

|| POLY-VERIFICATION USER GUIDE

This document contains the information on how to run the simulation through the Poly-Verification Suite. Once the installation and setup is completed you are ready to use the PolyVerif framework.

If you have not setup and installed the pre-requisites, then for reference please follow the **Setup_And_Installation** document.

This framework uses predefined metrics for the validation of the stacks which will be calculated after running the test cases. On the basis of the metrics threshold the success/failure of the stack is validated. Any user can set their own values by modifying the config.ini file in **adehome/Poly_Suite/config.ini**.

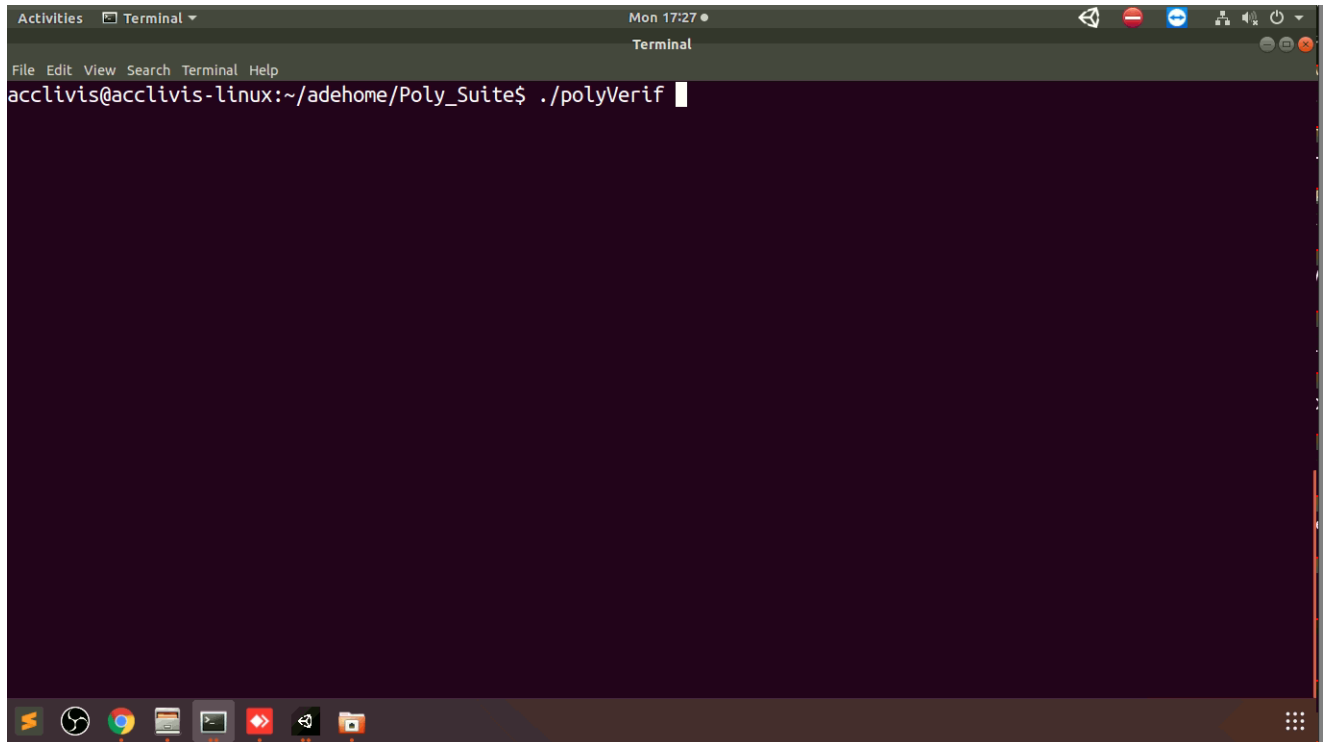
Below is the config.ini file parameters-

```
[autonomous_stack_config]
detection_max_threshold=40      #MAX value for detection validation
detection_min_threshold=30     #MIN value for detection validation
control_collion_count=0        #Collision count
localize_max_threshold=5       #MAX value for localization validation
localize_min_threshold=1       #MAX value for localization validation
planner_goalpose_max_deviation=5 #MAX value of goal position deviation
planner_goalpose_min_deviation=2 #MIN value of goal position deviation
```

Follow the steps given below-

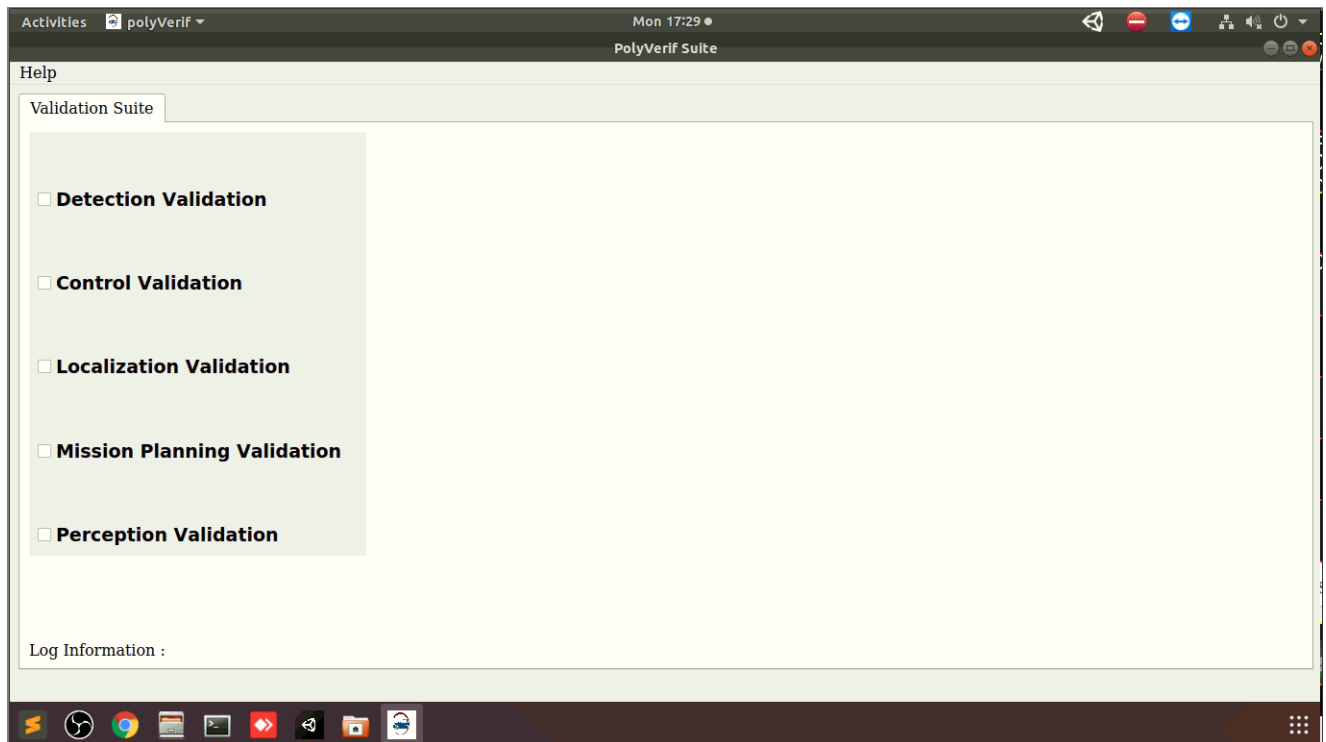
1. Go to the Poly_Suite directory and run

\$./polyVerif

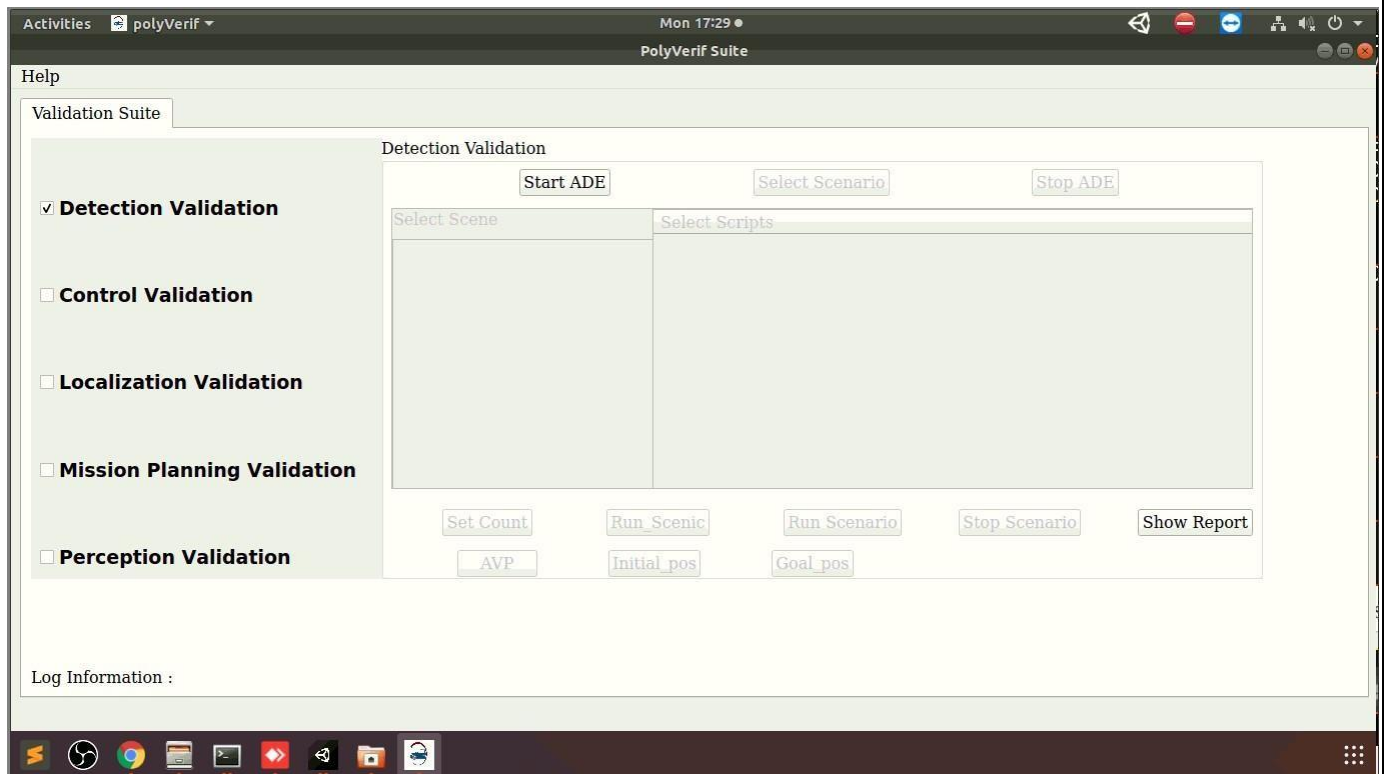


2. It will start the polyVerif framework
 - a. As of now four validations are working-
 - Detection Validation
 - Control Validation
 - Localization Validation
 - Mission Planning Validation

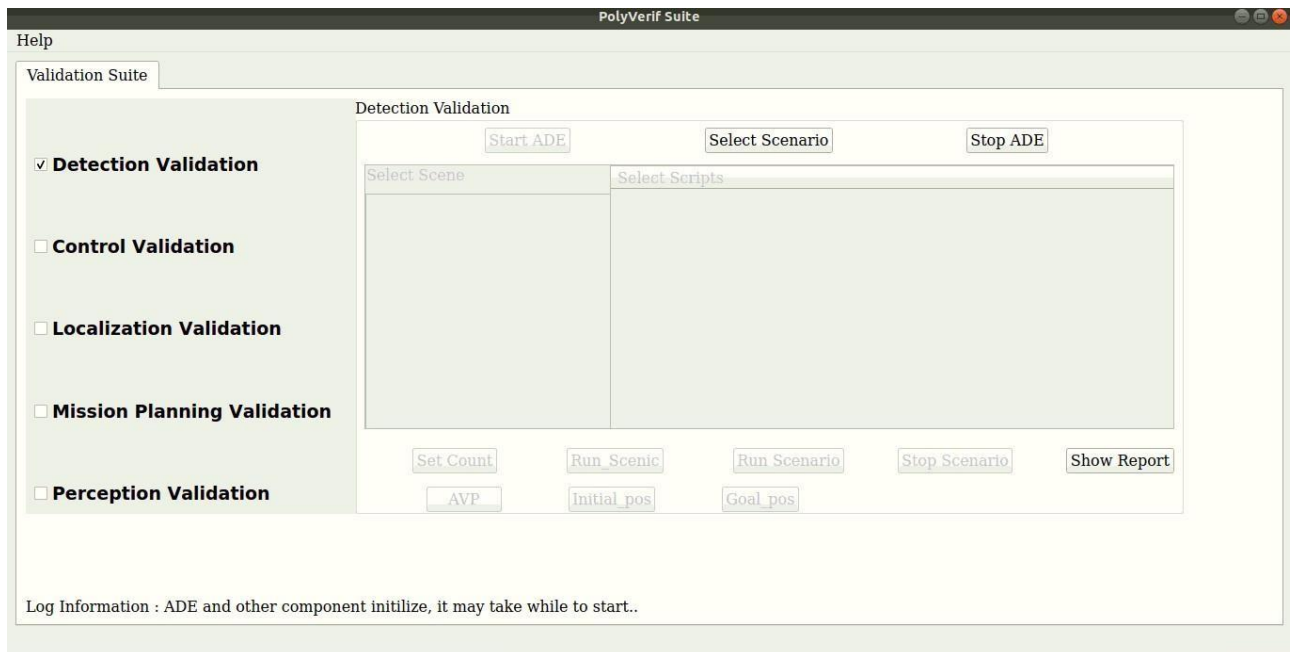
Click any of them (Screenshot is provided below for reference)-



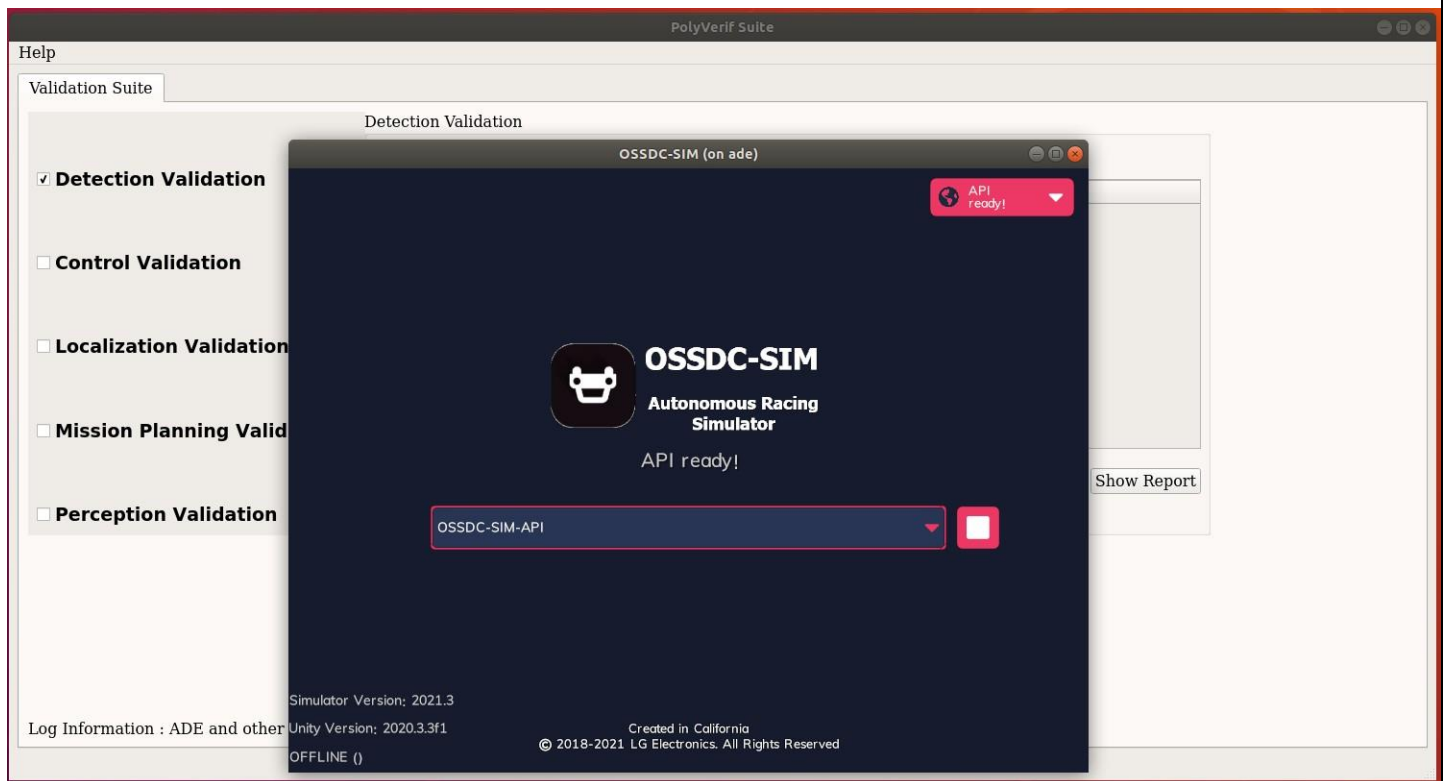
3. Detection Validation is for detecting the vehicle from perception stack.
 - a. Click on the Start ADE button, it will start all the required modules for example-
 - AutowareAuto
 - Perception Stack
 - OSSDC simulator
 - Rviz
 - Ros2-igsvl-bridge



4. It will take some time to start the ADE docker and other components please wait for a while.

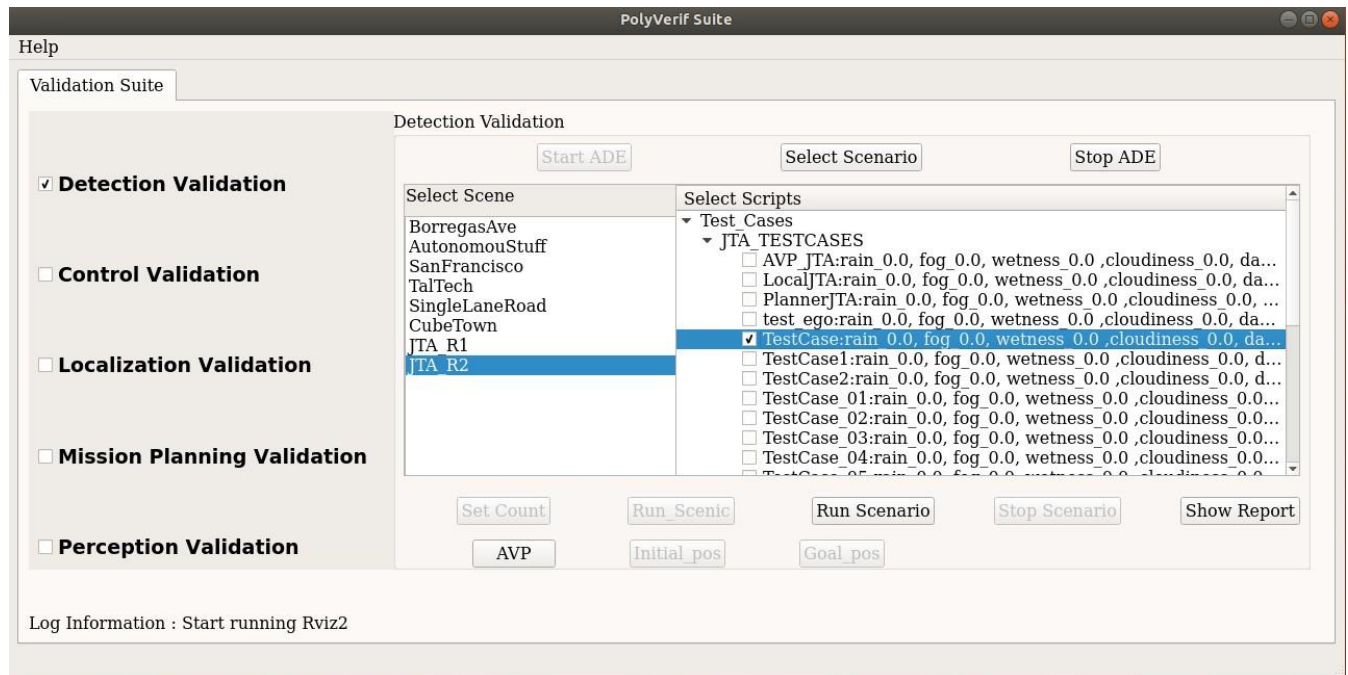


5. Now the simulator is in the **API_Mode** as shown.



6. Now click on the **Select Scenario** button

- Select **JTA_R2** map from the **Select Scene**
- Select any scenario from the **Select Scripts** list
- The selected script in the below screenshot is written in python which will add an ego vehicle in the JTA_R2 map which will be controlled by the AutowareAuto.



7. Once you select any script from the list it enables the respective button to run the script as there are two ways to run the script -

- Using Scenic
- Using PythonAPI

Since we have selected the python script , the **Run Scenario** button get enabled.

Note: After click on Run Scenario button if JTA map file is not in OSSDC directory you will get below error.

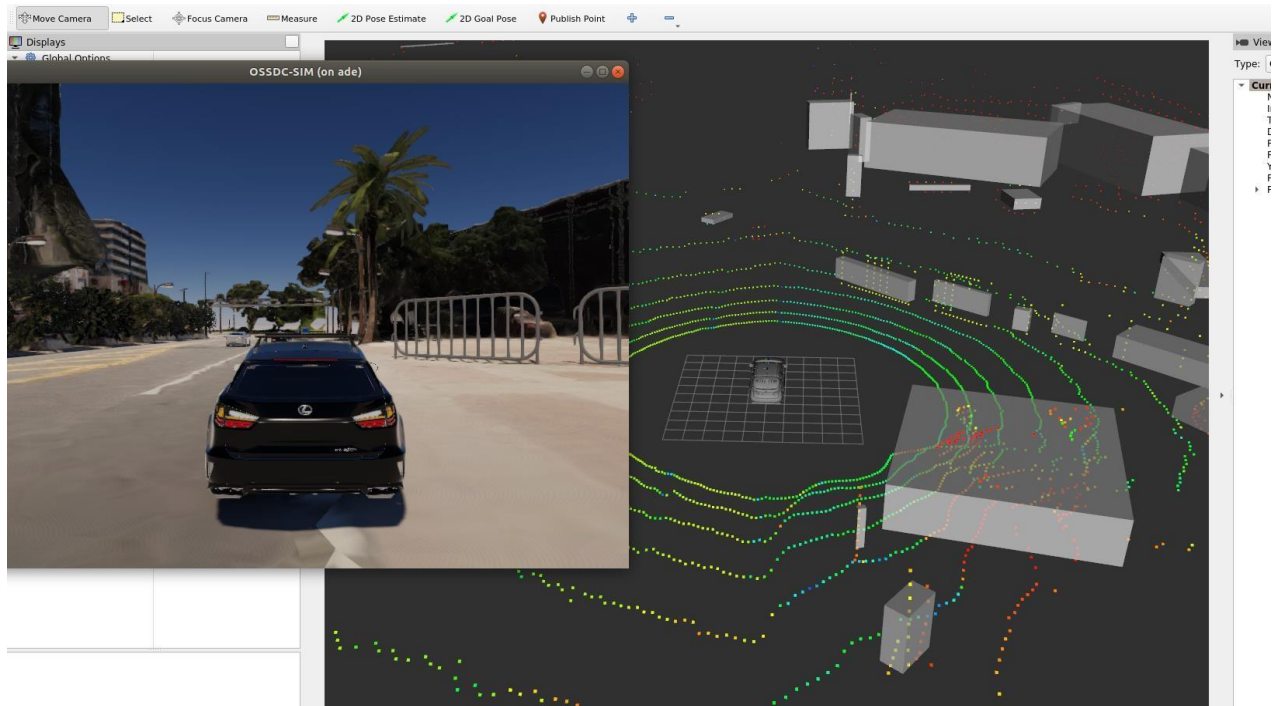
```
raise Exception(data["error"])
Exception: Download incomplete, received 0 of 0
ERROR: Command return non-zero exit code (see above): 1
```

For that need to download JTA map file and put it into below path:

JTA map download link - [Click Here..!](#)

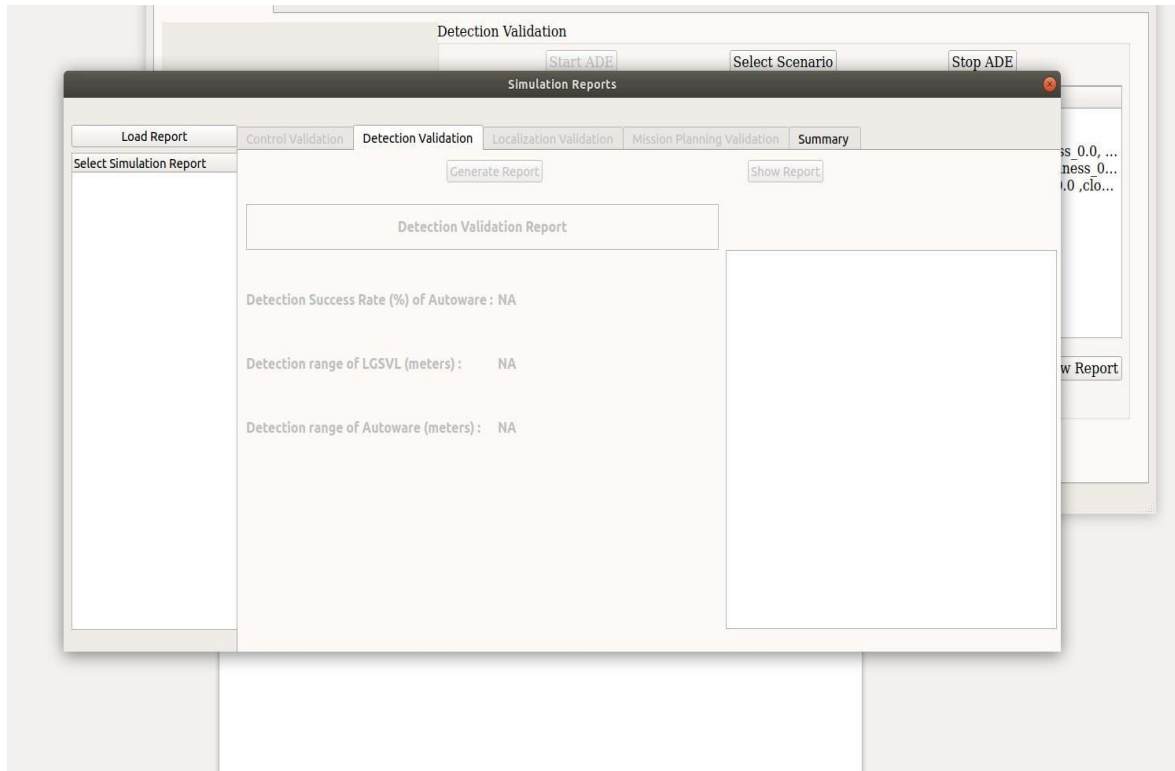
Path - /home/adehome/OSSDC-SIM-v1_1-Linux/2021.3-
ossdc/Environments/OSSDC_environment_JTA_R2

Note: After above steps stop ADE and again start it. After clicking Run Scenario wait for some time until vehicle locate on road because at first time simulator will download all sensors and vehicle assets.

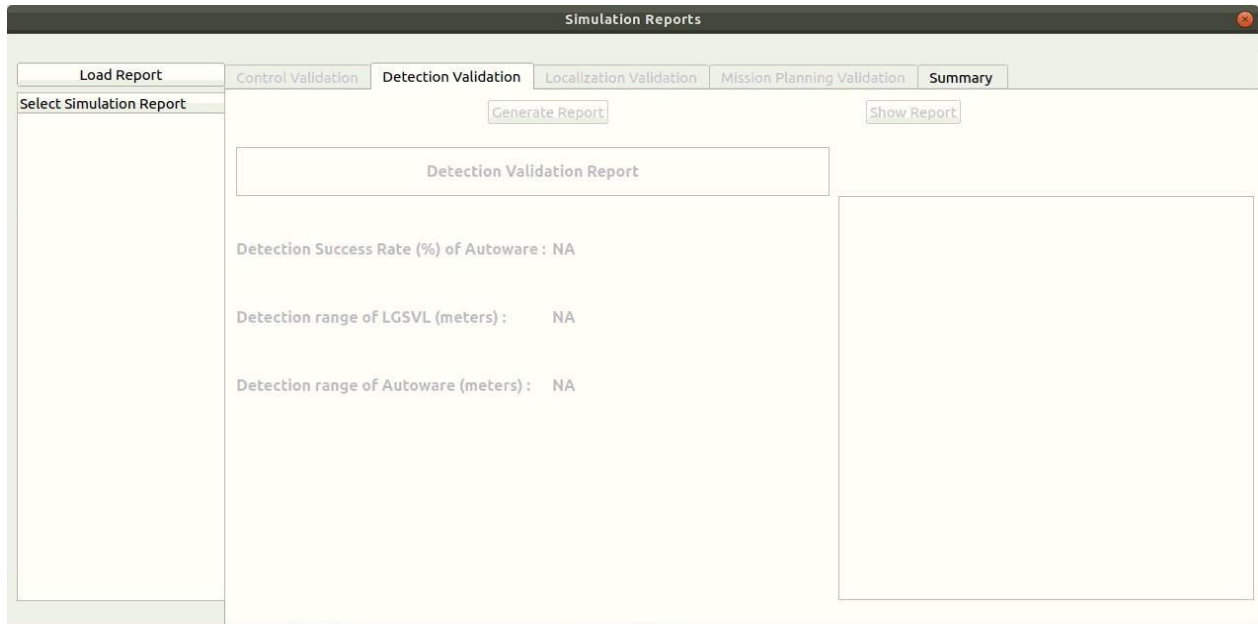


While running scenic script you can also set the number of random iterations for simulation through scenic using the **Set Count** button.

8. Once the simulation is completed or if you want to stop the simulation then Click on the **Stop Scenario** button and click on the **Show Report** button. It will redirect to another form where you can see the respective simulation reports.

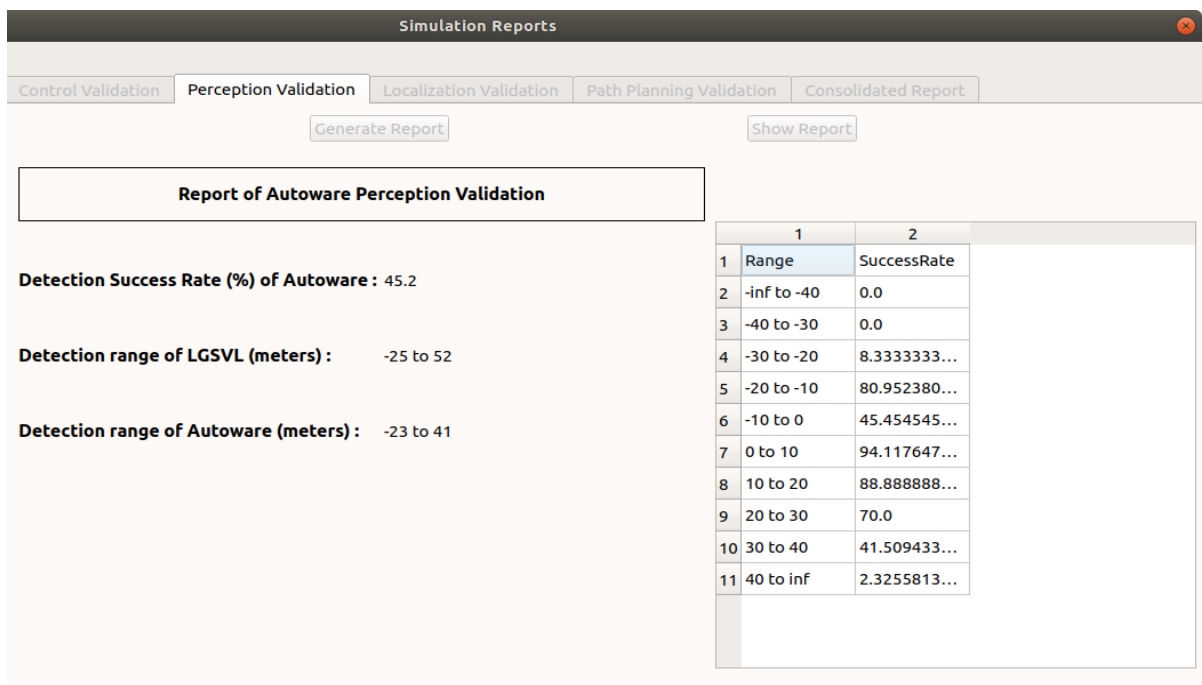


9. This form contains reports of the respective simulation. As we ran the simulation for Perception Validation so it will generate and show the perception report.
10. Click on the **Load Report** button, it will list all the simulation reports with date and time.



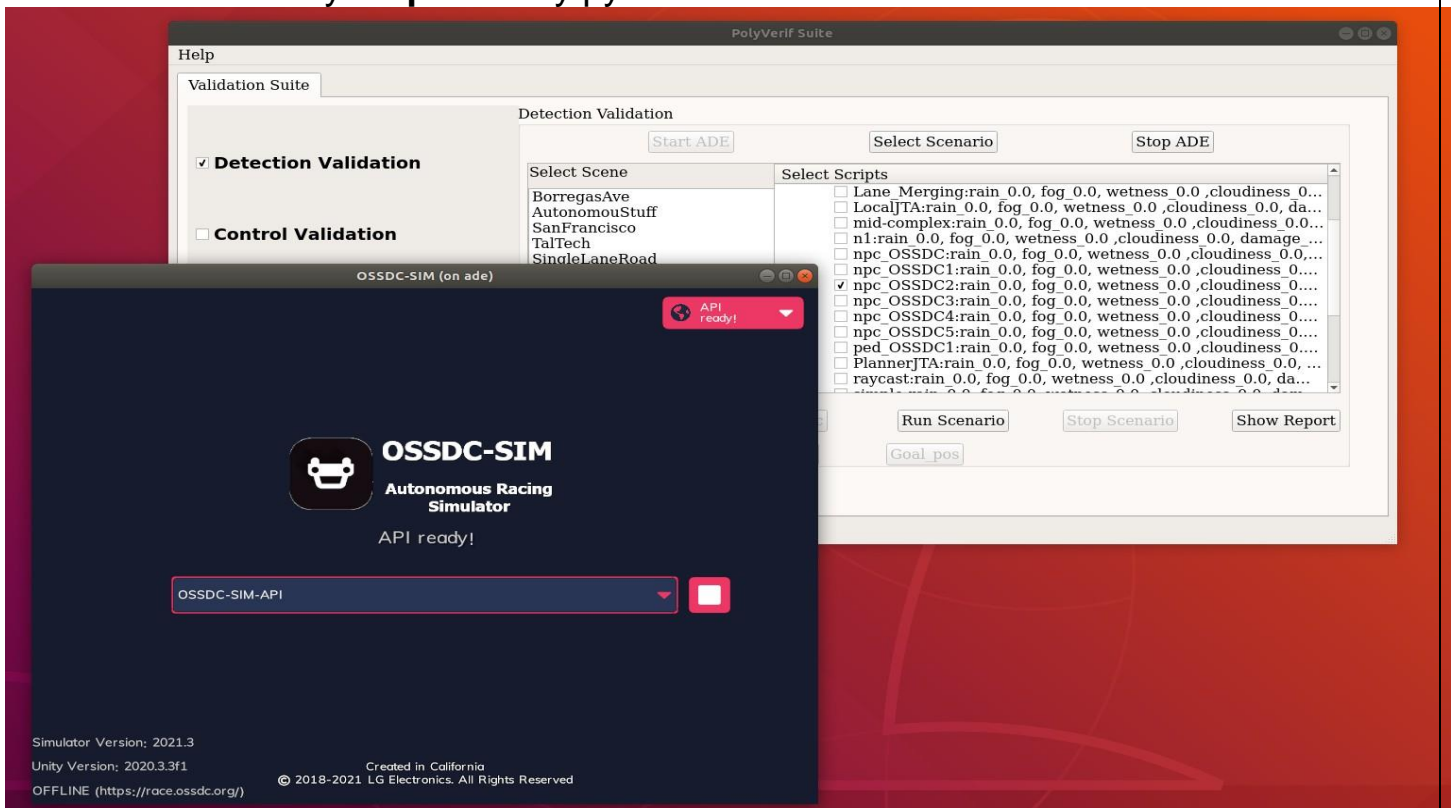
11. The last completed simulation report is set at the end of the list.

Select and click on the **Generate Report** button. It will take some time to generate the report. Once report is generated it will enable the **Show Report** button. By clicking on the **Show Report** button you will see few respective parameters on the UI.

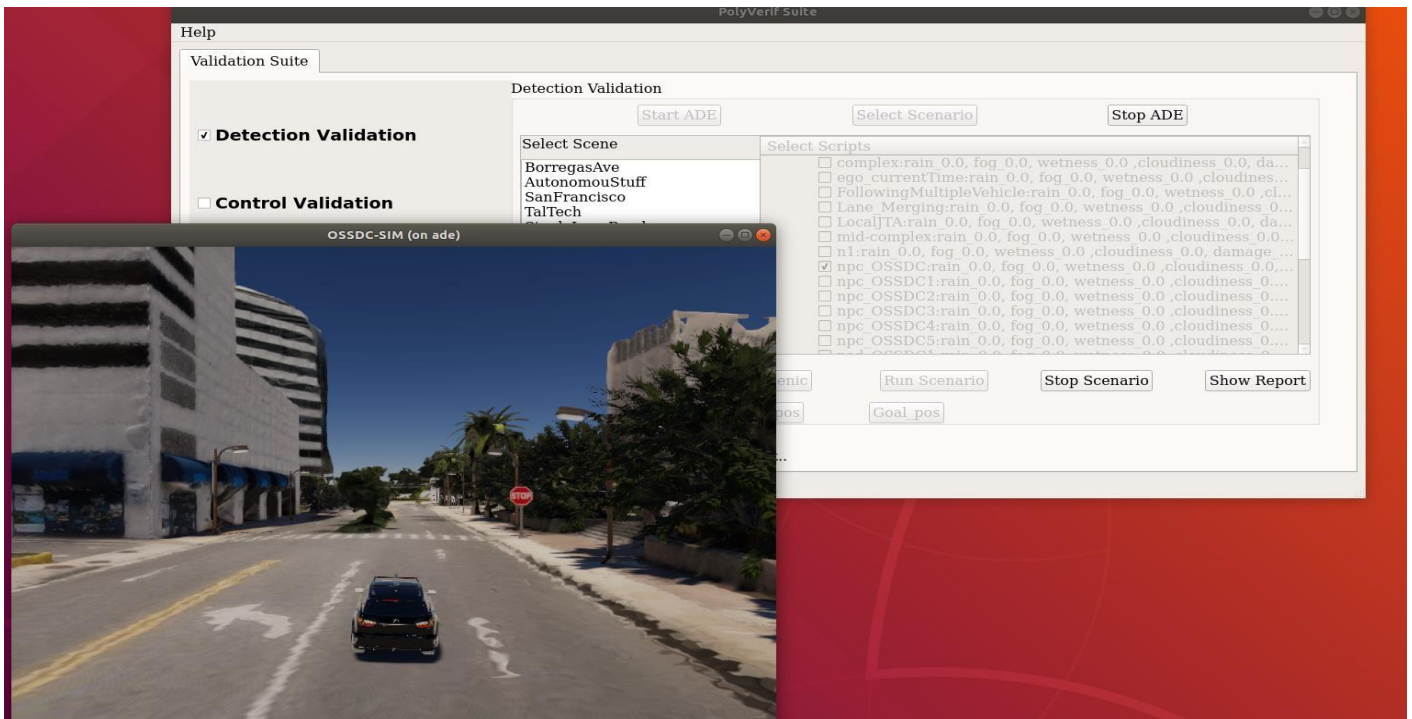


Steps to execute Python scenario –

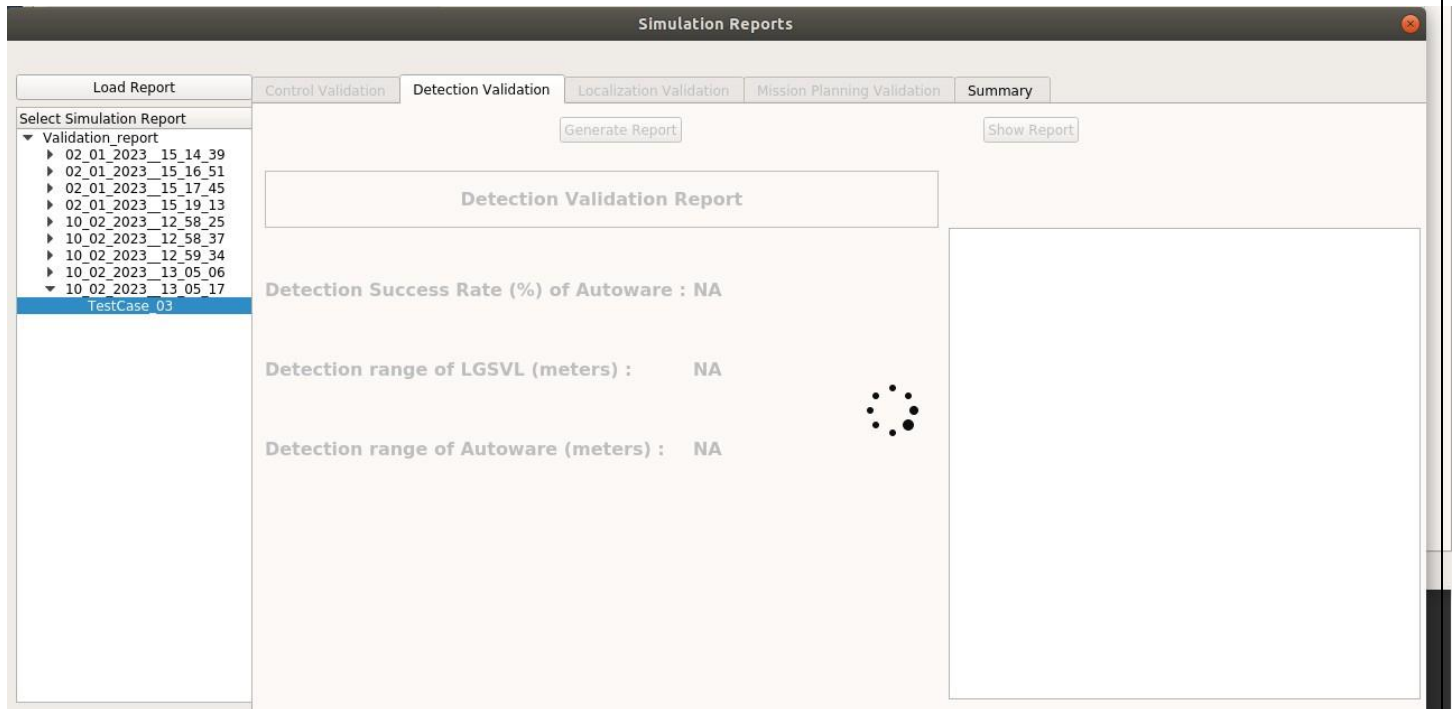
1. Select any **map** and any python scenario from the list.



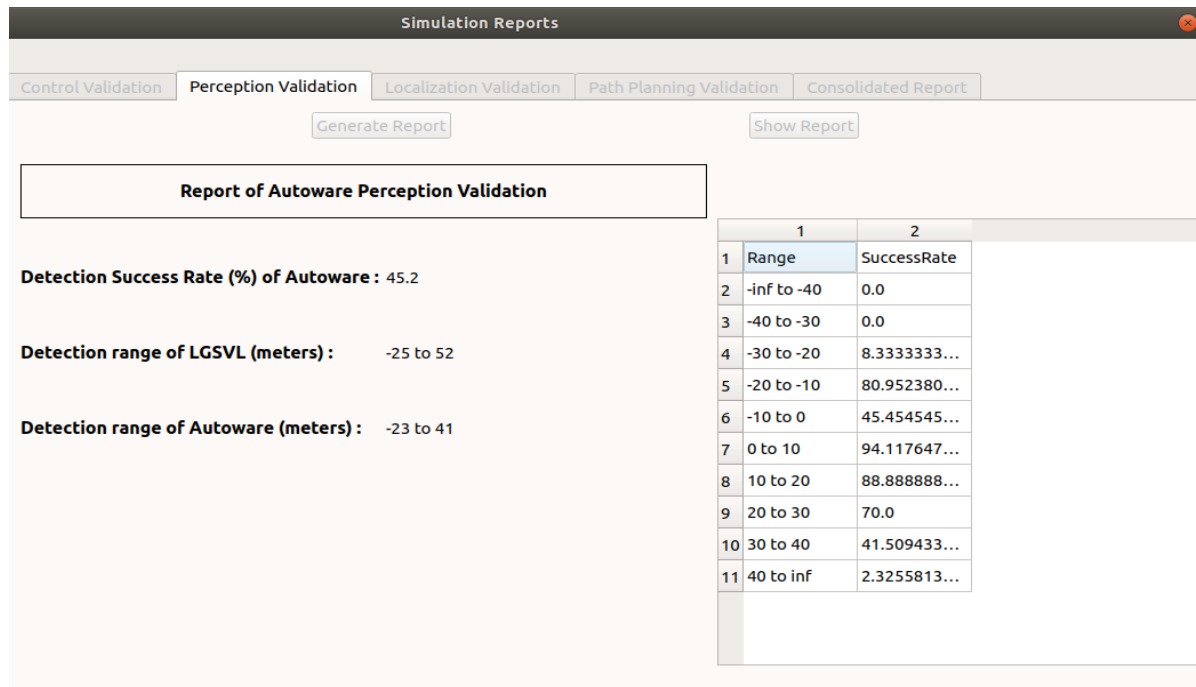
2. Now click on the **Run Scenario** button to run the selected scenario.



3. Once the simulation is completed, Click on the **Stop scenario** button and then Click on the **Show Report**.
4. Now click on the **Load Report** button and select the respective scenario in the list.
5. Now click on the **Generate Report** button, it will take some time to process the report.



6. Once generated data is processed, click on the **Show Report** button to see the final results.



Note: - You can run Control Test suite using the same steps as mentioned above. Firstly, you will need to stop the ADE by clicking the Stop ADE button and Select the Control Test Suite.

Assumptions/Issues:

- If there is only ego vehicle in the simulation then data will not compute.
- Sometimes Rviz crashes but the perception stack is running in the back ground
- While running scenario using scenic, it gets stuck or hangs while connecting to Ros2 Bridge. So we need to forcefully terminate using Ctrl+C and start again.
- Sometimes the machine might hang based on its system configuration

References:

- **OSSDC Simulator** - <https://github.com/OSSDC/OSSDCSIM/releases/tag/OSSDC-SIM-v1.1>
- **Scenic** - https://scenic-lang.readthedocs.io/en/latest/syntax_guide.html?highlight=facing#specifiers
- **PythonAPI** - <https://github.com/OSSDC/PythonAPI>
- **AutowareAuto avp demo** - <https://autowarefoundation.gitlab.io/autoware.auto/AutowareAuto/avpdemo.html>