
# USENIX Security '25 Artifact Appendix: Distributed Private Aggregation in Graph Neural Networks

Huanhuan Jia[†]
Southeast University

Yuanbo Zhao[†]
Southeast University

Kai Dong[*]
Southeast University

Zhen Ling
Southeast University

Ming Yang
Southeast University

Junzhou Luo
Southeast University

Xinwen Fu
University of Massachusetts Lowell

## A Artifact Appendix

### A.1 Abstract

This appendix describes the software artifact that implements the algorithms and evaluations proposed in "Distributed Private Aggregation in Graph Neural Networks". Specifically, the artifact includes the source code for `DPA-GNN`, the datasets used in the experiments, the bash scripts for executing the experiments, and the implementation of the comparison methods. These components are essential for reproducing the experiments (E1) to (E10) described in §7.2 of our paper.

### A.2 Description & Requirements

#### A.2.1 Security, privacy, and ethical concerns

Our research uses publicly available graph networks that contain no personal or sensitive data, avoiding privacy and ethical concerns. The artifact operates with standard benchmark datasets, adhering to ethical research standards and avoiding risks related to personal data. Additionally, it does not involve destructive actions or disable security mechanisms, ensuring users' machines and data remain secure.

#### A.2.2 How to access

The artifact is publicly available at https://doi.org/10.5281/zenodo.14710401.

#### A.2.3 Hardware dependencies

The artifact evaluation requires the following hardware configuration:

- **CPU:** Experiments use an Intel Xeon Gold 6430 processor (16 vCPUs), but comparable CPUs should also be sufficient.
- **RAM:** The server is equipped with 120 GB of RAM.
- **GPU (optional):** Experiments run on an NVIDIA GeForce RTX 4090 GPU, but similar consumer-grade GPUs are acceptable.
- **Storage:** At least 128 GB of storage is required for datasets, models, and experimental results.

#### A.2.4 Software dependencies

The artifact requires the following operating system and essential software packages:

- **OS:** Windows, Linux, or macOS.
- **Python Version:** Python 3.9.
- **Packages:**

```
- pytorch=2.0.0
- torch-geometric=2.4.0
- torch-scatter=2.1.2
- torch-sparse=0.6.18
- numpy=1.24.4
- pandas=2.0.3
- scikit-learn=1.3.2
- scipy=1.8.0
```

#### A.2.5 Benchmarks

**Datasets.** We conduct experiments on 6 public real-world datasets: Cora, CiteSeer, LastFM, Facebook, Amazon, and Reddit, as shown in §7.1.2 of the paper. These datasets are automatically downloaded during the first run of the script.

**Baselines.** We consider various baseline methods for comparison, as shown in §7.1.3 of the paper. The code for all baseline methods is provided in the `./baselines` directory. These baselines are categorized into three types: enhanced variants of existing methods, existing privacy preserving GNN methods, and centralized node-level DP methods, as detailed below:

[†]These authors contribute equally to this work.
[*]Corresponding author, email: dk@seu.edu.cn.

Enhanced variants of existing methods: Solitude+RR, LPGNN+RR, RGNN+RR

Existing privacy preserving GNN methods: Solitude, LPGNN, RGNN, DPRR, Blink.

Centralized node-level DP methods: DPSGD, GAP-NDP, PNPiGNNs, DPDGC.

## A.3 Set-up

### A.3.1 Installation

Please verify that the hardware and software dependencies are satisfied, and download the code from the repository. The environment setup required for this artifact is defined in the `requirements.txt` file. Further details, including the file structure and command-line parameters, are provided in the `README.md` file.

### A.3.2 Basic Test

To run the basic functionality test, execute the following command:

```
python train.py --c_rate 0.8
                --layerx 10
                --layery 8
                --epsilon_all 8
                --epsilon_rate 0.05
                --dataset cora
```

This command runs a basic test to construct a `DPA-GNN` using the Cora dataset, with a clipping rate of 0.8, 10 feature aggregation layers, and 8 label aggregation layers, under a total privacy budget of 8 and a feature privacy budget allocation ratio of 0.05.

If everything is functioning correctly, the results can be found in `output_results.txt`.

## A.4 Evaluation Workflow

### A.4.1 Major Claims

**(C1):** `DPA-GNN` achieves superior performance while preserving the privacy of node features, edges, and labels, demonstrating an effective utility-privacy trade-off. This is demonstrated by experiment (E1), as detailed in §7.2.1 of the paper, with results presented in Table 3.

**(C2):** `DPA-GNN` shows significant advantages when edges are private. This is demonstrated by experiments (E2) to (E5), as detailed in §7.2.2 of the paper, with results presented in Figures 3 to 6.

**(C3):** `DPA-GNN` achieves performance comparable to, or even surpassing, four competing methods when transitioning from distributed to centralized settings. This is demonstrated by experiment (E6), as detailed in §7.2.3 of the paper, with results shown in Figure 7.

**(C4):** We conduct experiments to evaluate the effect of hyperparameter settings. This is demonstrated by experiments (E7) to (E10), as detailed in §7.2.4 of the paper, with results presented in Figures 8 to 10 and Table 4.

### A.4.2 Experiments

**(E1):** *[Utility-privacy trade-off] [30 human-minutes + 60 compute-hours]: This experiment evaluates the trade-off between utility and privacy achieved by `DPA-GNN` and the baseline methods across six datasets under varying privacy budget settings. The results are presented in Table 3 of the paper.*

**Preparation:** Upon the first execution of the code, the datasets corresponding to the command will be automatically downloaded to the `./node-level` directory.

**Execution:** Run the `run_E1.sh` script to reproduce the results of `DPA-GNN` across all six datasets. We set the total privacy budget $\varepsilon$ to $\{2,4,6,8,10\}$, with the node representation privacy budget $\varepsilon_h$ set to 5%$\varepsilon$ and 20%$\varepsilon$. The optimal values for `c_rate`, `layerx`, and `layery` for each `dataset` are summarized in Table 1.

Table 1: Optimal parameter settings for different datasets.

| dataset | c_rate | layerx | layery |
|---|---|---|---|
| cora | 0.8 | 10 | 8 |
| citeseer | 0.8 | 8 | 10 |
| facebook | 0.7 | 10 | 6 |
| lastfm | 0.6 | 6 | 4 |
| amazon | 0.4 | 10 | 2 |
| reddit | 0.2 | 8 | 2 |

The same `run_E1.sh` script can also be used to reproduce the results of the privacy-enhanced variants of existing methods—Solitude+RR, LPGNN+RR, and RGNN+RR—whose source code is available in the `baselines` directory of our Zenodo repository.

**Results:** Each experiment is repeated 10 times, and the mean and standard deviation are reported. The results of the experiments are recorded in tabular format and saved to `./output/results_E1.csv` file. Note that due to numeric instability, exact numbers from the paper may be impossible to reproduce.

**(E2 to E5):** *[Ablation studies] [1 human-hour + 72 compute-hours]: These experiments conduct ablation studies by analyzing scenarios in which only certain data types are private, generating the results presented in Figures 3 to 6 of the paper.*

**Execution:** Run the scripts from `run_E2.sh` to `run_E5.sh` to reproduce the results of Experiments (E2) through (E5) for `DPA-GNN` and the existing privacy-preserving GNN methods across all six datasets.

The parameter settings for `DPA-GNN` differ slightly across the ablation experiments and are summarized as follows.

In Experiment (E2), both `layerx` and `layery` are set to 2. In Experiment (E4), only `layerx` is set to 2 for large-scale datasets. When features or labels are not considered private, the corresponding privacy budget parameters, `epsilon_x` or `epsilon_y`, are set to `inf`. The value of the `dummy_r` parameter is determined based on the $\varepsilon_e$ setting described in the paper. To implement the non-private edge setting in Experiment (E5), incorporate the following parameter into the execution command.

```
--execute_dummy False
```

**Results:** The results of the experiments are recorded in the output files `results_E2.csv` to `results_E5.csv`, located in the `./output` directory. Note that due to numeric instability, exact numbers from the paper may be impossible to reproduce.

**(E6):** *[Scenario transition] [30 human-minutes + 12 compute-hours]: This experiment compares the performance of DPA-GNN with baselines when transitioning from distributed to centralized settings and generates the results presented in Figure 6 of the paper.*

**Execution:** DPA-GNN operates without any changes to the model architecture or parameter settings when transitioning from distributed to centralized environments. Run the `run_E6.sh` script to reproduce the results of Experiments (E6) for both `DPA-GNN` and the enhanced variants of existing methods.

**Results:** The experimental results are logged in the `./output/results_E6.csv` file. Note that due to numeric instability, exact numbers from the paper may be impossible to reproduce.

**(E7 to E10):** *[Effect of parameters] [1 human-hour + 96 compute-hours]: These experiments evaluate the effect of parameters, including the number of layers $l_h$ and $l_y$, clipping rate, dummy parameter, and selective parameter, generating the results presented in Figures 8 to 10 and Table 4 of the paper.*

**Execution:** Run the `run_E7.sh` script to reproduce the results of Experiment (E7). This experiment requires modifying the execution command parameters, as shown below:

```
--layerx {LAYERX}
--layery {LAYERY}
```

Run `run_E8.sh` to reproduce the results of Experiment (E8) by modifying the parameter as shown below:

```
--c_rate {C_RATE}
```

Run `run_E9.sh` to reproduce the results of Experiment (E9) by adding the following parameter:

```
--dummy_r {DUMMY_R}
```

Run `run_E10.sh` to reproduce the results of Experiment (E10) by adding the following parameters:

```
--num_shares {NUM_SHARES}
--num_parties {NUM_PARTIES}
```

**Results:** The results of the experiments are recorded in the output files `results_E7.csv` through `results_E10.csv`, located in the `./output` directory. Note that due to numeric instability, exact numbers from the paper may be impossible to reproduce.

## A.5 Notes on Reusability

To use the artifact on other graph datasets, ensure that the dataset is pre-processed into the appropriate format. You can easily add new data loaders to accommodate new datasets.

## A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.