

CARLA Leave-Over Documentation

Hunter White

April 2024

1 Overview

This document is intended to serve as a resource for getting started with the "open-source driving simulator for autonomous driving research" known as Car Learning to Act CARLA. It includes a description of CARLA, the steps taken to install and configure CARLA on both lab computers, and the scripts, terminal commands, and workflow for interacting with CARLA that may be useful. A significant effort is made to include solutions to errors that may be encountered during the installation and initialization processes, as well as links to Stack-Overflow/Linux Forums/CARLA Documentation pages that proved useful in the troubleshooting and error correction processes.

2 Environment setup and package installation

2.1 Package and Environment Managers

Anaconda is used as the package and environment manager. Anaconda Navigator vastly simplifies the package and environment management problem by providing a relatively simple, if a little bloated, graphical user interface for managing virtual environments, with shortcuts to launch common programs with the conda environment activated. This proved especially useful when initially exploring the best (read: easiest to integrate with existing machine learning research on GitHub) version of python and CARLA to work with. Most packages can be installed via conda or pip, although some require one or the other. A short comparison between conda and pip is given on the conda blog [Special care should be taken to install all possible packages with one installer and then the other if necessary, as conda does not communicate with pip and vice-versa. A getting started guide for conda is available here.](#) When installing packages with conda, it will be useful to add the conda-forge channel. A guide on how to do that is provided here. Note: Occasionally, updating conda before installing all desired packages will result in a deadlocked environment. Unfortunately, the only way to fix the environment is to delete it and reinstall all desired packages in a new environment. Generally, I never updated conda unless I was about to

create a new environment or start a new project.

2.2 Required Packages

1. (Option 1: Individually) All required packages can be installed by navigating to each

```
requirements.txt
```

within the GitHub Repository and running the command

```
pip install -r requirements.txt
```

or

```
conda install --yes --file requirements.txt
```

in the command prompt with the desired conda environment activated.

2. (Option 2: Grouped) A single

```
requirements.txt
```

file is provided for easier installation with pip. This is installed the same way as described in Option 1. Alternatively, the conda equivalent

```
<environment_name>.yaml
```

file is also provided. To install the

```
<environment_name>.yaml
```

file, use the command

```
conda env create -f <environment-name>.yaml
```

in the command prompt or terminal.

3. (Note on Poetry) Poetry is included as an alternate package manager, and was used by Idrees Razak when creating the PPO agent and variational autoencoder that I used as a foundation. Poetry's main function in the repository is to download the legacy versions of PyTorch and CUDA that were used to train the PPO agent and VAE. To install these with Poetry, navigate to the Poetry folder using the command prompt or powershell, then run the command

```
poetry update
```

. As a disclaimer, I don't think this step is necessary if you would rather install the correct PyTorch and CUDA packages using pip or conda. The legacy versions can be found on the PyTorch website

3 How to Install CARLA