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

Yu Zheng[*], Chenang Li[*], Zhou Li[*§] and Qingsong Wang[†§]
[*] University of California, Irvine, Email: {yu.zheng, chenangl, zhou.li}@uci.edu
[†] University of California, San Diego, Email: qiw072@ucsd.edu

***Abstract*—Differential privacy (DP) has been integrated into graph neural networks (GNNs) to protect sensitive structural information, e.g., edges, nodes, and associated features across various applications. A prominent approach is to perturb the message-passing process, which forms the core of most GNN architectures. However, existing methods typically incur a privacy cost that grows linearly with the number of layers (e.g., GAP published in Usenix Security'23), ultimately requiring excessive noise to maintain a reasonable privacy level. This limitation becomes particularly problematic when multi-layer GNNs, which have shown better performance than one-layer GNN, are used to process graph data with sensitive information.**

In this paper, we theoretically establish that the privacy budget converges with respect to the number of layers by applying privacy amplification techniques to the message-passing process, exploiting the contractive properties inherent to standard GNN operations. Motivated by this analysis, we propose a simple yet effective *Contractive Graph Layer (CGL)* that ensures the contractiveness required for theoretical guarantees while preserving model utility. Our framework, CARIBOU[1], supports both training and inference, equipped with a contractive aggregation module, a privacy allocation module, and a privacy auditing module. Experimental evaluations demonstrate that CARIBOU significantly improves the privacy-utility trade-off and achieves superior performance in privacy auditing tasks.

I. Introduction

Graph neural networks (GNNs) [1], designed for operating over structural data, have achieved success in various domains, including social networks [2] and recommendation systems [3]. At their core, many GNN architectures are built upon the *message-passing paradigm*, where node representations are iteratively updated by aggregating information from their neighbors. However, graph structures often encode sensitive information about relationships and attributes. As a result, GNNs are vulnerable to privacy attacks, including membership inference [4–6] and attribute inference [7, 8]. These vulnerabilities highlight the urgent need for robust privacy protection mechanisms in graph learning.

[§]Co-corresponding authors.

[1]The full version of this paper, including the complete set of proofs, experiments, and artificial evaluation, can be found in https://arxiv.org/pdf/2506.22727. Our code is available at https://doi.org/10.5281/zenodo.17539660.

Network and Distributed System Security (NDSS) Symposium 2026
24-28 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.240255
www.ndss-symposium.org

Differential privacy (DP) [9–11] has emerged as a foundational framework to provide formal guarantees against data leakage, with widespread applications in machine learning [12, 13], synthetic data generation [14], and beyond. In the context of GNNs, recent works [15, 16] have advanced privacy protection through edge-level DP (EDP) [17] and node-level DP (NDP) [18, 19] guarantees. The primary approach employs *perturbed message passing*, which injects calibrated Gaussian or Laplace noise into aggregation layers to protect the edge or node memberships in a training graph. While these approaches have provided formal privacy guarantees, they share a critical limitation: *the privacy loss grows linearly with the number of layers K* or graph hops. This, in turn, severely degrades model utility.

Recent advances have shown that multi-layer GNNs, especially deeper GCNs, are essential in capturing complex relationships [20] and analyzing graphs with long-range interactions [21, 22]. For instance, in large biological networks, long-range dependencies influence protein functions, requiring more than 10 hops of message passing [23]. In social networks, privacy-sensitive relationships propagate through multi-hop neighborhoods [24, 25]. As reported in [23], increasing the network depth leads to a substantial improvement in accuracy from 72.5% to 88.2%. However, the aforementioned linear dependence on K is particularly challenging for multi-layer GNNs as larger K leads to larger privacy parameter ϵ, *a.k.a* weak privacy guarantee.

Interestingly, empirical studies [4] have shown that membership inference attacks are not particularly more successful against multi-layer GNNs, suggesting that the linear dependency of privacy cost on network depth might be an overestimation. This observation aligns with the phenomenon known as "over-smoothing" [26] in GNNs, where node representations become increasingly homogeneous as network depth increases and consequently making membership inference more challenging. This homogenization effect might actually provide inherent privacy benefits due to the contractive nature of GNN aggregation operations. This observation motivates our central research question:

Can we achieve differentially private graph learning with a ***convergent*** *(bounded) privacy budget, thereby improving the privacy-utility trade-off for deeper GNNs?*

In this work, we answer this question affirmatively. Prior perturbed message-passing mechanisms [17–19] assume that privacy loss grows linearly with depth, yet empirical results

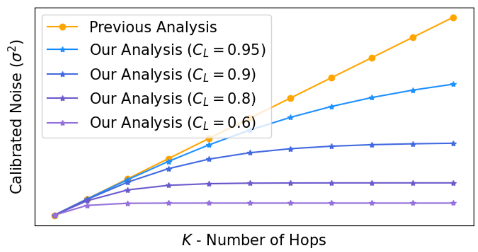


Fig. 1: Comparing Calibrated Noise for Perturbed Message Passing. Previous analysis requires $\sigma^2 \propto O(K)$, and our analysis demands $\sigma^2 \propto O\Big((1-C_{\mathsf{L}}^K)(1+C_{\mathsf{L}})/((1+C_{\mathsf{L}}^K)(1-C_{\mathsf{L}}))\Big)$, where C_{L} is Lipchitz constant. With sensitivity constrained to norm 1, the signal-to-noise ratio is markedly low as K increases, severely impacting utility.

show that deeper GNNs can be less vulnerable to membership inference. We attribute this phenomenon to the contractive nature of common aggregation operators. In theory, we analyze this contractive property underlying over-smoothing, which leads to bounded sensitivity, so the privacy budget converges with the number of layers K (see Figure 1) instead of growing linearly. This motivates CARIBOU, a privacy-preserving GNN framework that enforces *contractiveness* to mirror real-world GNN behaviors while achieving *convergent privacy budget* through privacy amplification.

A. Overview of Convergent Privacy Analysis

For a perturbed message-passing GNN with K layers, the standard approach analyzes the privacy loss at each step and then applies the DP composition theorem to aggregate the total privacy cost. This approach is common in existing privacy analyses of perturbed message-passing GNNs [15, 16, 27], resulting in a privacy budget that scales linearly with K, specifically $\epsilon = O(K/\sigma^2) + O(\sqrt{K/\sigma})$. Thus, as K increases, the amount of injected noise σ must also grow, leading to degraded utility particularly when a small ϵ is desired; see Section III for a motivating empirical study.

Inspired by recent advances in privacy amplification [28, 29] through hidden states and contractive iterative processes, we observe that a similar amplification effect can be exploited in GNNs. Here, *contractiveness* refers to the property that the distance between two inputs is reduced after applying the operation, implying reduced distinguishability of outputs from a privacy perspective. The potential privacy amplification in GNNs arises from the following two observations: (1) GNNs typically do not expose intermediate node embeddings during training or inference, focusing only on the final node representations. (2) Standard message-passing operations, such as those used in Graph Convolutional Networks (GCNs) [1] (the dominant model in practice and in empirical studies [4]), are inherently contractive, a property that also underlies the over-smoothing phenomenon [26]. We theoretically validate this insight by showing that the privacy loss of a K-layer perturbed message-passing process with contractive layers satisfies a *convergent privacy budget*. Specifically, instead of growing linearly with K, we show that the privacy budget follows a convergent form: $\epsilon = O\left(\frac{(1-C_{\mathsf{L}}^K)(1+C_{\mathsf{L}})}{(1+C_{\mathsf{L}}^K)(1-C_{\mathsf{L}})}\right)$, where C_{L} is the Lipschitz constant of the message-passing operator (see Theorem 3 for details). Our improved privacy analysis is achieved by recasting the multi-layer perturbed GNN as a *Contractive Noisy Iteration (CNI)* process [29] and applying the privacy convergence results established for CNIs.

To leverage this analysis in practice, we design a simple yet effective *Contractive Graph Layer (CGL)* that enforces the contractiveness required for our theoretical guarantees while maintaining model expressivity. The CGL layer builds upon standard GCN-style aggregation, augmented with residual connections [30] and mean aggregation normalization [31], ensuring expressiveness across many layers without exposing sensitive edge information.

We quantitatively characterize the privacy guarantees of CGL by carefully bounding the Lipchitz constant of the perturbed message-passing operation (Proposition 1) and the sensitivity of the perturbed message-passing with respect to both edge-level privacy (Theorem 4) and node-level privacy (Theorem 5). Together, these results allow us to explicitly quantify the privacy budget of the CGL layer using our general theory, culminating in the final privacy guarantee stated in Theorem 6.

B. CARIBOU*: Framework and Evaluation*

Building on perturbed CGL, we realize a private framework CARIBOU for GNN inference and training. CARIBOU includes contractive aggregation module, privacy allocation module, and privacy auditing module. Together, our design enables achieving convergent privacy guarantees while maintaining strong GNN performance across graphs with varying interaction ranges.

To evaluate CARIBOU, we conduct extensive experiments over nine graph datasets, including commonly-used real-world datasets and synthetic chain-structured datasets for developing configurable interaction ranges. The experimental results demonstrate that CARIBOU improves non-trivial utility over standard graph and chain-structured datasets. Compared with several SOTA baselines, CARIBOU's EDP and NDP show significant utility improvements, especially in high privacy regimes, and reasonable computational overhead. Table I presents a comprehensive comparison, which is explained in Section VIII. Ablation studies are provided to understand the relation between privacy-utility hops and various ranges of graph, and the choice of hyper-parameters of CGL. In addition to privacy verification, we perform auditing experiments based on two membership inference attacks [4, 34], demonstrating CARIBOU's robustness.

Contribution. In terms of our new insights (Section III), our contribution includes:

1) A novel privacy analysis for GNNs that leverages the contractiveness of message-passing operations to achieve convergent privacy costs, even for deep networks; (Section V)
2) The design of perturbed CGL and a practical differentially private GNN framework – CARIBOU with provable

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 (σ)	EDP Utility	NDP Utility
PertGraph [32, 17]	Graph perturbation	$O(\lvert\boldsymbol{V}\rvert^2)$	$\propto 1$	★★☆☆☆	★☆☆☆☆
DPDGC [27]	Decoupled graph with perturbation	$O(\lvert\boldsymbol{E}\rvert)$	$\propto \sqrt{K}$	★★★☆☆	★★★½☆
GAP [33]	Perturbed message passing	$O(\lvert\boldsymbol{E}\rvert)$	$\propto \sqrt{K}$	★★★★☆	★★★½☆
CARIBOU	Perturbed message passing	$O(\lvert\boldsymbol{E}\rvert)$	$\propto \sqrt{\min(K, \frac{1-C_{\mathrm{L}}^K}{1+C_{\mathrm{L}}^K}\frac{1+C_{\mathrm{L}}}{1-C_{\mathrm{L}}})}$	★★★★★	★★★★★

privacy guarantees and superior utility-privacy tradeoffs; (Section IV)

3) Extensive experimental validation across multiple graph datasets with varying structural properties, demonstrating significant improvements over state-of-the-art private GNN approaches. (Section VI)

II. Preliminary

A. Message Passing Graph Neural Networks

Graph Neural Networks (GNNs) are a class of neural networks that operate on graph-structured data. Most GNNs follow the message-passing paradigm [20], where nodes iteratively aggregate information from their neighbors to update their representations.

1) Message Passing Layers: Let $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ be a graph, where $\boldsymbol{V}$ denotes the set of vertices (or nodes) and $\boldsymbol{E}$ denotes the set of edges. Let $\boldsymbol{X}^{(k)} \in \mathbb{R}^{|\boldsymbol{V}|\times d}$ be the node feature matrix at layer k, where d is the dimension of the node features. Additionally, we use $\boldsymbol{X}_u^{(k)} \in \mathbb{R}^d$ to denote the feature vector of node u at layer k. Each layer of a message passing GNN can be generally written as,

$$\mathrm{MP}_G(\boldsymbol{X}_u^{(k)}) := \sigma\left(\psi\left(\boldsymbol{X}^{(k)}, \oplus_{v\in\mathcal{N}(u)}\phi(\boldsymbol{X}_u^{(k)}, \boldsymbol{X}_v^{(k)})\right)\right), \tag{1}$$

where σ is a non-linear activation function, $\mathcal{N}(u)$ is the set of neighbors of node u, ϕ is a function that computes the message from node v to node u, $\oplus$ represents the aggregation function that processes all messages from the neighbors of node u, and ψ is a function that updates the node feature vector of node u with the aggregated messages. GCN [35] and its variants are common examples of message passing GNNs.

2) Applications of Message Passing GNNs: Message passing GNNs leverage GNN layers to iteratively refine node representations, which are then employed in tasks like node classification [32], link prediction [36], and graph classification [37]. Multi-layer GNNs like deep GNNs [38, 39] are especially suitable to process long-range graphs [21, 22] by capturing dependencies between distant nodes, which is crucial for tasks like molecular property prediction [40], protein interaction modeling [41], and complex node interaction modeling [20].

B. Differential Privacy for GNNs

Definition 1 (Differential Privacy [42])**.** Given a data universe $\mathcal{D}$, two datasets $D, D' \subseteq \mathcal{D}$ are adjacent if they differ by only one data instance. A random mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if for all adjacent datasets D, D' and for all events S in the output space of $\mathcal{M}$, we have $\Pr(\mathcal{M}(D) \in S) \leq e^{\epsilon}\Pr(\mathcal{M}(D') \in S) + \delta$.

Intuitively, DP [42] theoretically quantifies the privacy of a model by measuring the indistinguishability of the outputs of a mechanism $\mathcal{M}$ on two adjacent datasets D and D'. It can be classified into bounded DP and unbounded DP depending on the construction of D', where the former is by replacing a data instance of D and the latter is by addition / removal of a data sample of D. The privacy budget ϵ is smaller representing a stronger privacy guarantee, while δ is a slackness quantity that relaxes the pure DP constraint.

1) Privacy Definition on Graphs: In the context of graph data, the notion of adjacency refers to the graph structure, which can be defined as edge-level adjacency (Definition 2) and node-level adjacency (Definition 3).

Definition 2 (Edge-level adjacency [43])**.** Two graphs $\boldsymbol{G}_1 = \{\boldsymbol{V}_1, \boldsymbol{E}_1\}$ and $\boldsymbol{G}_2 = \{\boldsymbol{V}_2, \boldsymbol{E}_2\}$ are considered as edge-level neighboring if they differ in a single edge (through addition or removal of the edge), *i.e.*, $(\boldsymbol{V}_2 = \boldsymbol{V}_1) \wedge (\neg(\boldsymbol{E}_2 \cap \boldsymbol{E}_1) = e_i)$ where $e_i \in \boldsymbol{E}_1$.

Definition 3 (Node-level adjacency [43])**.** Two graphs $\boldsymbol{G}_1 = \{\boldsymbol{V}_1, \boldsymbol{E}_1\}$ and $\boldsymbol{G}_2 = \{\boldsymbol{V}_2, \boldsymbol{E}_2\}$ are considered as node-level neighboring if they differ in a single node and its incident edges (through addition or removal of the node and its incident edges), *i.e.*, $\neg(\boldsymbol{V}_2 \cap \boldsymbol{V}_1) = \{n_i, \{e_{ij}\}_{\forall j}\}$ where $n_i \in (\boldsymbol{V}_1 \cup \boldsymbol{V}_2)$ and $\{e_{ij}\}_{\forall j}$ connects to n_i.

2) Perturbed Message Passing with DP: To incorporate DP into GNNs, one can add noise to the message passing layer, following the perturbed message passing approach [33]. Given a graph $\boldsymbol{G}$ and message passing function MP, we define a sequence $\{\boldsymbol{X}^{(k)}\}_{k=0}^K$ of node feature matrices by:

$$\boldsymbol{X}^{(k+1)} = \Pi_{\mathcal{K}}(\mathrm{MP}_{\boldsymbol{G}}(\boldsymbol{X}^{(k)}) + \boldsymbol{Z}^{(k)}) \tag{2}$$

where $\boldsymbol{X}^{(0)} = \boldsymbol{X}$ is the input feature matrix, $\boldsymbol{Z}^{(k)} \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise, and $\Pi_{\mathcal{K}}$ projects features back to bounded set $\mathcal{K}$ (typically constraining $\|\boldsymbol{X}_u\|_2 \leq 1$ for each node u). The privacy guarantees of perturbed message passing depend on the sensitivity of the mechanism:

Definition 4 (Sensitivity of Perturbed Message Passing)**.** Let $\mathrm{MP}_{\boldsymbol{G}}$, $\mathrm{MP}_{\boldsymbol{G}'}$ be the perturbed message passing mechanisms applied to neighboring graphs G, G'. Define the sensitivity as:

$$\Delta(\mathrm{MP}) = \max_{G,G'} \max_{\boldsymbol{X}\in\mathcal{K}} \|\mathrm{MP}_{\boldsymbol{G}}(\boldsymbol{X}) - \mathrm{MP}_{\boldsymbol{G}'}(\boldsymbol{X})\|_F \tag{3}$$

where the maximum is taken over all adjacent graphs and all node feature matrices in $\mathcal{K}$.

The sensitivity determines the scale of noise required for privacy guarantees. Lower sensitivity allows for less noise

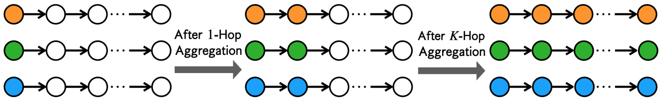


Fig. 2: Message Passing on Chain-structured Dataset

addition while maintaining the same privacy level, directly affecting the utility-privacy trade-off in differentially private GNNs.

3) Privacy Accounting: Privacy accounting process analyzes the total privacy budget for the composition of several (adaptive) private algorithms. A common approach for analyzing the Gaussian mechanism in perturbed message passing is through Rényi differential privacy (RDP) [9] and its composition theorem.

Definition 5 (Rényi differential privacy [9])**.** A randomized algorithm $\mathcal{M}$ is (α, ϵ)-RDP for $\alpha > 1$, $\epsilon > 0$ if for every adjacent dataset X, X', we have $D_\alpha(\mathcal{M}(X)\|\mathcal{M}(X')) \leq \epsilon$, where $D_\alpha(P\|Q)$ is the Rényi divergence of order α between probability distributions P and Q defined as:

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_{x\sim Q}\left[\left(\frac{P(x)}{Q(x)}\right)^\alpha\right]$$

Theorem 1 (Composition of RDP [9])**.** *If $\mathcal{M}_1, \ldots, \mathcal{M}_k$ are randomized algorithms satisfying, respectively, (α, ϵ_1)-RDP, ..., (α, ϵ_k)-RDP, then their composition defined as $(\mathcal{M}_1(S), \ldots, \mathcal{M}_k(S))$ satisfies $(\alpha, \epsilon_1 + \cdots + \epsilon_k)$-RDP.*

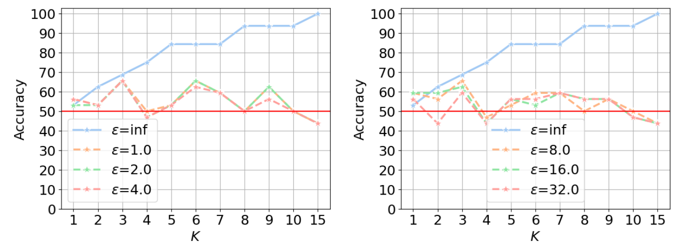
In this work, we present our privacy results in terms of RDP for ease of interpretation, while our underlying analysis employs the tighter f-DP framework. This analysis leverages recent advances in privacy amplification techniques [44, 29] to achieve stronger privacy guarantees. The technical details of our convergent privacy analysis are discussed in §III-B.

III. Private Multi-layer GNNs Initiative

Multi-layer GNNs [38, 39] are vital in tasks like modeling molecular structures, where some properties depend on long range interactions [21, 45, 22] of the nodes [46, 47]. Specifically, they require messages flowing across multiple hops before reaching a target node through stacking multiple layers to exchange information across K-hop neighborhoods. However, ensuring privacy in multi-layer GNNs poses key challenges. In this section, we outline and illustrate these challenges via a motivating case study (Section III-A), ultimately motivating our new design insights (Section III-B) based on *contractive message passing*.

A. Observations on Privacy Accumulation and Performance Degradation

1) A Case Study on GAP*'s Performance on Learning Long-Range Interactions:* To further investigate this phenomenon, we evaluate K-layer GAP on the *chain-structured* dataset (see Figure 2), which is adopted in [48, 49] to examine long-range interaction learning capabilities. Specifically, this dataset creates a controlled environment for evaluating GNN performance on long-range interactions. A model without learning from graph structure such as MLP would fail since most nodes in the dataset have zero-valued feature vectors regardless of their chain type. For a message-passing GNN to correctly classify nodes positioned K hops away from the informative first node, it must perform at least K propagation steps to transfer the meaningful features across the chain as shown in Figure 2. This requirement becomes particularly challenging in the private setting as noise must be injected after each message passing layer, potentially overwhelming the signal being propagated.



(a) Strong Privacy (b) Weak Privacy

Fig. 3: Motivating Experiments of Classification Model over Chain-structured Datasets.

Figure 3 compares GAP' accuracy of its non-private version (blue solid line) with its private versions (dashed lines), under different privacy budgets $\epsilon \in \{1, 2, 4, 8, 16, 32\}$. It shows a binary node classification over multiple 8-node chains as a case study. The red solid line in Figure 3 represents random guessing (50% accuracy).

Our exploration reveals a stark contrast between private and non-private settings:

- *Non-private setting.* GAP's accuracy consistently improves with increasing layer depth, as message passing enables feature propagation across the chain. The model achieves satisfactory performance after sufficient depth ($K \geq 8$), ultimately reaching perfect classification (100% accuracy) at $K = 15$ hops—demonstrating the necessity of deep architectures for capturing long-range dependencies.
- *Private setting.* Privacy protection dramatically degrades model utility. As shown in Figure 3, even with relatively generous privacy budgets ($\epsilon = 16$ or $\epsilon = 32$), performance remains marginally above random guessing (50%). Crucially, increasing depth offers no benefit and often harms performance, as noise accumulates exponentially across layers. This confirms our theoretical concerns: standard approaches to privacy in GNNs fundamentally limit the ability to learn long-range interactions.

2) Empirical (Counter-Intuitive) Observation: Deeper GNNs May Enhance Privacy: Recent work by [50] revealed a counter-intuitive phenomenon: deeper GNNs empirically exhibit lower vulnerability to membership and link inference attacks. This challenges standard privacy composition analysis, which suggests privacy risk increases with model depth due to multiple queries of the graph data. The insight stems from the *over-*

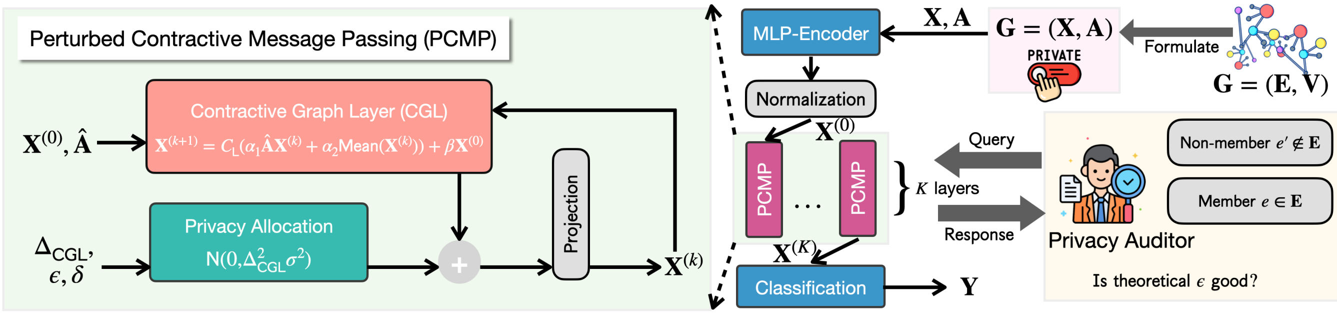


Fig. 4: Overview of CARIBOU. The framework integrates CAM, PAM, PDM (Section IV). Combining CAM and PAM can form PCMP. Together, these modules enable scalable and accurate private GNN learning under a convergent privacy budget.

smoothing[2] [26] phenomenon, where node representations become indistinguishably homogeneous phenomenon, where node representations converge toward similar values as depth increases, making it inherently difficult for adversaries to distinguish individual nodes or infer sensitive relationships.

This observation suggests that standard privacy analysis may be overly pessimistic. The conventional approach of adding noise that scales linearly with depth may be unnecessarily conservative, as the natural privacy amplification properties of over-smoothing could enable tighter privacy bounds with less noise per layer.

B. Core Idea for Convergent Privacy

Our insight is to identify and leverage the inherent privacy amplification that occurs in multi-layer GNNs through *contractiveness* (Definition 6). When nodes aggregate information from their neighbors (e.g., graph convolution), the resulting representations necessarily become more similar to each other.

Definition 6 (Contractive Map)**.** A map $f : \mathbb{R}^d \to \mathbb{R}^d$ is said to be contractive with respect to a norm $\|\cdot\|$ if there exists a constant $c < 1$ such that for all $x, y \in \mathbb{R}^d$: $\|f(x) - f(y)\| \leq c\|x - y\|$ where c is the contractiveness coefficient that governs the rate of contraction.

Designing Private GNNs with Convergent Privacy. We aim to leverage the privacy amplification properties of contractive map to design a new framework for private GNNs. This framework is inspired by the recent advances in differentially private gradient descent (DP-GD) [28, 29], which has shown that the privacy cost can *converge to a finite value* even with arbitrarily many iterations. Motivated by it, we aim to translate the advanced privacy analysis techniques from DP-GD to GNNs. We observe that perturbed message passing (Equation 2) in GNNs follows a strikingly mathematical parallel pattern as DP-GD: This parallel structure enables to leverage the insight of convergent privacy analysis in the context of GNNs, provided we ensure two critical conditions:

1) **Hidden intermediate embeddings:** Release only the final node representations $\boldsymbol{X}^{(K)}$ after K layers, concealing all intermediate states; (Section IV)
2) **Contractive message passing:** Design the message passing operation $\text{MP}_{\boldsymbol{G}}$ to be provably contractive with coefficient $c < 1$, ensuring $\|\text{MP}_{\boldsymbol{G}}(\boldsymbol{X}) - \text{MP}_{\boldsymbol{G}}(\boldsymbol{Y})\|_F \leq c\|\boldsymbol{X} - \boldsymbol{Y}\|_F$ for all node feature matrices $\boldsymbol{X}, \boldsymbol{Y}$. (Section IV-A)

When the perturbed message passing step is contractive with respect to the ℓ_2 norm, the distance between GNNs trained on neighboring datasets shrinks at each step. Consequently, the influence of individual data points diminishes, leading to the amplified privacy rooted from "over-smoothing" effect. Accordingly, our insight challenges the previous analysis on private GNNs that accumulates the privacy loss from multi-hop GNN aggregations linearly, and simultaneously removes the over-estimated privacy loss of finally released GNN model to derive a much tighter bound.

IV. Private GNNs with Contractiveness

Building on insights in Section III, we propose CARIBOU, a modular private GNN framework with convergent privacy budget (Figure 4), including three key modules:

1) **Contractive Aggregation Module (CAM, § IV-A):** We propose a *new design of message passing layers* with carefully controlled Lipschitz constants to ensure contractiveness so that we can add a reasonable, small yet sufficient noise to protect the computed message passing; Then, we realize a *new DP mechanism of perturbed message passing that only releases final node representations* $\boldsymbol{X}^{(K)}$, preventing adversary from exploiting intermediate states and thus amplifying the privacy guarantee rooted in hidden node embeddings.
2) **Privacy Allocation Module (PAM, § IV-B):** PAM ensures efficient privacy budget allocation for edge- and node-level DP guarantees, in which noise calibration is based on *new convergent privacy analysis*.
3) **Privacy Auditing Module (PDM, § IV-C):** PDM empirically audits graph DP through tests such as membership inference attacks, ensuring theoretical guarantees align with practical deployments.

Security Model. Aggregation-based GNNs such as GCN [35], GCN-II [30], and SAGE [51] reveal only the final

[2]Incidentally, another similar phrase is "over-fitting", which refers to a model performing well on training data but poorly on unseen data due to memorization. We emphasize that they are different concepts to avoid confusion.

CGL	Contractive Graph Layer.
RDP	Rényi differential privacy.
MIA	Membership Inference Attacks.
K	The number of layers or hops.
ϵ, σ	Privacy parameters: budget, noise scale.
C_{L}, Δ	Lipchitz constant, sensitivity.
$\alpha_1, \alpha_2, \beta$	Hyper-parameters of CGL.

TABLE II: Acronyms & Symbols

node embeddings, keeping intermediate embeddings hidden. The final node embeddings learned in this DP manner can be subsequently applied to downstream tasks such as node classification [52], link prediction [36], and graph classification [37]. Since DP introduces utility loss, GAP [33] seeks to improve model utility by concatenating all intermediate node embeddings from the K layers. However, it causes the noise scale σ to grow linearly as $O(\sqrt{K/\epsilon^2})$, and this privacy bound is loose. This work takes a step further beyond GAP, removing the assumption that all intermediate node embeddings must be revealed under a more realistic security model [53].

Overall, CARIBOU (Figure 4) achieves accurate and private multi-layer GNNs, which theory will be established in Section V. Table II summarizes acronyms and symbols.

A. Contractive Aggregation Module

Contractive operations in GNNs, such as the graph convolution (GConv) layer [1], inherently reduce privacy risks by mitigating the memorization of GNN parameters through the over-smoothing phenomenon [26]. This property aligns well with the need for K-layer aggregation in long-range graphs, where a target node aggregates embeddings from distant source nodes. However, directly stacking K GConv layers, as shown in Figure 3, is suboptimal due to limited expressive power [30] and heightened sensitivity to DP noise [54].

To address these limitations, we propose the Contractive Aggregation Module (CAM), centered on the Contractive Graph Layer (CGL). The CGL introduces adjustable coefficients to enhance flexibility and utility while maintaining privacy. Formally, the CGL is defined as:

$$\boldsymbol{X}^{(k+1)} = C_{\mathsf{L}}\big(\alpha_1 \hat{\boldsymbol{A}} \boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})\big) + \beta \boldsymbol{X}^{(0)}, \quad (4)$$

where $0 \le C_{\mathsf{L}} < 1$ ensures contractiveness, and $\alpha_1, \alpha_2, \beta > 0$ with $\alpha_1 + \alpha_2 = 1$ are hyperparameters. Here, $\hat{\boldsymbol{A}}$ is the symmetrically normalized adjacency matrix [3], $\mathsf{Mean}(\boldsymbol{X}^{(k)})$ computes the mean of node embeddings, and $\boldsymbol{X}^{(0)}$ represents the initial node features.

Analysis of hyperparameters. The hyperparameter C_{L} ensures the contractiveness by bounding the magnitude of the output embeddings. Specifically, C_{L} enforces a Lipschitz constraint, ensuring that small changes in the input embeddings $\boldsymbol{X}^{(k)}$ result in proportionally small changes in the output $\boldsymbol{X}^{(k+1)}$. The coefficients α_1, α_2 govern the relative contributions of the graph-based aggregation $\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)}$ and the mean-based aggregation $\mathsf{Mean}(\boldsymbol{X}^{(k)})$, respectively. By satisfying the constraint $\alpha_1 + \alpha_2 = 1$, these coefficients ensure a convex combination of the two components. A higher α_1 emphasizes the influence of the graph topology, leveraging structural information from the adjacency matrix $\hat{\boldsymbol{A}}$. The parameter β controls the residual connection $\beta\boldsymbol{X}^{(0)}$ calculated independently of the graph topology, incorporating the characteristics of the initial node $\boldsymbol{X}^{(0)}$ into the output. This residual term mitigates the vanishing gradient problem by preserving a direct path for gradient flow.

Together, the CGL combines three key components: (1) the graph-based aggregation $\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)}$, (2) the mean-based aggregation $\mathsf{Mean}(\boldsymbol{X}^{(k)})$, and (3) the residual connection $\beta\boldsymbol{X}^{(0)}$. This design balances the trade-off between privacy and utility by limiting the propagation of noise while preserving expressive power.

Comparison with existing design. GAP [33] enhances expressiveness by connecting all intermediate embeddings but lacks contractiveness, compromising utility and privacy. In contrast, CGL introduces adjustable coefficients, achieving a balanced trade-off between privacy, utility, and generalizability. Table I summarizes the general design characteristics between GAP and CARIBOU.

B. Privacy Allocation Module

For multi-hop aggregation, the features from the previous hop $\boldsymbol{X}^{(k-1)}$ are aggregated using the adjacency matrix $\hat{\boldsymbol{A}}$ to enable message passing to neighboring nodes. To ensure privacy, Gaussian noise $\mathcal{N}(\mu, \sigma^2)$ is added to the aggregated features, where the noise variance σ^2 is determined by the privacy budget ϵ, ensuring compliance with DP guarantees.

Building on Section IV-A, we integrate CAM and PAM to design the Perturbed Contractive Message Passing (PCMP) (Algorithm 1). By leveraging contractiveness, PCMP ensures bounded privacy loss for long K-hop graphs, eliminating the linear growth of privacy loss with K. A subsequent projection step enforces Lipschitz constraints, maintaining consistent scaling across hops. After K iterations, PCMP outputs private feature matrices $\hat{\mathbf{X}}^{(K)}$, which are passed to the classification module.

Privacy Budget Allocation. The privacy budget is distributed across K hops to ensure the total privacy loss adheres to the specified ϵ and δ. The noise scale is calibrated based on the sensitivity of graph operations and the desired DP guarantees, using a noise allocation mechanism (NAM) that limits noise accumulation under Lipschitz constraints. The maximum allowable noise calibration (see Corollary 1) is constrained by $K' = \min(K, (1 + C_{\mathsf{L}})/(1 - C_{\mathsf{L}}))$.

PCMP integrates contractive aggregation and privacy-preserving perturbation for private message passing. Noise calibration begins by determining the sensitivity of graph operations and computing the noise scale. At each hop, the CGL aggregates features while maintaining contractiveness through adjustable coefficients. Gaussian noise is added to ensure privacy, and embeddings are normalized to enforce

[3] $\hat{\boldsymbol{A}} = \boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{A} + \boldsymbol{I})\boldsymbol{D}^{-\frac{1}{2}}$ has been widely adopted after GCN emerged, where $\boldsymbol{D}$ is the degree matrix and $\boldsymbol{I}$ is the identity matrix.

Algorithm 1 Perturbed Contractive Message Passing (PCMP)

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with adjacency matrix $\boldsymbol{A}$; The number of hops K; Lipchitz constant C_{L}, DP parameters ϵ, δ; Initial normalized features $\mathbf{X}^{(0)}$;
Ensure: Private aggregated node embeddings $\hat{\boldsymbol{X}}^{(K)}$.

▷ *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 σ^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: $\mathbf{X}^{(K)}$

Lipschitz constraints. This iterative process produces private embeddings suitable for downstream tasks.

C. Privacy Auditing Module

Message passing mechanisms integrate graph structures by recursively aggregating and transforming node features from neighbors. Membership Inference Attacks (MIA) on graph examine the vulnerability of message passing to infer whether specific nodes or edges were part of a GNN's training set [34]. The adversaries exploit black-box access to the GNN, querying it with selected data and analyzing outputs (e.g., class probabilities) to infer membership. To align theoretical privacy guarantees with practical deployments, we propose an empirical auditing module. This module simulates an adversary to evaluate GNN privacy before deployment. Extending Carlini et al. [55]'s MIA framework to graphs, we define a graph-specific privacy auditing game (Definition 7) and implement two real-world attacks [4, 34] for privacy verification.

Definition 7 (Graph-based Privacy Auditing Game)**.** The game proceeds between a challenger $\mathcal{C}$ and an privacy auditor $\mathcal{A}$:

1) The challenger samples a training graph in the transductive setting (a set of subgraphs in the inductive setting) $G \leftarrow \mathbb{G}$ and trains a model $f_\theta \leftarrow \mathcal{T}(G)$ on the dataset G.
2) The challenger flips a bit b, and if $b = 0$, samples a fresh challenge point from the distribution $(x, y) \leftarrow \mathbb{G}$ (such that $(x, y) \notin G$). Otherwise, the challenger selects a point from the training set $(x, y) \overset{\$}{\leftarrow} G$.
3) The challenger sends (x, y) to the adversary.
4) The adversary gets query access to the distribution $\mathbb{G}$, and to the model f_θ, and outputs a bit $\hat{b} \leftarrow \mathcal{A}^{\mathbb{G}, f}(x, y)$.
5) Output 1 if $\hat{b} = b$, and 0 otherwise.

The attacker can output either a "hard prediction," or a continuous *confidence score*, thresholded as a reference to yield a membership prediction.

D. Putting it Together

As shown in Figure 4, before and after the perturbed message passing, CARIBOU employs an encoder and a classification module (CM) to process the node features. To ensure compliance with DP guarantees, the framework utilizes standard DP-SGD during pre-training.

Upon completing the private K-hop aggregations, the resulting private graph embeddings are passed to the CM. The CM integrates two key components: (1) the graph-agnostic node features $\boldsymbol{X}^{(0)}$, which capture individual node characteristics independent of the graph structure, and (2) the private, topology-aware aggregated features $\boldsymbol{X}'^{(K)}$, which encode structural information from the graph. This dual integration enhances the model's expressiveness while preserving privacy.

To improve classification accuracy, CARIBOU adopts a head MLP architecture proposed by Sajadmanesh et al. [33] as the CM. This design ensures that the CM effectively combines the information from both feature sets, enabling robust node classification. Furthermore, the CM guarantees efficient training by leveraging the graph-agnostic features $\boldsymbol{X}^{(0)}$, ensuring a lower-bound performance even in scenarios where graph topology is unavailable.

V. Convergent Privacy Analysis

In this section, we present a convergent privacy analysis for perturbed message-passing GNNs with respect to the number of hops. We review the standard privacy analysis for a one-layer perturbed message-passing GNN, and then observe that the privacy cost grows linearly with the number of layers under standard composition theorems. We then shift our focus to a *convergent* privacy analysis for perturbed GNNs whose message-passing layers are contractive. In particular, we draw upon the framework of *contractive noisy iteration* (CNI) from [29], recasting the multi-layer perturbed GNN as a CNI process. Our analysis reveals that, under hidden intermediate states and contractive message-passing layers, the privacy cost converges as the number of hops increases. Finally, we specialize this result to our proposed CARIBOU. We show that CARIBOU's message-passing operation is contractive, derive its sensitivity, and thereby establish concrete bounds on both edge-level and node-level differential privacy for arbitrarily many hops. All proofs are deferred to the Appendix A.

A. Contractive Noisy Iteration and Convergent Privacy

Many GNNs, such as GCN, use "mean-type" aggregation, mixing a node's representation with that of its neighbors. Intuitively, iterative mixing could "amplify" privacy, but existing analyses yield only linear compositions. Our key observation is that perturbed message passing in a contractive GNN layer behaves analogously to noisy gradient descent or *noisy iterative maps* [28, 56, 57], where recent work has demonstrated *privacy amplification via iteration*.

Below, we introduce the framework of *contractive noisy iteration* (CNI) from [29], and the meta theorem proved by [29] that provides a tight privacy guarantee for CNI processes.

Definition 8 (Contractive Noisy Iteration (CNI)[29, Definition 3.1])**.** Consider a sequence of random variables

$$\boldsymbol{X}^{(k+1)} = \Pi_{\mathcal{K}}\big(\, \phi_{k+1}\big(\boldsymbol{X}^{(k)}\big) + \boldsymbol{Z}^{(k)}\big), \tag{5}$$

where each map ϕ_k is Lipchitz continuous, $\boldsymbol{Z}^{(k)} \sim \mathcal{N}\big(0,\, \sigma^2\,\mathrm{I}\big)$ are i.i.d. Gaussian noise vectors independent of $\boldsymbol{X}^{(k)}$, and $\Pi_{\mathcal{K}}$ is a projection operator onto a convex feasible set $\mathcal{K} \subseteq \mathbb{R}^d$. This iterative process is called *contractive noisy iteration* (CNI).

A special case of CNI considered in [29] is the noisy gradient descent, where the contractive functions are the gradient update steps for some fixed loss function f, and the noise distribution ξ_k is Gaussian. This situation is similar to the perturbed message passing mechanism in a contractive GNN layer, where the contractive function $\phi_k \equiv \mathrm{MP}_G$ is the message passing operation for a fixed graph $\boldsymbol{G}$.

The complete privacy analysis of CNI processes in [29] involves concepts of *trade-off functions* and *Gaussian differential privacy* (GDP). GDP can be converted to the more familiar Rényi Differential Privacy (RDP) framework:

Lemma 1 (GDP Implies RDP [29, Lemma A.4])**.** *If a mechanism is μ-GDP, then it satisfies $\big(\alpha, \frac{1}{2}\,\alpha\,\mu^2\big)$-RDP for all $\alpha > 1$.*

With this connection established, we now state the key meta theorem from [29] that analyzes the CNI process and provides a tight privacy guarantee.

Theorem 2 (Meta Theorem on CNI [29, Theorem C.5])**.** *Let $\{\boldsymbol{X}^{(k)}\}_{k=0}^{K}$ and $\{\boldsymbol{X}'^{(k)}\}_{k=0}^{K}$ represent the output of CNI processes,*

$$\textit{CNI}(\boldsymbol{X}^{(0)}, \{\phi_k\}_{k=1}^{K}, \{\mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)\}_{k=1}^{K}, \mathcal{K}), \textit{ and}$$
$$\textit{CNI}(\boldsymbol{X}^{(0)}, \{\phi'_k\}_{k=1}^{K}, \{\mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)\}_{k=1}^{K}, \mathcal{K}).$$

Assume that:

- *ϕ_1 and ϕ'_1 are Lipschitz continuous,*
- *each ϕ_k, ϕ'_k is γ-Lipschitz, with $\gamma < 1$ for $k = 2, \ldots, K$,*
- *$\|\phi_k(x) - \phi'_k(x)\| \leq s$ for all x and $k = 1, \ldots, K$,*

Then the tradeoff function $T(\boldsymbol{X}^{(K)}, \boldsymbol{X}'^{(K)})$ satisfies,

$$T(\boldsymbol{X}^{(K)}, \boldsymbol{X}'^{(K)}) \geq T(\mathcal{N}(0,1), \mathcal{N}(\mu^{(K)}, 1)),$$

where,

$$\mu^{(K)} = \sqrt{\frac{1-\gamma^K}{1+\gamma^K}\frac{1+\gamma}{1-\gamma}}\frac{s}{\sigma}.$$

Remark 1. *The theorem above slightly generalizes the original result from [29] by relaxing the Lipschitz condition to require $\gamma < 1$ only for $k \geq 2$ rather than for all iterations. This relaxation is critical for analyzing our* CARIBOU *architecture, where the first message-passing layer includes the residual term $\beta \boldsymbol{X}^{(0)}$, potentially making it non-contractive while subsequent layers remain contractive. The proof extends the original argument by carefully tracking the influence of the first layer on the privacy guarantee.*

By utilize the above meta theorem and property of Gaussian tradeoff function, we can derive the privacy guarantee for a K-layer perturbed message-passing GNN with contractive message passing layers.

Theorem 3 ([29, Theorem 4.2] adapted for GNNs)**.** *Let* MP *be a message passing mechanism of a GNN such that the message passing operator* MP_G *for any graph $\boldsymbol{G}$ is contractive with Lipschitz constant $\gamma < 1$ for layers $k \geq 2$. Assume the sensitivity of* MP *is* $\Delta(\mathrm{MP})$ *and the noise scale is σ. Then, a K-layer perturbed message passing GNN with* MP *satisfies,*

$$\left(\alpha, \frac{1}{2}\alpha\frac{\Delta^2(\mathrm{MP})}{\sigma^2}\frac{1-\gamma^K}{1+\gamma^K}\frac{1+\gamma}{1-\gamma}\right)\textit{-RDP},$$

which is equivalent to $\left(\frac{1}{2}\alpha\frac{\Delta^2(\mathrm{MP})}{\sigma^2}\frac{1-\gamma^K}{1+\gamma^K}\frac{1+\gamma}{1-\gamma} + \frac{\log(1/\delta)}{\alpha-1}, \delta\right)$-DP.

The upshot is that as $K \to \infty$, the privacy parameter converges to $\frac{1}{2}\alpha\frac{\Delta^2(\mathrm{MP})}{\sigma^2}\frac{1+\gamma}{1-\gamma}$, rather than growing unbounded with K as in standard composition. This result enables meaningful privacy guarantees even for deep GNNs with many message-passing layers.

B. Edge- and Node-Level Privacy of CARIBOU

We now specialize to the CARIBOU architecture and establish both edge-level and node-level DP guarantees. Recall the *contractive graph layer (CGL)* of CARIBOU:

$$\text{CGL:} \qquad \boldsymbol{X}^{(k+1)} = C_{\mathsf{L}}(\alpha_1 \hat{\boldsymbol{A}}\boldsymbol{X}^{(k)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k)})) + \beta\boldsymbol{X}^{(0)},$$

where $0 \leq C_{\mathsf{L}} < 1$, and hyper-parameters, $\alpha_1, \alpha_2, \beta > 0, \alpha_1 + \alpha_2 = 1$. In order to establish the privacy guarantees of CARIBOU, we need to determine the sensitivity and the Lipschitz constant of the message passing layer CGL.

Proposition 1. *The message passing layer* CGL *of* CARIBOU *is contractive with Lipschitz constant $C_{\mathsf{L}} < 1$ with respect to the input $\boldsymbol{X}^{(k)}$, for any $k \geq 2$.*

The sensitivity of the message passing layer CGL regarding edge-level and node-level adjacency graphs is determined as follows.

Theorem 4 (Edge-level Sensitivity of CGL)**.** *Let $\boldsymbol{G}$ be a graph and $D_{\min}$ be the minimum node degree of $\boldsymbol{G}$. The edge level sensitivity Δ_e of the message passing layer* CGL *is*

$$\Delta_e(\mathsf{CGL}) = \sqrt{2}C_{\mathsf{L}}\alpha_1\Big(\frac{1}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})}{\sqrt{D_{\min}+1}} + \frac{1}{\sqrt{D_{\min}+2}\sqrt{D_{\min}+1}}\Big), \tag{6}$$

where $C(D_{\min})$ is a piecewise function defined as

$$C(D_{\min}) = \begin{cases} \frac{D_{\min}}{\sqrt{D_{\min}+1}} - \frac{D_{\min}}{\sqrt{D_{\min}+2}} & D_{\min} > 3 \\ (\frac{3}{\sqrt{4}} - \frac{3}{\sqrt{5}}) & 1 \leq D_{\min} \leq 3 \end{cases} \tag{7}$$

Intuitively, the edge-level sensitivity Δ_e will be smaller if the minimum degree $D_{\min}$ is larger or the Lipschitz constant C_{L} is smaller. The node level sensitivity Δ_n of the message passing layer CGL is determined as follows.

Theorem 5 (Node-level Sensitivity of CGL)**.** *Let $\boldsymbol{G}$ be a graph and $D_{\max}$ be the maximum node degree of $\boldsymbol{G}$. The node level sensitivity Δ_n of the message passing layer* CGL *is*

$$\Delta_n(\mathsf{CGL}) = 1 + \alpha_2 C_{\mathsf{L}} \frac{2|V|}{|V|+1} + \alpha_1 C_{\mathsf{L}} \left(\frac{\sqrt{D_{\max}}}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})\sqrt{D_{\max}}}{\sqrt{D_{\min}+1}} + \frac{1}{\sqrt{D_{\min}+2}} \right) \quad (8)$$

where $C(D_{\min})$ is defined as in equation 7.

From the above, we can see that the node-level sensitivity Δ_n will be smaller if the maximum degree $D_{\max}$ is smaller, the minimum degree $D_{\min}$ is larger, or the Lipschitz constant C_{L} is smaller.

Using the contractive constraint in Proposition 1 and the sensitivity results in Theorem 4 and Theorem 5, we can apply Theorem 3 to derive the following privacy guarantee (Corollary 1) for the message-passing layer CGL of CARIBOU. Specifically, for Lipchitz constant 0.8, a 10-layer GNN realizes $(\alpha, 3.62\alpha\Delta^2/\sigma^2)$-RDP (versus $(\alpha, 5\alpha\Delta^2/\sigma^2)$-RDP for GAP), meanwhile 2-layer GNN is $(\alpha, 0.99\alpha\Delta^2/\sigma^2)$-RDP (versus $(\alpha, \alpha\Delta^2/\sigma^2)$-RDP for GAP). That is, more privacy costs are saved as the number of layer increases.

Corollary 1 (DP Guarantees for CGL layers)**.** *Let $\boldsymbol{G}$ be a graph and K be the number of hops (*CGL *layers) in* CARIBOU*. Then the K-hop message passing of* CARIBOU *satisfies:*

$$\left(\alpha, \frac{\alpha}{2}\frac{\Delta^2}{\sigma^2}\min\left\{K, \frac{1-C_{\mathsf{L}}^K}{1+C_{\mathsf{L}}^K}\frac{1+C_{\mathsf{L}}}{1-C_{\mathsf{L}}}\right\}\right)\text{-}RDP,$$

where $\Delta = \Delta_e(\mathsf{CGL})$ from Theorem 4 for edge-level privacy, or $\Delta = \Delta_n(\mathsf{CGL})$ from Theorem 5 for node-level privacy. In particular, as $K \to \infty$, it converges to $\left(\alpha, \frac{\alpha}{2}\frac{\Delta^2}{\sigma^2}\frac{1+C_{\mathsf{L}}}{1-C_{\mathsf{L}}}\right)$-RDP. Also, the RDP guarantees imply the following DP guarantees:

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

For a complete analysis, we integrate the node-level privacy guarantees into an overall DP bound for the entire training process of CARIBOU. Specifically, we assume that the private DAE and CM satisfy $(\alpha, \epsilon_{\mathsf{DAE}}(\alpha))$-RDP and $(\alpha, \epsilon_{\mathsf{CM}}(\alpha))$-RDP, respectively. By privacy composition, CARIBOU's overall privacy budget $\epsilon_{\mathsf{CARIBOU}}$ is then derived as shown in Theorem 6. The terms $\epsilon_{\mathsf{DAE}}(\alpha)$ and $\epsilon_{\mathsf{CM}}(\alpha)$ individually quantify the privacy contributions from the DAE and CM modules, while the remaining aggregation term accounts for node-level privacy loss during K-hop message passing. Finally, the $\frac{\log(1/\delta)}{\alpha-1}$ term incorporates the privacy failure probability δ, ensuring conventional privacy guarantees across the inference process.

Theorem 6 (CARIBOU's Privacy Guarantee)**.** *For any $\alpha > 1$, let DAE training and CM training satisfy $(\alpha, \epsilon_{\mathsf{DAE}}(\alpha))$-RDP and $(\alpha, \epsilon_{\mathsf{CM}}(\alpha))$-RDP, respectively. Then, for the maximum hop $K \geq 1$, privacy failure probability $0 < \delta < 1$, Lipschitz constant $0 < C_{\mathsf{L}} < 1$, and noise variance σ^2,* CARIBOU *satisfies $(\epsilon_{\mathsf{CARIBOU}}, \delta)$-DP, where $\epsilon_{\mathsf{CARIBOU}} = \epsilon_{\mathsf{DAE}}(\alpha) + \epsilon_{\mathsf{CM}}(\alpha) + \frac{\alpha}{2}\frac{\Delta^2}{\sigma^2}\min\left\{K, \frac{1-C_{\mathsf{L}}^K}{1+C_{\mathsf{L}}^K}\frac{1+C_{\mathsf{L}}}{1-C_{\mathsf{L}}}\right\} + \frac{\log(1/\delta)}{\alpha-1}$. Here, $\Delta = \Delta_e(\mathsf{CGL})$ from Theorem 4 for edge-level privacy, or $\Delta = \Delta_n(\mathsf{CGL})$ from Theorem 5 for node-level privacy.*

VI. Experimental Evaluation

The empirical evaluation includes privacy-utility trade-offs, privacy audits, ablation studies of hops and hyper-parameters, and computational overhead.

Datasets. CARIBOU was tested over nine graph datasets. Five of the datasets have been broadly used to evaluate GNN methods, including `Photo` and `Computers` [58], `Cora` and `PubMed` [59], `Facebook` [60]. We also adjusted the synthetic chain-structured dataset developed under [48] into various scales, termed `Chain-S`, `Chain-M`, `Chain-L` and `Chain-X`. The chain-structured dataset has been used to understand the relations between privacy/utility and hops, as described in Section III-A. It is considered as an important benchmark to assist the development of long-range interaction GNNs by the ML community [21]. More details on datasets, model configurations, experiments, and privacy configurations are specialized in full-version paper.

Baselines. We compare CARIBOU with multiple baselines [27, 33, 17, 61, 32] about edge-level private GNNs and vanilla GNNs. To our best knowledge, GAP [33] and DPDGC [27] are the strongest baselines for perturbed message passing under Gaussian mechanisms in both edge/node-level DP GNNs. In addition, we consider another research direction, i.e., first perturbing the graph through randomized response and then training GNNs over the perturbed graph PertGraph [17]. For a comprehensive evaluation, we adopt both research lines of works as our baselines. MLP is a baseline commonly compared with GNNs to demonstrate how well GNNs utilize graph structures, as it provides a reference counterpart to which GNNs learn the only node features without graph topology.

A. Trade-offs of Privacy and Accuracy

We first compare CARIBOU with the baseline methods on all datasets for their downstream classification tasks and report top GNN model utility of both EDP and NDP. We run each model 3 times for each group of hyper-parameters, reporting the top classification accuracy in Table III. For the experiments about NDP, we set the max node degree $D_{\max}$ to 20, following the experiment setup of GAP.

Regarding the standard graph datasets, for `Computers`, `PubMed`, `Cora` and `Photo`, CARIBOU can outperform all the other baselines in most cases with varying ϵ, no matter for EDP or NDP. As established in Theorem 5, NDP requires injecting more noise compared to EDP under the same privacy budget, hence, the accuracy of NDP is often lower than EDP for standard datasets. In particular, CARIBOU is the only framework that can surpass MLP of most cases in NDP

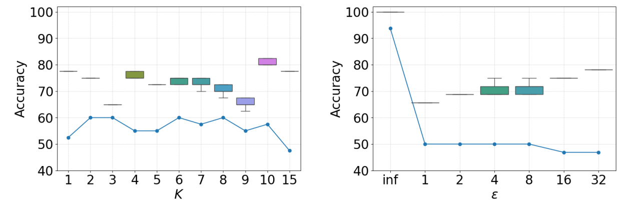
TABLE III: Comparison of Classification Accuracy for EDP and NDP. The best accuracy and the second-best accuracy are highlighted, respectively. The symbol ⬆ represents that the best accuracy improves the second-best accuracy by more than 10%. The symbol ▼ represents the accuracy less than 55%, close to random guess on the chain-structured datasets.

	Dataset	Computers	Facebook	PubMed	Cora	Photo	Chain-S	Chain-M	Chain-L	Chain-X
EDP										
$\epsilon = 1$	CARIBOU	92.0%	74.0%	88.2%	85.1%	95.8%	84.4% ⬆	82.5% ⬆	70.0%	66.0%
	DPDGC	88.0%	60.6%	88.3%	75.5%	92.5%	43.8% ▼	50.0% ▼	51.7% ▼	39.0% ▼
	GAP	87.2%	68.5%	87.4%	77.1%	93.0%	65.6%	67.5%	61.7%	67.0%
	PertGraph	77.8%	48.2% ▼	85.0%	60.0%	82.4%	53.1% ▼	45.0% ▼	53.3% ▼	51.0% ▼
$\epsilon = 2$	CARIBOU	92.1%	73.8%	89.1%	86.9%	95.9%	90.6% ⬆	82.5% ⬆	73.3% ⬆	68.0%
	DPDGC	88.2%	66.7%	88.2%	77.5%	93.3%	43.8% ▼	50.0% ▼	51.7% ▼	39.0% ▼
	GAP	88.3%	71.7%	87.5%	77.5%	93.4%	65.6%	67.5%	61.7%	68.0%
	PertGraph	76.1%	48.1% ▼	84.6%	60.1%	82.4%	43.8% ▼	45.0% ▼	53.3% ▼	51.0% ▼
$\epsilon = 4$	CARIBOU	92.2%	74.0%	89.5%	87.3% ⬆	95.9%	90.6% ⬆	82.5% ⬆	73.3% ⬆	69.0%
	DPDGC	88.9%	73.3%	88.4%	75.1%	94.2%	43.8% ▼	50.0% ▼	51.7% ▼	39.0% ▼
	GAP	87.2%	68.5%	87.4%	77.1%	93.0%	65.6%	67.5%	61.7%	67.0%
	PertGraph	79.1%	50.3% ▼	85.8%	63.3%	85.7%	50.0% ▼	47.5% ▼	51.7% ▼	54.0% ▼
$\epsilon = 8$	CARIBOU	92.2%	74.4%	89.8%	88.4%	96.0%	93.8% ⬆	82.5% ⬆	75.0% ⬆	70.0%
	DPDGC	89.4%	78.6%	88.6%	76.0%	94.6%	43.8% ▼	50.0% ▼	51.7% ▼	39.0% ▼
	GAP	90.1%	75.0%	88.1%	79.0%	94.6%	65.6%	60.0%	61.7%	67.0%
	PertGraph	87.9%	75.6%	84.8%	75.7%	92.2%	43.8% ▼	47.5% ▼	51.7% ▼	61.0%
$\epsilon = 16$	CARIBOU	92.2%	74.5%	89.8%	88.6%	96.0%	93.8%	85.0% ⬆	73.3% ⬆	73.0%
	DPDGC	90.4%	81.2%	88.9%	77.7%	93.8%	43.8% ▼	50.0% ▼	51.7% ▼	40.0% ▼
	GAP	90.1%	76.0%	88.5%	81.7%	94.6%	62.5%	70.0%	61.7%	68.0%
	PertGraph	90.9%	79.5%	87.6%	84.7%	94.1%	43.8% ▼	47.5% ▼	51.7% ▼	51.0% ▼
$\epsilon = 32$	CARIBOU	92.2%	74.0%	89.9%	88.6%	95.9%	93.8% ⬆	82.5% ⬆	78.3% ⬆	73.0%
	DPDGC	91.3%	82.9%	88.8%	79.9%	94.3%	43.8% ▼	50.0% ▼	51.7% ▼	40.0% ▼
	GAP	90.6%	77.0%	88.8%	82.5%	94.8%	59.4%	65.0%	61.7%	68.0%
	PertGraph	90.6%	79.9%	86.9%	85.2%	94.4%	43.8% ▼	47.5% ▼	51.7% ▼	51.0% ▼
NDP (max node degree = 20)										
$\epsilon = 1$	CARIBOU	91.91% ⬆	64.47% ⬆	72.94%	80.81% ⬆	94.83% ⬆	78.12% ⬆	85.00% ⬆	68.33% ⬆	66.00%
	DPDGC	56.72%	39.09% ▼	59.12%	33.58% ▼	46.39% ▼	58.06%	57.50%	57.63%	54.55% ▼
	GAP	36.71% ▼	35.07% ▼	55.06%	33.95% ▼	30.88% ▼	65.62%	55.0% ▼	58.33%	59.00%
	PertGraph	29.51% ▼	20.73% ▼	39.51% ▼	18.45% ▼	21.07% ▼	59.38%	60.00%	56.67%	55.00% ▼
	MLP	63.44%	45.39% ▼	73.52%	34.13% ▼	59.44%	56.25%	55.0% ▼	55.0% ▼	51.0% ▼
$\epsilon = 2$	CARIBOU	92.24% ⬆	68.10% ⬆	79.86% ⬆	83.39% ⬆	95.10% ⬆	84.38% ⬆	85.00%	70.00% ⬆	65.00%
	DPDGC	66.07%	45.37% ▼	68.60%	31.92% ▼	57.92%	58.06%	57.50%	57.63%	54.55% ▼
	GAP	47.92% ▼	39.97% ▼	67.82%	33.39% ▼	36.05% ▼	65.62%	55.00% ▼	58.33%	59.00%
	PertGraph	34.30% ▼	21.43% ▼	39.72% ▼	19.93% ▼	23.46% ▼	59.38%	60.00%	56.67%	55.00% ▼
	MLP	69.56%	47.95% ▼	80.6%	37.82% ▼	72.96%	56.25%	55.0% ▼	55.0% ▼	51.0% ▼
$\epsilon = 4$	CARIBOU	92.24% ⬆	70.83% ⬆	84.66%	85.61% ⬆	95.36% ⬆	90.62% ⬆	82.50%	71.67% ⬆	69.00% ⬆
	DPDGC	72.35%	48.81% ▼	79.71%	32.10% ▼	73.49%	58.06%	57.50%	57.63%	54.55% ▼
	GAP	61.84%	47.02% ▼	79.33%	33.39% ▼	45.33% ▼	65.62%	55.00% ▼	58.33%	59.00%
	PertGraph	36.12% ▼	23.01% ▼	40.30% ▼	21.59% ▼	25.78% ▼	59.38%	60.00%	56.67%	55.00% ▼
	MLP	73.94%	49.2% ▼	83.13%	50.18% ▼	78.99%	56.25%	55.0% ▼	55.0% ▼	51.0% ▼
$\epsilon = 8$	CARIBOU	92.39% ⬆	72.21% ⬆	87.12%	87.08% ⬆	95.29% ⬆	90.62% ⬆	87.50%	73.33% ⬆	69.00% ⬆
	DPDGC	76.50%	49.64% ▼	83.39%	43.54% ▼	79.66%	58.06%	57.50%	57.63%	59.60%
	GAP	68.52%	48.33% ▼	82.35%	31.55% ▼	68.46%	65.62%	55.00% ▼	58.33%	59.00%
	PertGraph	37.38% ▼	24.53% ▼	42.07% ▼	26.38% ▼	28.36% ▼	59.38%	60.00%	56.67%	55.00% ▼
	MLP	77.65%	49.96% ▼	84.66%	61.81%	83.96%	56.25%	55.0% ▼	55.0% ▼	51.0% ▼
$\epsilon = 16$	CARIBOU	92.43% ⬆	72.29% ⬆	88.41%	88.01% ⬆	95.29%	90.62% ⬆	82.50% ⬆	75.00% ⬆	70.00% ⬆
	DPDGC	78.47%	50.85% ▼	85.32%	56.09%	83.23%	58.06%	57.50%	57.63%	59.60%
	GAP	73.94%	49.98% ▼	83.67%	37.45% ▼	76.74%	65.62%	55.00% ▼	58.33%	59.00%
	PertGraph	40.13% ▼	26.88% ▼	44.33% ▼	27.68% ▼	33.47% ▼	59.38%	60.00%	56.67%	55.00% ▼
	MLP	80.25%	49.92% ▼	85.57%	67.53%	87.34%	56.25%	55.0% ▼	55.0% ▼	51.0% ▼
$\epsilon = 32$	CARIBOU	92.39% ⬆	73.33% ⬆	88.92%	87.82% ⬆	95.36%	87.50% ⬆	82.50% ⬆	75.00% ⬆	70.00% ⬆
	DPDGC	81.33%	51.19% ▼	85.90%	64.02%	86.94%	58.06%	57.50%	57.63%	58.59%
	GAP	76.95%	50.64% ▼	85.04%	57.01%	80.25%	65.62%	55.00% ▼	58.33%	59.00%
	PertGraph	46.40% ▼	31.30% ▼	47.53% ▼	29.89% ▼	41.68% ▼	59.38%	60.00%	56.67%	55.00% ▼
	MLP	81.89%	50.27% ▼	85.87%	70.85%	89.07%	56.25%	55.0% ▼	55.0% ▼	51.0% ▼
Non-Private										
Plain $\epsilon = \infty$	CARIBOU	92.4%	79.0%	89.9%	89.3%	96.0%	100.0%	100.0%	100.0%	100.0%
	DPDGC	92.8%	86.4%	88.1%	83.9%	96.2%	59.4%	77.5%	63.3%	73.0%
	GAP	91.0%	79.0%	89.3%	85.2%	95.5%	100.0%	100.0%	100.0%	100.0%
	PertGraph	91.6%	79.7%	87.0%	82.1%	94.2%	59.4%	55.0% ▼	60.0%	58.0%
	MLP	85.82%	51.35% ▼	87.45%	75.83%	91.98%	59.38%	55.0% ▼	53.33% ▼	51.0% ▼

settings, showing effective GNN learning over structural graphs. We leave ablation study for different max node degrees in Section VI-C5.

For standard benchmark datasets with informative node features, the utility of our model approaches that of non-private methods as the privacy ϵ increases. For chain-structured graphs, the learning task primarily relies on the underlying graph topology, which is more challenging. Accordingly, model utility is more sensitive to the added noise realized by perturbed message passing. This sensitivity is due to their sparse chain structure: non-zero features are present only at the first node of each chain. Information must propagate from this source, and it can be degraded by noise accumulation during propagation. In this case if the small training set, by chance, contains an imbalanced selection of nodes (e.g., sampling nodes only near the end of a chain, far from the feature source), the task becomes significantly more difficult. This can lead to higher variance in results. To ensure a fair comparison, we use the exact same data split for all models mentioned above.

> **Takeaway 1.** CARIBOU achieves a more favorable privacy-utility trade-off than other baselines across standard graph datasets, chain-structured datasets with various parameter settings.



(a) Different K when $\epsilon = 4$ (b) Different ϵ when $K = 10$

Fig. 5: Comparison between CARIBOU (colored boxes) and GAP (blue lines) for ablation study.

B. Privacy Auditing

Following the PAM outlined in Section IV-C, we focus on black-box, membership-based privacy audits that match the theoretical guarantees of our DP mechanisms and those of prior perturbed message-passing methods [27, 33]. Under this threat model, LinkTeller [4] and G-MIA [34] are canonical and complementary: LinkTeller targets edge-level membership by asking whether a specific edge exists in the training graph; G-MIA targets node-level membership by deciding whether a node and its connected edges were used during training. We adopt G-MIA's attacking settings of TSTF, where models have been trained on subgraph and tested on full graph. The adversary knows the whole graph $\boldsymbol{G}$ and all edges contained in $\boldsymbol{G}$ but has no access to the subgraphs used for early training. Both attacks (i) are specifically designed for GNNs, (ii) operate in the transductive setting considered in our analysis, (iii) require only query access to GNN models, and (iv) are publicly available and already used to evaluate DP-GNN defenses. This makes them ideal choices as mechanism-level auditing tools in CARIBOU.

In Table IV (in Appendix), we report the AUC (Area Under the Curve) score about the attack effectiveness, when CARIBOU is being attacked. AUC is a major metric to evaluate the membership inference attack [4, 62]. Specifically, AUC measures true positive rate against the false positive rate on various classification thresholds, and a score of 0.5 suggesting random guessing. We found CARIBOU is very effective against LinkTeller, by dropping the attack AUC from between 0.86 to 0.998 across all standard datasets ($\epsilon = \inf$) to less than 0.500 (ϵ ranges from 1 to 32). The similar effect was also observed by Wu et al. (e.g., less than 0.5 attack precision for 3-layer GCN for high density belief, shown in Table IX) [4] and Tang et al. (e.g., less than 0.5 attack AUC sometimes in Figure 10) [63]. For G-MIA, its AUC on the 5 datasets are already lower than LinkTeller by a notable margin when $\epsilon = \inf$ (between 0.567 to 0.702 for the 5 datasets), so the impact of CARIBOU is relatively small. But we observe on `Cora`, CARIBOU is able to drop AUC from 0.645 to 0.500.

> **Takeaway 2.** In privacy auditing, CARIBOU's shows effective resistance to membership inference attacks.

C. Ablation Study

1) Impact of K: Both CARIBOU and GAP perform K-hop aggregations under K aggregation layers. Here we evaluate the impact of K on accuracy on the chain-structured datasets (`Chain-S`, `Chain-M` and `Chain-L`), as their classification results highly depend on long-range interactions. In full-version paper, we compare CARIBOU and GAP on varying ϵ and varying K, respectively. The result of GAP is drawn with lines and the result of CARIBOU is illustrated with the colored boxes, because CARIBOU also depends on other hyper-parameters $C_{\mathsf{L}}, \alpha_1, \beta$ and we use the colored boxes to represent the interquartile range over 5 runs of their different value combinations.

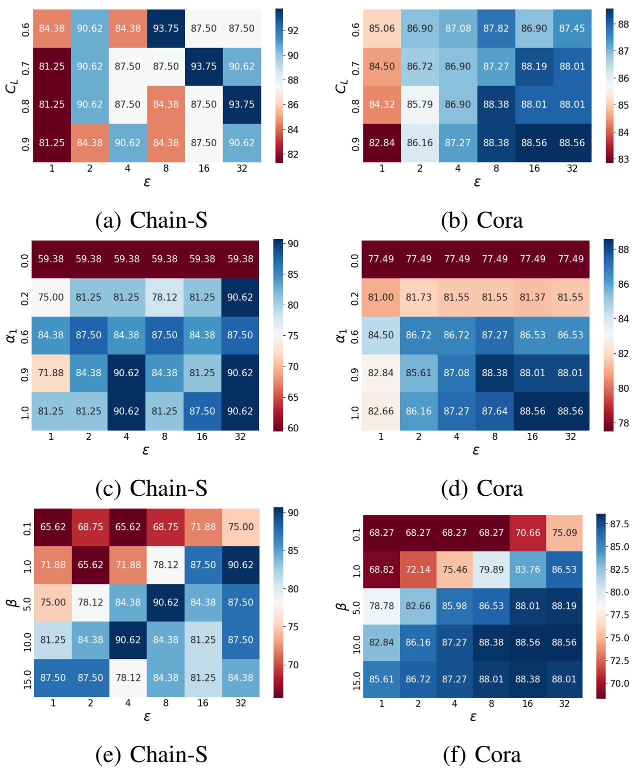
In Figure 5a, we show the result of one setting ($\epsilon = 4$ on `Chain-M`), and CARIBOU achieves higher accuracy at *every* K. In addition, the highest accuracy happens at $K = 10$ (close to the number of nodes per chain) for CARIBOU, and the classification accuracy fluctuates when K varies for both CARIBOU and GAP. Across all datasets, GAP's accuracy degrades monotonically with depth, consistent with its privacy noise variance growing linearly in the number of layers ($\sigma^2 \propto K$). In contrast, CARIBOU benefits from additional depth and then plateaus, owing to its convergent privacy cost with respect to depth. This behavior shows that CARIBOU can leverage deeper architectures and realize the contractive privacy amplification guaranteed by our analysis.

2) Impact of ϵ: We assess how ϵ affects the performance of CARIBOU and GAP, by flipping K and ϵ from the previous ablation study. Specifically, we evaluate the three chain-structured datasets (`Chain-S`, `Chain-M` and `Chain-L`), and for each dataset, we use the same K for both CARIBOU and GAP, and then change ϵ from 1 to 32.

We present the full results in the full-verison paper and one setting in Figure 5b ($K = 10$ on `Chain-S`). When K is much less than the number of nodes per chain, *e.g.*, $K = 1$ vs. 8-nodes chain, CARIBOU and GAP cannot realize satisfactory accuracy (both under 75%) even for non-private settings, as features from distant nodes cannot be effectively learnt. If K is near to or larger than the number of nodes per chain (e.g., when $K = 10$ for `Chain-S`, as shown in Figure 5b), though both GAP and CARIBOU see very high accuracy for non-private mode, the accuracy of GAP drops to 50% at $\epsilon = 1$ and further decreases with increased ϵ, suggesting the noise magnitude are not properly controlled. On the other hand, CARIBOU sees steady growth of accuracy along with increased ϵ, which is a desired outcome for privacy protection.

TABLE IV: Privacy Auditing via LinkTeller and G-MIA. AUC score is reported.

Dataset	LinkTeller							G-MIA						
	$\epsilon = \inf$	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$	$\epsilon = \inf$	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$
Facebook	0.977	0.463	0.483	0.482	0.478	0.472	0.462	0.567	0.587	0.587	0.587	0.583	0.587	0.588
PubMed	0.981	0.446	0.442	0.441	0.445	0.442	0.449	0.600	0.599	0.598	0.598	0.601	0.601	0.605
Cora	0.998	0.427	0.448	0.399	0.443	0.451	0.450	0.645	0.500	0.500	0.500	0.500	0.500	0.500
Photo	0.962	0.475	0.404	0.421	0.428	0.417	0.434	0.678	0.677	0.682	0.682	0.678	0.672	0.676
Computers	0.860	0.367	0.364	0.372	0.363	0.361	0.384	0.702	0.701	0.661	0.707	0.708	0.711	0.701



(a) Chain-S (b) Cora

(c) Chain-S (d) Cora

(e) Chain-S (f) Cora

Fig. 6: Classification Accuracy under $C_{\mathsf{L}}, \alpha_1, \beta$ of CGL.

> **Takeaway 3.** Private GNNs face fundamental trade-offs between privacy, utility, and model depth K. Model utility becomes more susceptible to the DP noise if GNNs are tightly coupled with the underlying graph structure.

3) Noise scaling with depth: Here we analyze how the noise scale changes with the number of layers K under standard linear composition and under convergent privacy analysis (Theorem 3). For a general study, we remove the effect of node degree $D_{\min}, D_{\max}$ derived from a particular dataset and fix the target DP parameters ϵ, δ. In this case, the dependence of the calibrated noise on depth is governed by depth K. To make this comparison concrete, we instantiate a representative setting by normalizing the sensitivity: $\Delta^2(\mathrm{MP}) = 1, \alpha = 6, C_{\mathsf{L}} = 0.9$. Setting $\Delta^2(\mathrm{MP}) = 1$ removes a common multiplicative factor and highlights the qualitative dependence on K. The choice $\alpha = 6$ is common and simplifies the expressions, while $C_{\mathsf{L}} = 0.9$ represents a standard contractive layer.

TABLE V: Noise Scale σ under Different K. We set $\Delta^2(\mathrm{MP}) = 1, \alpha = 6, C_{\mathsf{L}} = 0.9$ and $\epsilon = 4, \delta = 0.001$.

K	1	2	4	8	16	32	64	128
Linear	1.07	1.52	2.15	3.04	4.30	6.07	8.56	12.11
Convergent	1.07	1.52	2.14	3.00	4.07	4.66	4.66	4.67

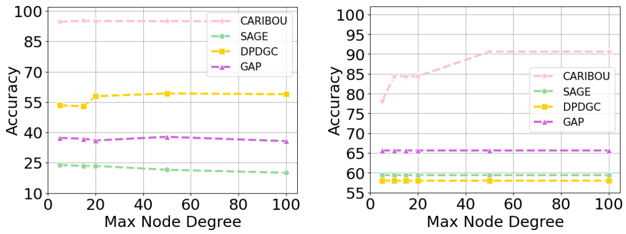
Table V reports the proportional values of σ for several representative depths K, assuming the same target privacy budget $\epsilon = 4$. Under linear composition, the required σ grows proportionally to K, becoming large for deep GNNs. In contrast, under CARIBOU's analysis, the required σ grows from 1 to a bounded constant (here approximately 4.7) and then saturates. This study highlights CARIBOU can support deep architectures without unbounded noise growth.

4) hyper-parameters related to contractiveness: We studied the impact of hyper-parameters $C_{\mathsf{L}}, \alpha_1, \beta$ in CGL. In Appendix, we draw Figure 6 of classification accuracy using Chain-Sand Cora. As C_{L} constrains the features learned at each aggregation, in the relatively weak privacy guarantee ($\epsilon = 16, 32$), Figures 6a and 6b empirically confirms that the accuracy improves with C_{L} increases. In contrast, for strong privacy guarantee ($\epsilon = 1, 2$), larger C_{L} reduces the model accuracy due to the accumulated large noise. Small C_{L} enforces strong contraction, accelerating privacy convergence and reducing effective sensitivity, but overly small values may reduce expressive power. Larger C_{L} increases representational capacity but slows contraction and slightly increases noise amplification.

Figures 6c and 6d describe the ratio (α_1) of learning from the graph, where $\alpha_1 = 1.0 (\alpha_2 = 0.0)$ means the information from adjacent matrix is utilized at the maximum degree. Larger α_1 leads to higher accuracy across varying ϵ in general, suggesting CARIBOU is able to achieve good balance between graph connectivity and privacy. The impact of β, which decides the power of residual connection between node features and CGL, is different on the two datasets. Since Chain-S is designed to tailor graph topology over node features, increasing β to a large value (e.g., 15) might hurt accuracy. For Cora with rich node features, the model accuracy is generally increased along with β.

> **Takeaway 4.** All parameters $\sigma, C_{\mathsf{L}}, \beta, \alpha_1$ in perturbed CGL contribute to the privacy-utility trade-off.

5) Impact of $D_{\max}, D_{\min}$: Figure 7 shows an example ($\epsilon = 2$) of the classification accuracy of NDP under different



(a) $\epsilon = 2$ (Photo) (b) $\epsilon = 2$ (Chain-S)

Fig. 7: NDP Accuracy with Varying Max Node Degree.

maximum node degrees, and more ablation study results are shown in full-version paper. As shown in Figure 7a on the Photo dataset, CARIBOU consistently realizes the highest accuracy under different maximum node degrees ranging from 5 to 100. Improving maximum node degree for DPDGC can slightly increase the classification accuracy when maximum node degree is 20, while the number of maximum node degree does not help for PertGraph and GAP. For the Chain-S dataset in Figure 7b, classification accuracy of CARIBOU is improved when the maximum node degree is increased and relatively small. In addition, CARIBOU outperforms baseline works significantly, *i.e.*, approximated 20%-25% higher than the second best GAP.

The sensitivity formulas in Theorems 4,5 explicitly depend on structural properties of the graph, particularly minimum degree $D_{\min}$ and maximum degree $D_{\max}$. Our empirical results (Table III) reflect these theoretical dependencies: (i) High-degree graphs such as Photo exhibit lower noise and higher accuracy; (ii) Low-degree or chain-like graphs incur higher sensitivity and lower accuracy, but CARIBOU mitigates the impact.

VII. Future Works and Discussions

More GNN models. CARIBOU is instantiated and evaluated primarily with commonly used message-passing architectures, where contractiveness naturally emerges or can be enforced by design. Extending our convergent privacy framework to a broader class of GNNs, including attention-based models (e.g., GAT), spectral convolution methods, and emerging graph transformers would be a natural next step, but new research problems will emerge. For instance, these models differ in their Lipschitz properties, aggregation operators, and feature mixing patterns, which may influence the achievable privacy amplification and the expressiveness–contractiveness trade-off. Developing a unified analysis for these graph families, or designing contractive variants of non-message-passing architectures, is an open and promising direction.

White-box attacks and defenses. Our privacy auditing focuses on black-box membership threats, which align with the theoretical guarantees of perturbed message passing. However, stronger adversaries with white-box access to gradients, intermediate embeddings, or partial training states can mount reconstruction, inversion, or property inference attacks that fall outside our current threat model. Prior work has shown that gradient-based attacks can recover fine-grained structural information, especially in over-parameterized models. Investigating the extent to which contractiveness mitigates these stronger threats, and designing DP mechanisms that remain robust under partial or full white-box exposure, warrant future research. Such analyses may require combining CARIBOU with complementary techniques such as gradient perturbation, secure aggregation, or private feature compression.

VIII. Related Works

Multi-layer GNNs. Recent literature shows that multi-layer GNNs hold significant potential for modeling long-range dependencies and complex relational structures crucial for many real-world applications. Node labels and attributes may depend on distant nodes, necessitating the aggregation of information over larger receptive fields [64] through multi-layer GNNs. Notably, Li et al. [23] demonstrated, through the 1000-layer GNN, that increasing the network depth attains substantial gains in accuracy, *e.g.*, from 72% with shallow GNNs to 88% with hundred- and thousand-layer GNNs, by capturing distant features. However, enforcing DP in multi-layer GNNs is particularly challenging, as these GNN models aggregate node embeddings over deeper layers and broader neighborhoods. Current research still lacks an effective solution to injecting DP noise to multi-layer GNNs with privacy-utility balance.

Differentially private GNNs. Graphs consist of edges and nodes. Corresponding to instance-level DP [9, 65, 53], the "instance" of graphs can be an edge or a node, naturally called edge-level DP (EDP) and node-level DP (NDP). GNNs have emerged as a key approach for applications over graph-structured data, such as intrusion detection [66, 67], social recommendation [68], and drug discovery [69]. Sharing trained GNN model can lead to privacy risks [7, 5], typically membership inference attack (MIA) [55, 5]. MIA stems from "overfitting", where models can memorize training memberships [70], either an edge or a node. Consequently, GNNs can leak sensitive information about their edge- or node-level neighbors.

To address these risks, existing research works [27, 33] have integrating DP with GNNs to achieve EDP and NDP. One research direction is to utilize graph perturbation (e.g., LPGNet [32] and LapGraph [17]) through a randomized response mechanism and adding discrete DP noise to the adjacency matrix. Then, the perturbed graph is passed to GNNs for subsequent training tasks, where the graph perturbation is required only once and also irrelevant to the GNN architectures. However, the GNN model utility is low when being trained over a perturbed graph when the privacy budget is tight, for example, $< 40\%$ accuracy of $\epsilon = 1, 2, 3, 4$ reported in LPGNet [32].

To improve utility, *perturbed message-passing mechanism* (PMP) [33] has been proposed by adding the calibrated Gaussian noise to the message-passing layer, and DPDGC perturbs the decoupled graph convolution [27]. As PMP realizes a better trade-off of privacy and utility, our work extends the research line of PMP. Table I presents a comprehensive comparison. Albeit their efforts on EDP and NDP, leveraging the contractive hidden node embeddings in private GNNs for amplifying privacy remains an underexplored avenue; thus, CARIBOU fills this gap.

IX. Conclusion

In this study, we provide a theoretical analysis establishing a convergent privacy budget for private deeper GNNs. Our analysis addresses a longstanding limitation in perturbed message-passing architectures, namely, the linear accumulation of noise with depth, by showing that privacy loss can remain bounded as the number of layers increases. Consequently, deeper models can be deployed with a significantly improved privacy-utility trade-off. Our analysis is broadly applicable, requiring only two conditions that are commonly satisfied in practice: the use of hidden intermediate states (also a standard design choice) and contractive message passing layers, which are often observed empirically.

To demonstrate the practical implications of our theory, we introduce a novel private GNN framework, CARIBOU, which incorporates a simple yet effective Contractive Graph Layer (CGL) that theoretically guarantees the contractiveness required by our analysis. CARIBOU further integrates optimized privacy budgeting, and modular auditing mechanisms to deliver strong privacy guarantees while preserving model utility. Empirical results show that CARIBOU substantially improves the privacy-utility trade-off and enhances robustness to membership inference attacks.

Acknowledgment

We thank the reviewers for their insightful and constructive comments. The authors of the University of California, Irvine are partially supported by NSF CNS-2220434 and TIP-2453148. Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations, ICLR*. OpenReview.net, 2017.

[2] Y. Zhang, H. Gao, J. Pei, and H. Huang, "Robust self-supervised structural graph neural network for social network prediction," in *The ACM Web Conference, WWW*, 2022, pp. 1352–1361.

[3] H. Chen, Y. Bei, Q. Shen, Y. Xu, S. Zhou, W. Huang, F. Huang, S. Wang, and X. Huang, "Macro graph neural networks for online billion-scale recommender systems," in *ACM on Web Conference, WWW*. ACM, 2024, pp. 3598–3608.

[4] F. Wu, Y. Long, C. Zhang, and B. Li, "LINKTELLER: recovering private edges from graph neural networks via influence analysis," in *IEEE Symposium on Security and Privacy, SP*. IEEE, 2022, pp. 2005–2024.

[5] J. Ye, A. Maddi, S. K. Murakonda, V. Bindschaedler, and R. Shokri, "Enhanced membership inference attacks against machine learning models," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022, pp. 3093–3106.

[6] X. He, R. Wen, Y. Wu, M. Backes, Y. Shen, and Y. Zhang, "Node-level membership inference attacks against graph neural networks," *CoRR*, vol. abs/2102.05429, 2021.

[7] Z. Zhang, M. Chen, M. Backes, Y. Shen, and Y. Zhang, "Inference attacks against graph neural networks," in *USENIX Security Symposium, USENIX Security*. USENIX Association, 2022, pp. 4543–4560.

[8] L. Sun, Y. Dou, C. Yang, K. Zhang, J. Wang, P. S. Yu, L. He, and B. Li, "Adversarial attack and defense on graph data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 7693–7711, 2023.

[9] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 2017, pp. 263–275.

[10] L. Wasserman and S. Zhou, "A statistical framework for differential privacy," *Journal of the American Statistical Association*, vol. 105, no. 489, pp. 375–389, 2010.

[11] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 84, no. 1, pp. 3–37, 2022.

[12] X. Tang, A. Panda, V. Sehwag, and P. Mittal, "Differentially private image classification by learning priors from random processes," in *Annual Conference on Neural Information Processing Systems*, 2023.

[13] Z. Yang, Y. Zhang, Y. Zheng, X. Tian, H. Peng, T. Liu, and B. Han, "Fedfed: Feature distillation against data heterogeneity in federated learning," in *Annual Conference on Neural Information Processing Systems*, 2023.

[14] D. Sun, J. Q. Chen, C. Gong, T. Wang, and Z. Li, "Netdpsyn: Synthesizing network traces under differential privacy," in *ACM on Internet Measurement Conference, IMC*. ACM, 2024, pp. 545–554.

[15] S. Sajadmanesh and D. Gatica-Perez, "When differential privacy meets graph neural networks," *arXiv preprint arXiv:2006.05535*, 2020.

[16] R. Wu, M. Zhang, L. Lyu, X. Xu, X. Hao, X. Fu, T. Liu, T. Zhang, and W. Wang, "Privacy-preserving design of graph neural networks with applications to vertical federated learning," *CoRR*, vol. abs/2310.20552, 2023.

[17] F. Wu, Y. Long, C. Zhang, and B. Li, "LINKTELLER: recovering private edges from graph neural networks via influence analysis," in *IEEE Symposium on Security and Privacy, SP*, 2022, pp. 2005–2024.

[18] M. Iftikhar and Q. Wang, "dk-projection: Publishing graph joint degree distribution with node differential privacy," in *Advances in Knowledge Discovery and Data Mining - Pacific-Asia Conference, PAKDD, Part II*, vol. 12713, 2021, pp. 358–370.

[19] W. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *International Conference on Management of Data, SIGMOD*. ACM, 2016, pp. 123–138.

[20] J. Feng, Y. Chen, F. Li, A. Sarkar, and M. Zhang,

"How powerful are k-hop message passing graph neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4776–4790, 2022.
[21] V. P. Dwivedi, L. Rampásek, M. Galkin, A. Parviz, G. Wolf, A. T. Luu, and D. Beaini, "Long range graph benchmark," in *Annual Conference on Neural Information Processing Systems*, 2022.
[22] J. Tönshoff, M. Ritzert, E. Rosenbluth, and M. Grohe, "Where did the gap go? reassessing the long-range graph benchmark," *Trans. Mach. Learn. Res.*, vol. 2024, 2024.
[23] G. Li, M. Müller, B. Ghanem, and V. Koltun, "Training graph neural networks with 1000 layers," in *International Conference on Machine Learning, ICML*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 6437–6449.
[24] D.-L. Nguyen, T.-H. Nguyen, T.-H. Do, and M. Yoo, "Probability-based multi-hop diffusion method for influence maximization in social networks," *Wireless Personal Communications*, vol. 93, no. 4, pp. 903–916, 2017.
[25] G. Liu, Q. Yang, H. Wang, X. Lin, and M. P. Wittie, "Assessment of multi-hop interpersonal trust in social networks by three-valued subjective logic," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, 2014, pp. 1698–1706.
[26] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *AAAI*. AAAI Press, 2020, pp. 3438–3445.
[27] 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.
[28] R. Chourasia, J. Ye, and R. Shokri, "Differential privacy dynamics of langevin diffusion and noisy gradient descent," in *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2021, pp. 14 771–14 781.
[29] J. Bok, W. J. Su, and J. Altschuler, "Shifted interpolation for differential privacy," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 4230–4266.
[30] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *International Conference on Machine Learning, ICML*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 1725–1735.
[31] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," *arXiv preprint arXiv:2006.05205*, 2020.
[32] 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.
[33] S. Sajadmanesh, A. S. Shamsabadi, A. Bellet, and D. Gatica-Perez, "Gap: Differentially private graph neural networks with aggregation perturbation," in *USENIX Security Symposium*, 2023.
[34] I. E. Olatunji, W. Nejdl, and M. Khosla, "Membership inference attack on graph neural networks," in *IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA*. IEEE, 2021, pp. 11–20.
[35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.
[36] A. Rao, N. A. Krishnan, and C. R. Rivero, "Using model calibration to evaluate link prediction in knowledge graphs," in *ACM on Web Conference, WWW*. ACM, 2024, pp. 2042–2051.
[37] P. Li, H. Wang, and C. Böhm, "Scalable graph classification via random walk fingerprints," in *IEEE International Conference on Data Mining, ICDM*. IEEE, 2024, pp. 231–240.
[38] M. Thürlemann and S. Riniker, "Anisotropic message passing: Graph neural networks with directional and long-range interactions," in *International Conference on Learning Representations, ICLR*. OpenReview.net, 2023.
[39] Z. Wu, P. Jain, M. A. Wright, A. Mirhoseini, J. E. Gonzalez, and I. Stoica, "Representing long-range context for graph neural networks with global attention," in *Annual Conference on Neural Information Processing Systems*, 2021, pp. 13 266–13 279.
[40] H. Ma, Y. Bian, Y. Rong, W. Huang, T. Xu, W. Xie, G. Ye, and J. Huang, "Cross-dependent graph neural networks for molecular property prediction," *Bioinform.*, vol. 38, no. 7, pp. 2003–2009, 2022.
[41] W. Zhong, C. He, C. Xiao, Y. Liu, X. Qin, and Z. Yu, "Long-distance dependency combined multi-hop graph neural networks for protein-protein interactions prediction," *BMC Bioinform.*, vol. 23, no. 1, p. 521, 2022.
[42] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography, Third Theory of Cryptography Conference, TCC*, ser. Lecture Notes in Computer Science, vol. 3876. Springer, 2006, pp. 265–284.
[43] V. Karwa, S. Raskhodnikova, A. D. Smith, and G. Yaroslavtsev, "Private analysis of graph structure," *Proc. VLDB Endow.*, vol. 4, no. 11, pp. 1146–1157, 2011.
[44] J. Schuchardt, M. Stoian, A. Kosmala, and S. Günnemann, "Unified mechanism-specific amplification by subsampling and group privacy amplification," in *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2024.
[45] D. Chen, T. H. Schulz, and K. M. Borgwardt, "Learning long range dependencies on graphs via random walks," *CoRR*, vol. abs/2406.03386, 2024.
[46] L. Zhang, H. Wang, M. C. Muniz, A. Z. Panagiotopoulos, R. Car *et al.*, "A deep potential model with long-range electrostatic interactions," *The Journal of Chemical Physics*, vol. 156, no. 12, 2022.
[47] X. Li, Z. Zhou, J. Yao, Y. Rong, L. Zhang, and B. Han, "Neural atoms: Propagating long-range interaction in molecular graphs through efficient communication chan-

nel," *arXiv preprint arXiv:2311.01276*, 2023.
[48] F. Gu, H. Chang, W. Zhu, S. Sojoudi, and L. E. Ghaoui, "Implicit graph neural networks," in *Annual Conference on Neural Information Processing Systems*, 2020.
[49] J. Liu, K. Kawaguchi, B. Hooi, Y. Wang, and X. Xiao, "Eignn: Efficient infinite-depth graph neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 762–18 773, 2021.
[50] R. Wu, G. Fang, M. Zhang, Q. Pan, T. Liu, and W. Wang, "On provable privacy vulnerabilities of graph representations," *Advances in Neural Information Processing Systems*, vol. 37, pp. 90 891–90 933, 2024.
[51] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.
[52] Z. Wang, Y. Li, B. Ding, Y. Li, and Z. Wei, "Exploring neural scaling law and data pruning methods for node classification on large-scale graphs," in *ACM on Web Conference, WWW*. ACM, 2024, pp. 780–791.
[53] J. Ye and R. Shokri, "Differentially private learning needs hidden state (or much faster convergence)," in *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2022.
[54] F. Wu, A. H. S. Jr., T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *International Conference on Machine Learning, ICML*, ser. Proceedings of Machine Learning Research, vol. 97, 2019, pp. 6861–6871.
[55] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr, "Membership inference attacks from first principles," in *IEEE Symposium on Security and Privacy, SP*, 2022, pp. 1897–1914.
[56] J. M. Altschuler and K. Talwar, "Privacy of noisy stochastic gradient descent: More iterations without more privacy loss," in *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2022.
[57] J. M. Altschuler, J. Bok, and K. Talwar, "On the privacy of noisy stochastic gradient descent for convex optimization," *SIAM J. Comput.*, vol. 53, no. 4, pp. 969–1001, 2024.
[58] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," *CoRR*, vol. abs/1811.05868, 2018.
[59] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.
[60] A. L. Traud, P. J. Mucha, and M. A. Porter, "Social structure of facebook networks," *CoRR*, vol. abs/1102.2166, 2011.
[61] A. Daigavane, G. Madan, A. Sinha, A. G. Thakurta, G. Aggarwal, and P. Jain, "Node-level differentially private graph neural networks," *CoRR*, vol. abs/2111.15521, 2021.
[62] X. He, J. Jia, M. Backes, N. Z. Gong, and Y. Zhang, "Stealing links from graph neural networks," in *USENIX security symposium*, 2021, pp. 2669–2686.
[63] T. Tang, Y. Niu, S. Avestimehr, and M. Annavaram, "Edge private graph neural networks with singular value perturbation," *Proc. Priv. Enhancing Technol.*, vol. 2024, no. 3, pp. 391–406, 2024.
[64] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.
[65] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *CCS*, 2016.
[66] G. Apruzzese, P. Laskov, and J. Schneider, "Sok: Pragmatic assessment of machine learning for network intrusion detection," in *IEEE European Symposium on Security and Privacy, EuroS&P*, 2023, pp. 592–614.
[67] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *IEEE Symposium on Security and Privacy, SP*, 2010, pp. 305–316.
[68] L. Xia, Y. Shao, C. Huang, Y. Xu, H. Xu, and J. Pei, "Disentangled graph social recommendation," in *International Conference on Data Engineering, ICDE*, 2023, pp. 2332–2344.
[69] Z. Zhong and D. Mottin, "Knowledge-augmented graph machine learning for drug discovery: From precision to interpretability," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*, 2023, pp. 5841–5842.
[70] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

Appendix

A. Proof of results in Section V-B

Proof of Proposition 1. Let $\boldsymbol{Y}^{(k-1)}, \boldsymbol{Y}'^{(k-1)} \in \mathcal{K}$ be two inputs to the message passing operator MP_G at layer k. Since $k \geq 2$, the residue term $\beta \boldsymbol{X}^{(0)}$ is independent of the input $\boldsymbol{Y}^{(k-1)}$ and $\boldsymbol{Y}'^{(k-1)}$, and thus does not affect the Lipschitz constant. We can write the difference between the outputs of CGL as follows:

$$
\begin{aligned}
&\|\mathsf{CGL}(\boldsymbol{Y}^{(k-1)}) - \mathsf{CGL}(\boldsymbol{Y}'^{(k-1)})\| \\
&\leq \|C_{\mathsf{L}}(\alpha_1 \hat{\boldsymbol{A}} \boldsymbol{Y}^{(k-1)} + \alpha_2 \mathsf{Mean}(\boldsymbol{Y}^{(k-1)})) + \beta \boldsymbol{X}^{(0)} \\
&\qquad - C_{\mathsf{L}}(\alpha_1 \hat{\boldsymbol{A}} \boldsymbol{Y}'^{(k-1)} + \alpha_2 \mathsf{Mean}(\boldsymbol{Y}'^{(k-1)})) - \beta \boldsymbol{X}^{(0)}\| \\
&\leq C_{\mathsf{L}} \|\alpha_1 (\hat{\boldsymbol{A}} \boldsymbol{Y}^{(k-1)} - \hat{\boldsymbol{A}} \boldsymbol{Y}'^{(k-1)}) \\
&\qquad + \alpha_2 (\mathsf{Mean}(\boldsymbol{Y}^{(k-1)}) - \mathsf{Mean}(\boldsymbol{Y}'^{(k-1)}))\| \\
&\leq C_{\mathsf{L}} (\alpha_1 + \alpha_2) \|\boldsymbol{Y}^{(k-1)} - \boldsymbol{Y}'^{(k-1)}\| \\
&= C_{\mathsf{L}} \|\boldsymbol{Y}^{(k-1)} - \boldsymbol{Y}'^{(k-1)}\|,
\end{aligned}
$$

where the second line follows from the fact that the operator norms of $\hat{\boldsymbol{A}}$ and Mean are bounded by 1. □

Proof of Theorem 4. Let $\boldsymbol{G}, \boldsymbol{G}'$ be two edge adjacent graphs and $\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}'$ be the corresponding adjacency matrices of $\boldsymbol{G}, \boldsymbol{G}'$ respectively. Without loss of generality, we assume that the edge e_{uv} is added to $\boldsymbol{G}$ to form $\boldsymbol{G}'$ for two nodes u and v. Then the CGL layer updates the node features as follows:

$$\boldsymbol{X}^{(k)} = C_{\mathsf{L}}(\alpha_1 \hat{\boldsymbol{A}}\boldsymbol{X}^{(k-1)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k-1)})) + \beta \boldsymbol{X}^{(0)},$$
$$\boldsymbol{X}'^{(k)} = C_{\mathsf{L}}(\alpha_1 \hat{\boldsymbol{A}}'\boldsymbol{X}'^{(k-1)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}'^{(k-1)})) + \beta \boldsymbol{X}^{(0)}.$$

The difference between the two outputs is given by the aggregation of $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{A}}'$. Then the edge-level sensitivity $\Delta_e(\mathsf{CGL})$ is the amount to bound $\|\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)} - \hat{\boldsymbol{A}}'\boldsymbol{X}'^{(k)}\|_F$. Since only one edge is added, the difference between $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{A}}'$ is only on the row corresponding to u and v.

For row u, we need to bound $\|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2$. For $(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u$, we can write it as

$$(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u = \frac{1}{d_u+1}\boldsymbol{X}_u^{(k)} + \sum_{w\in \boldsymbol{N}_u} \frac{1}{\sqrt{d_u+1}\sqrt{d_w+1}}\boldsymbol{X}_w^{(k)} \tag{9}$$

where d_u is the degree of node u in graph $\boldsymbol{G}$ and $\boldsymbol{N}_u$ is the neighbors of node u in graph $\boldsymbol{G}$. For $(\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u$, with the same notation for d_u and $\boldsymbol{N}_u$, we can write it as

$$\begin{aligned}(\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u &= \frac{1}{d_u+2}\boldsymbol{X}_u^{(k)} + \sum_{w\in \boldsymbol{N}_u} \frac{1}{\sqrt{d_u+2}\sqrt{d_w+1}}\boldsymbol{X}_w^{(k)} \\ &+ \frac{1}{\sqrt{d_u+2}\sqrt{d'_v+1}}\boldsymbol{X}_v^{(k)}\end{aligned} \tag{10}$$

where d'_v is the degree of node v in graph $\boldsymbol{G}'$.

Then there is

$$\begin{aligned}
&\|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2 \\
&\le \left\| \frac{1}{d_u+1}\boldsymbol{X}_u^{(k)} - \frac{1}{d_u+2}\boldsymbol{X}_u^{(k)} \right\|_2 \\
&+ \left\| \sum_{w\in \boldsymbol{N}_u} \frac{1}{\sqrt{d_u+1}\sqrt{d_w+1}}\boldsymbol{X}_w^{(k)} - \sum_{w\in \boldsymbol{N}_u} \frac{1}{\sqrt{d_u+2}\sqrt{d_w+1}}\boldsymbol{X}_w^{(k)} \right\|_2 \\
&+ \left\| \frac{1}{\sqrt{d_u+2}\sqrt{d'_v+1}}\boldsymbol{X}_v^{(k)} \right\|_2 \\
&\le \frac{1}{(d_u+1)(d_u+2)} + \sum_{w\in N_u} \frac{1}{\sqrt{d_w+1}}\left(\frac{1}{\sqrt{d_u+1}} - \frac{1}{\sqrt{d_u+2}}\right) \\
&+ \frac{1}{\sqrt{d_u+2}\sqrt{d'_v+1}} \\
&\le \frac{1}{(d_u+1)(d_u+2)} + \frac{d_u}{\sqrt{d_w+1}}\left(\frac{1}{\sqrt{d_u+1}} - \frac{1}{\sqrt{d_u+2}}\right) \\
&+ \frac{1}{\sqrt{d_u+2}\sqrt{d'_v+1}}
\end{aligned}$$

To bound $\frac{d_u}{\sqrt{d_w+1}}(\frac{1}{\sqrt{d_u+1}} - \frac{1}{\sqrt{d_u+2}})$, we study the monotonicity of the function $f(x) = \frac{x}{\sqrt{x+1}} - \frac{x}{\sqrt{x+2}}$ for $x > 0$. It turns out that $f(x)$ only has one positive critical point and is around $x = 2.9$, and when evaluated on integers, $f(x)$ increases from $x = 1$ to $x = 3$ and then decreases from $x = 3$ to ∞. Thus, when the minimum degree $D_{\min}$ of $\boldsymbol{G}$ is larger than 3, the function $f(x)$ is maximized at $x = D_{\min}$, and we have

$$\begin{aligned}
&\|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2 \\
&\qquad \le \frac{1}{(D_{\min}+1)(D_{\min}+2)} \\
&\qquad\quad + \frac{D_{\min}}{\sqrt{D_{\min}+1}}\left(\frac{1}{\sqrt{D_{\min}+1}} - \frac{1}{\sqrt{D_{\min}+2}}\right) \\
&\qquad\quad + \frac{1}{\sqrt{D_{\min}+2}\sqrt{D_{\min}+1}}
\end{aligned}$$

where we use the fact that the minimum degree of $\boldsymbol{G}'$ is larger than that of $\boldsymbol{G}$. When $1 \le D_{\min} \le 3$, we can bound the function $f(x)$ by $f(3) = \frac{3}{\sqrt{4}} - \frac{3}{\sqrt{5}}$. This results in

$$\begin{aligned}
&\|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2 \\
&\le \frac{1}{(D_{\min}+1)(D_{\min}+2)} + \left(\frac{3}{\sqrt{4}} - \frac{3}{\sqrt{5}}\right)\frac{1}{\sqrt{D_{\min}+1}} \\
&\quad + \frac{1}{\sqrt{D_{\min}+2}\sqrt{D_{\min}+1}}
\end{aligned}$$

For notation convenience, we use $C(D_{\min})$ to denote the piecewise function of $D_{\min}$, which is defined as

$$C(D_{\min}) = \begin{cases} \frac{D_{\min}}{\sqrt{D_{\min}+1}} - \frac{D_{\min}}{\sqrt{D_{\min}+2}} & D_{\min} > 3 \\ (\frac{3}{\sqrt{4}} - \frac{3}{\sqrt{5}}) & 1 \le D_{\min} \le 3 \end{cases} \tag{11}$$

Therefore, the effect of modifying an edge on a single node u of $\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)}$ is bounded by

$$\begin{aligned}
&\|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2 \\
&\le \frac{1}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})}{\sqrt{D_{\min}+1}} \\
&\quad + \frac{1}{\sqrt{D_{\min}+2}\sqrt{D_{\min}+1}}
\end{aligned} \tag{12}$$

The same analysis can be applied to the row v of $\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)}$ and $\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)}$. The edge sensitivity $\Delta_e(\mathsf{CARIBOU})$ can then be bounded as the following:

$$\begin{aligned}
&\Delta_e(\mathsf{CARIBOU}) := \max_{\boldsymbol{G},\boldsymbol{G}'} \|\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)} - \hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)}\|_F \\
&\le \alpha_1 C_{\mathsf{L}} \sqrt{\begin{aligned}&\|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2^2 \\ &+ \|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_v - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_v\|_2^2\end{aligned}} \\
&\le \sqrt{2}\alpha_1 C_{\mathsf{L}} \Bigg(\frac{1}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})}{\sqrt{D_{\min}+1}} \\
&\qquad + \frac{1}{\sqrt{D_{\min}+2}\sqrt{D_{\min}+1}} \Bigg)
\end{aligned}$$

□

Proof of Theorem 5. Let $\boldsymbol{G}, \boldsymbol{G}'$ be two node adjacent graphs and $\hat{\boldsymbol{A}}, \hat{\boldsymbol{A}}'$ be the corresponding adjacency matrices of $\boldsymbol{G}, \boldsymbol{G}'$ respectively. Without loss of generality, we assume that the node v is added to $\boldsymbol{G}$ to form $\boldsymbol{G}'$ and connected to nodes N_v in $\boldsymbol{G}$. The layer updates the node features as follows:

$$\boldsymbol{X}^{(k)} = C_{\mathsf{L}}(\alpha_1 \hat{\boldsymbol{A}}\boldsymbol{X}^{(k-1)} + \alpha_2 \mathsf{Mean}(\boldsymbol{X}^{(k-1)})) + \beta \boldsymbol{X}^{(0)},$$
$$\boldsymbol{X}'^{(k)} = C_{\mathsf{L}}(\alpha_1 \hat{\boldsymbol{A}}'\boldsymbol{X}'^{(k-1)} + \alpha_2 \mathsf{Mean}'(\boldsymbol{X}'^{(k-1)})) + \beta \boldsymbol{X}^{(0)}.$$

The difference between the two outputs is given by the aggregation of $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{A}}'$ as well as the mean operator Mean and Mean$'$ since $\boldsymbol{G}'$ has one more node than $\boldsymbol{G}$. Then the node-level sensitivity $\Delta_v(\mathsf{CGL})$ can be bounded as follows:

$$\begin{aligned}
&\Delta_n(\mathsf{CGL}) \\
&= \max_{\boldsymbol{G},\boldsymbol{G}'} \|\mathsf{CGL}(\boldsymbol{X}^{(k)}) - \mathsf{CGL}'(\boldsymbol{X}^{(k)})\|_F \\
&\le \|\boldsymbol{X}'_v\|_2 + \sum_{u\in \boldsymbol{N}_v} \alpha_1 C_{\mathsf{L}} \|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2 \\
&+ \quad \sum_{u\in \boldsymbol{N}_v} \alpha_2 C_{\mathsf{L}} \|\mathsf{Mean}(\boldsymbol{X}^{(k)})_u - \mathsf{Mean}'(\boldsymbol{X}^{(k)})_u\|_2
\end{aligned}$$

For the first term $\|\boldsymbol{X}'_v\|_2$, it is bounded by 1 by constraint of $\mathcal{K}$. For the second term, we can argue similar as in the proof of Theorem 4. For nodes $u \in \boldsymbol{N}_v$, there is

$$\begin{aligned}
&\|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2 \\
&\le \left\| \frac{1}{d_u+1}\boldsymbol{X}_u^{(k)} - \frac{1}{d_u+2}\boldsymbol{X}_u^{(k)} \right\|_2 \\
&+ \left\| \sum_{w\in \boldsymbol{N}_u} \frac{1}{\sqrt{d_u+1}\sqrt{d_w+1}}\boldsymbol{X}_w^{(k)} - \sum_{w\in \boldsymbol{N}_u} \frac{1}{\sqrt{d_u+2}\sqrt{d_w+1}}\boldsymbol{X}_w^{(k)} \right\|_2 \\
&+ \left\| \frac{1}{\sqrt{d_u+2}\sqrt{d'_v+1}}\boldsymbol{X}_v^{(k)} \right\|_2 \\
&\le \frac{1}{(d_u+1)(d_u+2)} + \sum_{w\in N_u} \frac{1}{\sqrt{d_w+1}}\left(\frac{1}{\sqrt{d_u+1}} - \frac{1}{\sqrt{d_u+2}}\right) \\
&+ \frac{1}{\sqrt{d_u+2}\sqrt{d'_v+1}} \\
&\le \frac{1}{(d_u+1)(d_u+2)} + \frac{d_u}{\sqrt{d_w+1}}\left(\frac{1}{\sqrt{d_u+1}} - \frac{1}{\sqrt{d_u+2}}\right) \\
&+ \frac{1}{\sqrt{d_u+2}\sqrt{d'_v+1}} \\
&\le \frac{1}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})}{\sqrt{D_{\min}+1}} \\
&+ \frac{1}{\sqrt{D_{\min}+2}\sqrt{d'_v+1}}
\end{aligned}$$

Then the summation term $\sum_{u\in \boldsymbol{N}_v} \|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2$ can be bounded by

$$\begin{aligned}
&\sum_{u\in \boldsymbol{N}_v} \|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2 \\
&\le \sum_{u\in \boldsymbol{N}_v} \left(\frac{1}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})}{\sqrt{D_{\min}+1}} \right. \\
&\qquad \left. + \frac{1}{\sqrt{D_{\min}+2\sqrt{d'_v+1}}} \right) \\
&\le |d'_v| \left(\frac{1}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})}{\sqrt{D_{\min}+1}} \right. \\
&\qquad \left. + \frac{1}{\sqrt{D_{\min}+2\sqrt{d'_v+1}}} \right) \\
&\le \frac{\sqrt{d'_v}}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})\sqrt{d'_v}}{\sqrt{D_{\min}+1}} \\
&\qquad + \frac{\sqrt{d'_v}}{\sqrt{D_{\min}+2\sqrt{d'_v+1}}} \\
&\le \frac{\sqrt{d'_v}}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})\sqrt{d'_v}}{\sqrt{D_{\min}+1}} + \frac{1}{\sqrt{D_{\min}+2}} \\
&\le \frac{\sqrt{D_{\max}}}{(D_{\min}+1)(D_{\min}+2)} + \frac{C(D_{\min})\sqrt{D_{\max}}}{\sqrt{D_{\min}+1}} + \frac{1}{\sqrt{D_{\min}+2}},
\end{aligned}$$

where $D_{\max}$ is the maximum degree of the graph $\boldsymbol{G}$ and d'_v is the degree of node v in graph $\boldsymbol{G}'$. Additionally, there is

$$\begin{aligned}
&\|\mathsf{Mean}(\boldsymbol{X}^{(k)})_u - \mathsf{Mean}'(\boldsymbol{X}^{(k)})_u\|_2 \\
&= \left\| \frac{1}{|V|}\sum_{w\in V} \boldsymbol{X}_w^{(k)} - \frac{1}{|V|+1}\sum_{w\in V} \boldsymbol{X}_w^{(k)} - \frac{1}{|V|+1}\boldsymbol{X}_{v'}^{(k)} \right\|_2 \\
&\le \left\| \frac{1}{|V|(|V|+1)}\sum_{w\in V} \boldsymbol{X}_w^{(k)} - \frac{1}{|V|+1}\boldsymbol{X}_{v'}^{(k)} \right\|_2 \\
&\le \frac{2}{|V|+1},
\end{aligned}$$

where $|V|$ is the number of nodes in graph $\boldsymbol{G}$.

Then for the node-level sensitivity of one layer of CGL, we have

$$\begin{aligned}
&\Delta_n(\mathsf{CGL}) \\
&= \max_{\boldsymbol{G},\boldsymbol{G}'} \|\mathsf{CGL}(\boldsymbol{X}^{(k)}) - \mathsf{CGL}'(\boldsymbol{X}^{(k)})\|_F \\
&\le \|\boldsymbol{X}'_v\|_2 + \sum_{u\in \boldsymbol{N}_v} \alpha_1 C_{\mathsf{L}} \|(\hat{\boldsymbol{A}}\boldsymbol{X}^{(k)})_u - (\hat{\boldsymbol{A}}'\boldsymbol{X}^{(k)})_u\|_2 \\
&+ \quad \sum_{u\in \boldsymbol{N}_v} \alpha_2 C_{\mathsf{L}} \|\mathsf{Mean}(\boldsymbol{X}^{(k)})_u - \mathsf{Mean}'(\boldsymbol{X}^{(k)})_u\|_2 \\
&\le 1 + \alpha_1 C_{\mathsf{L}} \left(\frac{\sqrt{D_{\max}}}{(D_{\min}+1)(D_{\min}+2)} \right. \\
&\qquad \left. + \frac{C(D_{\min})\sqrt{D_{\max}}}{\sqrt{D_{\min}+1}} + \frac{1}{\sqrt{D_{\min}+2}} \right) \\
&+ \quad \alpha_2 C_{\mathsf{L}} \frac{2|V|}{|V|+1}
\end{aligned}$$

□