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 Creating the Simulation Environment

2.1 Environment setup and package installation

2.1.1 Package and Environment Managers

Definitely Review For Consistency
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.

2.1.2   Required Packages

Be sure to update pip and conda before creating a new environment and downloading packages for CARLA or the provided code.
To run the provided code, the following packages and python version are/is necessary:

Package	Version	requirements filepath
CARLA	0.9.8	N/A
Python	3.7.*	N/A
numpy	1.21.1	(your-carla-filespath)/carla/PythonAPI/examples
pygame	2.1.2	(your-carla-filespath)/carla/PythonAPI/examples
matplotlib	*	(your-carla-filespath)/carla/PythonAPI/examples
open3d	*	(your-carla-filespath)/carla/PythonAPI/examples
pillow	9.4.0	(your-carla-filespath)/carla/PythonAPI/examples
future	0.18.3	(your-carla-filespath)/carla/PythonAPI/examples
networkx	*	(your-carla-filespath)/carla/PythonAPI/carla
distro	*	(your-carla-filespath)/carla/PythonAPI/carla
Shapely	1.7.*	(your-carla-filespath)/carla/PythonAPI/carla
psutil	*	(your-carla-filespath)/carla/PythonAPI/util
py-cpuinfo	*	(your-carla-filespath)/carla/PythonAPI/util
python-tr	*	(your-carla-filespath)/carla/PythonAPI/util
poetry	1.3.2	(your-repository-path)/requirements.txt

Table 1: Required packages for CARLA, CARLA examples, and the provided code

To install all of the above packages, you have a few options:

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>.yml
   ```

   file is also provided. To install the

   ```
   <environment_name>.yml
   ```

   file, use the command

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

   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 Download and Install CARLA

NOTE: CARLA requires a dedicated GPU for with at least 6GB of VRAM. This excludes most consumer laptops. More VRAM does not necessarily make CARLA run faster, but it does generally make CARLA more stable when adding large numbers of actors. CARLA also recommends having an addition GPU for any machine learning, although this is not necessary if your GPU has the capacity. (i.e. 12 GB of VRAM and from a recent generation of Vulkan compatible GPUs) CARLA is fairly easy to install, and the process is nearly identical for Windows and Linux (Ubuntu). Because we do not need the additional functionality gained by building CARLA from source, we can follow the quick getting started guide. **NOTE: There are different versions of documentation**

that correspond to each released version of CARLA. Make sure you use the documentation version that corresponds to the CARLA version you intend to use. Be sure to make sure that pip and/or conda are up to date before creating the environment.

4   Verifying CARLA installation

CARLA provides a number of examples that are useful for checking that CARLA and its required packages were installed correctly. These provided examples are also useful for understanding the CARLA Python API, and give some ideas on how to structure code that will interact with CARLA.