Exploring Generalized Visual Navigation using CARLA-ROS bridge

Student: Pragya Kaushik (a1840097)

Supervisor: Dr Sourav Garg

Problem Introduction

- Navigating robots autonomously in diverse environments is a challenge
- Historically, navigation solutions relied on geometric mapping and planning
- Modern complexities demand a more sophisticated approach

THE "GNM" PAPER

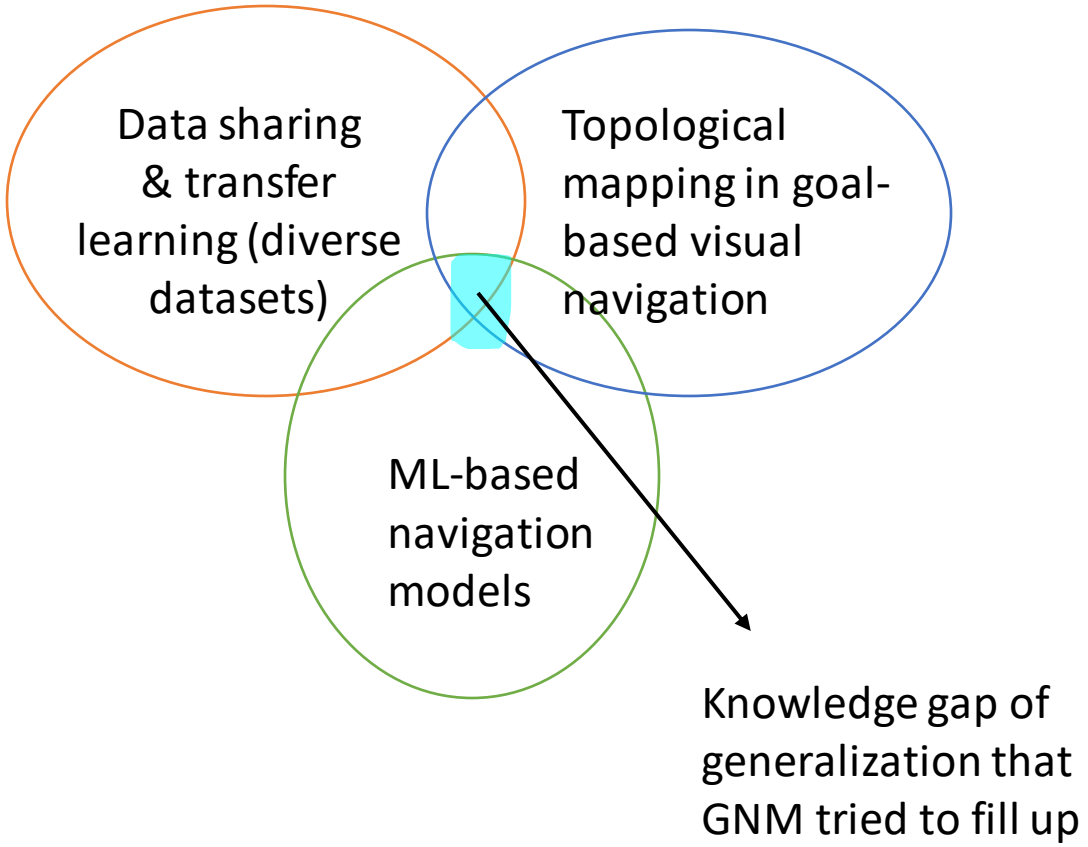
GNM Paper (2023): Shah et al.'s influential paper sparks our project exploration

GNM authors emphasize shortcomings in traditional approaches for real-world navigation challenges

Highlight the promise of machine learning—leveraging past experiences for informed navigation decisions

Introduces a groundbreaking method—training a general goal-oriented navigation model using diverse data from structurally dissimilar robots

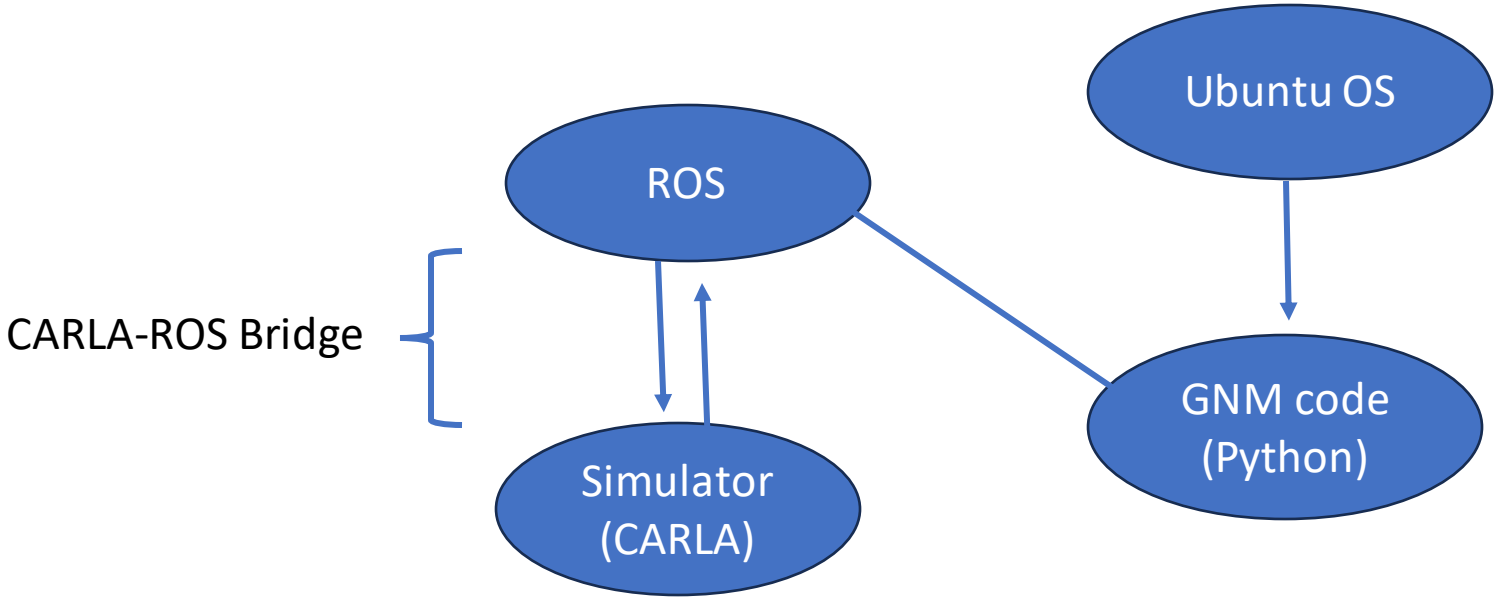
Problem Motivation



Project Objectives

- **Evaluate the effectiveness of GNM**, identify its limitations, and propose enhancements
- Execute the project **within a simulated environment using the Carla-ROS bridge**, allowing for controlled testing and analysis.
- Explore the compatibility and integration of GNM within the CARLA simulation environment through the ROS bridge

Project Infrastructure and Tools



Steps involved in the whole project

1. Familiarization with ROS and Robot Simulator
2. Teleoperation and Data Collection
3. Image Stitching for Dataset Creation
4. Data Processing via Topological Navigation
5. Model Execution and Deployment
6. Identification of Failure Modes
7. Cross-Robot Testing

FOR THE PURPOSE OF THIS COURSE; WOULD CONTINUE REST LATER ON

NOTE: HAVE NO PRIOR KNOWLEDGE WHATSOEVER

Project Progress and Challenges Faced

1. **Learning Fundamental Concepts:**

- Understanding the dynamics of robot navigation and delving into machine learning nuances, particularly within the GNM model.
- In-depth exploration of the GNM codebase, encompassing model training, deployment, and the intricacies of data processing pipelines.

Project Progress and Challenges Faced

2. Mastering Ubuntu OS:

- Familiarization with Ubuntu's command-line interface, file system structure, and command mastery for seamless execution of tasks.

- Ubuntu OS was a project prerequisite, providing the essential foundation for various tools and software components.

Project Progress and Challenges Faced

3. Navigating Compatibility Challenges:

- Sticking with Ubuntu 20.04 for compatibility with project tools.
- Addressing challenges like PyRobot compatibility issues and choosing an optimal path without deviating from project objectives.
- Shifting to ROS Noetic due to GNM's requirement of ROS kinetic, ensuring a harmonious software environment.

Project Progress and Challenges Faced

3. Navigating Compatibility Challenges:

- Troubleshooting and overcoming recurring segmentation errors, uncovering Python version incompatibility, and implementing corrective measures.

- Creating a dedicated virtual environment with Python 3.7 for seamless compatibility.

Project Progress and Challenges Faced

4. ROS and Carla Exploration:

- Immersive learning through tutorials and documentation on ROS workspace, nodes, rosbag, rostopic, and other fundamental concepts.

- Independent exploration and problem-solving through trial and error, emphasizing the exploratory nature of the project.

Project Progress and Challenges Faced

5. Carla-ROS Bridge Operations:

- Delving into the bridge's documentation, addressing challenges in tasks such as object and sensor deployment, manual control, Ackermann control, waypoint publishing, and AD agent usage.

- Optimization of data collection and teleoperation methods, with the CARLA AD agent identified as the most effective approach.

Project Milestone: Successful manual control and teleoperation within Carla Simulator, simultaneous data retrieval through rostopic commands, marking a significant project milestone.

Future Trajectory and Enhancements

- Open avenues for further exploration and enhancement, particularly in leveraging the capabilities of the CARLA AD agent.

- Approaches for optimizing teleoperation and data collection include:
  - Creation of a Python 3.7-compatible virtual environment
  - Integration of the OpenCV library

- Have laid solid foundation for future research and development, especially for the major next steps in the project

Thank you!