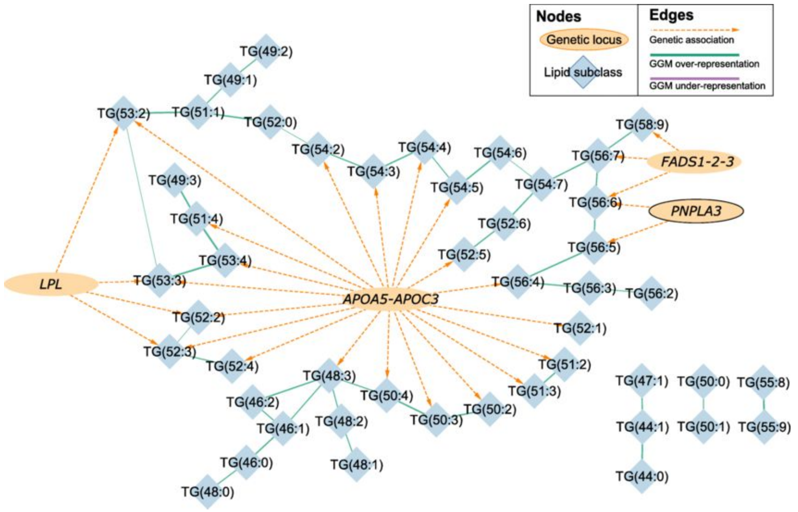
# Convergent Privacy Framework for Multi-layer GNNs through Contractive Message Passing

**Yu Zheng***, **Chenang Li***, **Zhou Li***, **Qingsong Wang**#

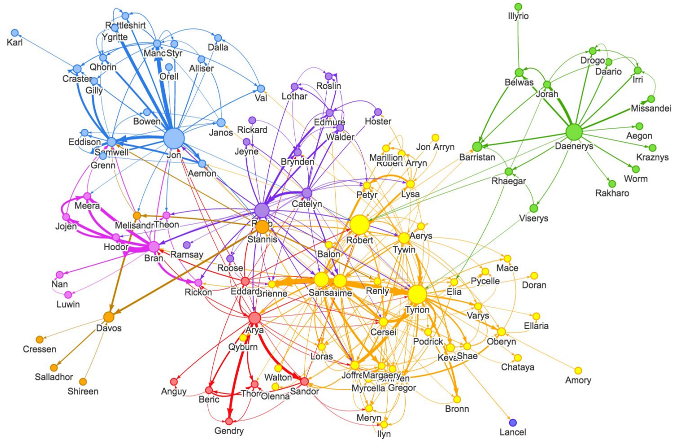
* University of California, Irvine
# University of California, San Diego

NDSS

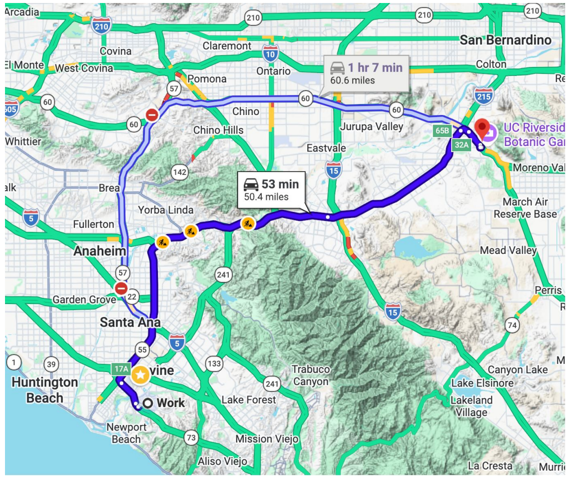
# Graphs are Everywhere




Genetic association



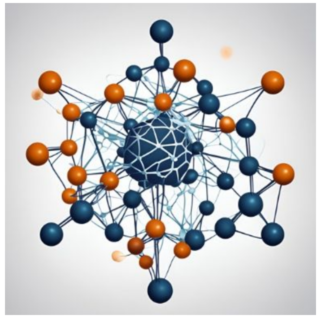
Social Networks



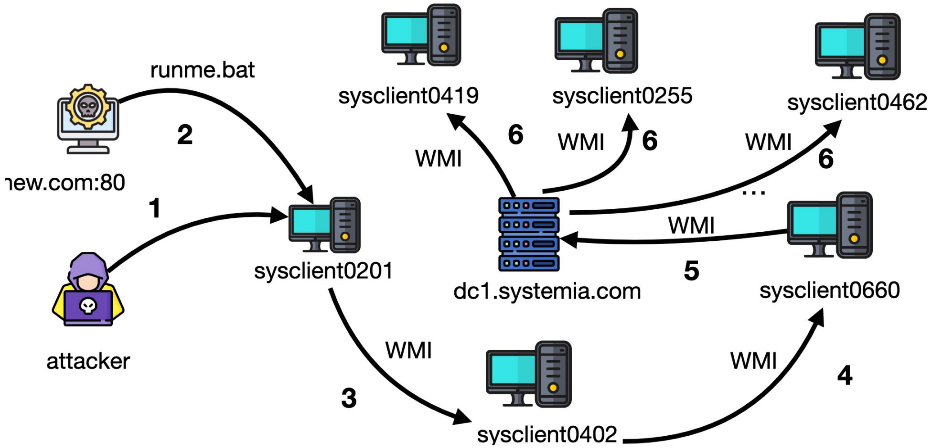

Maps

**Graph**

$G = (V, E)$



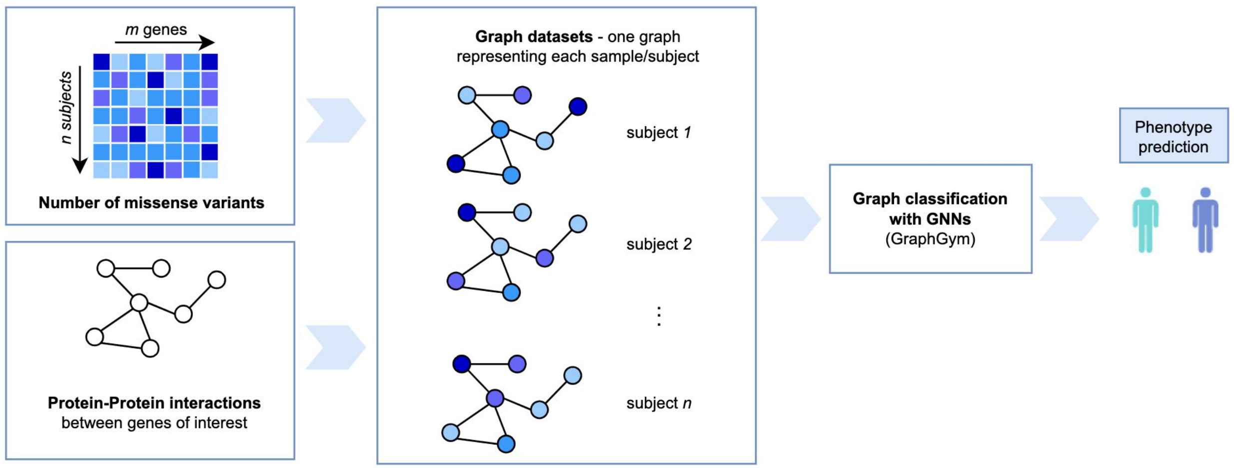
Chemical/Drug Molecules

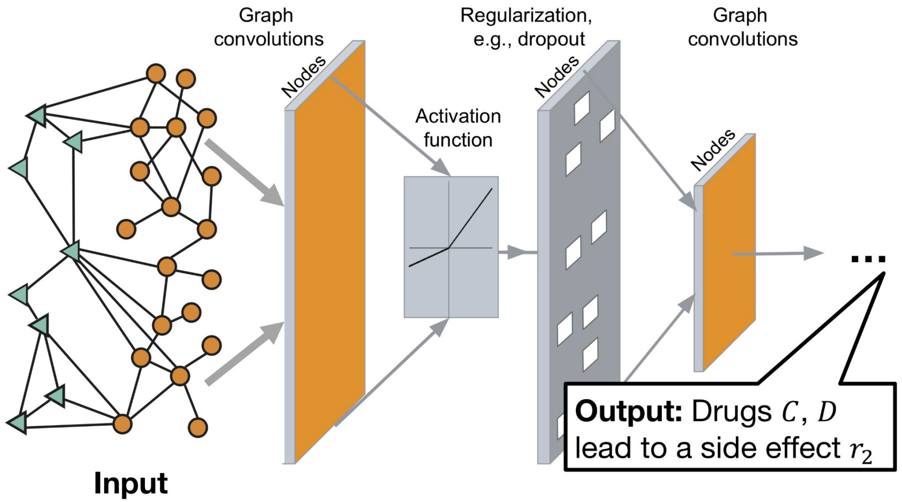



Network Logs

# Graph Neural Networks

## Graph Neural Networks (GNNs) are designed for structural data.



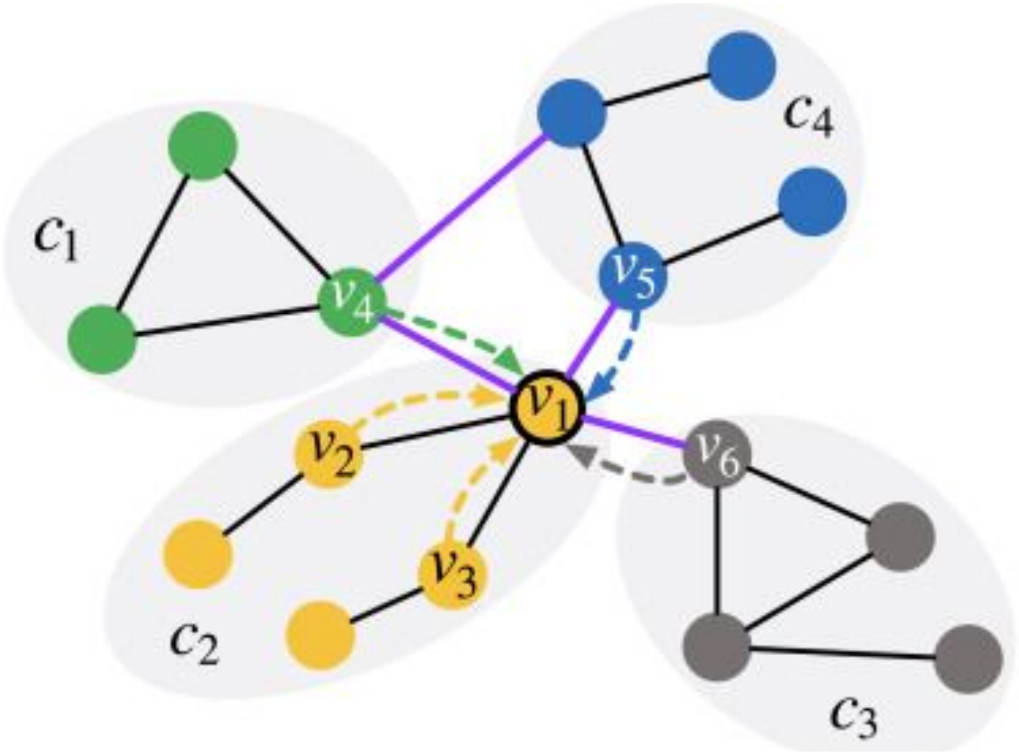




Early diagnosis of Alzheimer's disease

Drug analysis of side effects

[1] L. Hernández-Lorenzo. On the limits of graph neural networks for the early diagnosis of Alzheimer's disease. In Scientific Reports, 12(1), 17632.
[2] J. Leskovec. Graph Neural Networks for Multirelational Link Prediction. In CS224W: Machine Learning with Graphs.

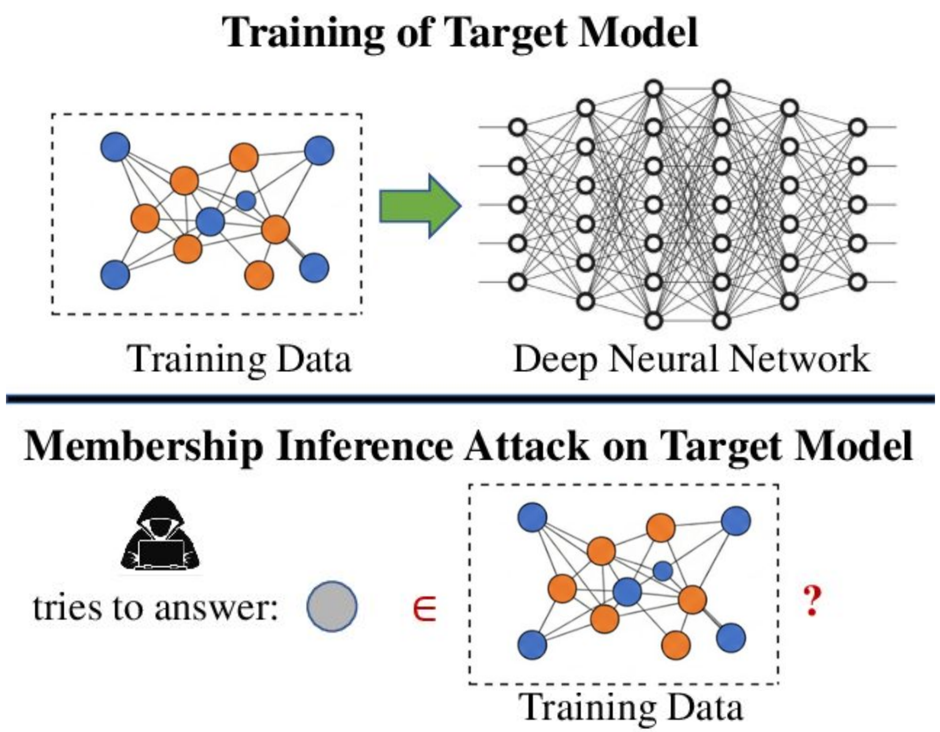
# Message-Passing GNNs




- Message-passing paradigm: aggregating information from their neighbors.
  - $v_1$: aggregated from neighbors.
- Aggregation layer by layer.

[2] Enhancing Graph Neural Networks by a High-quality Aggregation of Beneficial Information. Neural Networks, 2021.

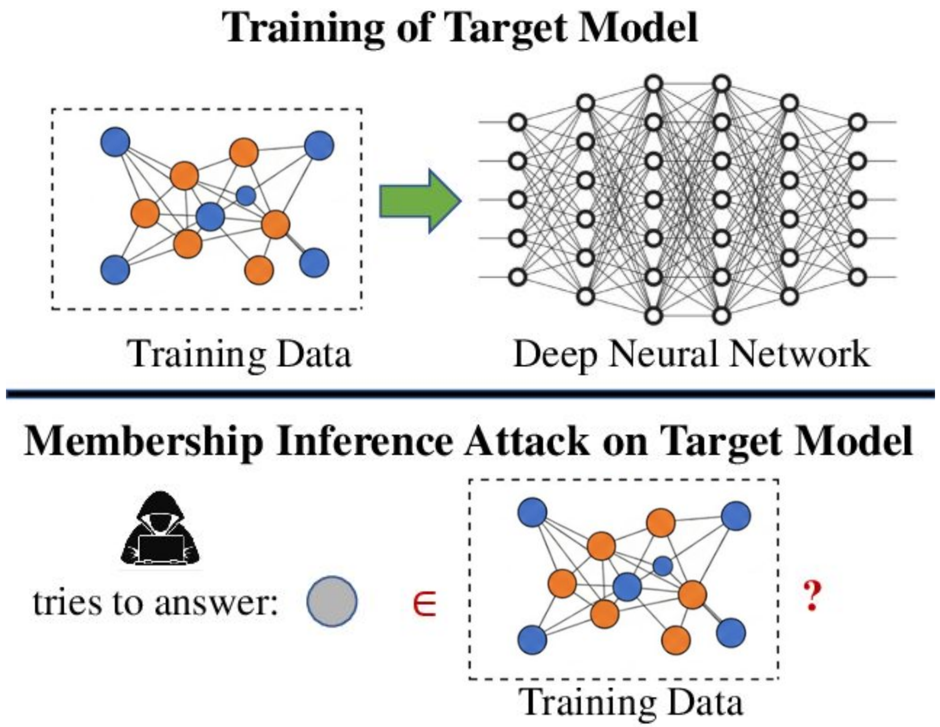
# Privacy Issue in GNNs




- Membership inference attacks (MIAs)
  - Node: is this user in the training social network?
  - Edge: is there a link between user A and B in training?
  - Subgraph: was this community structure used?

[1]F. Wu, Y. Long, C. Zhang, and B. Li, “LINKTELLER: recovering private edges from graph neural networks via influence analysis. IEEE S&P, 2022.
[2] J. Ye, A. Maddi, S. K. Murakonda, V. Bindschaedler, and R. Shokri, “Enhanced membership inference attacks against machine learning models.” In ACM CCS, 2022.

# Privacy Issue in GNNs




- Membership inference attacks (MIAs)
  - Node: is this user in the training social network?
  - Edge: is there a link between user A and B in training?
  - Subgraph: was this community structure used?

**Privacy attacks, e.g., MIAs, can recover sensitive information encoded within the graph via *black-box* model access.**

[1]F. Wu, Y. Long, C. Zhang, and B. Li, “LINKTELLER: recovering private edges from graph neural networks via influence analysis. IEEE S&P, 2022.
[2] J. Ye, A. Maddi, S. K. Murakonda, V. Bindschaedler, and R. Shokri, “Enhanced membership inference attacks against machine learning models.” In ACM CCS, 2022.

# Differential Privacy (DP) for Graphs

- Definition 1. Two datasets ***D***, ***D'*** are adjacent if they differ by only one data instance. A random mechanism ***M*** is ($\epsilon$, $\delta$)-differentially private if for all adjacent datasets ***D***, ***D'*** and for all events $S$ in the output space of ***M***, we have

$$\Pr(M(D) \in S) \leq e^{\epsilon} \Pr(M(D') \in S) + \delta.$$

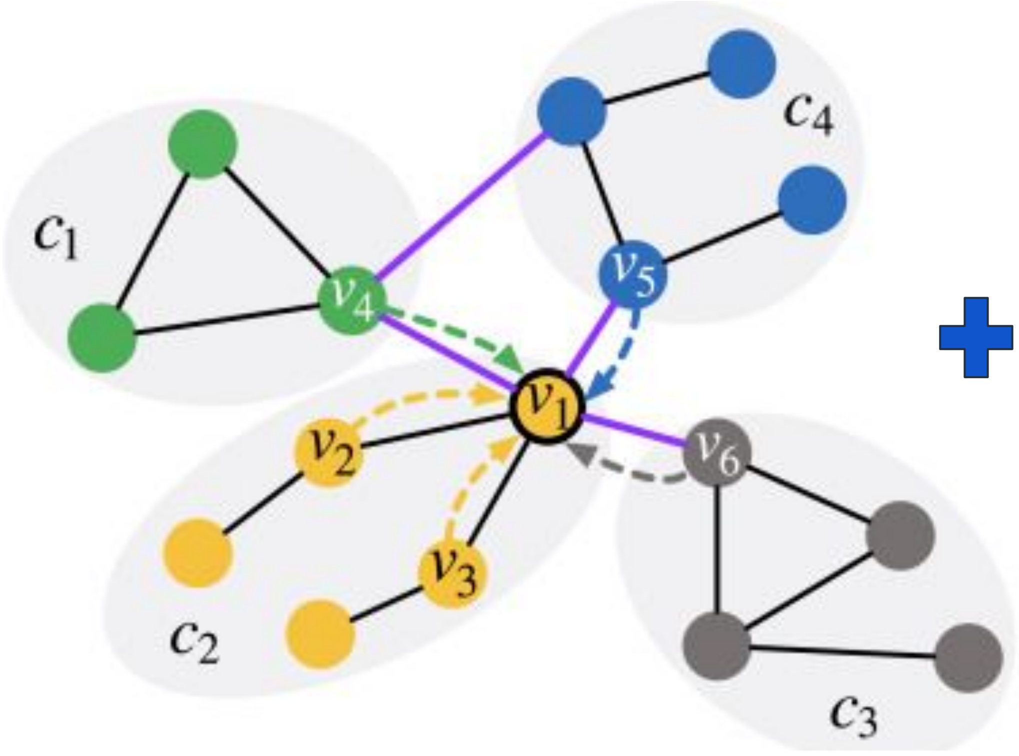
# Differential Privacy (DP) for Graphs

- Definition 1. Two datasets ***D***, ***D'*** are adjacent if they differ by only one data instance. A random mechanism ***M*** is ($\epsilon$, $\delta$)-differentially private if for all adjacent datasets ***D***, ***D'*** and for all events *S* in the output space of ***M***, we have

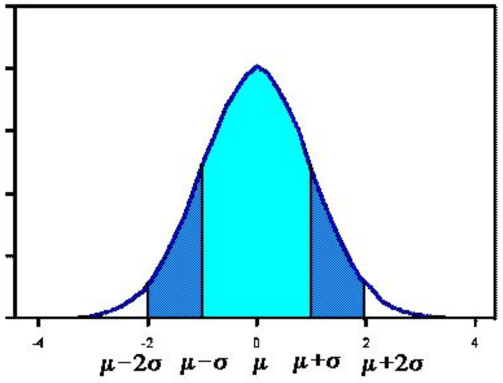
$$\Pr(M(D) \in S) \le e^{\epsilon} \Pr(M(D') \in S) + \delta.$$

- Edge-level neighboring & node-level neighboring.

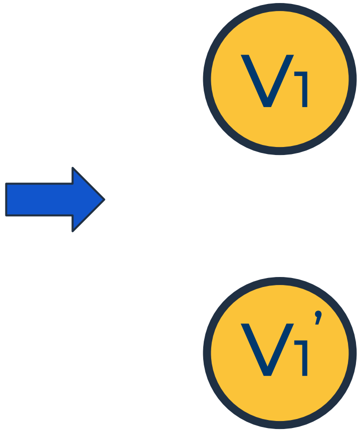
# Perturbed Message-Passing GNNs




+

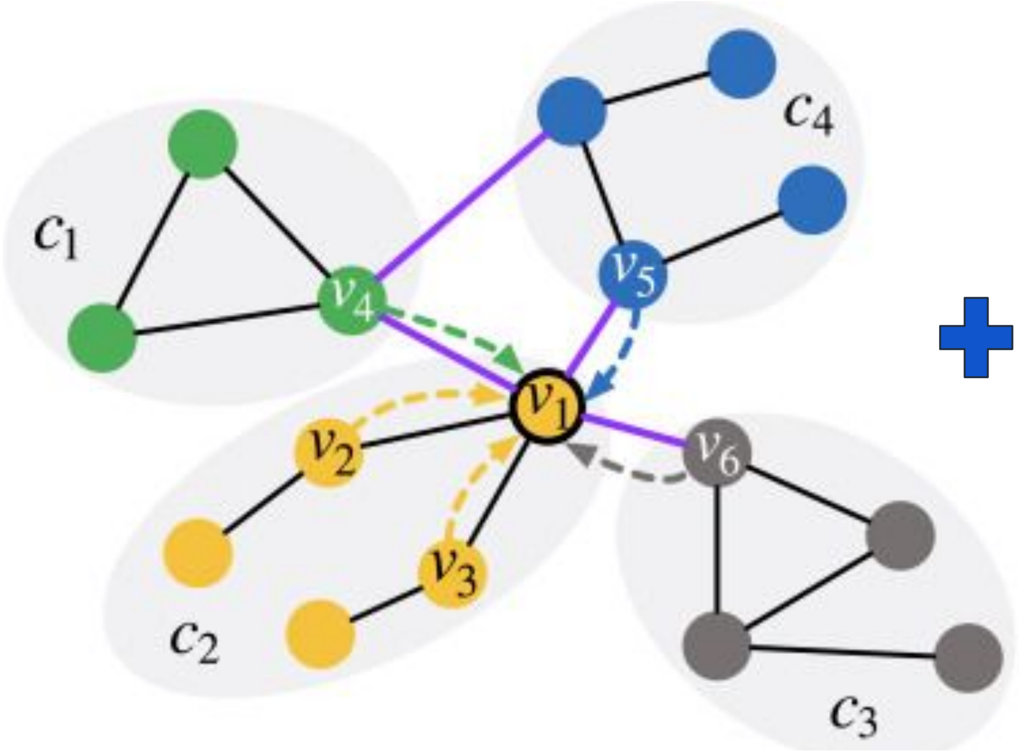


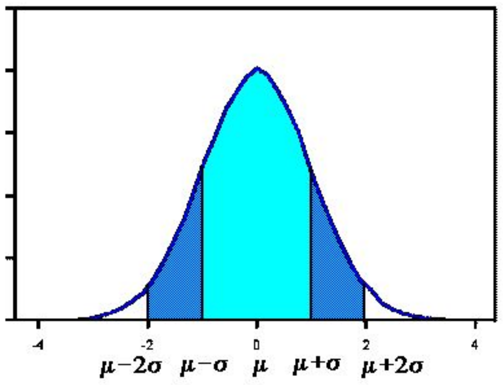

Add Gaussian noise.




Output node representations are similar.

# Perturbed Message-Passing GNNs




Add Gaussian noise.


Output node representations are similar.

# Perturbed Message-Passing GNNs

- **Formulation** on perturbed message passing:

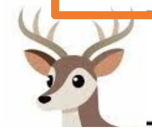
$$\boldsymbol{X}^{(k+1)} = \Pi(\mathrm{MP}_{G}(\boldsymbol{X}^{(k)}) + \boldsymbol{Z}^{(k)})$$

**Node representation/embeddings; $\boldsymbol{X}^{(0)}$: input feature matrix.**

**Gaussian noise ~ $N(0, \sigma^2)$.**

# State-of-the-Arts

TABLE I: Comparison between Private GNNs. EDP and NDP summarizes the results of private GNNs in Table III.

| Framework | Mechanism | Complexity per Layer | Calibrated Noise ($\sigma$) | EDP Utility | NDP Utility |
|---|---|---|---|---|---|
| PertGraph [45, 26] | Graph perturbation | $O(\|\boldsymbol{V}\|^2)$ | $\propto 1$ | ★★☆☆☆ | ★☆☆☆☆ |
| DPDGC [39] | Decoupled graph with perturbation | $O(\|\boldsymbol{E}\|)$ | $\propto \sqrt{K}$ | ★★★☆☆ | ★★★½☆ |
| GAP [46] | Perturbed message passing | $O(\|\boldsymbol{E}\|)$ | $\propto \sqrt{K}$ | ★★★★☆ | ★★★½☆ |
|  CARIBOU | Perturbed message passing | $O(\|\boldsymbol{E}\|)$ | $\propto \sqrt{\min(K, \frac{1-C_{\mathsf{L}}^K}{1+C_{\mathsf{L}}^K} \frac{1+C_{\mathsf{L}}}{1-C_{\mathsf{L}}})}$ | ★★★★★ | ★★★★★ |

- Share a critical limitation: the **privacy loss grows linearly with the number of layers K** or graph hops.
  - Require large amounts of noise to maintain a reasonable level of privacy guarantee; This, in turn, degrading model utility severely.

# Motivating Scenarios & Challenges

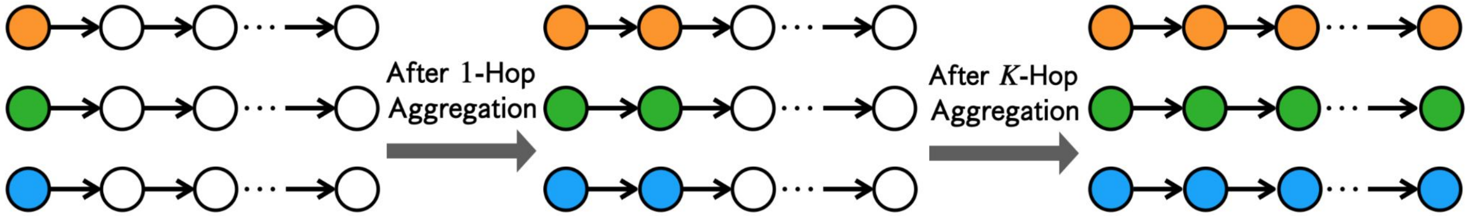
- Multi-layer GNNs: capture complex relations and analyze graphs with long-range interactions [3,4].
  - Example: ⬆ accuracy from **72.5%** to **88.2%** [4].

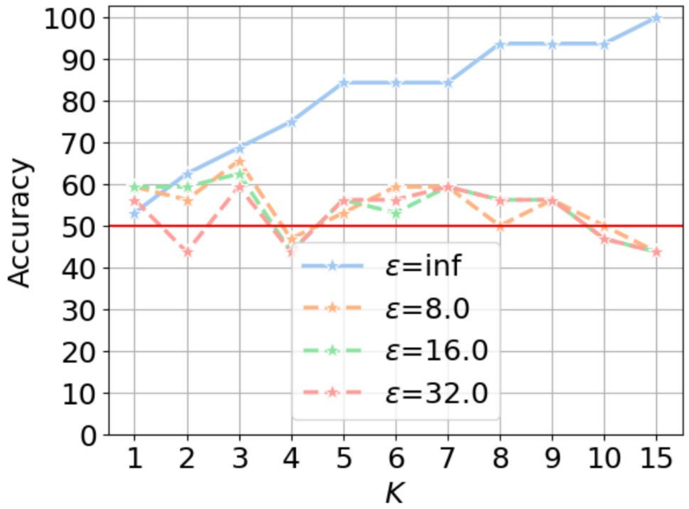
[3] How powerful are k-hop message passing graph neural networks. NeurIPS, 2022.
[4] Training graph neural networks with 1000 layers. ICML, 2021

# Motivating Scenarios & Challenges

- Multi-layer GNNs: capture complex relations and analyze graphs with long-range interactions [3,4].
  - Example: ⬆ accuracy from **72.5%** to **88.2%** [4].

- Challenges: larger *K* leads to larger privacy parameter $\epsilon$, a.k.a weak privacy guarantee.




[3] How powerful are k-hop message passing graph neural networks. NeurIPS, 2022.
[4] Training graph neural networks with 1000 layers. ICML, 2021.

# Research Question

- "**Over-smoothing**" phenomenon [5]: node representations become increasingly homogeneous as network depth increases and consequently making membership inference more challenging.

[5] Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. AAAI, 2020.

# Research Question

- "**Over-smoothing**" phenomenon [5]: node representations become increasingly homogeneous as network depth increases and consequently making membership inference more challenging.

⬇ Motivate

- Can we achieve differentially private graph learning with a **convergent (bounded)** privacy budget, thereby improving the privacy-utility trade-off for deeper GNNs?
  - **Not linearly increase with *K*!**

[5] Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. AAAI, 2020.

# Core Idea for Convergent Privacy

- Insight: leverage the inherent **privacy amplification** that occurs in multi-layer GNNs through **contractiveness**.
  - Motivated by DP-GD: **converge** to a finite value with **arbitrarily** many iterations.

# Core Idea for Convergent Privacy

- Insight: leverage the inherent **privacy amplification** that occurs in multi-layer GNNs through **contractiveness**.
  - Motivated by DP-GD: **converge** to a finite value with **arbitrarily** many iterations.

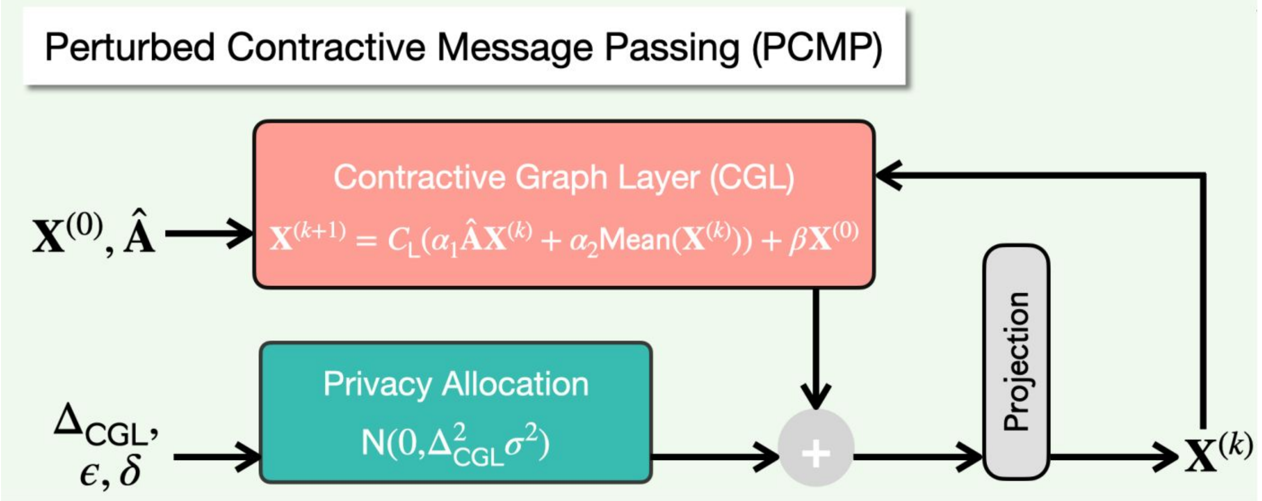
- When perturbed message passing is contractive, the **distance** between GNNs trained on neighboring datasets **shrinks** at each step.
  - *Translate* the advanced privacy analysis techniques from DP-GD to GNNs.

# Core Idea for Convergent Privacy

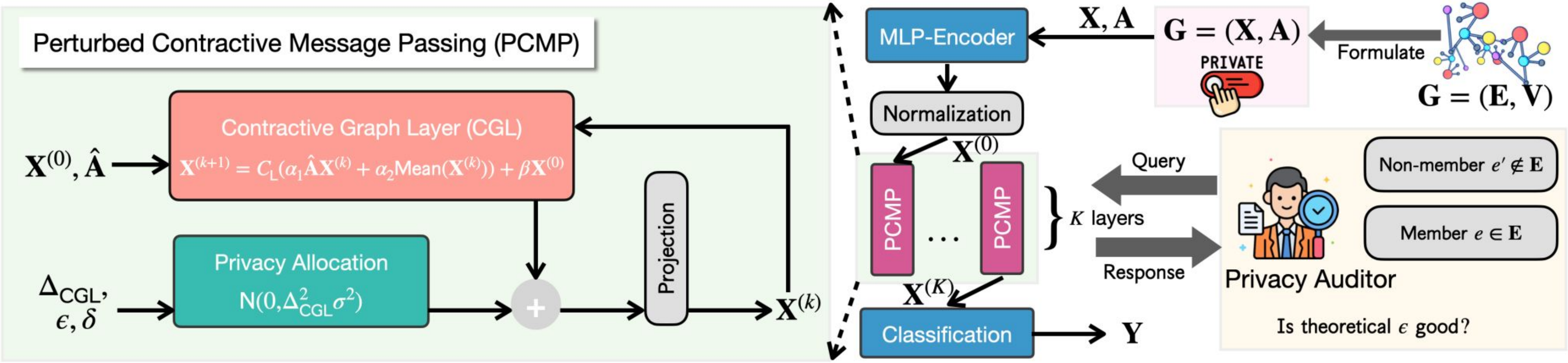
- Consequently, the influence of individual data points diminishes, leading to the amplified privacy rooted from “over-smoothing”.
  - Accordingly, remove the over-estimated privacy loss;
  - Derive a much tighter bound for finally released GNN model.

# Core Idea for Convergent Privacy

- Consequently, the influence of individual data points diminishes, leading to the amplified privacy rooted from "over-smoothing".
  - Accordingly, remove the over-estimated privacy loss;
  - Derive a much tighter bound for finally released GNN model.
- Two critical conditions:
  - Contractive message passing;
  - Release only $X^{(K)}$.

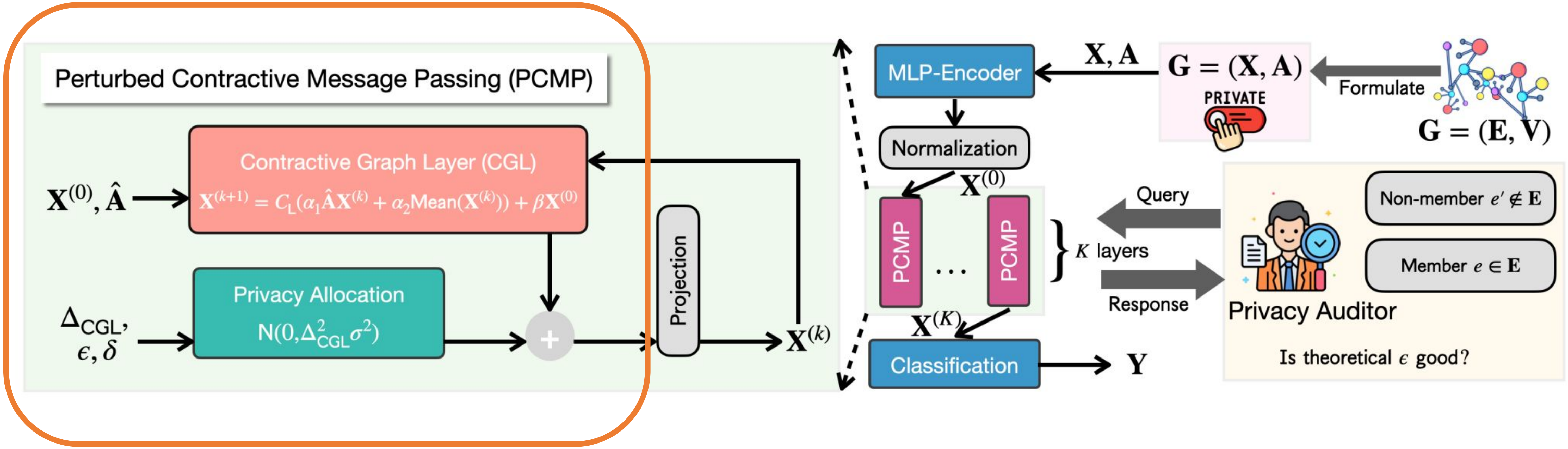


# Overview of Caribou



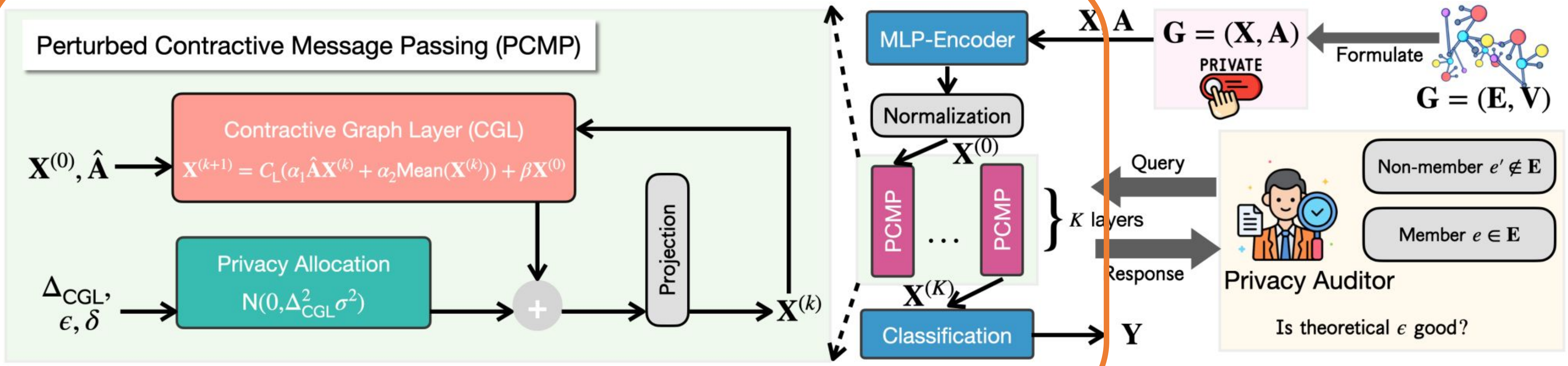
# Overview of Caribou

- **C**ontractive **A**ggregation **M**odule (CAM).
- **P**rivacy **A**llocation **M**odule (PAM).



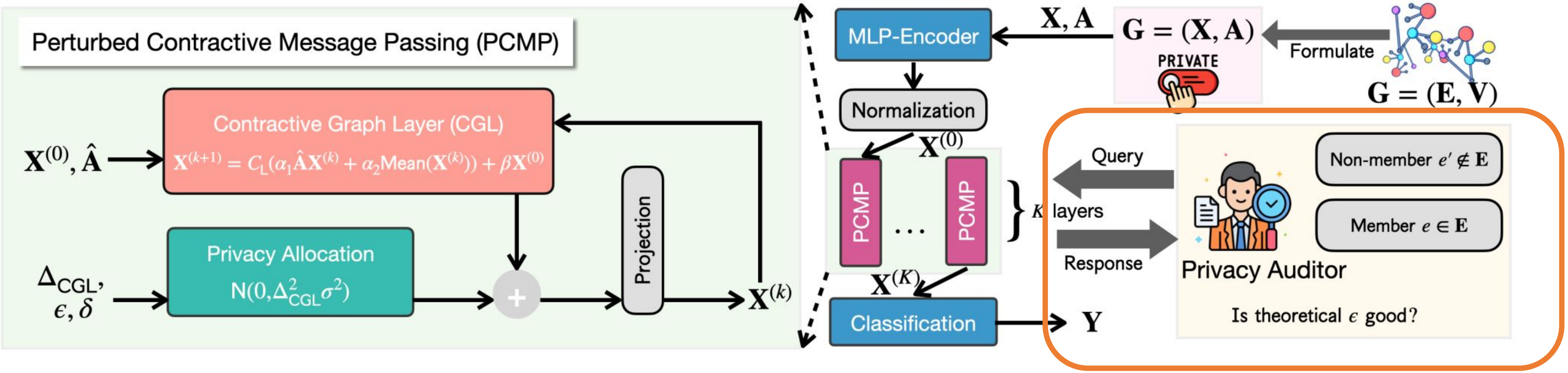
# Overview of Caribou

- **C**ontractive **A**ggregation **M**odule (CAM).
- **P**rivacy **A**llocation **M**odule (PAM).



# Overview of Caribou

- **C**ontractive **A**ggregation **M**odule (CAM).
- **P**rivacy **A**llocation **M**odule (PAM).
- **P**rivacy Au**d**iting **M**odule (PDM).



# Core Algorithm

- Step 1. Calculate Sensitivity;

▷ *Calculate the Required Noise Calibration (from PAM).*

**if** Edge-level privacy **then**
    Calculate $\Delta(\mathsf{CGL})$ through Equation 6
**else if** Node-level privacy **then**
    Calculate $\Delta(\mathsf{CGL})$ through Equation 8
**end if**
Calculate $\sigma^2$ through Theorem 6

▷ *Perturbed Contractive Message Passing (from CAM).*

**for** $k = 0, \ldots, K-1$ **do**
    $\boldsymbol{X}^{(k+1)} \leftarrow C_{\mathsf{L}}(\alpha_1 \hat{A} \boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})) + \beta \boldsymbol{X}^{(0)}$
    ▷ Contractive graph layer: compute node embeddings.
    $\boldsymbol{X}^{(k+1)} \leftarrow \boldsymbol{X}^{(k+1)} + \mathcal{N}(\mu, (\Delta(\mathsf{CGL}))^2 \sigma^2)$
    ▷ DP Perturbation.
    $\boldsymbol{X}^{(k+1)} \leftarrow \Pi_{\mathcal{K}}(\boldsymbol{X}^{(k)})$ ▷ Projection with norm 1.
**end for**

**Return**: $\mathbf{X}^{(K)}$

# Core Algorithm

- Step 1. Calculate Sensitivity;
- Step 2. Contractive message passing;

▷ *Calculate the Required Noise Calibration (from PAM).*
**if** Edge-level privacy **then**
  Calculate $\Delta(\mathsf{CGL})$ through Equation 6
**else if** Node-level privacy **then**
  Calculate $\Delta(\mathsf{CGL})$ through Equation 8
**end if**
Calculate $\sigma^2$ through Theorem 6

▷ *Perturbed Contractive Message Passing (from CAM).*
**for** $k = 0, \ldots, K-1$ **do**
  $\boldsymbol{X}^{(k+1)} \leftarrow C_{\mathsf{L}}(\alpha_1 \hat{A} \boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})) + \beta \boldsymbol{X}^{(0)}$
  ▷ Contractive graph layer: compute node embeddings.
  $\boldsymbol{X}^{(k+1)} \leftarrow \boldsymbol{X}^{(k+1)} + \mathcal{N}(\mu, (\Delta(\mathsf{CGL}))^2 \sigma^2)$
  ▷ DP Perturbation.
  $\boldsymbol{X}^{(k+1)} \leftarrow \Pi_{\mathcal{K}}(\boldsymbol{X}^{(k)})$ ▷ Projection with norm 1.
**end for**

**Return**: $\mathbf{X}^{(K)}$

# Core Algorithm

- Step 1. Calculate sensitivity;
- Step 2. Contractive message passing;
- Step 3. Add appropriate DP noise;

▷ *Calculate the Required Noise Calibration (from PAM).*

**if** Edge-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 6

**else if** Node-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 8

**end if**

Calculate $\sigma^2$ through Theorem 6

▷ *Perturbed Contractive Message Passing (from CAM).*

**for** $k = 0, \ldots, K-1$ **do**

  $\boldsymbol{X}^{(k+1)} \leftarrow C_{\mathsf{L}}(\alpha_1 \hat{A} \boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})) + \beta \boldsymbol{X}^{(0)}$

  ▷ Contractive graph layer: compute node embeddings.

  $\boldsymbol{X}^{(k+1)} \leftarrow \boldsymbol{X}^{(k+1)} + \mathcal{N}(\mu, (\Delta(\mathsf{CGL}))^2 \sigma^2)$

  ▷ DP Perturbation.

  $\boldsymbol{X}^{(k+1)} \leftarrow \Pi_{\mathcal{K}}(\boldsymbol{X}^{(k)})$  ▷ Projection with norm 1.

**end for**

**Return**: $\mathbf{X}^{(K)}$

# Core Algorithm

- Step 1. Calculate sensitivity;
- Step 2. Contractive message passing;
- Step 3. Add appropriate DP noise;
- Step 4. Calculate projection.

```
▷ Calculate the Required Noise Calibration (from PAM).
if Edge-level privacy then
    Calculate Δ(CGL) through Equation 6
else if Node-level privacy then
    Calculate Δ(CGL) through Equation 8
end if
Calculate σ^2 through Theorem 6

▷ Perturbed Contractive Message Passing (from CAM).
for k = 0, ..., K − 1 do
    X^(k+1) ← C_L(α_1 Â X^(k) + α_2 Mean(X^(k))) + β X^(0)
    ▷ Contractive graph layer: compute node embeddings.
    X^(k+1) ← X^(k+1) + N(μ, (Δ(CGL))^2 σ^2)
                                        ▷ DP Perturbation
    X^(k+1) ← Π_K(X^(k))               ▷ Projection with norm 1.
end for

Return: X^(K)
```

# Core Algorithm

- Step 1. Calculate sensitivity;
- Step 2. Contractive message passing;
- Step 3. Add appropriate DP noise;
- Step 4. Calculate projection.

- Repeat Steps 1-4 for all layers.

- Return the final node representation.

▷ *Calculate the Required Noise Calibration (from PAM).*
**if** Edge-level privacy **then**
  Calculate $\Delta(\mathsf{CGL})$ through Equation 6
**else if** Node-level privacy **then**
  Calculate $\Delta(\mathsf{CGL})$ through Equation 8
**end if**
Calculate $\sigma^2$ through Theorem 6

▷ *Perturbed Contractive Message Passing (from CAM).*
**for** $k = 0, \ldots, K-1$ **do**
  $\boldsymbol{X}^{(k+1)} \leftarrow C_{\mathsf{L}}(\alpha_1 \hat{A} \boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})) + \beta \boldsymbol{X}^{(0)}$
  ▷ Contractive graph layer: compute node embeddings.
  $\boldsymbol{X}^{(k+1)} \leftarrow \boldsymbol{X}^{(k+1)} + \mathcal{N}(\mu, (\Delta(\mathsf{CGL}))^2 \sigma^2)$
  ▷ DP Perturbation.
  $\boldsymbol{X}^{(k+1)} \leftarrow \Pi_{\mathcal{K}}(\boldsymbol{X}^{(k)})$ ▷ Projection with norm 1.
**end for**

**Return**: $\mathbf{X}^{(K)}$

# Core Algorithm

- Step 1. Calculate Sensitivity;
- Step 2. Calculate cotractive message passing;
- Step 3. Add appropriate DP noise;
- Step 4. Calculate projection.

**Algorithm 1:** Private Multi-hop Aggregation

**Input** : Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with adjacency matrix $\mathbf{A}$; initial normalized features $\check{\mathbf{X}}^{(0)}$; max hop $K$; noise variance $\sigma^2$;

**Output** : Private aggregated node feature matrices $\check{\mathbf{X}}^{(1)}, \ldots, \check{\mathbf{X}}^{(K)}$

```
for k ∈ {1,...,K} do
    X^(k) ← A^T · X̌^(k−1)                    // aggregate
    X̃^(k) ← X^(k) + N(σ²I)                   // perturb
    for v ∈ V do
        X̌_v^(k) ← X̃_v^(k) / ||X̃_v^(k)||_2      // normalize
    end
end
return X̌^(1), ..., X̌^(K)
```

rs.

```
▷ Calculate the Required Noise Calibration (from PAM).
if Edge-level privacy then
    Calculate Δ(CGL) through Equation 6
else if Node-level privacy then
    Calculate Δ(CGL) through Equation 8
end if
Calculate σ² through Theorem 6

▷ Perturbed Contractive Message Passing (from CAM).
for k = 0, ..., K − 1 do
    X^(k+1) ← C_L(α1 Â X^(k) + α2 Mean(X^(k))) + β X^(0)
    ▷ Contractive graph layer: compute node embeddings.
    X^(k+1) ← X^(k+1) + N(μ, (Δ(CGL))² σ²)
                                        ▷ DP Perturbation.
    X^(k+1) ← Π_K(X^(k))                ▷ Projection with norm 1.
end for

Return: X^(K)
```

# Core Theory

**Theorem 1 [DP guarantee for CGL layers].** Let ***G*** be a graph and *K* be the number of contractive graph layers in CARIBOU. Let $C_L < 1$ be Lipschitz constant. Then, the *K*-hop message passing of CARIBOU satisfies:

$$\left(\frac{\alpha}{2}\frac{\Delta^2}{\sigma^2}\min\left\{K, \frac{1-C_L^K}{1+C_L^K}\frac{1+C_L}{1-C_L}\right\} + \frac{\log(1/\delta)}{\alpha-1}, \delta\right)\text{-}DP.$$

# Core Theory

**Theorem 1 [DP guarantee for CGL layers].** Let ***G*** be a graph and *K* be the number of contractive graph layers in CARIBOU. Let $C_L < 1$ be Lipschitz constant. Then, the *K*-hop message passing of CARIBOU satisfies:

$$\left(\frac{\alpha}{2}\frac{\Delta^2}{\sigma^2}\min\left\{K, \frac{1-C_L^K}{1+C_L^K}\frac{1+C_L}{1-C_L}\right\} + \frac{\log(1/\delta)}{\alpha-1}, \delta\right)\text{-}DP.$$

# Experiments

## Accuracy: Edge-DP over the Cora dataset.

| $\epsilon$ | 1 | 2 | 4 | 8 | 16 | 32 |
|---|---|---|---|---|---|---|
| **CARIBOU** | **85%** | **87%** | **87%** | **88%** | **89%** | **89%** |
| GAP [1] | 77% | 78% | 77% | 79% | 82% | 83% |
| DPDGC [2] | 76% | 78% | 75% | 76% | 78% | 80% |
| PertGraph [3] | 60% | 60% | 63% | 76% | 85% | 85% |

[1] S. Sajadmanesh, A. S. Shamsabadi, A. Bellet, and D. Gatica-Perez, “Gap: Differentially private graph neural networks with aggregation perturbation,” in USENIX Security 2023.
[2] E. Chien, W.-N. Chen, C. Pan, P. Li, A. Ozgur, and O. Milenkovic, “Differentially private decoupled graph convolutions for multigranular topology protection,” in Advances in Neural Information Processing Systems,vol. 36, 2023.
[3] A. Kolluri, T. Baluta, B. Hooi, and P. Saxena, “Lpgnet: Link private graph networks for node classification,” in ACM SIGSAC Conference on Computer and Communications Security, CCS, 2022, pp. 1813–1827.

# Experiments

Accuracy: Node-DP over the Cora dataset.

| $\epsilon$ | 1 | 2 | 4 | 8 | 16 | 32 |
|---|---|---|---|---|---|---|
| **CARIBOU** | **81%** | **83%** | **86%** | **87%** | **88%** | **88%** |
| GAP | 34% | 32% | 32% | 44% | 56% | 64% |
| DPDGC | 34% | 34% | 33% | 32% | 28% | 30% |
| PertGraph | 19% | 20% | 22% | 26% | 28% | 30% |

# Conclusion

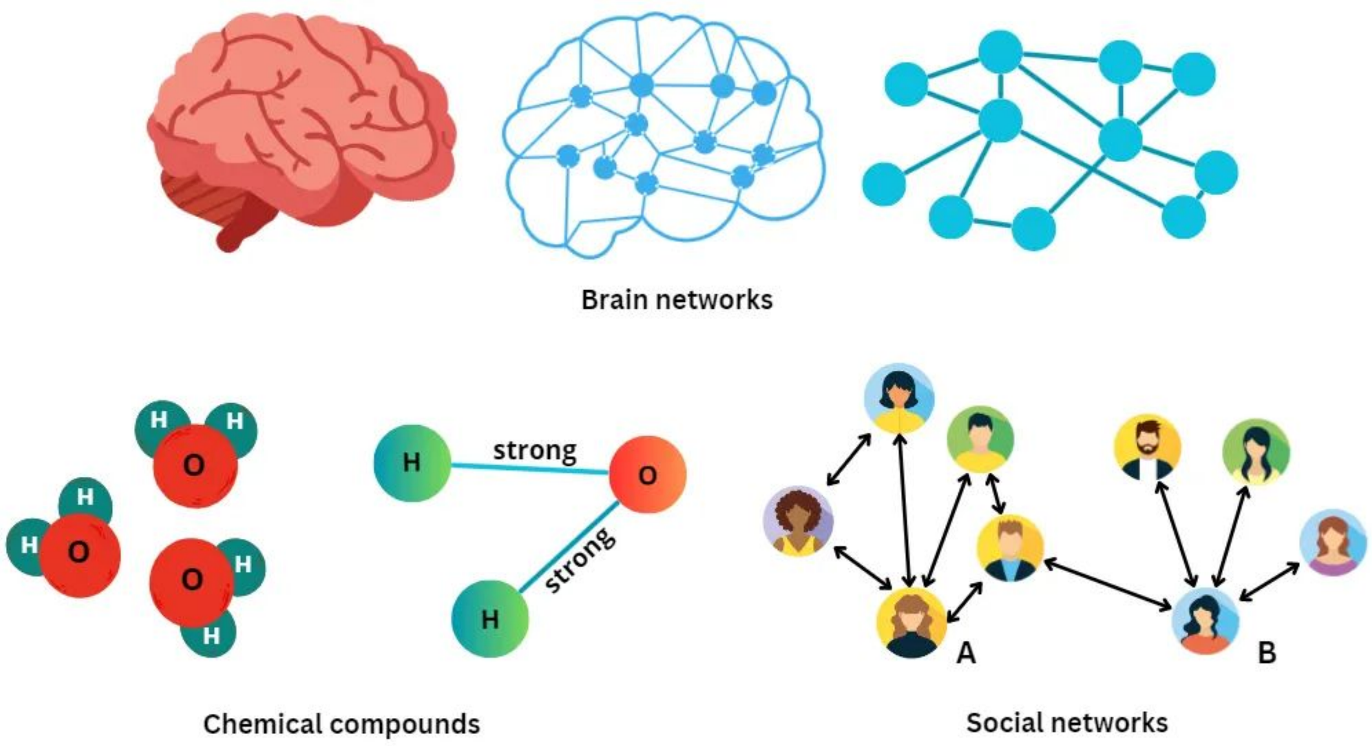
- A novel privacy analysis for GNNs that leverages the contractiveness of message-passing operations to achieve convergent privacy costs.
- The design of perturbed CGL and a practical differentially private GNN framework – CARIBOU.

**Full-version Paper**

**Code**

Artifact Evaluated
NDSS SYMPOSIUM
Available
Functional
Reproduced

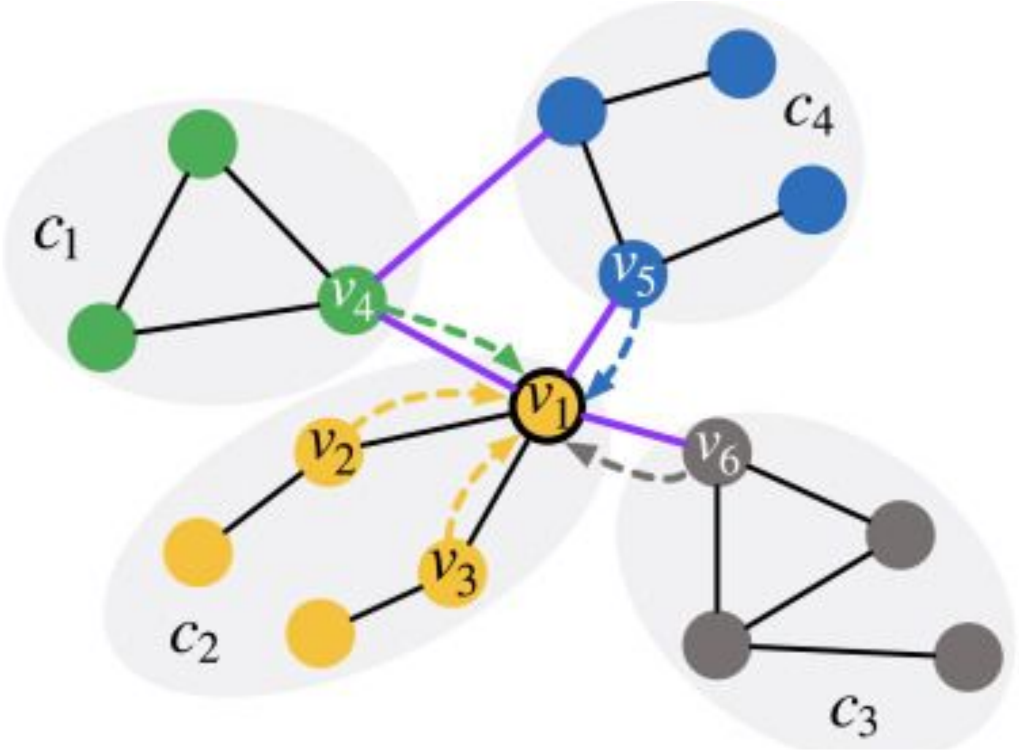
# Graphs and Graph Neural Networks




- Graph Neural Networks (GNNs) are designed for structural data.
  - Graph $G = (V, E)$
- Examples: GCN [1]

[1] Semi-supervised classification with graph convolutional networks. ICLR, 2017.

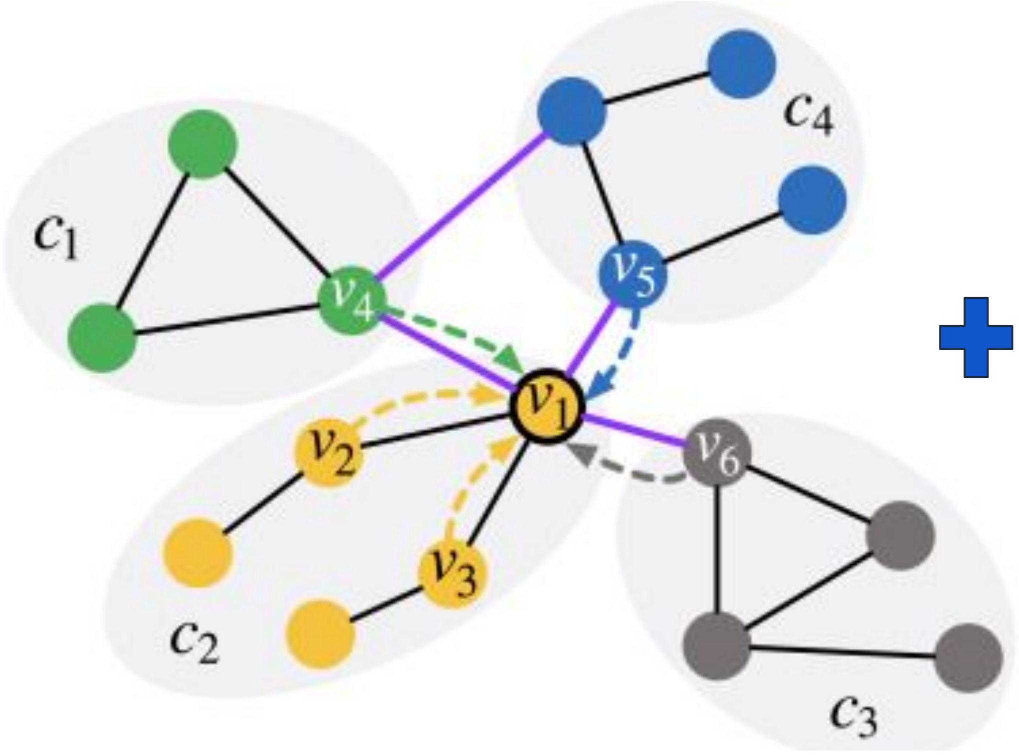
# Message-Passing GNNs

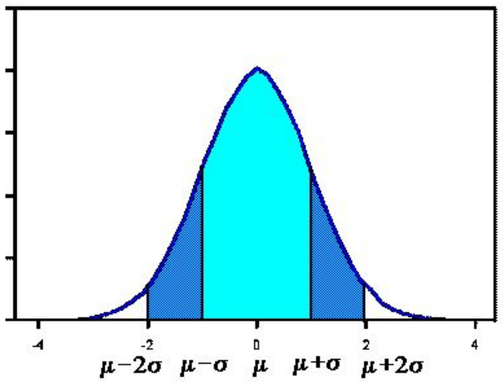



- Message-passing paradigm: aggregating information from their neighbors.
  - $v_1$: aggregated from neighbors.
- Aggregation layer by layer.

[2] Enhancing Graph Neural Networks by a High-quality Aggregation of Beneficial Information. Neural Networks, 2021.

# Perturbed Message-Passing GNNs

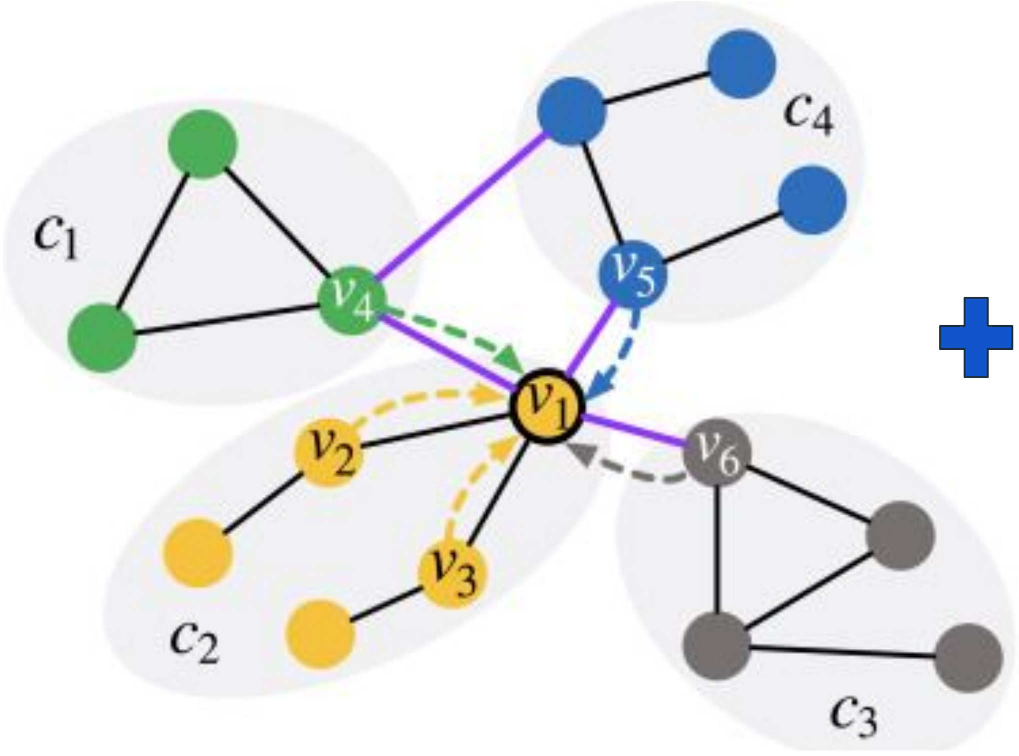


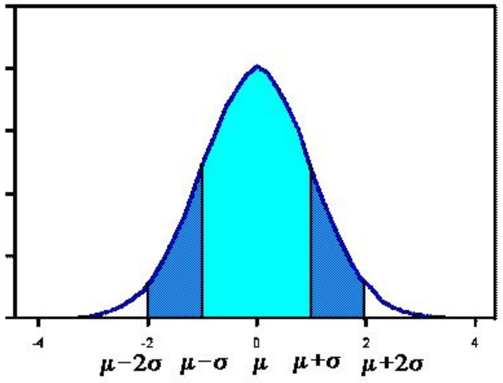




Add Gaussian noise.


Output node representations are similar.

# Perturbed Message-Passing GNNs




Add Gaussian noise.


Output node representations are similar.

# Differential Privacy (DP) for Graphs

- Definition 1. Two datasets $D$, $D'$ are adjacent if they differ by only one data instance. A random mechanism $M$ is $(\epsilon, \delta)$-differentially private if for all adjacent datasets $D$, $D'$ and for all events $S$ in the output space of $M$, we have

$$\Pr(M(\mathbf{D}) \in S) \le e^{\epsilon} \Pr(M(D') \in S) + \delta.$$

# Differential Privacy (DP) for Graphs

- Definition 1. Two datasets ***D***, ***D'*** are adjacent if they differ by only one data instance. A random mechanism ***M*** is ($\boldsymbol{\epsilon}$, $\boldsymbol{\delta}$)-differentially private if for all adjacent datasets ***D***, ***D'*** and for all events *S* in the output space of ***M***, we have

$$\Pr(\boldsymbol{M}(\mathbf{D}) \in \boldsymbol{S}) \leq e^{\epsilon} \Pr(\boldsymbol{M}(\boldsymbol{D'}) \in \boldsymbol{S}) + \boldsymbol{\delta}.$$

- Edge-level neighboring & node-level neighboring.

# Differential Privacy (DP) for Graphs

- Definition 1. Two datasets ***D***, ***D'*** are adjacent if they differ by only one data instance. A random mechanism ***M*** is (***ϵ***, ***δ***)-differentially private if for all adjacent datasets ***D***, ***D'*** and for all events *S* in the output space of ***M***, we have

$$\Pr(\boldsymbol{M}(\mathbf{D}) \in \boldsymbol{S}) \leq e^{\epsilon} \Pr(\boldsymbol{M}(\boldsymbol{D}') \in \boldsymbol{S}) + \boldsymbol{\delta}.$$

- Edge-level neighboring & node-level neighboring.

- **Formulation** on perturbed message passing:

$$\boldsymbol{X}^{(k+1)} = \Pi(\mathrm{MP}_{\boldsymbol{G}}(\boldsymbol{X}^{(k)}) + \boldsymbol{Z}^{(k)})$$

Node representation/embeddings; $\boldsymbol{X}^{(0)}$: input feature matrix.

Gaussian noise ~ $N(0, \sigma^2)$.

# State-of-the-Arts

TABLE I: Comparison between Private GNNs. EDP and NDP summarizes the results of private GNNs in Table III.

| Framework | Mechanism | Complexity per Layer | Calibrated Noise ($\sigma$) | EDP Utility | NDP Utility |
|---|---|---|---|---|---|
| PertGraph [45, 26] | Graph perturbation | $O(\lvert\boldsymbol{V}\rvert^2)$ | $\propto 1$ | ★★☆☆☆ | ★☆☆☆☆ |
| DPDGC [39] | Decoupled graph with perturbation | $O(\lvert\boldsymbol{E}\rvert)$ | $\propto \sqrt{K}$ | ★★★☆☆ | ★★★½☆ |
| GAP [46] | Perturbed message passing | $O(\lvert\boldsymbol{E}\rvert)$ | $\propto \sqrt{K}$ | ★★★★☆ | ★★★½☆ |
| CARIBOU | Perturbed message passing | $O(\lvert\boldsymbol{E}\rvert)$ | $\propto \sqrt{\min(K, \frac{1-C_{\mathsf{L}}^K}{1+C_{\mathsf{L}}^K}\frac{1+C_{\mathsf{L}}}{1-C_{\mathsf{L}}})}$ | ★★★★★ | ★★★★★ |

- Share a critical limitation: the **privacy loss grows linearly with the number of layers K** or graph hops.
  - Require large amounts of noise to maintain a reasonable level of privacy guarantee; This, in turn, degrading model utility severely.

# Motivating Scenarios & Challenges

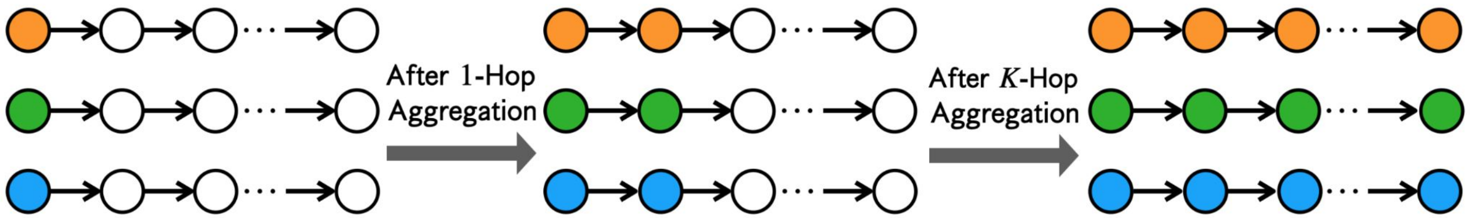
- Multi-layer GNNs: capture complex relations and analyze graphs with long-range interactions [3,4].
  - Example: ⬆ accuracy from **72.5%** to **88.2%** [4].

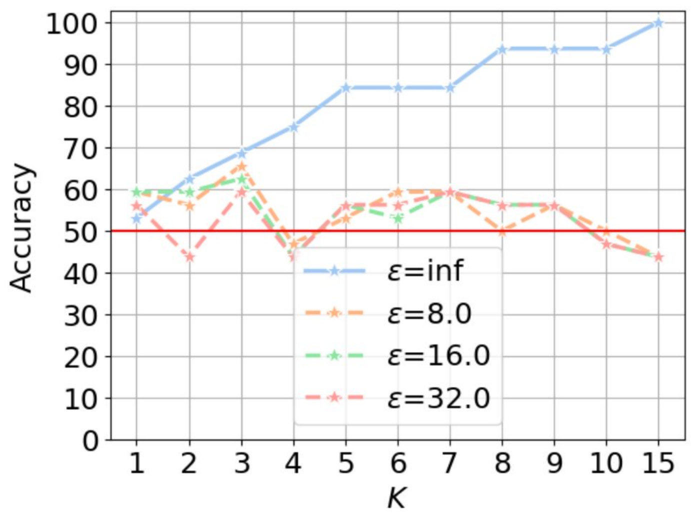
[3] How powerful are k-hop message passing graph neural networks. NuerIPS, 2022.
[4] Training graph neural networks with 1000 layers. ICML, 2021

# Motivating Scenarios & Challenges

- Multi-layer GNNs: capture complex relations and analyze graphs with long-range interactions [3,4].
  - Example: ⬆ accuracy from **72.5%** to **88.2%** [4].

- Challenges: larger *K* leads to larger privacy parameter $\epsilon$, a.k.a weak privacy guarantee.




[3] How powerful are k-hop message passing graph neural networks. NuerIPS, 2022.
[4] Training graph neural networks with 1000 layers. ICML, 2021.

# Research Question

- “Over-smoothing” phenomenon: node representations become increasingly homogeneous as network depth increases and consequently making membership inference more challenging.

# Research Question

- "Over-smoothing" phenomenon: node representations become increasingly homogeneous as network depth increases and consequently making membership inference more challenging.

⬇ Motivate

- Can we achieve differentially private graph learning with a **convergent (bounded)** privacy budget, thereby improving the privacy-utility trade-off for deeper GNNs?
  - Not linearly increase with $K$.

# Core Idea for Convergent Privacy

- Insight: leverage the inherent **privacy amplification** that occurs in multi-layer GNNs through contractiveness.
  - Motivated by DP-GD: **converge** to a finite value with **arbitrarily** many iterations.

# Core Idea for Convergent Privacy

- Insight: leverage the inherent **privacy amplification** that occurs in multi-layer GNNs through contractiveness.
  - Motivated by DP-GD: **converge** to a finite value with **arbitrarily** many iterations.

- When perturbed message passing is contractive, the distance between GNNs trained on neighboring datasets shrinks at each step.
  - *Translate* the advanced privacy analysis techniques from DP-GD to GNNs.

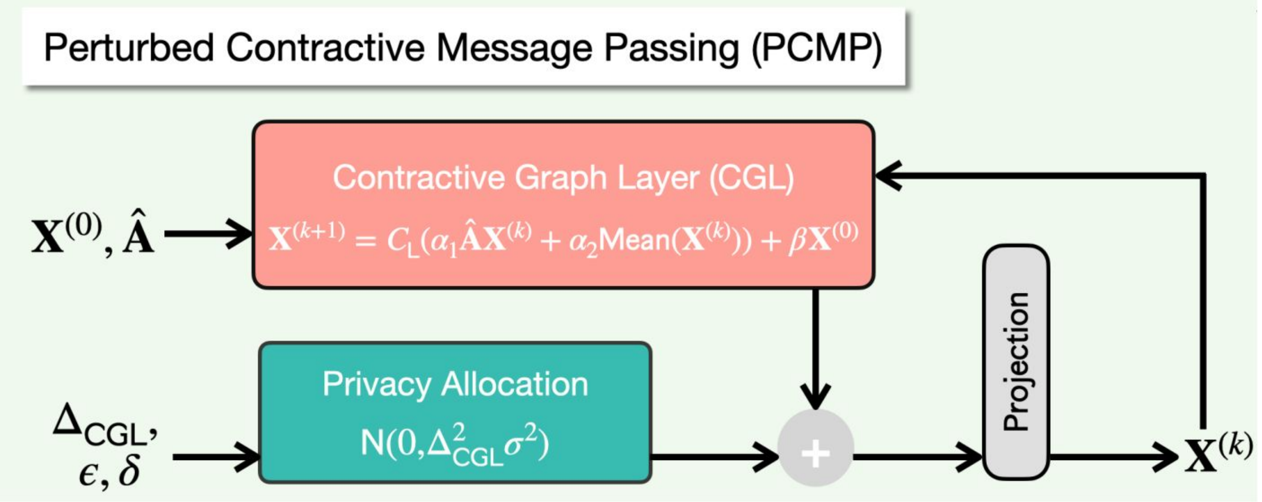
# Core Idea for Convergent Privacy

- Consequently, the influence of individual data points diminishes, leading to the amplified privacy rooted from “over-smoothing”.
  - Accordingly, remove the over-estimated privacy loss;
  - Derive a much tighter bound for finally released GNN model.

# Core Idea for Convergent Privacy

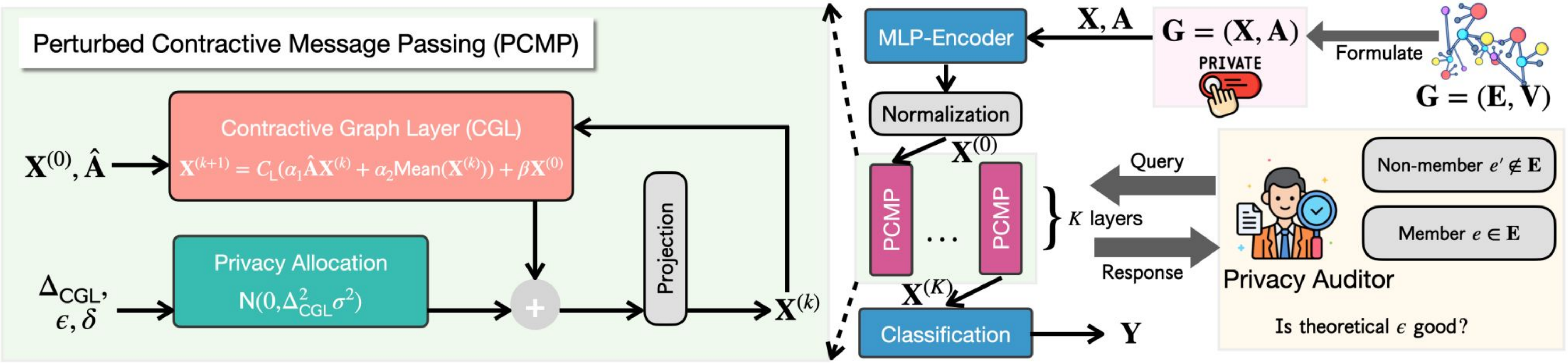
- Consequently, the influence of individual data points diminishes, leading to the amplified privacy rooted from “over-smoothing”.
  - Accordingly, remove the over-estimated privacy loss;
  - Derive a much tighter bound for finally released GNN model.

- Two critical conditions:
  - Contractive message passing;
  - Release only $X^{(K)}$.



# Overview of CARIBOU

- **C**ontractive **A**ggregation **M**odule (CAM).
- **P**rivacy **A**llocation **M**odule (PAM).
- **P**rivacy Au**d**iting **M**odule (PDM).



# Core Algorithm

- Step 1. Calculate Sensitivity;

▷ *Calculate the Required Noise Calibration (from PAM).*

**if** Edge-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 6

**else if** Node-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 8

**end if**

Calculate $\sigma^2$ through Theorem 6

▷ *Perturbed Contractive Message Passing (from CAM).*

**for** $k = 0, \ldots, K-1$ **do**

  $\boldsymbol{X}^{(k+1)} \leftarrow C_{\mathsf{L}}(\alpha_1 \hat{A}\boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})) + \beta \boldsymbol{X}^{(0)}$

  ▷ Contractive graph layer: compute node embeddings.

  $\boldsymbol{X}^{(k+1)} \leftarrow \boldsymbol{X}^{(k+1)} + \mathcal{N}(\mu, (\Delta(\mathsf{CGL}))^2\sigma^2)$

  ▷ DP Perturbation.

  $\boldsymbol{X}^{(k+1)} \leftarrow \Pi_{\mathcal{K}}(\boldsymbol{X}^{(k)})$ ▷ Projection with norm 1.

**end for**

**Return**: $\mathbf{X}^{(K)}$

# Core Algorithm

- Step 1. Calculate Sensitivity;
- Step 2. Calculate cotractive message passing;

▷ *Calculate the Required Noise Calibration (from PAM).*

**if** Edge-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 6

**else if** Node-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 8

**end if**

Calculate $\sigma^2$ through Theorem 6

▷ *Perturbed Contractive Message Passing (from CAM).*

**for** $k = 0, \ldots, K-1$ **do**

  $\boldsymbol{X}^{(k+1)} \leftarrow C_{\mathsf{L}}(\alpha_1 \hat{A} \boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})) + \beta \boldsymbol{X}^{(0)}$

  ▷ Contractive graph layer: compute node embeddings.

  $\boldsymbol{X}^{(k+1)} \leftarrow \boldsymbol{X}^{(k+1)} + \mathcal{N}(\mu, (\Delta(\mathsf{CGL}))^2 \sigma^2)$

  ▷ DP Perturbation.

  $\boldsymbol{X}^{(k+1)} \leftarrow \Pi_{\mathcal{K}}(\boldsymbol{X}^{(k)})$ ▷ Projection with norm 1.

**end for**

**Return**: $\mathbf{X}^{(K)}$

# Core Algorithm

- Step 1. Calculate Sensitivity;
- Step 2. Calculate cotractive message passing;
- Step 3. Add appropriate DP noise;

```
▷ Calculate the Required Noise Calibration (from PAM).
if Edge-level privacy then
    Calculate Δ(CGL) through Equation 6
else if Node-level privacy then
    Calculate Δ(CGL) through Equation 8
end if
Calculate σ^2 through Theorem 6

▷ Perturbed Contractive Message Passing (from CAM).
for k = 0, ..., K − 1 do
    X^(k+1) ← C_L(α_1 Â X^(k) + α_2 Mean(X^(k))) + β X^(0)
    ▷ Contractive graph layer: compute node embeddings.
    X^(k+1) ← X^(k+1) + N(μ, (Δ(CGL))^2 σ^2)
                                        ▷ DP Perturbation.
    X^(k+1) ← Π_K(X^(k))                ▷ Projection with norm 1.
end for

Return: X^(K)
```

# Core Algorithm

- Step 1. Calculate Sensitivity;
- Step 2. Calculate cotractive message passing;
- Step 3. Add appropriate DP noise;
- Step 4. Calculate projection.

▷ *Calculate the Required Noise Calibration (from PAM).*

**if** Edge-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 6

**else if** Node-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 8

**end if**

Calculate $\sigma^2$ through Theorem 6

▷ *Perturbed Contractive Message Passing (from CAM).*

**for** $k = 0, \ldots, K-1$ **do**

  $\boldsymbol{X}^{(k+1)} \leftarrow C_{\mathsf{L}}(\alpha_1 \hat{A} \boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})) + \beta \boldsymbol{X}^{(0)}$

  ▷ Contractive graph layer: compute node embeddings.

  $\boldsymbol{X}^{(k+1)} \leftarrow \boldsymbol{X}^{(k+1)} + \mathcal{N}(\mu, (\Delta(\mathsf{CGL}))^2 \sigma^2)$

  ▷ DP Perturbation

  $\boldsymbol{X}^{(k+1)} \leftarrow \Pi_{\mathcal{K}}(\boldsymbol{X}^{(k)})$ ▷ Projection with norm 1.

**end for**

**Return**: $\mathbf{X}^{(K)}$

# Core Algorithm

- Step 1. Calculate Sensitivity;
- Step 2. Calculate cotractive message passing;
- Step 3. Add appropriate DP noise;
- Step 4. Calculate projection.

- Repeat Steps 1-4 for all layers.

- Return the final node representation.

▷ *Calculate the Required Noise Calibration (from PAM).*

**if** Edge-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 6

**else if** Node-level privacy **then**

  Calculate $\Delta(\mathsf{CGL})$ through Equation 8

**end if**

Calculate $\sigma^2$ through Theorem 6

▷ *Perturbed Contractive Message Passing (from CAM).*

**for** $k = 0, \ldots, K-1$ **do**

  $\boldsymbol{X}^{(k+1)} \leftarrow C_{\mathsf{L}}(\alpha_1 \hat{A} \boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})) + \beta \boldsymbol{X}^{(0)}$

  ▷ Contractive graph layer: compute node embeddings.

  $\boldsymbol{X}^{(k+1)} \leftarrow \boldsymbol{X}^{(k+1)} + \mathcal{N}(\mu, (\Delta(\mathsf{CGL}))^2 \sigma^2)$

  ▷ DP Perturbation.

  $\boldsymbol{X}^{(k+1)} \leftarrow \Pi_{\mathcal{K}}(\boldsymbol{X}^{(k)})$ ▷ Projection with norm 1.

**end for**

**Return**: $\mathbf{X}^{(K)}$

# Core Theory

Theorem 1 [DP guarantee for CGL layers]. Let ***G*** be a graph and $K$ be the number of contractive graph layers in CARIBOU. Let $C_L < 1$ be Lipschitz constant. Then, the $K$-hop message passing of CARIBOU satisfies:

# Core Theory

Theorem 1 [DP guarantee for CGL layers]. Let ***G*** be a graph and *K* be the number of contractive graph layers in CARIBOU. Let $C_L < 1$ be Lipschitz constant. Then, the *K*-hop message passing of CARIBOU satisfies:

$$\left(\frac{\alpha}{2}\frac{\Delta^2}{\sigma^2}\min\left\{K, \frac{1-C_L^K}{1+C_L^K}\frac{1+C_L}{1-C_L}\right\} + \frac{\log(1/\delta)}{\alpha-1}, \delta\right)\text{-}DP.$$

UCI Samueli School of Engineering

# Core Theory

Theorem 1 [DP guarantee for CGL layers]. Let ***G*** be a graph and *K* be the number of contractive graph layers in CARIBOU. Let $C_L < 1$ be Lipschitz constant. Then, the *K*-hop message passing of CARIBOU satisfies:

$$\left(\frac{\alpha}{2}\frac{\Delta^2}{\sigma^2}\min\left\{K, \frac{1-C_L^K}{1+C_L^K}\frac{1+C_L}{1-C_L}\right\} + \frac{\log(1/\delta)}{\alpha-1}, \delta\right)\text{-}DP.$$

# Experiments

Accuracy: EDP over the Cora dataset.

| $\epsilon$ | 1 | 2 | 4 | 8 | 16 | 32 |
|---|---|---|---|---|---|---|
| CARIBOU | 85% | 87% | 87% | 88% | 89% | 89% |
| GAP | 76% | 78% | 75% | 76% | 78% | 80% |

Accuracy: NDP over the Cora dataset.

| $\epsilon$ | 1 | 2 | 4 | 8 | 16 | 32 |
|---|---|---|---|---|---|---|
| CARIBOU | 81% | 83% | 86% | 87% | 88% | 88% |
| GAP | 34% | 32% | 32% | 44% | 56% | 64% |

# Conclusion

- A novel privacy analysis for GNNs that leverages the contractiveness of message-passing operations to achieve convergent privacy costs.
- The design of perturbed CGL and a practical differentially private GNN framework – CARIBOU.

**Full-version Paper**

**Code**

Artifact Evaluated
NDSS SYMPOSIUM
Available
Functional
Reproduced