptability across different robot embodiments.

High-Level Pipeline

The VLMaps pipeline implemented in this notebook consists of the following stages:

1. **RGB-D Data Acquisition**

- The agent collects RGB images, depth maps, and pose information while exploring an indoor environment.

2. **Visual-Language Feature Extraction**
   - Each RGB observation is encoded into a high-dimensional semantic embedding using a vision–language model operating in a shared image–text embedding space.
3. **3D Reconstruction and Back-Projection**
   - Depth pixels are back-projected into 3D space using camera intrinsics and transformed into a global coordinate frame using the agent pose.
4. **VLMap Construction**
   - Visual-language embeddings are projected onto a 2D top-down grid map.
   - Multiple embeddings falling into the same grid cell are aggregated to produce a robust semantic representation.
5. **Language Querying and Landmark Identification**
   - Natural language queries are encoded and matched against the VLMap using semantic similarity.
   - High-scoring regions correspond to language-referenced landmarks.
6. **Navigation and Execution**
   - Identified goal regions are used to generate navigation plans and action sequences for the robot.

Mathematical Formulation

Let an RGB-D observation at time step t be defined as:

$$\left(I_t, D_t, T_t\right)$$

where I_t is the RGB image, D_t is the depth image, and T_t is the camera-to-world transformation.

Visual-Language Encoding

Each RGB image is encoded using a visual encoder:

$$f_t = E_v\left(I_t\right)$$

A language query q is encoded using a text encoder:

$$l_q = E_l\left(q\right)$$

Both embeddings lie in a shared semantic space.

Back-Projection to 3D

For a pixel $u = \left(u, v\right)$, the 3D point in the camera frame is computed as:

$$P_k = D\left(u\right) K^{-1} \tilde{u}$$

where K is the camera intrinsic matrix and $\tilde{u}$ is the homogeneous pixel coordinate.

The point is transformed into the world frame using:

$$P_W = T_{W\,k} P_k$$

VLMap Representation

The environment is discretized into a 2D grid map $M \in R^{H \times W \times C}$, where each cell stores an aggregated semantic embedding:

$$M(x, y) = \frac{1}{n} \sum_{i=1}^{n} f_i$$

Semantic Matching

To localize a language query in the map, cosine similarity is computed:

$$s(x, y) = \cos\left(M(x, y), l_q\right)$$

Cells with the highest similarity scores correspond to language-grounded targets.

Open-Vocabulary Navigation

Because VLMaps operate in a shared image–language embedding space, the system supports **open-vocabulary landmark indexing** without requiring predefined object categories. This enables robots to follow complex, multi-step instructions and reason about spatial relationships between objects.

In addition, language instructions can be decomposed into navigation primitives and executed sequentially, enabling zero-shot navigation behavior.

Results Summary

The implementation demonstrates that:

- visual-language embeddings can be reliably fused into a spatial map,
- landmarks can be identified using free-form language queries,
- and navigation actions can be generated based on semantic reasoning.

The resulting VLMaps enable robust object-goal navigation and illustrate the potential of combining vision–language models with spatial mapping for intuitive robot control.

Notebook Structure

The remainder of this notebook is organized as follows:

1. Dataset preparation and environment setup
2. VLMap creation and visualization
3. Landmark identification using language queries
4. Navigation experiments and action generation
5. Discussion and observations

Citation

This work is based on the following paper:

Visual Language Maps for Robot Navigation
Chenguang Huang, Oier Mees, Andy Zeng, Wolfram Burgard
Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2023
London, UK

Project website: https://vlmaps.github.io/

```
@inproceedings{huang23vlmaps,
  title     = {Visual Language Maps for Robot Navigation},
  author    = {Chenguang Huang and Oier Mees and Andy Zeng and Wolfram
Burgard},
  booktitle = {Proceedings of the IEEE International Conference on
Robotics and Automation (ICRA)},
  year      = {2023},
  address   = {London, UK}
}
```

Dataset Preparation and Repository Setup

This cell prepares the workspace by downloading the required dataset and setting up the codebase used for Visual Language Map (VLMap) generation.

First, a preprocessed RGB-D dataset corresponding to an indoor scene is downloaded. This dataset contains:

- RGB images
- depth maps
- camera pose information

These components are required to construct VLMaps by back-projecting visual-language features into a spatial map.

Next, the official VLMaps implementation repository is cloned. This repository provides the core utilities for:

- visual-language feature extraction,

- VLMap construction,
- landmark indexing,
- and navigation-related processing.

After cloning, the working directory is switched to the repository root to access the provided scripts, configuration files, and utilities used throughout the subsequent stages of the pipeline.

At the end of this step, the environment contains:

- the downloaded RGB-D dataset,
- the VLMaps codebase,
- and all necessary files to begin VLMap creation.

```
%cd /content/
!gdown 1n81Lm8ywg5xTp8uwWEtzXvmDHB_9Sboi
!pip install tqdm
!unzip -o jh4fc5c5qoQ_1.zip | tqdm --desc extracted --unit files --
unit_scale --total `unzip -l jh4fc5c5qoQ_1.zip | tail -n 1 | xargs
echo -n | cut -d' ' -f2` > /dev/null
!git clone https://github.com/Aman-Chandak/vlmaps.git
!git checkout master
%cd vlmaps
!ls

/content
Downloading...
From (original): https://drive.google.com/uc?
id=1n81Lm8ywg5xTp8uwWEtzXvmDHB_9Sboi
From (redirected): https://drive.google.com/uc?
id=1n81Lm8ywg5xTp8uwWEtzXvmDHB_9Sboi&confirm=t&uuid=7d1f11fc-c016-
49fe-b7c0-cb2c3e184e5e
To: /content/jh4fc5c5qoQ_1.zip
100% 1.54G/1.54G [00:10<00:00, 144MB/s]
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (4.66.6)
extracted: 2.02kfiles [01:05, 30.7files/s]
Cloning into 'vlmaps'...
remote: Enumerating objects: 40, done.ote: Counting objects: 100%
(40/40), done.ote: Compressing objects: 100% (37/37), done.ote: Total
40 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)aps
examples  install.bash     lseg  pyproject.toml  requirements-dev.txt
requirements.txt  utils
```

Dependency Installation and Simulation Stack Setup

This cell installs all required dependencies for running the Visual Language Maps (VLMaps) pipeline end-to-end.

The implementation relies on a combination of:

- deep learning frameworks for visual-language embedding,

- computer vision utilities for image and depth processing,
- mapping and visualization libraries,
- and a simulation environment for embodied navigation.

Key components installed in this step include:

- **CLIP** for joint vision–language embeddings,
- **LSeg / encoding utilities** for semantic feature extraction,
- **PyTorch Lightning** for structured model execution,
- **OpenCV and ImageIO** for image handling,
- **Habitat-Sim and Habitat-Lab** for RGB-D simulation and navigation experiments.

The Habitat simulation stack is installed from source and pinned to a specific commit to ensure compatibility with the VLMaps implementation. This setup enables the agent to access RGB-D observations, camera poses, and environment metadata required for VLMap construction.

At the end of this step, the environment is fully configured to:

- process RGB-D data,
- generate visual-language embeddings,
- construct VLMaps,
- and perform language-conditioned navigation experiments.

```
# install dependencies
!pip install git+https://github.com/openai/CLIP.git
!pip install torch-encoding
!pip install ipython
!pip install pytorch-lightning
!pip install opencv-python
!pip install imageio
!pip install ftfy regex tqdm
!pip install altair
!pip install --upgrade protobuf
!pip install timm
!pip install matplotlib
!pip install wandb
!pip install h5py

# install habitat
!pip install cmake==3.14.4
!pip install tensorflow-gpu==2.9.1
!conda install habitat-sim -c conda-forge -c aihabitat
%cd ~
!git clone https://github.com/facebookresearch/habitat-lab.git
!pip install gym==0.22.0
%cd habitat-lab
!git checkout bfba72f47800819d858a6859b14cfa26122c2762
!python setup.py develop --all
```

```
!pip install openai==0.8.0
!pip install grad-cam
```

```
Collecting git+https://github.com/openai/CLIP.git
  Cloning https://github.com/openai/CLIP.git to /tmp/pip-req-build-
hheciqpn
  Running command git clone --filter=blob:none --quiet
https://github.com/openai/CLIP.git /tmp/pip-req-build-hheciqpn
  Resolved https://github.com/openai/CLIP.git to commit
dcba3cb2e2827b402d2701e7e1c7d9fed8a20ef1
  Preparing metadata (setup.py) ...  clip==1.0)
  Downloading ftfy-6.3.1-py3-none-any.whl.metadata (7.3 kB)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from clip==1.0) (24.2)
Requirement already satisfied: regex in
/usr/local/lib/python3.10/dist-packages (from clip==1.0) (2024.9.11)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from clip==1.0) (4.66.6)
Requirement already satisfied: torch in
/usr/local/lib/python3.10/dist-packages (from clip==1.0) (2.5.1+cu121)
Requirement already satisfied: torchvision in
/usr/local/lib/python3.10/dist-packages (from clip==1.0)
(0.20.1+cu121)
Requirement already satisfied: wcwidth in
/usr/local/lib/python3.10/dist-packages (from ftfy->clip==1.0)
(0.2.13)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch->clip==1.0)
(3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch->clip==1.0)
(4.12.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch->clip==1.0)
(3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch->clip==1.0)
(3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch->clip==1.0)
(2024.10.0)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch->clip==1.0)
(1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch-
>clip==1.0) (1.3.0)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from torchvision->clip==1.0)
(1.26.4)
```

```
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/usr/local/lib/python3.10/dist-packages (from torchvision->clip==1.0)
(11.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch-
>clip==1.0) (3.0.2)
Downloading ftfy-6.3.1-py3-none-any.whl (44 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.8/44.8 kB 2.3 MB/s eta
0:00:00
e=clip-1.0-py3-none-any.whl size=1369489
sha256=e7ed1c19e56a93012b05292a971c512c3ae885c8e9a840203ff20bd145c4981
3
  Stored in directory:
/tmp/pip-ephem-wheel-cache-caju5v8u/wheels/da/2b/4c/d6691fa9597aac8bb8
5d2ac13b112deb897d5b50f5ad9a37e4
Successfully built clip
Installing collected packages: ftfy, clip
Successfully installed clip-1.0 ftfy-6.3.1
Collecting torch-encoding
  Downloading torch_encoding-1.2.1-py2.py3-none-any.whl.metadata (4.3
kB)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from torch-encoding) (1.26.4)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from torch-encoding) (4.66.6)
Collecting nose (from torch-encoding)
  Downloading nose-1.3.7-py3-none-any.whl.metadata (1.7 kB)
Collecting portalocker (from torch-encoding)
  Downloading portalocker-3.0.0-py3-none-any.whl.metadata (8.5 kB)
Requirement already satisfied: torch>=1.4.0 in
/usr/local/lib/python3.10/dist-packages (from torch-encoding)
(2.5.1+cu121)
Requirement already satisfied: torchvision>=0.5.0 in
/usr/local/lib/python3.10/dist-packages (from torch-encoding)
(0.20.1+cu121)
Requirement already satisfied: Pillow in
/usr/local/lib/python3.10/dist-packages (from torch-encoding) (11.0.0)
Requirement already satisfied: scipy in
/usr/local/lib/python3.10/dist-packages (from torch-encoding) (1.13.1)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from torch-encoding) (2.32.3)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->torch-
encoding) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->torch-
encoding) (4.12.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->torch-
```

```
encoding) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->torch-
encoding) (3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->torch-
encoding) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.4.0->torch-
encoding) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1-
>torch>=1.4.0->torch-encoding) (1.3.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->torch-
encoding) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests->torch-
encoding) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->torch-
encoding) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->torch-
encoding) (2024.8.30)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.4.0-
>torch-encoding) (3.0.2)
Downloading torch_encoding-1.2.1-py2.py3-none-any.whl (126 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 126.2/126.2 kB 4.8 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 154.7/154.7 kB 10.8 MB/s eta
0:00:00
ent already satisfied: ipython in /usr/local/lib/python3.10/dist-
packages (7.34.0)
Requirement already satisfied: setuptools>=18.5 in
/usr/local/lib/python3.10/dist-packages (from ipython) (75.1.0)
Collecting jedi>=0.16 (from ipython)
  Downloading jedi-0.19.2-py2.py3-none-any.whl.metadata (22 kB)
Requirement already satisfied: decorator in
/usr/local/lib/python3.10/dist-packages (from ipython) (4.4.2)
Requirement already satisfied: pickleshare in
/usr/local/lib/python3.10/dist-packages (from ipython) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in
/usr/local/lib/python3.10/dist-packages (from ipython) (5.7.1)
Requirement already satisfied: prompt-toolkit!=3.0.0,!
=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from
ipython) (3.0.48)
Requirement already satisfied: pygments in
```

```
/usr/local/lib/python3.10/dist-packages (from ipython) (2.18.0)
Requirement already satisfied: backcall in
/usr/local/lib/python3.10/dist-packages (from ipython) (0.2.0)
Requirement already satisfied: matplotlib-inline in
/usr/local/lib/python3.10/dist-packages (from ipython) (0.1.7)
Requirement already satisfied: pexpect>4.3 in
/usr/local/lib/python3.10/dist-packages (from ipython) (4.9.0)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in
/usr/local/lib/python3.10/dist-packages (from jedi>=0.16->ipython)
(0.8.4)
Requirement already satisfied: ptyprocess>=0.5 in
/usr/local/lib/python3.10/dist-packages (from pexpect>4.3->ipython)
(0.7.0)
Requirement already satisfied: wcwidth in
/usr/local/lib/python3.10/dist-packages (from prompt-toolkit!=3.0.0,!
=3.0.1,<3.1.0,>=2.0.0->ipython) (0.2.13)
Downloading jedi-0.19.2-py2.py3-none-any.whl (1.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 24.0 MB/s eta
0:00:00
etadata (21 kB)
Requirement already satisfied: torch>=2.1.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(2.5.1+cu121)
Requirement already satisfied: tqdm>=4.57.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(4.66.6)
Requirement already satisfied: PyYAML>=5.4 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(6.0.2)
Requirement already satisfied: fsspec>=2022.5.0 in
/usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0-
>pytorch-lightning) (2024.10.0)
Collecting torchmetrics>=0.7.0 (from pytorch-lightning)
  Downloading torchmetrics-1.6.0-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(24.2)
Requirement already satisfied: typing-extensions>=4.4.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(4.12.2)
Collecting lightning-utilities>=0.10.0 (from pytorch-lightning)
  Downloading lightning_utilities-0.11.9-py3-none-any.whl.metadata
(5.2 kB)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
/usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0-
>pytorch-lightning) (3.11.9)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from lightning-
utilities>=0.10.0->pytorch-lightning) (75.1.0)
```

```
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-
lightning) (3.16.1)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-
lightning) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-
lightning) (3.1.4)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-
lightning) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1-
>torch>=2.1.0->pytorch-lightning) (1.3.0)
Requirement already satisfied: numpy>1.20.0 in
/usr/local/lib/python3.10/dist-packages (from torchmetrics>=0.7.0-
>pytorch-lightning) (1.26.4)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (2.4.4)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (1.3.1)
Requirement already satisfied: async-timeout<6.0,>=4.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (4.0.3)
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (6.1.0)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (0.2.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (1.18.3)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=2.1.0-
>pytorch-lightning) (3.0.2)
Requirement already satisfied: idna>=2.0 in
/usr/local/lib/python3.10/dist-packages (from yarl<2.0,>=1.17.0-
>aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-
lightning) (3.10)
```

```
Downloading pytorch_lightning-2.4.0-py3-none-any.whl (815 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 815.2/815.2 kB 20.4 MB/s eta
0:00:00
etrics-1.6.0-py3-none-any.whl (926 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 926.4/926.4 kB 54.3 MB/s eta
0:00:00
etrics, pytorch-lightning
Successfully installed lightning-utilities-0.11.9 pytorch-lightning-
2.4.0 torchmetrics-1.6.0
Requirement already satisfied: opencv-python in
/usr/local/lib/python3.10/dist-packages (4.10.0.84)
Requirement already satisfied: numpy>=1.21.2 in
/usr/local/lib/python3.10/dist-packages (from opencv-python) (1.26.4)
Requirement already satisfied: imageio in
/usr/local/lib/python3.10/dist-packages (2.36.1)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from imageio) (1.26.4)
Requirement already satisfied: pillow>=8.3.2 in
/usr/local/lib/python3.10/dist-packages (from imageio) (11.0.0)
Requirement already satisfied: ftfy in /usr/local/lib/python3.10/dist-
packages (6.3.1)
Requirement already satisfied: regex in
/usr/local/lib/python3.10/dist-packages (2024.9.11)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (4.66.6)
Requirement already satisfied: wcwidth in
/usr/local/lib/python3.10/dist-packages (from ftfy) (0.2.13)
Requirement already satisfied: altair in
/usr/local/lib/python3.10/dist-packages (4.2.2)
Requirement already satisfied: entrypoints in
/usr/local/lib/python3.10/dist-packages (from altair) (0.4)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from altair) (3.1.4)
Requirement already satisfied: jsonschema>=3.0 in
/usr/local/lib/python3.10/dist-packages (from altair) (4.23.0)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from altair) (1.26.4)
Requirement already satisfied: pandas>=0.18 in
/usr/local/lib/python3.10/dist-packages (from altair) (2.2.2)
Requirement already satisfied: toolz in
/usr/local/lib/python3.10/dist-packages (from altair) (0.12.1)
Requirement already satisfied: attrs>=22.2.0 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair)
(24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair)
(2024.10.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair)
```

```
(0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair)
(0.22.3)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas>=0.18->altair)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas>=0.18->altair)
(2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas>=0.18->altair)
(2024.2)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->altair) (3.0.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas>=0.18->altair) (1.16.0)
Requirement already satisfied: protobuf in
/usr/local/lib/python3.10/dist-packages (4.25.5)
Collecting protobuf
  Downloading protobuf-5.29.1-cp38-abi3-
manylinux2014_x86_64.whl.metadata (592 bytes)
Downloading protobuf-5.29.1-cp38-abi3-manylinux2014_x86_64.whl (319
kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 319.7/319.7 kB 7.9 MB/s eta
0:00:00
pting uninstall: protobuf
    Found existing installation: protobuf 4.25.5
    Uninstalling protobuf-4.25.5:
      Successfully uninstalled protobuf-4.25.5
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
tensorflow 2.17.1 requires protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3, but you have protobuf
5.29.1 which is incompatible.
tensorflow-metadata 1.13.1 requires protobuf<5,>=3.20.3, but you have
protobuf 5.29.1 which is incompatible.
Successfully installed protobuf-5.29.1
Requirement already satisfied: timm in /usr/local/lib/python3.10/dist-
packages (1.0.12)
Requirement already satisfied: torch in
/usr/local/lib/python3.10/dist-packages (from timm) (2.5.1+cu121)
Requirement already satisfied: torchvision in
/usr/local/lib/python3.10/dist-packages (from timm) (0.20.1+cu121)
Requirement already satisfied: pyyaml in
/usr/local/lib/python3.10/dist-packages (from timm) (6.0.2)
Requirement already satisfied: huggingface_hub in
```

```
/usr/local/lib/python3.10/dist-packages (from timm) (0.26.3)
Requirement already satisfied: safetensors in
/usr/local/lib/python3.10/dist-packages (from timm) (0.4.5)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from huggingface_hub->timm)
(3.16.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface_hub->timm)
(2024.10.0)
Requirement already satisfied: packaging>=20.9 in
/usr/local/lib/python3.10/dist-packages (from huggingface_hub->timm)
(24.2)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from huggingface_hub->timm)
(2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in
/usr/local/lib/python3.10/dist-packages (from huggingface_hub->timm)
(4.66.6)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface_hub->timm)
(4.12.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch->timm) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch->timm) (3.1.4)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch->timm) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch-
>timm) (1.3.0)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from torchvision->timm)
(1.26.4)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/usr/local/lib/python3.10/dist-packages (from torchvision->timm)
(11.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch->timm)
(3.0.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests-
>huggingface_hub->timm) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests-
>huggingface_hub->timm) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests-
>huggingface_hub->timm) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
```

```
/usr/local/lib/python3.10/dist-packages (from requests-
>huggingface_hub->timm) (2024.8.30)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.55.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: numpy<2,>=1.21 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib) (1.16.0)
Requirement already satisfied: wandb in
/usr/local/lib/python3.10/dist-packages (0.18.7)
Requirement already satisfied: click!=8.0.0,>=7.1 in
/usr/local/lib/python3.10/dist-packages (from wandb) (8.1.7)
Requirement already satisfied: docker-pycreds>=0.4.0 in
/usr/local/lib/python3.10/dist-packages (from wandb) (0.4.0)
Requirement already satisfied: gitpython!=3.1.29,>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from wandb) (3.1.43)
Requirement already satisfied: platformdirs in
/usr/local/lib/python3.10/dist-packages (from wandb) (4.3.6)
Requirement already satisfied: protobuf!=4.21.0,!=5.28.0,<6,>=3.19.0
in /usr/local/lib/python3.10/dist-packages (from wandb) (5.29.1)
Requirement already satisfied: psutil>=5.0.0 in
/usr/local/lib/python3.10/dist-packages (from wandb) (5.9.5)
Requirement already satisfied: pyyaml in
/usr/local/lib/python3.10/dist-packages (from wandb) (6.0.2)
Requirement already satisfied: requests<3,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from wandb) (2.32.3)
Requirement already satisfied: sentry-sdk>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from wandb) (2.19.0)
Requirement already satisfied: setproctitle in
/usr/local/lib/python3.10/dist-packages (from wandb) (1.3.4)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from wandb) (75.1.0)
```

```
Requirement already satisfied: typing-extensions<5,>=4.4 in
/usr/local/lib/python3.10/dist-packages (from wandb) (4.12.2)
Requirement already satisfied: six>=1.4.0 in
/usr/local/lib/python3.10/dist-packages (from docker-pycreds>=0.4.0-
>wandb) (1.16.0)
Requirement already satisfied: gitdb<5,>=4.0.1 in
/usr/local/lib/python3.10/dist-packages (from gitpython!
=3.1.29,>=1.0.0->wandb) (4.0.11)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.0.0-
>wandb) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.0.0-
>wandb) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.0.0-
>wandb) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.0.0-
>wandb) (2024.8.30)
Requirement already satisfied: smmap<6,>=3.0.1 in
/usr/local/lib/python3.10/dist-packages (from gitdb<5,>=4.0.1-
>gitpython!=3.1.29,>=1.0.0->wandb) (5.0.1)
Requirement already satisfied: h5py in /usr/local/lib/python3.10/dist-
packages (3.12.1)
Requirement already satisfied: numpy>=1.19.3 in
/usr/local/lib/python3.10/dist-packages (from h5py) (1.26.4)
Collecting cmake==3.14.4
  Downloading cmake-3.14.4-py3-none-manylinux2010_x86_64.whl.metadata
(5.4 kB)
Downloading cmake-3.14.4-py3-none-manylinux2010_x86_64.whl (15.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15.6/15.6 MB 101.0 MB/s eta
0:00:00
ake
  Attempting uninstall: cmake
    Found existing installation: cmake 3.30.5
    Uninstalling cmake-3.30.5:
      Successfully uninstalled cmake-3.30.5
Successfully installed cmake-3.14.4
Collecting tensorflow-gpu==2.9.1
  Downloading tensorflow_gpu-2.9.1-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.0 kB)
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(1.6.3)
Collecting flatbuffers<2,>=1.12 (from tensorflow-gpu==2.9.1)
```

```
  Downloading flatbuffers-1.12-py2.py3-none-any.whl.metadata (872
bytes)
Collecting gast<=0.4.0,>=0.2.1 (from tensorflow-gpu==2.9.1)
  Downloading gast-0.4.0-py3-none-any.whl.metadata (1.1 kB)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(1.68.1)
Requirement already satisfied: h5py>=2.9.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(3.12.1)
Collecting keras<2.10.0,>=2.9.0rc0 (from tensorflow-gpu==2.9.1)
  Downloading keras-2.9.0-py2.py3-none-any.whl.metadata (1.3 kB)
Collecting keras-preprocessing>=1.1.1 (from tensorflow-gpu==2.9.1)
  Downloading Keras_Preprocessing-1.1.2-py2.py3-none-any.whl.metadata
(1.9 kB)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(18.1.1)
Requirement already satisfied: numpy>=1.20 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(1.26.4)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(3.4.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(24.2)
Collecting protobuf<3.20,>=3.9.2 (from tensorflow-gpu==2.9.1)
  Downloading protobuf-3.19.6-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (787 bytes)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(75.1.0)
Requirement already satisfied: six>=1.12.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(1.16.0)
Collecting tensorboard<2.10,>=2.9 (from tensorflow-gpu==2.9.1)
  Downloading tensorboard-2.9.1-py3-none-any.whl.metadata (1.9 kB)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(0.37.1)
Collecting tensorflow-estimator<2.10.0,>=2.9.0rc0 (from tensorflow-
gpu==2.9.1)
  Downloading tensorflow_estimator-2.9.0-py2.py3-none-any.whl.metadata
(1.3 kB)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
```

```
(2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow-gpu==2.9.1)
(1.17.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0-
>tensorflow-gpu==2.9.1) (0.45.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.10,>=2.9-
>tensorflow-gpu==2.9.1) (2.27.0)
Collecting google-auth-oauthlib<0.5,>=0.4.1 (from
tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1)
  Downloading google_auth_oauthlib-0.4.6-py2.py3-none-any.whl.metadata
(2.7 kB)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.10,>=2.9-
>tensorflow-gpu==2.9.1) (3.7)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.10,>=2.9-
>tensorflow-gpu==2.9.1) (2.32.3)
Collecting tensorboard-data-server<0.7.0,>=0.6.0 (from
tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1)
  Downloading tensorboard_data_server-0.6.1-py3-none-
manylinux2010_x86_64.whl.metadata (1.1 kB)
Collecting tensorboard-plugin-wit>=1.6.0 (from tensorboard<2.10,>=2.9-
>tensorflow-gpu==2.9.1)
  Downloading tensorboard_plugin_wit-1.8.1-py3-none-any.whl.metadata
(873 bytes)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.10,>=2.9-
>tensorflow-gpu==2.9.1) (3.1.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1) (5.5.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1) (0.4.1)
Requirement already satisfied: rsa<5,>=3.1.4 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1)
(1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
```

```
>tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1) (2024.8.30)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1-
>tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1) (3.0.2)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in
/usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1-
>google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1)
(0.6.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.10/dist-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard<2.10,>=2.9->tensorflow-gpu==2.9.1) (3.2.2)
Downloading tensorflow_gpu-2.9.1-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (511.7 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 511.7/511.7 MB 1.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 69.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.6/42.6 kB 3.5 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 52.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.8/5.8 MB 105.2 MB/s eta
0:00:00
ator-2.9.0-py2.py3-none-any.whl (438 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 438.7/438.7 kB 35.9 MB/s eta
0:00:00
anylinux2010_x86_64.whl (4.9 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.9/4.9 MB 102.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 781.3/781.3 kB 47.0 MB/s eta
0:00:00
ator, tensorboard-data-server, protobuf, keras-preprocessing, gast,
google-auth-oauthlib, tensorboard, tensorflow-gpu
  Attempting uninstall: keras
    Found existing installation: keras 3.5.0
    Uninstalling keras-3.5.0:
      Successfully uninstalled keras-3.5.0
  Attempting uninstall: flatbuffers
```

```
    Found existing installation: flatbuffers 24.3.25
    Uninstalling flatbuffers-24.3.25:
      Successfully uninstalled flatbuffers-24.3.25
  Attempting uninstall: tensorboard-data-server
    Found existing installation: tensorboard-data-server 0.7.2
    Uninstalling tensorboard-data-server-0.7.2:
      Successfully uninstalled tensorboard-data-server-0.7.2
  Attempting uninstall: protobuf
    Found existing installation: protobuf 5.29.1
    Uninstalling protobuf-5.29.1:
      Successfully uninstalled protobuf-5.29.1
  Attempting uninstall: gast
    Found existing installation: gast 0.6.0
    Uninstalling gast-0.6.0:
      Successfully uninstalled gast-0.6.0
  Attempting uninstall: google-auth-oauthlib
    Found existing installation: google-auth-oauthlib 1.2.1
    Uninstalling google-auth-oauthlib-1.2.1:
      Successfully uninstalled google-auth-oauthlib-1.2.1
  Attempting uninstall: tensorboard
    Found existing installation: tensorboard 2.17.1
    Uninstalling tensorboard-2.17.1:
      Successfully uninstalled tensorboard-2.17.1
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
google-ai-generativelanguage 0.6.10 requires protobuf!=4.21.0,!
=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but
you have protobuf 3.19.6 which is incompatible.
google-cloud-aiplatform 1.73.0 requires protobuf!=4.21.0,!=4.21.1,!
=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but you have
protobuf 3.19.6 which is incompatible.
google-cloud-bigquery-connection 1.16.1 requires protobuf!=4.21.0,!
=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but
you have protobuf 3.19.6 which is incompatible.
google-cloud-bigquery-storage 2.27.0 requires protobuf!=3.20.0,!
=3.20.1,!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!
=4.21.5,<6.0.0dev,>=3.20.2, but you have protobuf 3.19.6 which is
incompatible.
google-cloud-bigtable 2.27.0 requires protobuf!=4.21.0,!=4.21.1,!
=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but you have
protobuf 3.19.6 which is incompatible.
google-cloud-datastore 2.20.1 requires protobuf!=3.20.0,!=3.20.1,!
=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!
=4.21.5,<6.0.0dev,>=3.20.2, but you have protobuf 3.19.6 which is
incompatible.
google-cloud-firestore 2.19.0 requires protobuf!=3.20.0,!=3.20.1,!
=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!
=4.21.5,<6.0.0dev,>=3.20.2, but you have protobuf 3.19.6 which is
```

```
incompatible.
google-cloud-functions 1.18.1 requires protobuf!=4.21.0,!=4.21.1,!
=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but you have
protobuf 3.19.6 which is incompatible.
google-cloud-iam 2.16.1 requires protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but you have protobuf
3.19.6 which is incompatible.
google-cloud-language 2.15.1 requires protobuf!=4.21.0,!=4.21.1,!
=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but you have
protobuf 3.19.6 which is incompatible.
google-cloud-pubsub 2.27.1 requires protobuf!=4.21.0,!=4.21.1,!
=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but you have
protobuf 3.19.6 which is incompatible.
google-cloud-resource-manager 1.13.1 requires protobuf!=4.21.0,!
=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but
you have protobuf 3.19.6 which is incompatible.
google-cloud-translate 3.17.0 requires protobuf!=4.21.0,!=4.21.1,!
=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but you have
protobuf 3.19.6 which is incompatible.
googleapis-common-protos 1.66.0 requires protobuf!=3.20.0,!=3.20.1,!
=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0.dev0,>=3.20.2, but
you have protobuf 3.19.6 which is incompatible.
grpc-google-iam-v1 0.13.1 requires protobuf!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.2, but you have protobuf
3.19.6 which is incompatible.
grpcio-status 1.62.3 requires protobuf>=4.21.6, but you have protobuf
3.19.6 which is incompatible.
pandas-gbq 0.24.0 requires google-auth-oauthlib>=0.7.0, but you have
google-auth-oauthlib 0.4.6 which is incompatible.
tensorflow 2.17.1 requires flatbuffers>=24.3.25, but you have
flatbuffers 1.12 which is incompatible.
tensorflow 2.17.1 requires keras>=3.2.0, but you have keras 2.9.0
which is incompatible.
tensorflow 2.17.1 requires protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3, but you have protobuf
3.19.6 which is incompatible.
tensorflow 2.17.1 requires tensorboard<2.18,>=2.17, but you have
tensorboard 2.9.1 which is incompatible.
tensorflow-datasets 4.9.7 requires protobuf>=3.20, but you have
protobuf 3.19.6 which is incompatible.
tensorflow-metadata 1.13.1 requires protobuf<5,>=3.20.3, but you have
protobuf 3.19.6 which is incompatible.
Successfully installed flatbuffers-1.12 gast-0.4.0 google-auth-
oauthlib-0.4.6 keras-2.9.0 keras-preprocessing-1.1.2 protobuf-3.19.6
tensorboard-2.9.1 tensorboard-data-server-0.6.1 tensorboard-plugin-
wit-1.8.1 tensorflow-estimator-2.9.0 tensorflow-gpu-2.9.1
```


```
{"id":"ee3450767eb5487d8f9bb96d0436fa37","pip_warning":{"packages":
["google"]}}
```

```
/bin/bash: line 1: conda: command not found
/root
Cloning into 'habitat-lab'...
remote: Enumerating objects: 65024, done.ote: Counting objects: 100%
(992/992), done.ote: Compressing objects: 100% (429/429), done.ote:
Total 65024 (delta 645), reused 844 (delta 554), pack-reused 64032
(from 1)==0.22.0
  Downloading gym-0.22.0.tar.gz (631 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 631.1/631.1 kB 14.3 MB/s eta
0:00:00
ents to build wheel ... etadata (pyproject.toml) ... ent already
satisfied: numpy>=1.18.0 in /usr/local/lib/python3.10/dist-packages
(from gym==0.22.0) (1.26.4)
Requirement already satisfied: cloudpickle>=1.2.0 in
/usr/local/lib/python3.10/dist-packages (from gym==0.22.0) (3.1.0)
Requirement already satisfied: gym_notices>=0.0.4 in
/usr/local/lib/python3.10/dist-packages (from gym==0.22.0) (0.0.8)
Building wheels for collected packages: gym
  Building wheel for gym (pyproject.toml) ... : filename=gym-0.22.0-
py3-none-any.whl size=708390
sha256=56439d41fc2114d3df4b8110e1f7060a72f8c8fe27a807fbcf9dad0505f90fd
8
  Stored in directory:
/root/.cache/pip/wheels/42/e8/e8/6dfbc92a1dcd76c1a5e2bb982750fd6b7e792
239f46039e6b1
Successfully built gym
Installing collected packages: gym
  Attempting uninstall: gym
    Found existing installation: gym 0.25.2
    Uninstalling gym-0.25.2:
      Successfully uninstalled gym-0.25.2
Successfully installed gym-0.22.0
/root/habitat-lab
Note: switching to 'bfba72f47800819d858a6859b14cfa26122c2762'.

You are in 'detached HEAD' state. You can look around, make
experimental
changes and commit them, and you can discard any commits you make in
this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you
may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -
```

```
Turn off this advice by setting config variable advice.detachedHead to
false

HEAD is now at bfba72f4 [CI] Fixing pytorch restoration from cache
(#921)
/usr/local/lib/python3.10/dist-packages/setuptools/__init__.py:94:
_DeprecatedInstaller: setuptools.installer and fetch_build_eggs are
deprecated.
!!


********************************************************************
**********
        Requirements should be satisfied by a PEP 517 installer.
        If you are using pip, you can try `pip install --use-pep517`.

********************************************************************
**********

!!
  dist.fetch_build_eggs(dist.setup_requires)
/usr/local/lib/python3.10/dist-packages/setuptools/_distutils/dist.py:
261: UserWarning: Unknown distribution option: 'tests_require'
  warnings.warn(msg)
running develop
/usr/local/lib/python3.10/dist-packages/setuptools/command/develop.py:
41: EasyInstallDeprecationWarning: easy_install command is deprecated.
!!


********************************************************************
**********
        Please avoid running ``setup.py`` and ``easy_install``.
        Instead, use pypa/build, pypa/installer or other
        standards-based tools.

        See https://github.com/pypa/setuptools/issues/917 for details.

********************************************************************
**********

!!
  easy_install.initialize_options(self)
/root/habitat-lab/setup.py:57: SetuptoolsDeprecationWarning: setup.py
install is deprecated.
!!


********************************************************************
**********
```

```
        Please avoid running ``setup.py`` directly.
        Instead, use pypa/build, pypa/installer or other
        standards-based tools.

        See https://blog.ganssle.io/articles/2021/10/setup-py-
deprecated.html for details.

********************************************************************
**********

!!
  super().initialize_options()
running egg_info
creating habitat.egg-info
writing habitat.egg-info/PKG-INFO
writing dependency_links to habitat.egg-info/dependency_links.txt
writing requirements to habitat.egg-info/requires.txt
writing top-level names to habitat.egg-info/top_level.txt
writing manifest file 'habitat.egg-info/SOURCES.txt'
reading manifest file 'habitat.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
adding license file 'LICENSE'
writing manifest file 'habitat.egg-info/SOURCES.txt'
running build_ext
Creating /usr/local/lib/python3.10/dist-packages/habitat.egg-link
(link to .)
Adding habitat 0.2.2 to easy-install.pth file

Installed /root/habitat-lab
Processing dependencies for habitat==0.2.2
Searching for lmdb>=0.98
Reading https://pypi.org/simple/lmdb/
Downloading
https://files.pythonhosted.org/packages/2d/39/24b71e2b10edf46a1bc0c46e
aca5c5253bbc7fea059c0626628c0754223c/lmdb-1.5.1-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl#sha256=8f5796c4ad57bce4
95395af7a24fd5f7f307b2ac30f91caa6c8046e90d40edba
Best match: lmdb 1.5.1
Processing lmdb-1.5.1-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Installing lmdb-1.5.1-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl to
/usr/local/lib/python3.10/dist-packages
Adding lmdb 1.5.1 to easy-install.pth file

Installed /usr/local/lib/python3.10/dist-packages/lmdb-1.5.1-py3.10-
linux-x86_64.egg
Searching for webdataset==0.1.40
Reading https://pypi.org/simple/webdataset/
Downloading
```

```
https://files.pythonhosted.org/packages/16/bc/8b98d07eb97a51584ff305b4
68a13628f3905964d597c62513f6beacb4a4/webdataset-0.1.40-py3-none-
any.whl#sha256=f20e3f1143395a321fad7d6cc10e07c4b269f0270ef6fdc1d4ed944
21bce1bf0
Best match: webdataset 0.1.40
Processing webdataset-0.1.40-py3-none-any.whl
Installing webdataset-0.1.40-py3-none-any.whl to
/usr/local/lib/python3.10/dist-packages
Adding webdataset 0.1.40 to easy-install.pth file
detected new path './lmdb-1.5.1-py3.10-linux-x86_64.egg'

Installed /usr/local/lib/python3.10/dist-packages/webdataset-0.1.40-
py3.10.egg
Searching for ifcfg
Reading https://pypi.org/simple/ifcfg/
Downloading
https://files.pythonhosted.org/packages/d8/35/d5d8da34881946022e311693
a875e2d162045f20ea17d7d33707ab929eb8/ifcfg-0.24-py2.py3-none-
any.whl#sha256=950f31ddc3fdd877724fb4154b744434471106088ecf10170b96063
c24b3c0b7
Best match: ifcfg 0.24
Processing ifcfg-0.24-py2.py3-none-any.whl
Installing ifcfg-0.24-py2.py3-none-any.whl to
/usr/local/lib/python3.10/dist-packages
Adding ifcfg 0.24 to easy-install.pth file
detected new path './webdataset-0.1.40-py3.10.egg'

Installed /usr/local/lib/python3.10/dist-packages/ifcfg-0.24-
py3.10.egg
Searching for yacs>=0.1.8
Reading https://pypi.org/simple/yacs/
Downloading
https://files.pythonhosted.org/packages/38/4f/fe9a4d472aa867878ce3bb7e
fb16654c5d63672b86dc0e6e953a67018433/yacs-0.1.8-py3-none-
any.whl#sha256=99f893e30497a4b66842821bac316386f7bd5c4f47ad35c9073ef08
9aa33af32
Best match: yacs 0.1.8
Processing yacs-0.1.8-py3-none-any.whl
Installing yacs-0.1.8-py3-none-any.whl to
/usr/local/lib/python3.10/dist-packages
Adding yacs 0.1.8 to easy-install.pth file
detected new path './ifcfg-0.24-py3.10.egg'

Installed /usr/local/lib/python3.10/dist-packages/yacs-0.1.8-
py3.10.egg
Searching for numpy-quaternion>=2019.3.18.14.33.20
Reading https://pypi.org/simple/numpy-quaternion/
Downloading
https://files.pythonhosted.org/packages/a2/56/7bf6619eafac834c252a05cd
f502421dc31131c09d894b1d5d8709044bf3/numpy_quaternion-2024.0.3-cp310-
```

```
cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux
2014_x86_64.whl#sha256=ce23a7555ec8ab1eab6d28487068038565799d4fa61e9bf
dab04d919b49ddc74
Best match: numpy-quaternion 2024.0.3
Processing numpy_quaternion-2024.0.3-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux
2014_x86_64.whl
Installing numpy_quaternion-2024.0.3-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux
2014_x86_64.whl to /usr/local/lib/python3.10/dist-packages
Adding numpy-quaternion 2024.0.3 to easy-install.pth file
detected new path './yacs-0.1.8-py3.10.egg'

Installed /usr/local/lib/python3.10/dist-packages/numpy_quaternion-
2024.0.3-py3.10-linux-x86_64.egg
Searching for tensorboard==2.8.0
Reading https://pypi.org/simple/tensorboard/
Downloading
https://files.pythonhosted.org/packages/f7/fd/67c61276de025801cfa8a1b9
af2d7c577e7f27c17b6bff2baca20bf03543/tensorboard-2.8.0-py3-none-
any.whl#sha256=65a338e4424e9079f2604923bdbe301792adce2ace1be68da6b3ddf
005170def
Best match: tensorboard 2.8.0
Processing tensorboard-2.8.0-py3-none-any.whl
Installing tensorboard-2.8.0-py3-none-any.whl to
/usr/local/lib/python3.10/dist-packages
Adding tensorboard 2.8.0 to easy-install.pth file
detected new path './numpy_quaternion-2024.0.3-py3.10-linux-
x86_64.egg'
Installing tensorboard script to /usr/local/bin

Installed /usr/local/lib/python3.10/dist-packages/tensorboard-2.8.0-
py3.10.egg
Searching for protobuf==3.20.1
Reading https://pypi.org/simple/protobuf/
Downloading
https://files.pythonhosted.org/packages/ef/c8/2e7f7feaf804b7206e6cc8fa
3f0f49834a78f7cb127813d2c45e42d5f7bf/protobuf-3.20.1-py2.py3-none-
any.whl#sha256=adfc6cf69c7f8c50fd24c793964eef18f0ac321315439d949458206
12849c388
Best match: protobuf 3.20.1
Processing protobuf-3.20.1-py2.py3-none-any.whl
Installing protobuf-3.20.1-py2.py3-none-any.whl to
/usr/local/lib/python3.10/dist-packages
Adding protobuf 3.20.1 to easy-install.pth file
detected new path './tensorboard-2.8.0-py3.10.egg'

Installed /usr/local/lib/python3.10/dist-packages/protobuf-3.20.1-
py3.10.egg
```

```
Searching for objectio
Reading https://pypi.org/simple/objectio/
Downloading
https://files.pythonhosted.org/packages/86/e3/a132a91c4e9fd5e59c947263
c7ef4e3415640fa151344f858e2def8c1726/objectio-0.2.29-py3-none-
any.whl#sha256=2577426a393d0e8a5b27370798e493b89f2d8c5619614199ea5521e
4feee6f11
Best match: objectio 0.2.29
Processing objectio-0.2.29-py3-none-any.whl
Installing objectio-0.2.29-py3-none-any.whl to
/usr/local/lib/python3.10/dist-packages
Adding objectio 0.2.29 to easy-install.pth file
detected new path './protobuf-3.20.1-py3.10.egg'
Installing obj script to /usr/local/bin

Installed /usr/local/lib/python3.10/dist-packages/objectio-0.2.29-
py3.10.egg
Searching for braceexpand
Reading https://pypi.org/simple/braceexpand/
Downloading
https://files.pythonhosted.org/packages/fa/93/e8c04e80e82391a6e51f218c
a49720f64236bc824e92152a2633b74cf7ab/braceexpand-0.1.7-py2.py3-none-
any.whl#sha256=91332d53de7828103dcae5773fb43bc34950b0c8160e35e0f44c442
7a3b85014
Best match: braceexpand 0.1.7
Processing braceexpand-0.1.7-py2.py3-none-any.whl
Installing braceexpand-0.1.7-py2.py3-none-any.whl to
/usr/local/lib/python3.10/dist-packages
Adding braceexpand 0.1.7 to easy-install.pth file
detected new path './objectio-0.2.29-py3.10.egg'

Installed /usr/local/lib/python3.10/dist-packages/braceexpand-0.1.7-
py3.10.egg
Searching for simplejson
Reading https://pypi.org/simple/simplejson/
Downloading
https://files.pythonhosted.org/packages/61/20/0035a288deaff05397d6cc01
45b33f3dd2429b99cdc880de4c5eca41ca72/simplejson-3.19.3-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux
2014_x86_64.whl#sha256=f8efb03ca77bd7725dfacc9254df00d73e6f43013cf39bd
37ef1a8ed0ebb5165
Best match: simplejson 3.19.3
Processing simplejson-3.19.3-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux
2014_x86_64.whl
Installing simplejson-3.19.3-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux
2014_x86_64.whl to /usr/local/lib/python3.10/dist-packages
Adding simplejson 3.19.3 to easy-install.pth file
detected new path './braceexpand-0.1.7-py3.10.egg'
```

```
Installed /usr/local/lib/python3.10/dist-packages/simplejson-3.19.3-
py3.10-linux-x86_64.egg
Searching for moviepy==1.0.3
Best match: moviepy 1.0.3
Adding moviepy 1.0.3 to easy-install.pth file
detected new path './simplejson-3.19.3-py3.10-linux-x86_64.egg'

Using /usr/local/lib/python3.10/dist-packages
Searching for torch==2.5.1+cu121
Best match: torch 2.5.1+cu121
Adding torch 2.5.1+cu121 to easy-install.pth file
Installing convert-caffe2-to-onnx script to /usr/local/bin
Installing convert-onnx-to-caffe2 script to /usr/local/bin
Installing torchfrtrace script to /usr/local/bin
Installing torchrun script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for attrs==24.2.0
Best match: attrs 24.2.0
Adding attrs 24.2.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for scipy==1.13.1
Best match: scipy 1.13.1
Adding scipy 1.13.1 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for numba==0.60.0
Best match: numba 0.60.0
Adding numba 0.60.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for numpy==1.26.4
Best match: numpy 1.26.4
Adding numpy 1.26.4 to easy-install.pth file
Installing f2py script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for gym==0.22.0
Best match: gym 0.22.0
Adding gym 0.22.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for opencv-python==4.10.0.84
Best match: opencv-python 4.10.0.84
Adding opencv-python 4.10.0.84 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for tqdm==4.66.6
```

```
Best match: tqdm 4.66.6
Adding tqdm 4.66.6 to easy-install.pth file
Installing tqdm script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for imageio==2.36.1
Best match: imageio 2.36.1
Adding imageio 2.36.1 to easy-install.pth file
Installing imageio_download_bin script to /usr/local/bin
Installing imageio_remove_bin script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for imageio-ffmpeg==0.5.1
Best match: imageio-ffmpeg 0.5.1
Adding imageio-ffmpeg 0.5.1 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for PyYAML==6.0.2
Best match: PyYAML 6.0.2
Adding PyYAML 6.0.2 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for msgpack==1.1.0
Best match: msgpack 1.1.0
Adding msgpack 1.1.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for pillow==11.0.0
Best match: pillow 11.0.0
Adding pillow 11.0.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for proglog==0.1.10
Best match: proglog 0.1.10
Adding proglog 0.1.10 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for requests==2.32.3
Best match: requests 2.32.3
Adding requests 2.32.3 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for decorator==4.4.2
Best match: decorator 4.4.2
Adding decorator 4.4.2 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for sympy==1.13.1
Best match: sympy 1.13.1
Adding sympy 1.13.1 to easy-install.pth file
```

```
Installing isympy script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for fsspec==2024.10.0
Best match: fsspec 2024.10.0
Adding fsspec 2024.10.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for jinja2==3.1.4
Best match: jinja2 3.1.4
Adding jinja2 3.1.4 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for networkx==3.4.2
Best match: networkx 3.4.2
Adding networkx 3.4.2 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for typing-extensions==4.12.2
Best match: typing-extensions 4.12.2
Adding typing-extensions 4.12.2 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages/setuptools/_vendor
Searching for filelock==3.16.1
Best match: filelock 3.16.1
Adding filelock 3.16.1 to easy-install.pth file
detected new path './setuptools/_vendor'

Using /usr/local/lib/python3.10/dist-packages
Searching for llvmlite==0.43.0
Best match: llvmlite 0.43.0
Adding llvmlite 0.43.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for wheel==0.45.1
Best match: wheel 0.45.1
Adding wheel 0.45.1 to easy-install.pth file
Installing wheel script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for werkzeug==3.1.3
Best match: werkzeug 3.1.3
Adding werkzeug 3.1.3 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for tensorboard-plugin-wit==1.8.1
Best match: tensorboard-plugin-wit 1.8.1
Adding tensorboard-plugin-wit 1.8.1 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for tensorboard-data-server==0.6.1
```

```
Best match: tensorboard-data-server 0.6.1
Adding tensorboard-data-server 0.6.1 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for setuptools==75.1.0
Best match: setuptools 75.1.0
Adding setuptools 75.1.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for Markdown==3.7
Best match: Markdown 3.7
Adding Markdown 3.7 to easy-install.pth file
Installing markdown_py script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for google-auth-oauthlib==0.4.6
Best match: google-auth-oauthlib 0.4.6
Adding google-auth-oauthlib 0.4.6 to easy-install.pth file
Installing google-oauthlib-tool script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for google-auth==2.27.0
Best match: google-auth 2.27.0
Adding google-auth 2.27.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for grpcio==1.68.1
Best match: grpcio 1.68.1
Adding grpcio 1.68.1 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for absl-py==1.4.0
Best match: absl-py 1.4.0
Adding absl-py 1.4.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for gym-notices==0.0.8
Best match: gym-notices 0.0.8
Adding gym-notices 0.0.8 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for cloudpickle==3.1.0
Best match: cloudpickle 3.1.0
Adding cloudpickle 3.1.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for typer==0.15.0
Best match: typer 0.15.0
Adding typer 0.15.0 to easy-install.pth file
Installing typer script to /usr/local/bin
```

```
Using /usr/local/lib/python3.10/dist-packages
Searching for certifi==2024.8.30
Best match: certifi 2024.8.30
Adding certifi 2024.8.30 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for urllib3==2.2.3
Best match: urllib3 2.2.3
Adding urllib3 2.2.3 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for idna==3.10
Best match: idna 3.10
Adding idna 3.10 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for charset-normalizer==3.4.0
Best match: charset-normalizer 3.4.0
Adding charset-normalizer 3.4.0 to easy-install.pth file
Installing normalizer script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for mpmath==1.3.0
Best match: mpmath 1.3.0
Adding mpmath 1.3.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for MarkupSafe==3.0.2
Best match: MarkupSafe 3.0.2
Adding MarkupSafe 3.0.2 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for requests-oauthlib==1.3.1
Best match: requests-oauthlib 1.3.1
Adding requests-oauthlib 1.3.1 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for rsa==4.9
Best match: rsa 4.9
Adding rsa 4.9 to easy-install.pth file
Installing pyrsa-decrypt script to /usr/local/bin
Installing pyrsa-encrypt script to /usr/local/bin
Installing pyrsa-keygen script to /usr/local/bin
Installing pyrsa-priv2pub script to /usr/local/bin
Installing pyrsa-sign script to /usr/local/bin
Installing pyrsa-verify script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for pyasn1-modules==0.4.1
Best match: pyasn1-modules 0.4.1
```

```
Adding pyasn1-modules 0.4.1 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for cachetools==5.5.0
Best match: cachetools 5.5.0
Adding cachetools 5.5.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for rich==13.9.4
Best match: rich 13.9.4
Adding rich 13.9.4 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for shellingham==1.5.4
Best match: shellingham 1.5.4
Adding shellingham 1.5.4 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for click==8.1.7
Best match: click 8.1.7
Adding click 8.1.7 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for oauthlib==3.2.2
Best match: oauthlib 3.2.2
Adding oauthlib 3.2.2 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for pyasn1==0.6.1
Best match: pyasn1 0.6.1
Adding pyasn1 0.6.1 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for pygments==2.18.0
Best match: pygments 2.18.0
Adding pygments 2.18.0 to easy-install.pth file
Installing pygmentize script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for markdown-it-py==3.0.0
Best match: markdown-it-py 3.0.0
Adding markdown-it-py 3.0.0 to easy-install.pth file
Installing markdown-it script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for mdurl==0.1.2
Best match: mdurl 0.1.2
Adding mdurl 0.1.2 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
```

```
Finished processing dependencies for habitat==0.2.2
Collecting openai==0.8.0
  Downloading openai-0.8.0.tar.gz (147 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 147.6/147.6 kB 5.1 MB/s eta
0:00:00
etadata (setup.py) ... ent already satisfied: requests>=2.20 in
/usr/local/lib/python3.10/dist-packages (from openai==0.8.0) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from openai==0.8.0) (4.66.6)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.20-
>openai==0.8.0) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.20-
>openai==0.8.0) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.20-
>openai==0.8.0) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.20-
>openai==0.8.0) (2024.8.30)
Building wheels for collected packages: openai
  Building wheel for openai (setup.py) ... e=openai-0.8.0-py3-none-
any.whl size=158497
sha256=51467692d1c1a4d0a49f77b517aafc3f88e7451cd91d1f5e2d41fe858dcf3d3
8
  Stored in directory:
/root/.cache/pip/wheels/fc/98/16/109a7a1e372570618944b8fd5e497200863c5
cc94a131562f3
Successfully built openai
Installing collected packages: openai
  Attempting uninstall: openai
    Found existing installation: openai 1.54.5
    Uninstalling openai-1.54.5:
      Successfully uninstalled openai-1.54.5
Successfully installed openai-0.8.0
Collecting grad-cam
  Downloading grad-cam-1.5.4.tar.gz (7.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.8/7.8 MB 69.1 MB/s eta
0:00:00
ents to build wheel ... etadata (pyproject.toml) ... ent already
satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from
grad-cam) (1.26.4)
Requirement already satisfied: Pillow in
/usr/local/lib/python3.10/dist-packages (from grad-cam) (11.0.0)
Requirement already satisfied: torch>=1.7.1 in
/usr/local/lib/python3.10/dist-packages (from grad-cam) (2.5.1+cu121)
Requirement already satisfied: torchvision>=0.8.2 in
/usr/local/lib/python3.10/dist-packages (from grad-cam) (0.20.1+cu121)
```

```
Collecting ttach (from grad-cam)
  Downloading ttach-0.0.3-py3-none-any.whl.metadata (5.2 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from grad-cam) (4.66.6)
Requirement already satisfied: opencv-python in
/usr/local/lib/python3.10/dist-packages (from grad-cam) (4.10.0.84)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (from grad-cam) (3.8.0)
Requirement already satisfied: scikit-learn in
/usr/local/lib/python3.10/dist-packages (from grad-cam) (1.5.2)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch>=1.7.1->grad-cam)
(3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.7.1->grad-cam)
(4.12.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch>=1.7.1->grad-cam)
(3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.7.1->grad-cam)
(3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch>=1.7.1->grad-cam)
(2024.10.0)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.7.1->grad-cam)
(1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1-
>torch>=1.7.1->grad-cam) (1.3.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->grad-cam)
(1.3.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->grad-cam)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->grad-cam)
(4.55.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->grad-cam)
(1.4.7)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->grad-cam)
(24.2)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->grad-cam)
(3.2.0)
```

```
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->grad-cam)
(2.8.2)
Requirement already satisfied: scipy>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->grad-cam)
(1.13.1)
Requirement already satisfied: joblib>=1.2.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->grad-cam)
(1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->grad-cam)
(3.5.0)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib->grad-cam) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.7.1-
>grad-cam) (3.0.2)
Downloading ttach-0.0.3-py3-none-any.whl (9.8 kB)
Building wheels for collected packages: grad-cam
  Building wheel for grad-cam (pyproject.toml) ... :
filename=grad_cam-1.5.4-py3-none-any.whl size=39588
sha256=f594b7e4b73401c307d13766338289c433c942e09c5f01c3b7d4261bd6b7fcf
f
  Stored in directory:
/root/.cache/pip/wheels/50/b0/82/1f97b5348c7fe9f0ce0ba18497202cafa5dec
4562bd5292680
Successfully built grad-cam
Installing collected packages: ttach, grad-cam
Successfully installed grad-cam-1.5.4 ttach-0.0.3
```

Helper Functions for RGB-D Video Generation and Visualization

This cell defines a set of helper functions used to **process, visualize, and export RGB-D observations** collected from the simulated environment.

During VLMap construction and navigation experiments, the agent generates:

- sequences of RGB images, and
- corresponding depth maps

These modalities are useful not only for map generation, but also for **qualitative inspection of agent behavior**, scene structure, and depth perception.

The utilities defined here support the following functionality:

- **Depth Loading**
  Load depth data stored in NumPy format for further processing and visualization.

- **Efficient Video Writing**
  Create video writers that automatically select between GPU-accelerated or CPU-based encoding depending on availability. This allows fast generation of MP4 videos from long RGB-D sequences.

- **RGB and Depth Video Creation**
  Convert ordered RGB frames and depth maps into synchronized video streams. Depth values are normalized and color-mapped to improve interpretability.

- **Inline Video Display**
  Render generated videos directly within the notebook for quick inspection, enabling side-by-side visualization of RGB and depth observations.

These visualization utilities are used throughout the notebook to:

- verify dataset integrity,
- inspect environment coverage during exploration,
- and qualitatively analyze navigation and mapping results.

```
# @title Helper functions for video creation and display. This would
combine the RGB images and heat maps sequentially which was created
using Matterport3D and Habitat simulator.
import sys
import os
import imageio
import numpy as np
import cv2
import tqdm
from IPython.display import HTML
from base64 import b64encode

if "google.colab" in sys.modules:
    os.environ["IMAGEIO_FFMPEG_EXE"] = "/usr/bin/ffmpeg"

# @markdown if the colab instance doesn't have GPU, untick the
following checkbox
has_gpu = True # @param {type: "boolean"}
codec = "h264"
if has_gpu:
  codec = "h264_nvenc"

def load_depth(depth_filepath):
    with open(depth_filepath, 'rb') as f:
        depth = np.load(f)
    return depth



def get_fast_video_writer(video_file: str, fps: int = 60):
    if (
```

```
        "google.colab" in sys.modules
        and os.path.splitext(video_file)[-1] == ".mp4"
        and os.environ.get("IMAGEIO_FFMPEG_EXE") == "/usr/bin/ffmpeg"
    ):
        # USE GPU Accelerated Hardware Encoding
        writer = imageio.get_writer(
            video_file,
            fps=fps,
            codec=codec,
            mode="I",
            bitrate="1000k",
            format="FFMPEG",
            ffmpeg_log_level="info",
            quality=10,
            output_params=["-minrate", "500k", "-maxrate", "5000k"],
        )
    else:
        # Use software encoding
        writer = imageio.get_writer(video_file, fps=fps)
    return writer

def create_video(data_dir: str, output_dir: str, fps: int = 30):

    rgb_dir = os.path.join(data_dir, "rgb")
    depth_dir = os.path.join(data_dir, "depth")
    rgb_out_path = os.path.join(output_dir, "rgb.mp4")
    depth_out_path = os.path.join(output_dir, "depth.mp4")
    rgb_writer = get_fast_video_writer(rgb_out_path, fps=fps)
    depth_writer = get_fast_video_writer(depth_out_path, fps=fps)

    rgb_list = sorted(os.listdir(rgb_dir), key=lambda x: int(
        x.split("_")[-1].split(".")[0]))
    depth_list = sorted(os.listdir(depth_dir), key=lambda x: int(
        x.split("_")[-1].split(".")[0]))

    rgb_list = [os.path.join(rgb_dir, x) for x in rgb_list]
    depth_list = [os.path.join(depth_dir, x) for x in depth_list]
    pbar = tqdm.tqdm(total=len(rgb_list), position=0, leave=True)
    for i, (rgb_path, depth_path) in enumerate(zip(rgb_list,
depth_list)):
        bgr = cv2.imread(rgb_path)
        rgb = cv2.cvtColor(bgr, cv2.COLOR_BGR2RGB)

        depth = load_depth(depth_path)
        depth_vis = (depth / 10 * 255).astype(np.uint8)

        depth_color = cv2.applyColorMap(depth_vis, cv2.COLORMAP_JET)

        rgb_writer.append_data(rgb)
        depth_writer.append_data(depth_color)
```

```
        pbar.update(1)
    rgb_writer.close()
    depth_writer.close()

def show_video(video_path, video_width = 1080):

    video_file = open(video_path, "r+b").read()

    video_url = f"data:video/mp4;base64,
{b64encode(video_file).decode()}"
    return HTML(f"""<video width={video_width} autoplay
controls><source src="{video_url}"></video>""")

def show_videos(video_paths, video_width = 1080):
    html = ""
    for video_path in video_paths:
        video_file = open(video_path, "r+b").read()

        video_url = f"data:video/mp4;base64,
{b64encode(video_file).decode()}"
        html += f"""<video width={video_width} autoplay
controls><source src="{video_url}"></video>
                """
    return HTML(html)
```

Generating RGB and Depth Videos from the Dataset

This cell uses the previously defined helper functions to generate video visualizations from the RGB-D dataset.

The function processes:

- the sequence of RGB images, and
- the corresponding depth maps

stored in the dataset directory, and converts them into two synchronized MP4 videos:

- an RGB video showing the visual appearance of the environment, and
- a depth video showing the scene structure encoded as color-mapped depth values.

These videos provide a qualitative overview of the environment and serve as a useful tool for verifying the integrity of the dataset before proceeding with VLMap construction and navigation experiments.

```
create_video("/content/jh4fc5c5qoQ_1", "/content/", fps=30)
```

```
  0%|          | 0/500 [00:00<?, ?it/s]WARNING:imageio_ffmpeg:IMAGEIO
FFMPEG_WRITER WARNING: input image is not divisible by
macro_block_size=16, resizing from (1080, 720) to (1088, 720) to
```

```
ensure video compatibility with most codecs and players. To prevent
resizing, make your input image divisible by the macro_block_size or
set the macro_block_size to 1 (risking incompatibility).
WARNING:imageio_ffmpeg:IMAGEIO FFMPEG_WRITER WARNING: input image is
not divisible by macro_block_size=16, resizing from (1080, 720) to
(1088, 720) to ensure video compatibility with most codecs and
players. To prevent resizing, make your input image divisible by the
macro_block_size or set the macro_block_size to 1 (risking
incompatibility).
100%|██████████| 500/500 [00:23<00:00, 21.01it/s]
```

Visualizing RGB and Depth Sequences

This cell displays the generated RGB and depth videos directly within the notebook.

The RGB video illustrates the visual appearance of the environment as observed by the agent, while the depth video highlights the scene geometry and spatial structure through color-mapped depth values.

Visualizing these modalities together helps verify:

- correct alignment between RGB and depth data,
- consistency across sequential frames,
- and overall scene coverage prior to VLMap construction.

```
# show rgb and depth
show_videos(["/content/rgb.mp4", "/content/depth.mp4"])
```

```
<IPython.core.display.HTML object>
```

Downloading Pretrained LSeg Model Checkpoints

This cell prepares the **LSeg semantic segmentation module** by downloading the required pretrained model checkpoints.

LSeg is a language-driven semantic segmentation model that operates in a shared vision–language embedding space. It is a core component of the VLMaps pipeline, as it enables extraction of dense visual-language features from RGB observations.

The downloaded checkpoint is stored in the expected directory structure so that it can be directly loaded by the VLMaps utilities during map construction.

At the end of this step, the environment contains:

- the pretrained LSeg weights,
- a correctly initialized checkpoint directory,
- and all necessary resources for semantic feature extraction.

```
%cd /content/vlmaps/lseg
!mkdir checkpoints
```

```
%cd checkpoints
!gdown 1ayk6NXURI_vIPlym16f_RG3ffxBWHxvb
!ls
%cd /content/vlmaps
```

```
/content/vlmaps/lseg
/content/vlmaps/lseg/checkpoints
Downloading...
From (original): https://drive.google.com/uc?
id=1ayk6NXURI_vIPlym16f_RG3ffxBWHxvb
From (redirected): https://drive.google.com/uc?
id=1ayk6NXURI_vIPlym16f_RG3ffxBWHxvb&confirm=t&uuid=681ab39f-8fa1-
4703-9221-a431413351ac
To: /content/vlmaps/lseg/checkpoints/demo_e200.ckpt
100% 3.10G/3.10G [00:23<00:00, 129MB/s]
demo_e200.ckpt
/content/vlmaps
```

VLMap Construction Parameters

This cell defines the core parameters used for constructing the Visual Language Map (VLMap) from RGB-D observations.

The parameters control the spatial resolution of the map, the scale at which the environment is discretized, and how depth data is sampled during back-projection. Together, these settings determine the fidelity, size, and semantic density of the resulting VLMap.

Key parameters include:

- **Cell size (meters per cell):** controls the physical resolution of the map.
- **Map resolution:** defines the overall spatial extent of the top-down grid.
- **Camera height:** used to filter floor points during 3D reconstruction.
- **Depth subsampling rate:** reduces computational load by sampling depth pixels.
- **Data directory:** specifies the location of RGB, depth, and pose data, as well as where generated maps and intermediate outputs are stored.

These parameters remain fixed throughout the VLMap generation process to ensure consistent spatial alignment and semantic aggregation.

```
# setup parameters
# @markdown meters per cell size
cs = 0.1 # @param {type: "number"}
# @markdown map resolution (gs x gs)
gs = 1800 # @param {type: "integer"}
# @markdown camera height (used for filtering out points on the floor)
camera_height = 1.8 # @param {type: "number"}
# @markdown depth pixels subsample rate
depth_sample_rate = 150 # @param {type: "integer"}
```

```
# @markdown data where rgb, depth, pose are loaded and map are saved
data_dir = "/content/jh4fc5c5qoQ_1/" # @param {type: "string"}
```

Helper Functions for Visual Language Map (VLMap) Creation

This cell defines the core helper functions required to construct a **Visual Language Map (VLMap)** from RGB-D observations.

The functions implemented here perform the following key operations:

1. Visual-Language Feature Extraction

- A pretrained **LSeg model** is used to extract dense, pixel-level visual-language embeddings from RGB images.
- LSeg operates in the CLIP embedding space, allowing visual features to be directly aligned with natural language concepts.
- CLIP text embeddings are generated for a predefined set of semantic labels and normalized for similarity comparison.

2. Depth Back-Projection and Coordinate Transformation

- Depth maps are converted into 3D point clouds using camera intrinsics.
- Points are transformed from the camera frame into a global reference frame using pose information.
- A configurable subsampling strategy is applied to reduce computational cost while preserving spatial coverage.

3. VLMap Construction

- The environment is discretized into a top-down 2D grid.
- Each grid cell stores:
  - an aggregated visual-language embedding,
  - semantic ground-truth labels (when available),
  - color information for visualization,
  - and obstacle occupancy information.
- Multiple observations projecting into the same grid cell are averaged to produce a stable semantic representation.

4. Obstacle and Semantic Map Generation

- Points above the floor plane are filtered using camera height constraints.
- An obstacle map is constructed to distinguish free space from occupied regions.
- Semantic labels are mapped from object IDs to class IDs for consistent indexing.

5. Batch Processing of RGB-D Frames

- RGB images, depth maps, semantic annotations, and poses are processed sequentially.
- Progress tracking is used to monitor map construction over long sequences.

- Intermediate and final maps are saved to disk for later querying and navigation.

Together, these utilities implement the full VLMap generation pipeline described in the original paper, enabling the creation of persistent, open-vocabulary spatial maps that can be queried using natural language.

```
# @title Helper functions for VLMap Creation

import os
import math

import numpy as np
import cv2
from tqdm import tqdm
import torch
import torchvision.transforms as transforms
import clip

from utils.clip_mapping_utils import load_pose, load_semantic,
load_obj2cls_dict, save_map, cvt_obj_id_2_cls_id, depth2pc,
transform_pc, get_sim_cam_mat, pos2grid_id, project_point

from lseg.modules.models.lseg_net import LSegEncNet
from lseg.additional_utils.models import resize_image, pad_image,
crop_image

def create_lseg_map_batch(img_save_dir, camera_height, cs=0.05,
gs=1000, depth_sample_rate=100):
    mask_version = 1 # 0, 1

    crop_size = 480 # 480
    base_size = 520 # 520
    lang = "door,chair,ground,ceiling,other"
    labels = lang.split(",")

    # loading models
    device = "cuda" if torch.cuda.is_available() else "cpu"
    print(device)
    clip_version = "ViT-B/32"
    clip_feat_dim = {'RN50': 1024, 'RN101': 512, 'RN50x4': 640,
'RN50x16': 768,
                     'RN50x64': 1024, 'ViT-B/32': 512, 'ViT-B/16': 512,
'ViT-L/14': 768}[clip_version]
    print("Loading CLIP model...")
    clip_model, preprocess = clip.load(clip_version)  #
clip.available_models()
    clip_model.to(device).eval()
    lang_token = clip.tokenize(labels)
    lang_token = lang_token.to(device)
    with torch.no_grad():
```

```
        text_feats = clip_model.encode_text(lang_token)
        text_feats = text_feats / text_feats.norm(dim=-1,
keepdim=True)
    text_feats = text_feats.cpu().numpy()
    model = LSegEncNet(lang, arch_option=0,
                       block_depth=0,
                       activation='lrelu',
                       crop_size=crop_size)
    model_state_dict = model.state_dict()
    pretrained_state_dict =
torch.load("lseg/checkpoints/demo_e200.ckpt")
    pretrained_state_dict = {k.lstrip('net.'): v for k, v in
pretrained_state_dict['state_dict'].items()}
    model_state_dict.update(pretrained_state_dict)
    model.load_state_dict(pretrained_state_dict)

    model.eval()
    model = model.cuda()

    norm_mean= [0.5, 0.5, 0.5]
    norm_std = [0.5, 0.5, 0.5]
    padding = [0.0] * 3
    transform = transforms.Compose(
        [
            transforms.ToTensor(),
            transforms.Normalize([0.5, 0.5, 0.5], [0.5, 0.5, 0.5]),
        ]
    )

    print(f"loading scene {img_save_dir}")
    rgb_dir = os.path.join(img_save_dir, "rgb")
    depth_dir = os.path.join(img_save_dir, "depth")
    pose_dir = os.path.join(img_save_dir, "pose")
    semantic_dir = os.path.join(img_save_dir, "semantic")
    obj2cls_path = os.path.join(img_save_dir, "obj2cls_dict.txt")

    rgb_list = sorted(os.listdir(rgb_dir), key=lambda x: int(
        x.split("_")[-1].split(".")[0]))
    depth_list = sorted(os.listdir(depth_dir), key=lambda x: int(
        x.split("_")[-1].split(".")[0]))
    pose_list = sorted(os.listdir(pose_dir), key=lambda x: int(
        x.split("_")[-1].split(".")[0]))
    #pose_list = sorted(os.listdir(pose_dir), key=lambda x: int(
     #   x.split("_")[-1].split(".")[0]))
    semantic_list = sorted(os.listdir(semantic_dir), key=lambda x:
int(
        x.split("_")[-1].split(".")[0]))

    rgb_list = [os.path.join(rgb_dir, x) for x in rgb_list]
    depth_list = [os.path.join(depth_dir, x) for x in depth_list]
```

```
    pose_list = [os.path.join(pose_dir, x) for x in pose_list]
    semantic_list = [os.path.join(semantic_dir, x) for x in
semantic_list]


    map_save_dir = os.path.join(img_save_dir, "map")
    os.makedirs(map_save_dir, exist_ok=True)
    color_top_down_save_path = os.path.join(map_save_dir,
f"color_top_down_{mask_version}.npy")
    gt_save_path = os.path.join(map_save_dir,
f"grid_{mask_version}_gt.npy")
    grid_save_path = os.path.join(map_save_dir,
f"grid_lseg_{mask_version}.npy")
    weight_save_path = os.path.join(map_save_dir,
f"weight_lseg_{mask_version}.npy")
    obstacles_save_path = os.path.join(map_save_dir, "obstacles.npy")

    obj2cls = load_obj2cls_dict(obj2cls_path)

    # initialize a grid with zero position at the center
    color_top_down_height = (camera_height + 1) * np.ones((gs, gs),
dtype=np.float32)
    color_top_down = np.zeros((gs, gs, 3), dtype=np.uint8)
    gt = np.zeros((gs, gs), dtype=np.int32)
    grid = np.zeros((gs, gs, clip_feat_dim), dtype=np.float32)
    obstacles = np.ones((gs, gs), dtype=np.uint8)
    weight = np.zeros((gs, gs), dtype=float)

    save_map(color_top_down_save_path, color_top_down)
    save_map(gt_save_path, gt)
    save_map(grid_save_path, grid)
    save_map(weight_save_path, weight)
    save_map(obstacles_save_path, obstacles)

    tf_list = []
    data_iter = zip(rgb_list, depth_list, semantic_list, pose_list)
    pbar = tqdm(total=len(rgb_list))
    # load all images and depths and poses
    for data_sample in data_iter:
        rgb_path, depth_path, semantic_path, pose_path = data_sample

        bgr = cv2.imread(rgb_path)
        rgb = cv2.cvtColor(bgr, cv2.COLOR_BGR2RGB)

        # read pose
        pos, rot = load_pose(pose_path)  # z backward, y upward, x to
the right
        rot_ro_cam = np.eye(3)
        rot_ro_cam[1, 1] = -1
        rot_ro_cam[2, 2] = -1
```

```
        rot = rot @ rot_ro_cam
        pos[1] += camera_height


        pose = np.eye(4)
        pose[:3, :3] = rot
        pose[:3, 3] = pos.reshape(-1)

        tf_list.append(pose)
        if len(tf_list) == 1:
            init_tf_inv = np.linalg.inv(tf_list[0])

        tf = init_tf_inv @ pose

        # read depth
        depth = load_depth(depth_path)

        # read semantic
        semantic = load_semantic(semantic_path)
        semantic = cvt_obj_id_2_cls_id(semantic, obj2cls)

        pix_feats = get_lseg_feat(model, rgb, labels, transform,
crop_size, base_size, norm_mean, norm_std)

        # transform all points to the global frame
        pc, mask = depth2pc(depth)
        shuffle_mask = np.arange(pc.shape[1])
        np.random.shuffle(shuffle_mask)
        shuffle_mask = shuffle_mask[::depth_sample_rate]
        mask = mask[shuffle_mask]
        pc = pc[:, shuffle_mask]
        pc = pc[:, mask]
        pc_global = transform_pc(pc, tf)

        rgb_cam_mat = get_sim_cam_mat(rgb.shape[0], rgb.shape[1])
        feat_cam_mat = get_sim_cam_mat(pix_feats.shape[2],
pix_feats.shape[3])

        # project all point cloud onto the ground
        for i, (p, p_local) in enumerate(zip(pc_global.T, pc.T)):
            x, y = pos2grid_id(gs, cs, p[0], p[2])

            # ignore points projected to outside of the map and points
that are 0.5 higher than the camera (could be from the ceiling)
            if x >= obstacles.shape[0] or y >= obstacles.shape[1] or \
                x < 0 or y < 0 or p_local[1] < -0.5:
                continue

            rgb_px, rgb_py, rgb_pz = project_point(rgb_cam_mat,
p_local)
```

```
            rgb_v = rgb[rgb_py, rgb_px, :]
            semantic_v = semantic[rgb_py, rgb_px]
            if semantic_v == 40:
                semantic_v = -1

            # when the projected location is already assigned a color
value before, overwrite if the current point has larger height
            if p_local[1] < color_top_down_height[y, x]:
                color_top_down[y, x] = rgb_v
                color_top_down_height[y, x] = p_local[1]
                gt[y, x] = semantic_v

            # average the visual embeddings if multiple points are
projected to the same grid cell
            px, py, pz = project_point(feat_cam_mat, p_local)
            if not (px < 0 or py < 0 or px >= pix_feats.shape[3] or py
>= pix_feats.shape[2]):
                feat = pix_feats[0, :, py, px]
                grid[y, x] = (grid[y, x] * weight[y, x] + feat) /
(weight[y, x] + 1)
                weight[y, x] += 1

            # build an obstacle map ignoring points on the floor (0
means occupied, 1 means free)
            if p_local[1] > camera_height:
                continue
            obstacles[y, x] = 0
        pbar.update(1)

    save_map(color_top_down_save_path, color_top_down)
    save_map(gt_save_path, gt)
    save_map(grid_save_path, grid)
    save_map(weight_save_path, weight)
    save_map(obstacles_save_path, obstacles)


def get_lseg_feat(model: LSegEncNet, image: np.array, labels,
transform, crop_size=480, \
                  base_size=520, norm_mean=[0.5, 0.5, 0.5],
norm_std=[0.5, 0.5, 0.5]):
    vis_image = image.copy()
    image = transform(image).unsqueeze(0).cuda()
    img = image[0].permute(1,2,0)
    img = img * 0.5 + 0.5

    batch, _, h, w = image.size()
    stride_rate = 2.0/3.0
    stride = int(crop_size * stride_rate)

    long_size = base_size
```

```
    if h > w:
        height = long_size
        width = int(1.0 * w * long_size / h + 0.5)
        short_size = width
    else:
        width = long_size
        height = int(1.0 * h * long_size / w + 0.5)
        short_size = height


    cur_img = resize_image(image, height, width, **{'mode':
'bilinear', 'align_corners': True})

    if long_size <= crop_size:
        pad_img = pad_image(cur_img, norm_mean,
                            norm_std, crop_size)
        print(pad_img.shape)
        with torch.no_grad():
            outputs, logits = model(pad_img, labels)
        outputs = crop_image(outputs, 0, height, 0, width)
    else:
        if short_size < crop_size:
            # pad if needed
            pad_img = pad_image(cur_img, norm_mean,
                                norm_std, crop_size)
        else:
            pad_img = cur_img
        _,_,ph,pw = pad_img.shape #.size()
        assert(ph >= height and pw >= width)
        h_grids = int(math.ceil(1.0 * (ph-crop_size)/stride)) + 1
        w_grids = int(math.ceil(1.0 * (pw-crop_size)/stride)) + 1
        with torch.cuda.device_of(image):
            with torch.no_grad():
                outputs = image.new().resize_(batch,
model.out_c,ph,pw).zero_().cuda()
                logits_outputs = image.new().resize_(batch,
len(labels),ph,pw).zero_().cuda()
            count_norm =
image.new().resize_(batch,1,ph,pw).zero_().cuda()
        # grid evaluation
        for idh in range(h_grids):
            for idw in range(w_grids):
                h0 = idh * stride
                w0 = idw * stride
                h1 = min(h0 + crop_size, ph)
                w1 = min(w0 + crop_size, pw)
                crop_img = crop_image(pad_img, h0, h1, w0, w1)
                # pad if needed
                pad_crop_img = pad_image(crop_img, norm_mean,
                                         norm_std, crop_size)
```

```
                with torch.no_grad():
                    output, logits = model(pad_crop_img, labels)
                cropped = crop_image(output, 0, h1-h0, 0, w1-w0)
                cropped_logits = crop_image(logits, 0, h1-h0, 0, w1-
w0)
                outputs[:,:,h0:h1,w0:w1] += cropped
                logits_outputs[:,:,h0:h1,w0:w1] += cropped_logits
                count_norm[:,:,h0:h1,w0:w1] += 1
        assert((count_norm==0).sum()==0)
        outputs = outputs / count_norm
        logits_outputs = logits_outputs / count_norm
        outputs = outputs[:,:,:height,:width]
        logits_outputs = logits_outputs[:,:,:height,:width]
    outputs = outputs.cpu()
    outputs = outputs.numpy() # B, D, H, W
    predicts = [torch.max(logit, 0)[1].cpu().numpy() for logit in
logits_outputs]
    pred = predicts[0]

    return outputs

__file__:  /content/vlmaps/examples/context.py
imported path: /content/vlmaps
```

Creating the Visual Language Map (VLMap)

This cell executes the full **VLMap construction pipeline** using the previously defined parameters and helper functions.

The function processes the RGB-D dataset frame by frame to:

- extract dense visual-language embeddings using LSeg,
- back-project depth pixels into 3D space,
- transform points into a global coordinate frame using pose information,
- and aggregate semantic embeddings into a top-down spatial grid.

During this process, multiple maps are generated and stored:

- a **color top-down map** for visualization,
- a **semantic grid** encoding object class information,
- a **visual-language embedding grid** representing the VLMap,
- a **weight map** tracking feature aggregation,
- and an **obstacle map** distinguishing free space from occupied regions.

The resulting VLMap serves as a persistent spatial representation that enables open-vocabulary landmark identification and language-conditioned navigation in subsequent stages.

```
create_lseg_map_batch(data_dir, camera_height=camera_height, cs=cs,
gs=gs, depth_sample_rate=depth_sample_rate)
```

```
cuda
Loading CLIP model...

100%|████████████████████████████████████████| 338M/338M [00:02<00:00,
166MiB/s]
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_auth.py
:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your
settings tab (https://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to
access public models or datasets.
  warnings.warn(

{"model_id":"e01c7fdda92a43988c16230a965eebeb","version_major":2,"vers
ion_minor":0}

<ipython-input-7-1e86ff46a78f>:46: FutureWarning: You are using
`torch.load` with `weights_only=False` (the current default value),
which uses the default pickle module implicitly. It is possible to
construct malicious pickle data which will execute arbitrary code
during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  pretrained_state_dict =
torch.load("lseg/checkpoints/demo_e200.ckpt")
The cache for model files in Transformers v4.22.0 has been updated.
Migrating your old cache. This is a one-time only operation. You can
interrupt this and resume the migration later on by calling
`transformers.utils.move_cache()`.

{"model_id":"cb09643de5b343be87afcc552eac7fc4","version_major":2,"vers
ion_minor":0}

loading scene /content/jh4fc5c5qoQ_1/
/content/jh4fc5c5qoQ_1/map/color_top_down_1.npy is saved.
/content/jh4fc5c5qoQ_1/map/grid_1_gt.npy is saved.
/content/jh4fc5c5qoQ_1/map/grid_lseg_1.npy is saved.
/content/jh4fc5c5qoQ_1/map/weight_lseg_1.npy is saved.
/content/jh4fc5c5qoQ_1/map/obstacles.npy is saved.
```

```
100%|██████████| 500/500 [09:51<00:00,  1.17s/it]

/content/jh4fc5c5qoQ_1/map/color_top_down_1.npy is saved.
/content/jh4fc5c5qoQ_1/map/grid_1_gt.npy is saved.
/content/jh4fc5c5qoQ_1/map/grid_lseg_1.npy is saved.
/content/jh4fc5c5qoQ_1/map/weight_lseg_1.npy is saved.
/content/jh4fc5c5qoQ_1/map/obstacles.npy is saved.

 100%|██████████| 500/500 [10:54<00:00,  1.31s/it]
```

Loading Utilities for VLMap Querying and Visualization

This cell imports the utilities required to **load, analyze, and visualize** the Visual Language Maps generated in the previous step.

The imported modules support:

- loading saved VLMap artifacts (semantic grids, embedding maps, obstacle maps),
- generating color palettes for semantic visualization,
- encoding natural language queries into the shared vision–language embedding space,
- and rendering results using matplotlib.

These utilities enable qualitative inspection of the generated maps and form the foundation for **language-based landmark querying** and navigation analysis in the subsequent cells.

```
%matplotlib inline
import os
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from utils.clip_mapping_utils import load_map, get_new_pallete,
get_new_mask_pallete
from utils.clip_utils import get_text_feats
from utils.mp3dcat import mp3dcat
import clip
```

Selecting and Loading the Generated VLMap

This cell configures the paths used to load the Visual Language Map (VLMap) artifacts for querying and visualization.

A flag is used to determine whether the pipeline should:

- load the **self-built VLMap** generated in this notebook, or
- fall back to an alternative precomputed map directory (if available).

Based on this selection, the corresponding file paths are defined for:

- the **color top-down map** used for visualization,
- the **visual-language embedding grid** used for semantic querying,

- and the **obstacle map** used to identify free and occupied space.

These paths are used in subsequent cells to load the VLMap and perform language-conditioned landmark identification and analysis.

```
use_self_built_map = True # @param {type: "boolean"}
map_save_dir = os.path.join(data_dir, "map_correct")
if use_self_built_map:
    map_save_dir = os.path.join(data_dir, "map")
os.makedirs(map_save_dir, exist_ok=True)

color_top_down_save_path = os.path.join(map_save_dir,
f"color_top_down_1.npy")
grid_save_path = os.path.join(map_save_dir, f"grid_lseg_1.npy")
obstacles_save_path = os.path.join(map_save_dir, "obstacles.npy")
```

Visualizing the Obstacle Map

This cell loads and visualizes the **obstacle map** generated during VLMap construction.

The obstacle map encodes navigability information in a top-down grid:

- **0** indicates occupied space (obstacles),
- **1** indicates free space.

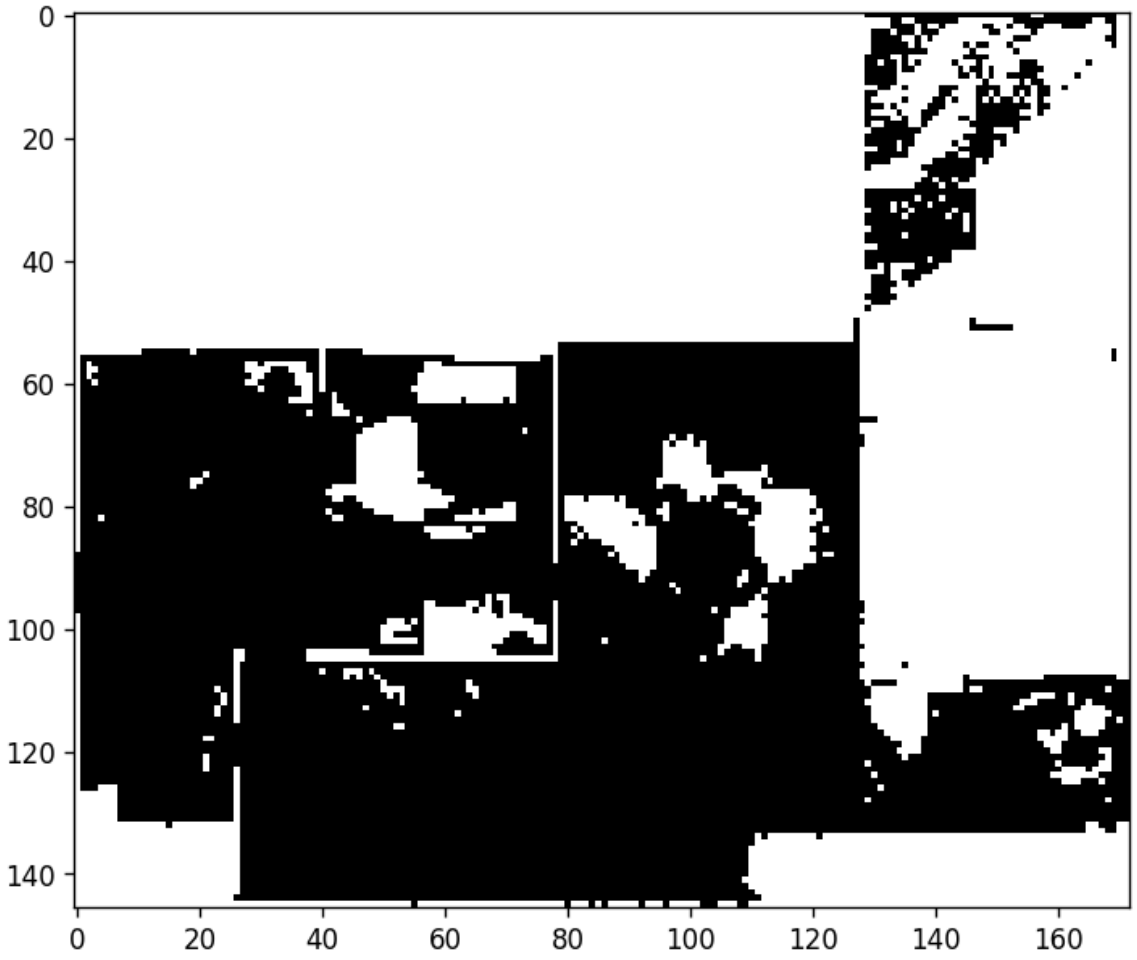
To focus on the relevant region of the environment, the map is cropped to the bounding box that contains all occupied cells. This removes unused areas of the grid and improves visual clarity.

The resulting visualization provides a clear view of:

- free-space structure,
- obstacle layout,
- and navigable regions that will be used during navigation planning.

```
obstacles = load_map(obstacles_save_path)
x_indices, y_indices = np.where(obstacles == 0)

xmin = np.min(x_indices)
xmax = np.max(x_indices)
ymin = np.min(y_indices)
ymax = np.max(y_indices)

print(np.unique(obstacles))
obstacles_pil = Image.fromarray(obstacles[xmin:xmax+1, ymin:ymax+1])
plt.figure(figsize=(8, 6), dpi=120)
plt.imshow(obstacles_pil, cmap='gray')
plt.show()
```

```
[0 1]
```



It can be observed that the obstacles are white and the free space is black. You can also observe noisy black points lying on the ground. These points are caused by the slope or different floor heights in the scene.

Visualizing the Top-Down Color Map

This cell visualizes the **top-down color map** generated during VLMap construction.

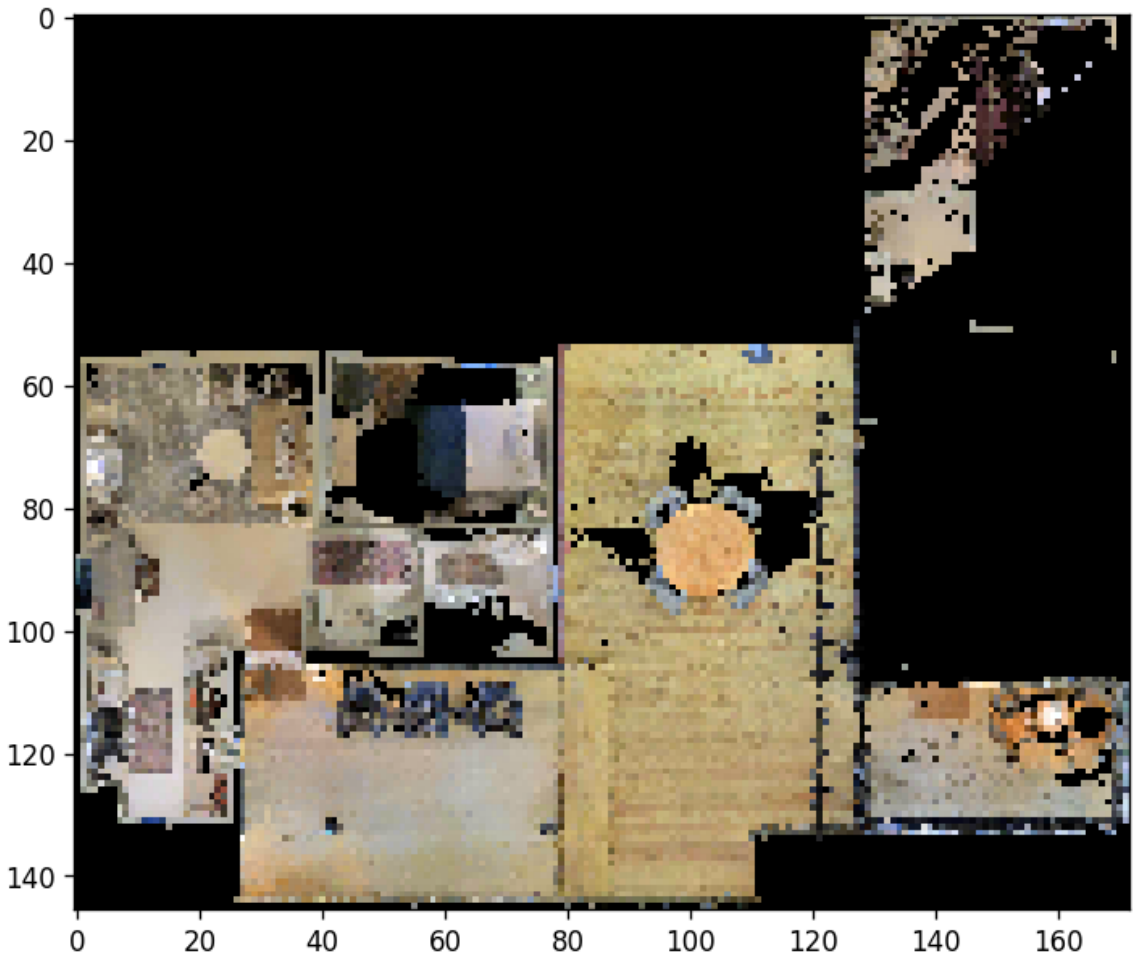
The color map is created by projecting RGB observations from multiple viewpoints onto a 2D ground-plane grid. For each grid cell, the color corresponding to the closest visible surface is retained, producing a compact visual summary of the environment layout.

The map is cropped using the same spatial bounds as the obstacle map to ensure consistent alignment across all visualizations.

This representation provides:

- an intuitive overview of the environment structure,
- visual context for obstacle and semantic maps,
- and a reference for interpreting language-grounded landmarks in subsequent steps.

```
color_top_down = load_map(color_top_down_save_path)
color_top_down = color_top_down[xmin:xmax+1, ymin:ymax+1]
color_top_down_pil = Image.fromarray(color_top_down)
plt.figure(figsize=(8, 6), dpi=120)
plt.imshow(color_top_down_pil)
plt.show()
```



The color map is created by projecting the color-augmented point cloud to the floor.

Loading CLIP and Preparing the VLMap for Language Querying

This cell initializes the **CLIP vision–language model** and loads the visual-language embedding grid generated during VLMap construction.

CLIP provides a shared embedding space for images and text, enabling semantic comparison between natural language queries and spatial map representations. The model is loaded in evaluation mode and moved to the appropriate compute device.

The saved VLMap embedding grid is then loaded and spatially cropped using the same bounds applied to the obstacle and color maps. This ensures consistent alignment across all map representations.

At the end of this step, the notebook is ready to:

- encode natural language queries,
- compute semantic similarity against the VLMap,
- and identify language-grounded landmarks within the environment.

```
%%capture
device = "cuda" if torch.cuda.is_available() else "cpu"
clip_version = "ViT-B/32"
clip_feat_dim = {'RN50': 1024, 'RN101': 512, 'RN50x4': 640, 'RN50x16':
768,
                 'RN50x64': 1024, 'ViT-B/32': 512, 'ViT-B/16': 512,
'ViT-L/14': 768}[clip_version]
clip_model, preprocess = clip.load(clip_version)  #
clip.available_models()
clip_model.to(device).eval()

grid = load_map(grid_save_path)
grid = grid[xmin:xmax+1, ymin:ymax+1]
```

This uses the text form of all semantic categories provided by the Matterport3D dataset as prompts to the VLMap and localize all categories in the map. We filter out "floor" category.

Semantic Map Generation via Language Querying

This cell performs **open-vocabulary semantic labeling** over the constructed VLMap using natural language queries.

The process consists of the following steps:

- **Text Embedding Generation**
  A predefined set of semantic categories is encoded into the CLIP embedding space. These text embeddings represent language concepts that will be grounded onto the map.

- **Semantic Similarity Computation**
  Each grid cell's visual-language embedding is compared against all text embeddings using dot-product similarity. This produces a semantic score for every cell–label pair.

- **Label Assignment**
  Each grid cell is assigned the label with the highest semantic similarity, resulting in a dense semantic segmentation of the environment.

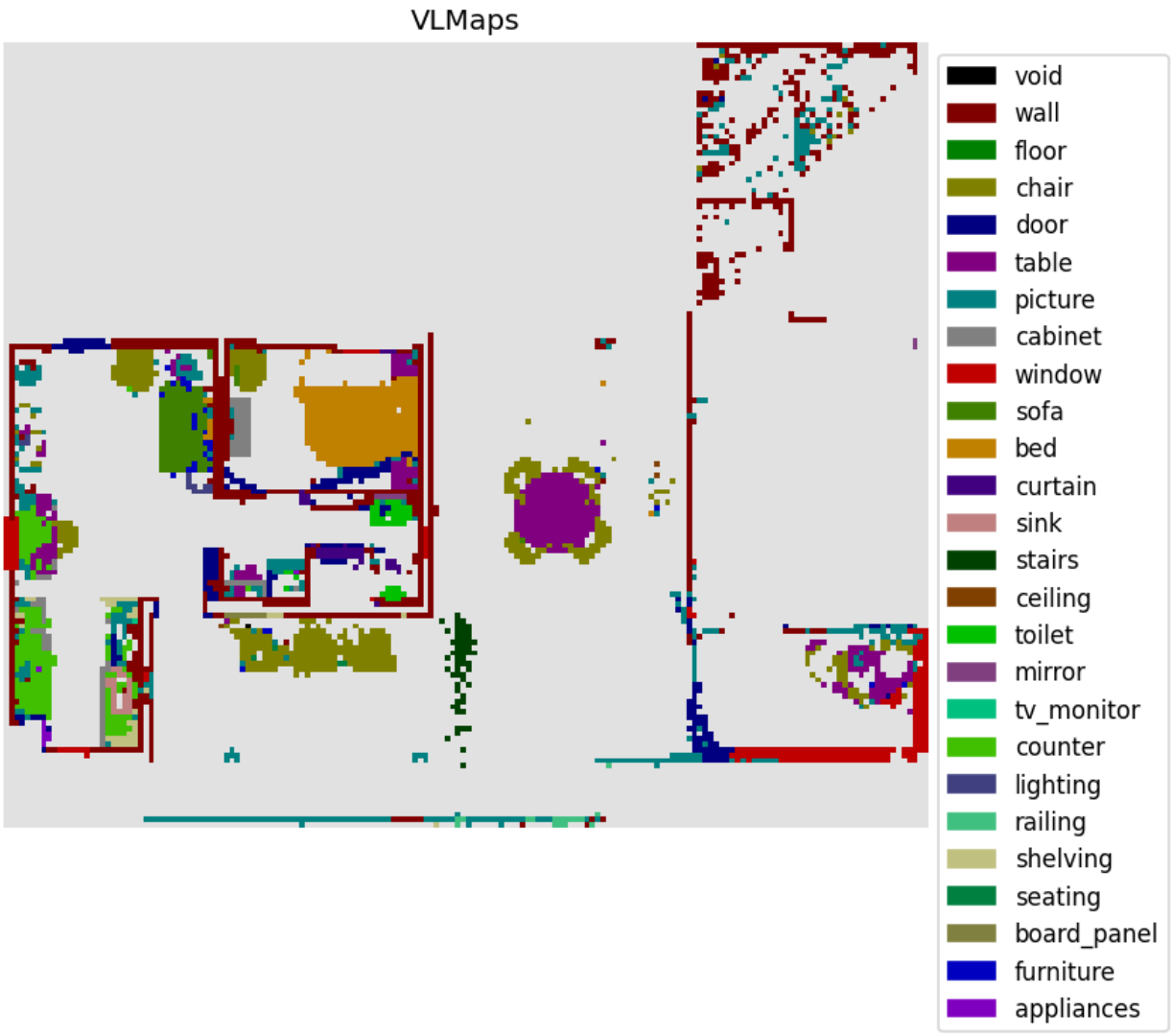
- **Masking Non-Navigable Regions**
  Cells corresponding to obstacles or floor regions are masked to improve interpretability and ensure that only valid semantic regions are visualized.

- **Visualization**
  The resulting semantic map is rendered using a color palette, with a legend indicating the predicted semantic categories for each region.

This visualization represents the final **Visual Language Map**, demonstrating how natural language concepts are grounded into a persistent spatial representation.

```
no_map_mask = obstacles[xmin:xmax+1, ymin:ymax+1] > 0
obstacles_rgb = np.repeat(obstacles[xmin:xmax+1, ymin:ymax+1, None],
3, axis=2)
print(no_map_mask.shape)

lang = mp3dcat
text_feats = get_text_feats(lang, clip_model, clip_feat_dim)

map_feats = grid.reshape((-1, grid.shape[-1]))
scores_list = map_feats @ text_feats.T

predicts = np.argmax(scores_list, axis=1)
predicts = predicts.reshape((xmax - xmin + 1, ymax - ymin + 1))
floor_mask = predicts == 2

new_pallete = get_new_pallete(len(lang))
mask, patches = get_new_mask_pallete(predicts, new_pallete,
out_label_flag=True, labels=lang)
seg = mask.convert("RGBA")
seg = np.array(seg)
seg[no_map_mask] = [225, 225, 225, 255]
seg[floor_mask] = [225, 225, 225, 255]
seg = Image.fromarray(seg)
plt.figure(figsize=(10, 6), dpi=120)
plt.legend(handles=patches, loc='upper left', bbox_to_anchor=(1., 1),
prop={'size': 10})
plt.axis('off')
plt.title("VLMaps")
plt.imshow(seg)
plt.show()
```

```
(146, 172)
```



Open-Vocabulary Landmark Querying with Custom Language Prompts

This cell demonstrates **interactive, open-vocabulary querying** of the Visual Language Map using a user-defined natural language prompt.

A list of object or region names is provided as a comma-separated string. Each term is encoded into the shared CLIP embedding space and compared against the visual-language embeddings stored in the VLMap.

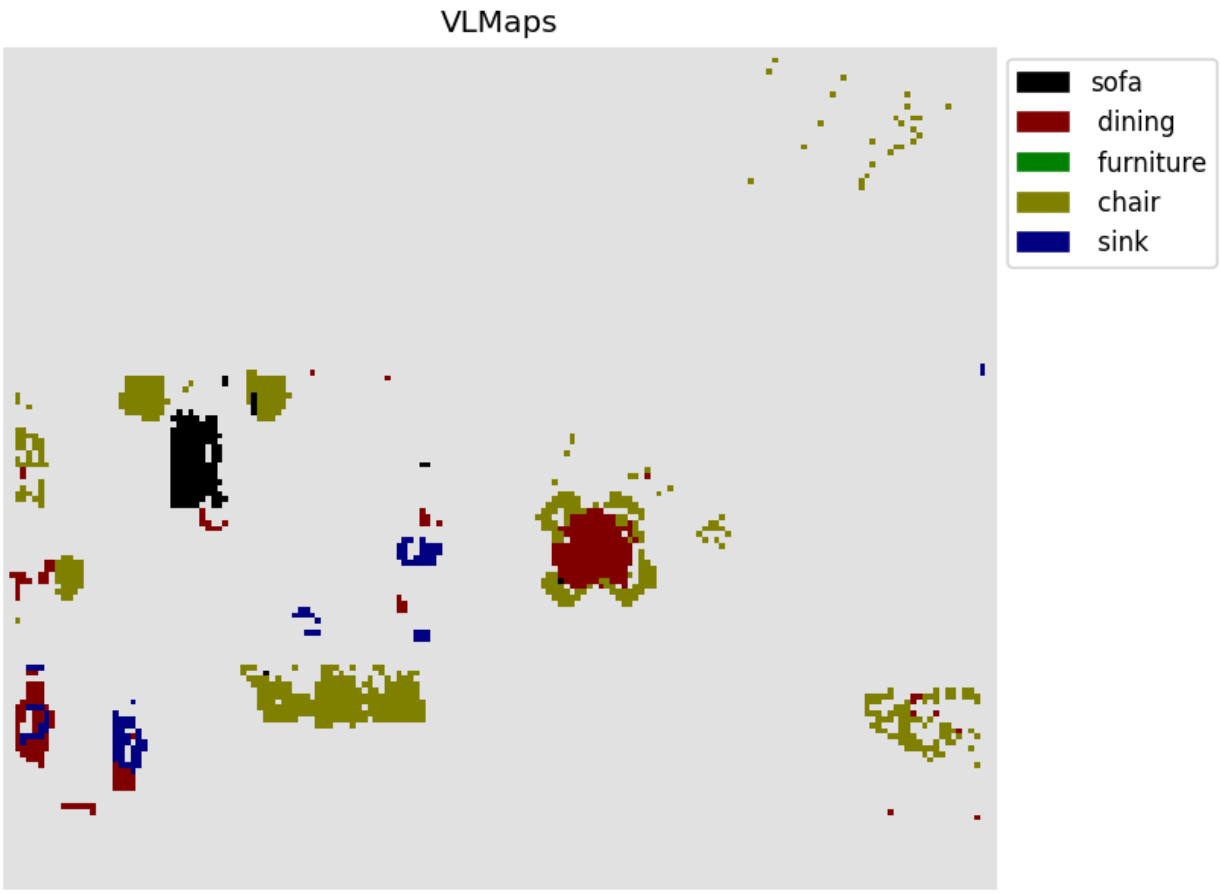
For each grid cell:

- semantic similarity scores are computed against all query terms,
- the highest-scoring label is selected,
- and the resulting semantic assignment is visualized as a colored top-down map.

Non-navigable regions such as obstacles and floor areas are masked to improve interpretability.

This step highlights a key capability of VLMaps: **the ability to localize arbitrary, previously unseen language concepts directly within a spatial map—without retraining or predefined object categories.**

```
# @markdown Input the prompt as a string of object names separated by
","
lang = "sofa, dining, furniture, chair, sink" # @param {type:
"string"}
lang = lang.split(",")
text_feats = get_text_feats(lang, clip_model, clip_feat_dim)

map_feats = grid.reshape((-1, grid.shape[-1]))
scores_list = map_feats @ text_feats.T

predicts = np.argmax(scores_list, axis=1)
predicts = predicts.reshape((xmax - xmin + 1, ymax - ymin + 1))
floor_mask = predicts == 2

new_pallete = get_new_pallete(len(lang))
mask, patches = get_new_mask_pallete(predicts, new_pallete,
out_label_flag=True, labels=lang)
seg = mask.convert("RGBA")
seg = np.array(seg)
seg[no_map_mask] = [225, 225, 225, 255]
seg[floor_mask] = [225, 225, 225, 255]
seg = Image.fromarray(seg)
plt.figure(figsize=(10, 6), dpi=120)
plt.legend(handles=patches, loc='upper left', bbox_to_anchor=(1., 1),
prop={'size': 10})
plt.axis('off')
plt.title("VLMaps")
plt.imshow(seg)
plt.show()
```



Conclusion and Discussion

This notebook presented a complete reimplementation of **Visual Language Maps (VLMaps)** for language-conditioned robot navigation, demonstrating how visual–language embeddings can be fused with spatial representations to enable open-vocabulary semantic reasoning.

Through the construction of a VLMap from RGB-D observations, the system successfully:

- integrated geometric structure with semantic understanding,
- produced persistent top-down representations of indoor environments,
- and enabled landmark identification using free-form natural language queries.

The results show that VLMaps allow a robot to localize and reason about objects such as chairs, sofas, tables, and other household landmarks without relying on predefined object categories or task-specific retraining. By operating in a shared vision–language embedding space, the framework supports **zero-shot generalization**, allowing new concepts to be introduced at inference time purely through language.

In addition, the generated obstacle maps and top-down color maps demonstrate how VLMaps can support navigation planning by distinguishing free space from occupied regions while retaining semantic context. This unified representation addresses a key limitation of traditional SLAM-based systems, which lack the ability to interpret semantic relationships and natural language instructions.

Overall, the findings confirm that fusing visual-language features with spatial maps provides a robust foundation for intuitive and flexible robot navigation. While performance may be affected by noise and clutter in complex environments, the approach highlights strong potential for extension to multi-robot systems, different robot embodiments, and more advanced language-driven planning pipelines.

This work demonstrates the effectiveness of VLMaps as a scalable framework for bridging semantic understanding and spatial reasoning, and reinforces their applicability as a core building block for future research in language-guided robotic navigation.

Credits and Contributions

This project was completed as a collaborative effort. Contributions were divided as follows:

Aman Chandak

Ayushman Mishra

Both contributors collaborated on validating results and ensuring alignment with the original *Visual Language Maps for Robot Navigation* paper.

Repository Link: **https://github.com/aymisxx/vlmaps-reimplementation**