**Maintenance and Implementation of Future Changes in MaskGAE model**

**System Overview**

The system consists of Python files that include the architecture of a Graph Autoencoder (GAE) and various Python libraries for graph processing, deep learning, and mathematical computations. Key libraries used in the project include torch, torch\_geometric, numpy, scipy, and texttable. The model is designed to handle graph datasets, supporting tasks like link prediction and graph reconstruction for training and evaluation purposes.

# Requirements Make sure that the following libraries and frameworks are installed. Higher versions should be compatible:

# numpy == 1.21.6

# torch == 1.12.1+cu102

# torch-cluster == 1.6.0

# torch\_geometric >= 2.4.0

# torch-scatter == 2.0.9

# torch-sparse == 0.6.14

# scipy == 1.7.3

# texttable == 1.6.2

# CUDA == 10.2

# CUDNN == 7.6.0

To install the required dependencies, use the following command:  
pip install $library\_name==$version

Make sure to have CUDA and CUDNN installed for GPU acceleration, as the project relies on them for efficient computation.

# Codebase Maintenance

**Version Control** (GIT)

The code is available in git at the following path: <https://github.com/gilseg10/BraudeProject/tree/main>

For version control you can open a new branch, work on the code, and make a pull request which we will approve. Alternatively, you can download the code to a local environment and make the desired changes in it (without updating in git).

**Dependency Management**

Python libraries tend to change and be updated over time, therefore it is useful to check every period of time whether an update or improvement has been made in one of the functions of the libraries used in the model and update accordingly. Both for the purpose of improving the model and for a necessary dependency adjustment.

**Reproduction:**

To reproduce the results or run experiments, first, clone the project repository. Then, you can run the following commands for the **link prediction** tasks. For example,link prediction experiments with Cora data set:

1. With default values for parameters:

**python train\_linkpred.py --dataset Cora --bn**

2. With edge masking and all other parameters set to default:

**python train\_linkpred.py --dataset Cora --bn --mask Edge**