**Graph Neural Network**

Graph Neural Networks (GNNs) are a class of neural networks designed to operate on graph-structured data. Unlike traditional neural networks that operate on data in the form of grid (images) or sequential data (text), GNNs are tailored for data represented in the form of graphs, which consist of nodes connected by edges. GNNs are used in predicting nodes, edges, and graph-based tasks.

**Frameworks**

PyTorch Geometric: Provides utilities for deep learning on irregularly structured input data such as graphs.

Deep Graph Library (DGL): Built for easy implementation of graph neural network models in various frameworks like PyTorch, and TensorFlow.

PyTorch Lightning: Simplifies the training process and provides useful abstractions for organizing code, and handling distributed training.

Spektral: Classify the users of a social network, generate new graphs, cluster nodes, predict links, and any other task where data is described by graphs

Graph nets: Offers a flexible and composable framework for constructing GNNs

**Advantages of Frameworks**

Efficiency: Optimized for efficient execution on modern hardware, including CPUs, GPUs, and specialized accelerators

Integration with Deep Learning Tools: Integrate with other deep learning tools and libraries, such as TensorFlow, PyTorch, and MXNet. This integration allows users to combine GNNs with traditional neural network architectures.

Flexibility and Customization: Support various GNN architectures like Graph Convolutional Networks, Graph Attention Networks, and more. They allow users to customize and extend these architectures with different aggregation functions, activation functions.  
**Disadvantages of Frameworks**

Complexity: Frameworks may not handle all types of graphs where they are large, dynamic, or heterogeneous. Users may need to preprocess and manipulate the graph data.

Compatibility: Evolve with updates, and new features, which can lead to compatibility issues or changes in behavior between different versions.

Learning curve: Learning to use GNN frameworks effectively may require some time and effort, especially for users who are new to graph-based machine learning.

**Popular Frameworks**

PyTorch Geometric and Deep Graph Library (DGL) were among the most popular frameworks for working with Graph Neural Networks (GNNs), with PyTorch Geometric being particularly favored by the research community due to its ease of use and extensive functionality for graph-based tasks.

**Existing Papers that use the frameworks**

-Deep Graph Library Optimizations For Intel® X86 Architecture –

-PocketFinderGNN: A manufacturing feature recognition software based on Graph Neural Networks (GNNs) using PyTorch Geometric and NetworkX

**Tools that use the frameworks**

**Geon-GCN:** A library that implements various Graph Convolutional Networks (GCNs) using PyTorch Geometric

**OGB:** provides datasets, model implementations, and leaderboards for benchmarking graph-based methods, often leveraging PyTorch Geometric