1. Hands-On Introduction

In this hands-on, we will continue practicing on of the deep graph neural network frame, [DGL](https://github.com/dmlc/dgl), and mainly focus on stochastic training GNN with single and multiple GPUs. After this hands-on, you will feel free to use DGL to build your own models and train your own dataset at ease. Please read the following content for more detail.

1. DGL
   1. Introduction

In previous hands-on, we have learned the basic usage of DGL, and build some classical models on several well-known graph datasets. However, in real world, it is not always the graph data are small enough such that we can fit them into GPU directly. We need to process our data with neighbor sampling, and sometimes we also need to use whole neighbors to do inference. In this hands-on, we will learn how to perform stochastic training and inference in our data layer by layer.

* 1. Tutorials

Please follow the tutorial in [KDD20 hands-on](https://github.com/dglai/KDD20-Hands-on-Tutorial) and run the first and second “ipynb” files in “4-large\_graph” folder (‘1 - Node Classification.ipynb’, ‘2 - Unsupervised Learning and Link Prediction.ipynb’). It is also welcome to run the third file if you want to learn how to customize your sampling method and run your model on multiple GPU.

* 1. Tasks

In this hands-on, you need to build a GNN model to do supervised node classification task in stochastic training. Here we recommend you to use GraphSAGE model training on “[ogbn-products](https://ogb.stanford.edu/docs/nodeprop/#ogbn-products)” in [Open Graph Benchmark](https://ogb.stanford.edu/) (OGB). You can see the examples on OGB official webpage, [DGL tutorial](https://docs.dgl.ai/guide/data-loadogb.html#guide-data-pipeline-loadogb), or even some [existed code](https://ogb.stanford.edu/docs/leader_nodeprop/#ogbn-products). If you want to practice more, try to modify your supervised version code to unsupervised setting, or try to use more GPU to train your model. Good luck!