CEGA: A Cost-Effective Approach for Graph-Based Model Extraction and Acquisition

Zebin Wang [1] **Menghan Lin** [2] **Bolin Shen** [3] **Ken Anderson** [3] **Molei Liu** [4] **Tianxi Cai** [1] **Yushun Dong** [3]

Abstract

Graph Neural Networks (GNNs) have demonstrated remarkable utility across diverse applications, and their growing complexity has made Machine Learning as a Service (MLaaS) a viable platform for scalable deployment. However, this accessibility also exposes GNN to serious security threats, most notably model extraction attacks (MEAs), in which adversaries strategically query a deployed model to construct a high-fidelity replica. In this work, we evaluate the vulnerability of GNNs to MEAs and explore their potential for cost-effective model acquisition in non-adversarial research settings. Importantly, adaptive node querying strategies can also serve a critical role in research, particularly when labeling data is expensive or time-consuming. By selectively sampling informative nodes, researchers can train high-performing GNNs with minimal supervision, which is particularly valuable in domains such as biomedicine, where annotations often require expert input. To address this, we propose a node querying strategy tailored to a highly practical yet underexplored scenario, where bulk queries are prohibited, and only a limited set of initial nodes is available. Our approach iteratively refines the node selection mechanism over multiple learning cycles, leveraging historical feedback to improve extraction efficiency. Extensive experiments on benchmark graph datasets demonstrate our superiority over comparable baselines on accuracy, fidelity, and F1 score under strict query-size constraints. These results highlight both the susceptibility of deployed GNNs to extraction attacks and the promise of ethical, efficient GNN acquisition methods to support low-resource research environments. Our implementation is publicly available at `https://github.com/LabRAI/CEGA`.

1. Introduction

Graph Neural Networks (GNNs) have achieved remarkable performance in a variety of applications powered by graph learning, such as molecular graph structure analysis (Sun et al., 2022; Wang et al., 2023; Zang et al., 2023; Zhao et al., 2024), fraud detection (Qin et al., 2022; Cheng et al., 2024; Motie & Raahemi, 2024; Lou et al., 2025), and healthcare diagnostics (Ahmedt-Aristizabal et al., 2021; Lu & Uddin, 2021; Zafeiropoulos et al., 2023; Paul et al., 2024). However, as GNNs grow in complexity and computational demands, training them from scratch becomes increasingly prohibitive due to rising computational costs (Abbahaddou et al., 2024; Kose et al., 2024). To address this, graph-based Machine Learning as a Service (MLaaS) has emerged as a cost-effective alternative, allowing users to access powerful pre-trained GNN models via APIs provided by service providers (Liu et al., 2022; Wu et al., 2023a; 2024).

Nevertheless, despite the advantages of graph-based MLaaS, such an inference paradigm also exposes GNN models to serious security risks, with GNN-based model extraction attacks (MEAs) posing a particularly significant threat (Wu et al., 2023a; 2024). Specifically, the goal of a model extraction attacker is to replicate the functionality of a GNN model owned by the service provider (i.e., the target model) by strategically querying it and using the responses to construct a local replica (i.e., the extracted model) (Shen et al., 2022; Wu et al., 2022). Such graph-based MEAs can lead to severe consequences such as copyright violations and patent infringement, especially in high-stake applications. For example, in the pharmaceutical industry, GNNs are widely used to predict molecular-level drug-target interactions (DTIs) (Wieder et al., 2020; Zhang et al., 2022; Tran et al., 2022). In this context, graph-based MLaaS provides

[1]Department of Biostatistics, T. H. Chan School of Public Health, Harvard University, Boston, Massachusetts, USA [2]Department of Statistics, Florida State University, Tallahassee, Florida, USA [3]Department of Computer Science, Florida State University, Tallahassee, Florida, USA [4]Department of Biostatistics, Mailman School of Public Health, Columbia University, New York, New York, USA. Correspondence to: Tianxi Cai <tcai@hsph.harvard.edu>, Yushun Dong <yd24f@fsu.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

pharmaceutical companies with a cost-effective and efficient platform for conducting related studies (Ahmedt-Aristizabal et al., 2021; Lu & Uddin, 2021; Vora et al., 2023). However, MEAs targeting such GNNs pose a serious risk to proprietary data, threatening trade secrets and potentially enabling unauthorized redistribution and unfair competition. These concerns may ultimately result in substantial financial and reputational damage (Bessen & Meurer, 2008; Nealey et al., 2015; Armstrong, 2016). Consequently, appropriately understanding and managing the threat of MEAs against GNNs has become a pressing concern (Zhao et al., 2025).

Beyond malicious uses, the research-driven acquisition of GNN functionality offers significant value, particularly for tailoring models to specialized downstream applications. A compelling example is the analysis of knowledge graphs (KGs) constructed from electronic health records (EHRs), where nodes represent clinical concepts—such as diagnoses, medications, and procedures—and edges capture relationships based on medical ontologies or empirical patterns of co-occurrence (Wang et al., 2014; Hong et al., 2021). In EHR-based KG research, deploying graph-based models for specific inference tasks within local health systems is often hampered by practical constraints, including the incompleteness of local database, limitations on data sharing across institutions, and heterogeneity in clinical practice patterns and patient populations, which can significantly limit model generalizability (Zhou et al., 2022; 2025). In such settings, effectively extracting and acquiring a well-trained target model developed on large-scale EHR data presents a powerful alternative to the costly and often impractical process of training from scratch using local EHR data (Lin et al., 2023a; Gan et al., 2025). This strategy not only improves computational efficiency but also enables advanced applications such as non-linear statistical inference on clinical knowledge graphs and ontology-informed learning for downstream applications (Xu et al., 2023; Liu et al., 2024).

In response to the pressing need outlined above, it is essential to systematically investigate strategies for extracting and acquiring graph-based model functionality. On the one hand, such efforts enable rigorous assessment of the severity of MEA threats to MLaaS platforms and inform the development of robust defense mechanisms (Zhang & Zitnik, 2020; Mujkanovic et al., 2022; Ennadir et al., 2023; Dong et al., 2024; Cheng et al., 2025). On the other hand, they also support the efficiency and feasibility of research-oriented GNN acquisition, as demonstrated in recent work on surrogate learning and transfer-based GNN extraction (Huo et al., 2023; Oloulade et al., 2023). However, despite the urgency and potential impact of such research, designing systematic strategies for extracting and acquiring GNN functionality remains a non-trivial task. In particular, we face two fundamental challenges:

(1) *Stringent budget and query batch size constraints.* First, excessive querying incurs substantial computational and financial costs under the pay-per-query basis, making large-scale extraction on well-trained MLaaS models economically unfeasible (Hou et al., 2019; Gong et al., 2020; Wu et al., 2023b). Second, querying in bulky batches risks violating MLaaS user agreements or triggering security alerts, as many providers implement monitoring mechanisms to identify and block potentially adversarial queries (Brundage et al., 2018; Juuti et al., 2019).

(2) *Structural dependency between nodes.* First, nodes can naturally exhibit various types of dependencies between each other in real-world graphs, depending on what types of semantics the edges encode (Zhou et al., 2020; Wu et al., 2021). Second, these dependencies across a broad localized area in the graph topology can collectively influence the information that the extracted model can acquire (Ju et al., 2024; Kahn et al., 2025).

Multiple research works have taken early steps to explore model extraction against GNNs for node-level graph learning tasks (DeFazio & Ramesh, 2019; Shen et al., 2022; Wu et al., 2022). However, these studies overlook the practical constraints on budget and batch size. More recently, several research works attempted to handle query budget limitations on MEAs (Shi et al., 2017; Liu et al., 2023; Karmakar & Basu, 2023). However, these approaches can hardly be generalized to graph learning tasks, as they often overlook the fact that GNNs can embed deeper information to the graph data during processing, even when certain features are absent or filtered out from the input (Dong et al., 2025). Therefore, the study of addressing the two practical challenges above specifically tailored for node-level graph learning tasks remains nascent.

To address the aforementioned challenges, we propose a targeted approach for the extraction and acquisition of GNN functionality, termed Cost-Efficient Graph Acquisition (CEGA). Our framework is specifically designed to balance effectiveness and efficiency in acquiring GNNs under realistic constraints. Without loss of generality, we focus on GNNs performing node classification, one of the most widely studied and fundamental tasks in node-level graph learning. Specifically, to overcome the challenge of budget and query batch size constraints, CEGA is designed to incorporate historical information from the initial and previous queries, starting with a very limited number of queries, in each of its iterations to improve its informativeness in node selection. To overcome the challenge of structural dependency between nodes, we prioritize nodes with high structural centrality to ensure queries can capture information that aligns with the localized graph topology. Furthermore, we introduce a diversity metric to prevent query clustering at the structural level and improve stability. Extensive

empirical experiments on real-world benchmark datasets demonstrated the superiority of the proposed CEGA framework in extracting the target GNN models under realistic query constraints, delivering key practical significance over existing alternatives.

In summary, the contributions of this paper are three-fold:

- **Novel Problem Formulation**: We introduce the problem of *GNN Model Extraction With Limited Budgets* specifically within the context of node-level graph learning tasks. This formulation offers a more realistic and practical setting for GNN model extraction compared to prior work in model extraction.
- **Comprehensive Methodology Design**: We present a novel framework for GNN model extraction and acquisition that dynamically identifies the most informative queries throughout the training process. Our approach integrates guidance based on three complementary criteria: *representativeness*, *uncertainty*, and *diversity*, enabling efficient and effective query selection.
- **Extensive Empirical Evaluation**. We conduct comprehensive experiments on real-world graph datasets to demonstrate the effectiveness of the proposed CEGA framework. Evaluation metrics include both model faithfulness and downstream utility, showing CEGA's superiority over existing alternatives.

2. Preliminaries

Notations. Suppose that we have a GNN model f_{T} trained on the target graph $\mathcal{G}_{\mathrm{T}} = \{\mathcal{V}_{\mathrm{T}}, \mathcal{E}_{\mathrm{T}}\}$, where $\mathcal{E}_{\mathrm{T}}$ denotes the edges of $\mathcal{G}_{\mathrm{T}}$. In the acquisition process, we assume knowledge to a pool of candidate nodes for querying, denoted as $\mathcal{V}_{\mathrm{a}}$, and a respective graph structure, denoted as $\mathcal{G}_{\mathrm{a}}$. We consider an iterative node querying approach with Γ learning cycles in total. We denote the initial query set as $\mathcal{V}_0$, with budget $\mathcal{I} = |\mathcal{V}_0|$. On the γth iterative cycle with $\gamma \in \{1, 2, ..., \Gamma\}$, we use $\mathcal{V}_{\gamma-1}$ to denote the collection of nodes queried in previous cycles, where $\mathcal{V}_0 \subsetneq \mathcal{V}_1 \subsetneq \mathcal{V}_2 \subsetneq ... \subsetneq \mathcal{V}_\Gamma \subsetneq \mathcal{V}_{\mathrm{a}}$. In this cycle, we query κ nodes from the candidate set $\mathcal{V}_{\mathrm{a}} \backslash \mathcal{V}_{\gamma-1}$, with capacity $n_{\gamma-1} = \left|\mathcal{V}_{\mathrm{a}} \backslash \mathcal{V}_{\gamma-1}\right|$. The attributes of the nodes belonging to $\mathcal{V}_{\mathrm{a}} \backslash \mathcal{V}_{\gamma-1}$ are denoted as $\mathcal{X}_{\gamma-1} = \{\mathbf{x}_{\gamma-1}^{(1)}, \mathbf{x}_{\gamma-1}^{(2)}, ..., \mathbf{x}_{\gamma-1}^{(n_{\gamma-1})}\}$, where each $\mathbf{x}_{\gamma-1}^{(j)}$ is an attribute vector with d-dimensions. For convenience, we denote the respective attribute of some node $v \in \mathcal{V}_{\mathrm{a}} \backslash \mathcal{V}_{\gamma-1}$ as $\mathbf{x}_{\gamma-1}^{(v)}$. The respective outcome for node v in a model f is denoted as $\widehat{\mathbf{y}}_v = f(\mathbf{x}^{(v)}, \mathcal{G}_{\mathrm{a}}) = \{\widehat{y}_v^{(1)}, \widehat{y}_v^{(2)}, ..., \widehat{y}_v^{(C)}\}$, where $\widehat{\mathbf{y}}_v$ is a probability vector of length C representing the softmax scores for each class. Our notation system is compatible with existing works in the context of MEAs against GNNs, as highlighted by existing work such as (Wu et al., 2022; Shen et al., 2022; Wu et al., 2023a).

Background. Existing research works have rarely discussed GNN-based MEA contexts in real-world settings. In our paper, we expect to gain practical significance by considering a realistic setup to extract GNNs. In our setting, we pose upper limits on (1) the initial query budget $\mathcal{I}$, (2) the per-cycle query budget κ, and (3) the overall budget B. Our setting is more realistic compared with existing ones that rely on simultaneous high-volume queries since excessive queries impose significant costs, and frequent large-scale queries are likely to alert the maintainers of the target model. We focus on node classification tasks, which are among the most widely studied problems in node-level graph learning, as previously investigated by (Dong et al., 2023; Luan et al., 2023; Zhang et al., 2023; Li et al., 2024).

Goal of Acquisition. The researchers aim to extract a model f_{a} that closely replicates the behavior of target GNN model f_{T} using a limited number of queries constrained by an initial query budget $\mathcal{I}$, a per-cycle query budget κ, and an overall query budget B. Here, the similarity in their behavior is generally measured by the ratio of the same input-output pairs.

We then formulate the problem of *GNN Model Extraction With Limited Budgets* in Problem 1.

Problem 1. *(GNN Model Extraction With Limited Budgets). Given a target GNN model f_{T} and available prior knowledge $\{\mathcal{X}_{\mathrm{a}}, \mathcal{G}_{\mathrm{a}}\}$, the objective is to achieve an extracted model f_{a} with behaviors being as similar to f_{T} as possible for any given test node $v \in \mathcal{V}_{\mathrm{T}}$, while adhering to the constraints of limited initial query budget $\mathcal{I}$, per-cycle query budget κ, and total query budget B.*

3. Methodology

3.1. Overview

In this section, we introduce CEGA, an active sampling framework designed to extract and acquire GNN behaviors efficiently. CEGA employs a multilevel analysis strategy that iteratively selects informative nodes by leveraging prior heuristics derived from the initial query set $\mathcal{V}_0$ and the $\gamma - 1$ batches of previously queried nodes. these historical insights are summarized by an interim model $f_{\gamma-1}$, which guides the selection process in iteration γ.

Specifically, CEGA is designed to conduct node selection by incorporating three key objectives: (1) *Representativeness:* The queried nodes should capture the structural essence of the graph, facilitating an accurate reconstruction of model behavior across the network. (2) *Uncertainty:* Nodes with high uncertainty, as indicated by historical predictions, are prioritized, as they likely reside near decision boundaries of the interim prediction model f_γ. (3) *Diversity:* To avoid excessive clustering, selected nodes should be diverse in their distribution across the graph, ensuring a comprehensive

exploration of the underlying structure.

To achieve these goals, CEGA is equipped with three objectives, $\mathcal{L}_1^\gamma(v, \mathcal{G}_\text{a})$, $\mathcal{L}_2^\gamma(v, \mathcal{G}_\text{a})$, and $\mathcal{L}_3^\gamma(v, \mathcal{G}_\text{a})$, evaluating the tendency of selecting a node v from the candidate set $\mathcal{V}_\text{a} \backslash \mathcal{V}_{\gamma-1}$ in each querying cycle γ. Nodes are adaptively ranked and selected based on their combined ranking across these three criteria, ensuring an efficient and cost-effective querying strategy.

3.2. The Proposed Framework of CEGA

The CEGA framework begins by building a primitive initial GNN-based prediction model f_0 with $\mathcal{I}$ initial queried nodes. In each learning cycle, CEGA selects κ nodes with the highest comprehensive rank through an adaptive node selection method based on the representativeness, uncertainty, and diversity of the nodes. A new interim model f_γ involving all nodes queried in the past and new nodes selected for query in the same cycle is trained as a summarization of existing historical information to guide further queries. We perform such cycles iteratively until the budget limit B is reached. Finally, we evaluate the performance of CEGA by training GNN models with queried nodes.

Initialization. In CEGA, we randomly select $\mathcal{I}$ initial nodes from the node pool for acquisition $\mathcal{V}_\text{a}$. A random selection of initial nodes reduces systematic bias and ensures comparability between CEGA and other approaches mentioned in the existing literature (Cai et al., 2017; Zhang et al., 2021).

Graph Structure-Based Analysis for Representativeness. To ensure that the queried nodes in each cycle are representative of the overall graph structure, we rank nodes based on structural indices that capture their relative importance within the network. Among such indices, we specifically incorporate PageRank (Page et al., 1999), where the objective function $\mathcal{L}_1^\gamma$ is given as

$$\mathcal{L}_1^\gamma(v, \mathcal{G}_\text{a}) = \frac{1-\xi}{N} + \xi \sum_{w \in \text{in}(v)} \frac{\mathcal{L}_1^\gamma(w, \mathcal{G}_\text{a})}{L(w)}. \tag{1}$$

In (1), $N = \left|\mathcal{V}_\text{a}\right|$ represents the total number of nodes in the extraction subgraph $\mathcal{G}_\text{a}$, $\text{in}(v)$ represents the collection of nodes with edges pointing to node v, $L(w)$ represents the number of outbound edges from node w, and ξ represents a damping factor typically set at 0.85. The evaluation of $\mathcal{L}_1^\gamma$ is inherently recursive in computation, relying on iterative processes to update the scores of the nodes given their neighborhood until convergence. The rank of nodes according to their representativeness in the γth cycle is denoted as $\mathcal{R}_1^\gamma$.

History-Based Analysis for Uncertainty. To evaluate the uncertainty of nodes in an interim GNN model $f_{\gamma-1}$ on the classification task, we evaluate the entropy of the nodes in the pool $\mathcal{V}_\text{a} \backslash \mathcal{V}_{\gamma-1}$ as their sensitivity against changes in the attributes of its neighbors. In particular, we give the objective function as

$$\mathcal{L}_2^\gamma(v, \mathcal{G}_\text{a}) = -\sum_{i=1}^{C} \widehat{y}_{v;\gamma-1}^{(i)} \log(\widehat{y}_{v;\gamma-1}^{(i)}). \tag{2}$$

In (2), $\widehat{y}_{v;\gamma-1}^{(i)}$ is the ith entry of $\widehat{\mathbf{y}}_{v;\gamma-1} = f_{\gamma-1}(\mathbf{x}_{\gamma-1}^{(v)}, \mathcal{G}_\text{a})$. For downstream tasks that are less sensitive to time and space complexity, we propose a theory-backed alternative ranking mechanism that measures a node's resilience in maintaining its predicted label under moderate Gaussian perturbation. In practice, we consider a series of perturbation $\boldsymbol{\tau}^{(j)} \overset{i.i.d}{\sim} \mathcal{N}(\mathbf{0}, \epsilon^2 \mathbf{I})$ where $\mathbf{I} \in \mathbb{R}^{d \times d}$ is an identity matrix, and obtain the perturbed version of the attributes $\mathcal{X}_{\gamma-1}$, denoted as

$$\mathcal{T}_{\gamma-1} = \left\{\mathbf{x}_{\gamma-1}^{(1)} + \boldsymbol{\tau}^{(1)}, \mathbf{x}_{\gamma-1}^{(2)} + \boldsymbol{\tau}^{(2)}, ..., \mathbf{x}_{\gamma-1}^{(n_{\gamma-1})} + \boldsymbol{\tau}^{(n_{\gamma-1})}\right\}.$$

We repeat the perturbation S times and obtain the perturb data $\mathcal{T}_{\gamma-1}^\ell$ where $\ell \in [S]$. For any node v, the respective probability for $\mathcal{T}_{\gamma-1}^\ell$ is denoted as $\widehat{\mathbf{y}}_{v;\gamma-1}^\ell = f_{\gamma-1}(\mathbf{x}_{\gamma-1}^{(v)} + \boldsymbol{\tau}_\ell^{(v)}, \mathcal{G}_\text{a})$, where $\ell \in [S]$. The alternative objective function $\mathcal{L}_{2;\text{alt}}^\gamma(v, \mathcal{G}_\text{a})$ is given as

$$\mathcal{L}_{2;\text{alt}}^\gamma(v, \mathcal{G}_\text{a}) = \sum_{\ell=1}^{S} \mathbb{I}_{\left\{\arg\max\{\widehat{\mathbf{y}}_{v;\gamma-1}\} = \arg\max\{\widehat{\mathbf{y}}_{v;\gamma-1}^\ell\}\right\}}, \tag{3}$$

where $\widehat{\mathbf{y}}_{v;\gamma-1}$ and $\widehat{\mathbf{y}}_{v;\gamma-1}^\ell$ are outputs of $f_{\gamma-1}$, which takes the subgraph $\mathcal{G}_\text{a}$ as an input. The rank of nodes based on their uncertainty on the prediction of the interim model $f_{\gamma-1}$ in the γth cycle of CEGA is denoted as $\mathcal{R}_2^\gamma$.

In Section 3.3, we provide theoretical insights into the time and space complexity of CEGA to further justify its suitability to measure uncertainty in node selection. Furthermore, we present the theoretical guarantee for the existence of an appropriate perturbation parameter ϵ involved in the alternative approach. The parameter ϵ is expected to be sufficiently large to capture the sensitivity of the interim model's predictions while remaining small enough to preserve the stability and effectiveness of the interim model.

Distance-Based Analysis for Diversity. Finally, we evaluate the diversity of a node in the pool $\mathcal{V}_\text{a} \backslash \mathcal{V}_{\gamma-1}$ compared to the queried nodes $\mathcal{V}_{\gamma-1}$. To start, we apply the K-Means algorithm for the embedding of queried nodes with $K = C$, where C is the number of classes for the graph dataset. We then compare the embeddings of the nodes belong to $\mathcal{V}_\text{a} \backslash \mathcal{V}_{\gamma-1}$ with the clusters formed by the queried nodes to determine whether they align with a category that is over-represented in $\mathcal{V}_{\gamma-1}$. To do this, we measure the distance between some node $v \in \mathcal{V}_\text{a} \backslash \mathcal{V}_{\gamma-1}$ and the C centroids. We assign node v to the class $j \in \{1, 2, ..., C\}$ such that the distance between its embedding $\mathcal{E}_v$ and the centroid $\mathcal{C}_j$ is

minimized. Here, $\mathcal{E}_v$ is an output of the interim model $f_{\gamma-1}$, with the graph structure $\mathcal{G}_\mathrm{a}$ serving as a necessary input. We then establish the objective function $\mathcal{L}_3^\gamma(v, \mathcal{G}_\mathrm{a})$ as

$$\mathcal{L}_3^\gamma(v, \mathcal{G}_\mathrm{a}) = \rho\varphi_{[0,1]}\Big(\frac{1}{1+\delta_v}\Big) + (1-\rho)\varphi_{[0,1]}\Big(\frac{1}{1+|\mathcal{Q}_v|}\Big). \tag{4}$$

Here δ_v is the minimal distance between the embedding $\mathcal{E}_v$ for node v and centroids $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_C$, and we have

$$\delta_v = \min_{c\in\{1,2,...,C\}} \big\|\mathcal{E}_v - \mathcal{C}_c\big\|_2.$$

On the other hand, $\mathcal{Q}_v$ represents the collection of queried nodes that belong to the same centroid $\mathcal{C}_{c^*}$ as node v, where

$$c^* = \operatorname*{arg\,min}_{c\in\{1,2,...,C\}} \big\|\mathcal{E}_v - \mathcal{C}_c\big\|_2.$$

The number of nodes belong to $\mathcal{Q}_v$ is denoted as $|\mathcal{Q}_v|$. $\varphi_{[0,1]}(\cdot)$ represents the min-max scaling function. ρ is a hyperparameter subject to tuning. The rationale behind the setup of $\mathcal{L}_3^\gamma$ is to guide the selection of nodes from $\mathcal{V}_\mathrm{a}\backslash\mathcal{V}_{\gamma-1}$ that represent the embedding patterns of labels that are underrepresented in the queried nodes. By using $\mathcal{L}_3^\gamma$, we rank the nodes $v \in \mathcal{V}_\mathrm{a}\backslash\mathcal{V}_{\gamma-1}$ in the order $\mathcal{R}_3^\gamma$.

Adaptive Node Selection Method. Once we obtain the ranking of all candidate nodes according to the objective functions $\mathcal{L}_1^\gamma$, $\mathcal{L}_2^\gamma$, and $\mathcal{L}_3^\gamma$, we compute a weighted average ranking of the nodes in the three categories and query the top-κ nodes $\mathcal{V}_\gamma^Q$ based on this weighted average, as those nodes are expected to guide further informative queries to the target GNN model.

The weighted average ranking for cycle γ is expressed as

$$\mathcal{R}^\gamma = \omega_1(\gamma)\mathcal{R}_1^\gamma + \omega_2(\gamma)\mathcal{R}_2^\gamma + \omega_3(\gamma)\mathcal{R}_3^\gamma.$$

The cycle-specific weights ω_1, ω_2, and ω_3 are subject to adaptive optimization according to the principle, as inspired by (Cai et al., 2017), that the representativeness rank $\mathcal{R}_1$ does not rely on the interim model $f_{\gamma-1}$, while $\mathcal{R}_2$ and $\mathcal{R}_3$ rely on the $f_{\gamma-1}$. The weight ω_1 is assigned a higher value when γ is small, reflecting the relatively poor performance of the interim model during the earlier stages of querying. As γ increases, ω_2 and ω_3 are progressively raised, resonating the improved performance of the interim model in later querying cycles. The dynamic node selection ensures that the weights can fit CEGA's progressive querying process.

Learning to Guide Queries and Output. After obtaining queried nodes $\mathcal{V}_\gamma$ in the γth cycle, we train the new interim model f_γ based on $\{\mathcal{V}_\gamma, \mathcal{G}_\mathrm{a}\}$ and the previous interim model $f_{\gamma-1}$ for E epochs. The updated model f_γ then guides node selection for further queries in the $(\gamma+1)$th cycle. After completion of the Γ querying cycles, CEGA returns the collection of queried nodes $\{\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_\Gamma\}$. We summarize the algorithmic routine of CEGA in Algorithm 1.

Algorithm 1 The Proposed Framework of CEGA

Initialization: Query initial nodes $\mathcal{V}_0$, where $|\mathcal{V}_0| = \mathcal{I}$, from $\mathcal{V}_\mathrm{a}$.
Train the initial model f_0 on $\{\mathcal{V}_0, \mathcal{G}_\mathrm{a}\}$.
for Cycle γ from 1 to Γ **do**
  if $\mathcal{I} + (\gamma-1)\kappa < B$ **then**
    Evaluate the representativeness score $\mathcal{L}_1^\gamma$, uncertainty score $\mathcal{L}_2^\gamma$, and diversity score $\mathcal{L}_3^\gamma$ for all candidate nodes in $\mathcal{V}_\mathrm{a}\backslash\mathcal{V}_{\gamma-1}$.
    Obtain node ranks $\mathcal{R}_1^\gamma$, $\mathcal{R}_2^\gamma$, and $\mathcal{R}_3^\gamma$.
    Select and query top-κ nodes $\mathcal{V}_\gamma^Q$ via the adaptive selection method.
    Set $\mathcal{V}_\gamma = \mathcal{V}_{\gamma-1} \cup \mathcal{V}_\gamma^Q$.
  else
    Set $\mathcal{V}_\gamma = \mathcal{V}_{\gamma-1}$.
  end if
  Train the new interim model f_γ based on $\{\mathcal{V}_\gamma, \mathcal{G}_\mathrm{a}\}$ and $f_{\gamma-1}$ for E epochs.
end for
Return the nodes collection $\{\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_\Gamma\}$.

3.3. Theoretical Analysis

In this section, we summarize the theoretical results for CEGA's measurements with respect to the uncertainty of candidate nodes indicated by history-inspired interim models. As outlined in Section 3.2, we address two core aspects: CEGA's efficiency in referring to the history guide and the existence of an appropriate perturbation parameter ϵ for the alternative. The detailed proofs are given in Appendix A.

First, we provide a thorough analysis of the time and space complexity of CEGA's uncertainty measurement approach. Proposition 3.1 highlights the feasibility of our approach, indicating its resource-friendly feature required for graph data with complex attributes and subgraph structure.

Proposition 3.1 (Evaluation of Complexity)**.** *Suppose that the base model of CEGA is a L-layer GCN. Under the conditions such that*

1. *The number of nodes queried by CEGA in each cycle, indicated by κ, is $\Theta(1)$;*
2. *$d \gg h = \Theta(C)$, where d indicates the dimension of the attributes for the graph data, h indicates the dimension of the node embeddings, and C indicates the number of classes in the softmax score output,*

CEGA's entropy-based approach introduces an additional time complexity of $O(CN + N\log N)$ and space complexity of $O(CN)$, building on the $O(LN^2d + LNd^2)$ time complexity and $O(N^2 + Ld^2 + LNd)$ space complexity required for CEGA to compute embeddings and softmax scores via forward propagation for the analysis in Section 3.2.

Remarks for Proposition 3.1. CEGA's entropy-based approach provides superior scalability and adaptability for large, complex graph datasets, introducing minimal additional time and space complexity beyond attribute propagation through the interim model for embeddings and softmax scores. Notably, these added complexities are of significantly lower order than the training and forward propagation costs of the interim model.

In Theorem 3.2, we show the existence of an appropriate perturbation intensity ϵ that is sufficiently small to maintain the overall stability of the history guide f_γ. This ensures the feasibility of perturbation-based alternative uncertainty evaluation in CEGA by guaranteeing that the approach can reliably capture prediction uncertainty from history without causing inconsistency in the interim model, even when random noise is applied to the node attributes.

Theorem 3.2 (Existence of Feasible Perturbation)**.** *Consider the perturbation scheme for the evaluation of node uncertainty under the interim GNN model f_γ in CEGA. We show that there exists some perturbation intensity ϵ such that $\left\|f_\gamma(\mathbf{x}_i, \mathcal{G}_\mathrm{a}) - f_\gamma(\widetilde{\mathbf{x}}_i, \mathcal{G}_\mathrm{a})\right\|_2$ holds the stability conditions, where $\widetilde{\mathbf{x}}_\tau$ is the perturbation of $\mathbf{x}_\tau$ where $\widetilde{\mathbf{x}}_\tau - \mathbf{x}_\tau \sim \mathcal{N}(0, \epsilon^2)$. Specifically, for any $\zeta > 0$, there exists some $\epsilon = \epsilon(\zeta, \delta)$ such that $\left\|f_\gamma(\mathbf{x}_i, \mathcal{G}_\mathrm{a}) - f_\gamma(\widetilde{\mathbf{x}}_i, \mathcal{G}_\mathrm{a})\right\|_2 \leq \zeta$ with probability at least $1 - O(\delta)$.*

4. Experimental Evaluation

In this section, we present an in-depth experimental evaluation of CEGA, addressing three key research questions: **RQ1**: How effective is CEGA in graph-based model extraction and acquisition tasks compared to baseline methods on a fixed querying budget? **RQ2**: How well does CEGA recover the target model on a limited query budget, as opposed to querying all nodes that can be queried? **RQ3**: What is the contribution of each evaluation module of CEGA to its overall performance?

4.1. Experimental Settings

Graph Learning Task and Datasets. We evaluate CEGA on the extraction task for graph-based node classification models, assuming that the extraction side has access to the attributes of the candidate nodes $\mathcal{V}_\mathrm{a}$ and the structure of subgraph $\mathcal{G}_\mathrm{a}$ that involves $\mathcal{V}_\mathrm{a}$. This setup is categorized as *Attack 0* in MEA literature such as (Wu et al., 2022). Our experiments are conducted on 6 widely used benchmark datasets: (1) Coauthorship networks where nodes are authors and edges represent collaboration, including *Coauthor-CS* and *Coauthor-Physics*; (2) Co-purchase graphs with nodes as products and edges as items frequently purchased together, including *Amazon-Computer* and *Amazon-Photo*; and (3) Academic citation and collaboration network, including *Cora-Full* and *DBLP*. These datasets vary in size, complexity, and formality of node attributes, providing a comprehensive basis for evaluating CEGA's performance. The dataset statistics are provided in Appendix B.1.

Training Protocol. In our experiment, we consider two models trained on the datasets of interest. The *full subgraph model* is trained on $\left\{\mathcal{V}_\mathrm{a}, \mathcal{G}_\mathrm{a}\right\}$, where the subgraph known to the extractors $\mathcal{G}_\mathrm{a}$ satisfies $\mathcal{G}_\mathrm{a} \subsetneq \mathcal{G}_\mathrm{T}$, to establish the upper limit of performance that model extraction approaches can achieve in their respective task under our assumptions. In the *budget-constrained model* trained on $\left\{\mathcal{V}_\Gamma, \mathcal{G}_\mathrm{a}\right\}$, where $|\mathcal{V}_\Gamma| \ll |\mathcal{V}_\mathrm{a}|$ in many practices, the model extraction task is conducted on a more restrictive but realistic scenario with budget constraints, as we have defined in Section 2. In response to the constraints, CEGA and the baseline models, as detailed in the following sections, progressively select and query the most informative nodes from the pool of candidate nodes for optimal performance. To show the superiority of CEGA, we compare the performance of all models tested under a budget-constrained setup and examine the performance incrementation of the full subgraph model over the budget-constrained model for each approach.

Evaluation Metrics. We evaluate the accuracy and F1 score of budget-constrained models built under nodes selectively queried by all the tested approaches. Furthermore, we evaluate the faithfulness of these models to the target model f_T, using fidelity as the metric. The measurements on accuracy, F1 score, and fidelity are further compared between the full subgraph model and the budget-constrained model to highlight the relative efficiency of the strategies tested. Finally, we evaluate the mean and standard deviation of accuracy, fidelity, and F1 score for budget-constrained models trained on nodes queried using CEGA and its variants with certain components ablated. All empirical evaluations are based on consistent settings with commonly used ones. The query budget is set as the number of label classes for each graph dataset multiplied by a fixed factor, ranging from $2C$ to $20C$, following widely accepted prior work, such as (Yang et al., 2016; Cai et al., 2017; Zhang et al., 2021).

Baselines. In our experiment, the performance of CEGA is compared against the *Random* baseline, where all the queried nodes are selected randomly from $\mathcal{V}_\mathrm{a}$. Furthermore, we compare CEGA with the state-of-the-art active learning (AL) techniques specially designed for GNN in a query-by-training process consistent with CEGA for fair comparison, including *AGE* ((Cai et al., 2017)), *GRAIN (NN-D)*, and *GRAIN (ball-D)* (Zhang et al., 2021). To ensure consistency, all baseline models adhere to the same query constraints initialization setup. Details on the hyperparameter setup for all baseline methods and the proposed framework CEGA are included in Appendix B.2.

Table 1. Test accuracy, fidelity, and F1 score on different datasets using $20C$ queried nodes. Dataset abbreviations: CoCS (Coauthor-CS), CoP (Coauthor-Physics), AmzC (Amazon-Computer), AmzP (Amazon-Photo), Cora-Full, and DBLP. All numerical values are reported in percentage. The best results are in **bold**.

		CoCS	CoP	AmzC	AmzP	Cora_Full	DBLP
Accuracy	**Random**	88.75 ± 0.7	91.50 ± 1.3	83.79 ± 0.9	90.15 ± 2.6	49.73 ± 0.3	69.14 ± 1.9
	GRAIN(NN-D)	89.77 ± 0.6	93.37 ± 0.8	83.89 ± 1.7	90.98 ± 0.3	51.57 ± 1.0	68.37 ± 0.9
	GRAIN(ball-D)	89.43 ± 0.6	93.37 ± 1.0	82.48 ± 2.1	90.01 ± 1.2	51.27 ± 1.3	68.57 ± 1.0
	AGE	**90.68 ± 0.4**	93.69 ± 0.3	85.13 ± 0.6	90.79 ± 2.6	50.59 ± 0.3	72.41 ± 2.2
	CEGA	90.57 ± 0.5	**93.90 ± 0.4**	**85.98 ± 0.4**	**91.95 ± 0.3**	**52.74 ± 0.6**	**73.29 ± 0.9**
Fidelity	**Random**	91.43 ± 0.8	93.15 ± 1.4	88.45 ± 1.0	93.31 ± 2.7	74.06 ± 0.8	73.86 ± 2.2
	GRAIN(NN-D)	92.41 ± 0.8	95.11 ± 0.9	88.65 ± 2.0	94.17 ± 0.6	76.65 ± 1.6	72.71 ± 1.1
	GRAIN(ball-D)	92.00 ± 0.7	95.19 ± 1.2	86.89 ± 2.4	92.93 ± 1.4	76.18 ± 1.5	73.35 ± 1.2
	AGE	**93.61 ± 0.5**	95.55 ± 0.4	90.10 ± 0.7	93.97 ± 2.9	75.67 ± 0.8	77.18 ± 2.4
	CEGA	93.40 ± 0.6	**95.83 ± 0.5**	**90.81 ± 0.4**	**95.33 ± 0.5**	**77.90 ± 0.9**	**78.50 ± 0.9**
F1	**Random**	81.44 ± 1.6	87.70 ± 2.4	78.95 ± 1.8	86.54 ± 5.3	27.56 ± 0.3	57.46 ± 5.0
	GRAIN(NN-D)	85.61 ± 1.4	90.93 ± 1.0	80.29 ± 3.5	88.06 ± 0.8	28.93 ± 1.0	58.72 ± 3.7
	GRAIN(ball-D)	85.38 ± 0.9	90.97 ± 1.4	74.47 ± 5.9	86.99 ± 2.0	28.62 ± 1.1	60.87 ± 3.5
	AGE	**87.65 ± 0.4**	91.58 ± 0.5	78.37 ± 3.5	89.14 ± 3.2	29.28 ± 0.5	65.72 ± 3.2
	CEGA	87.41 ± 0.5	**91.78 ± 0.7**	**82.57 ± 1.4**	**90.06 ± 0.6**	**31.20 ± 0.8**	**67.35 ± 1.5**

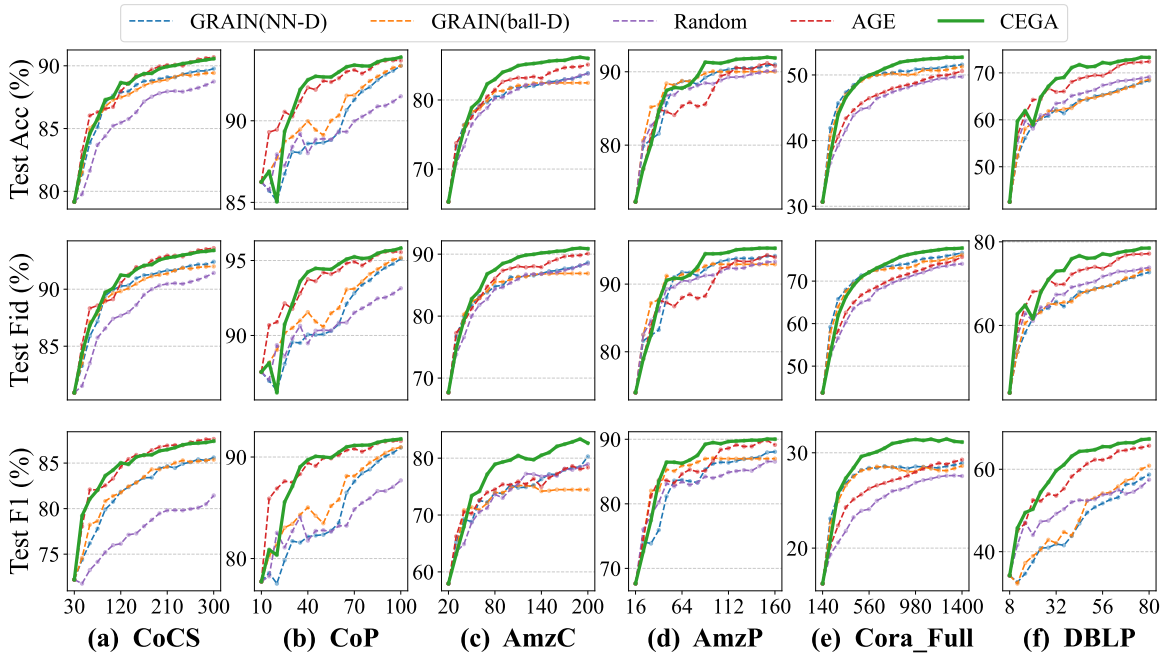

Figure 1. The trajectory of test accuracy, fidelity, and F1 score on different datasets using $2C$ to $20C$ queried nodes. The performance trajectory of CEGA is bolded in green, showing significant superiority over the alternatives across different number of queried nodes.

4.2. Evaluation on Budget-Constrained Model

To answer **RQ1**, we evaluate the performance of CEGA and various AL-based baseline methods on history-based progressive node querying in all the 6 datasets under different labeling budget scenarios. We incrementally raise the query budget from $2C$ to $20C$. Our evaluation primarily focuses on the fidelity of the budget-constrained subgraph model to the target model, while also considering accuracy and F1 score as supplementary metrics. To account for variability due to randomized initialization, each method is evaluated five times, and the mean performance is reported.

Figure 1 presents the trajectory of measured metrics with

the extracted model trained with $2C$ to $20C$ queried nodes. Table 1 presents a direct comparison between the performance of CEGA and the baseline models using $20C$ queried nodes in the 6 datasets of interest. We summarize the key observations below: (1) From the perspective of comparative performance, CEGA consistently achieves significant improvements in accuracy, fidelity, and F1 score across varying budget levels, demonstrating its capability to closely mimic the target model under stringent query budget constraints. CEGA also shows strong adaptability to the graph datasets being tested, which is considered one of its main advantages over the baselines. (2) From the perspective of the progressive nature of the models, as the budget increases from $2C$ to 10-15C, CEGA further extends its advantage over baseline methods among all the metrics tested, particularly for the Amazon-Computer and Cora-Full datasets. This indicates that CEGA's node selection strategy effectively identifies the most informative nodes throughout the iterative querying process, leading to superior alignment with the target model as more queries become available.

4.3. Comparison between Full Subgraph Model and Budget-Constrained Model

To answer **RQ2**, we highlight the effectiveness of CEGA in detecting the most informative nodes by comparing the accuracy, fidelity, and F1 score recovery performance between the budget-constrained model and the full subgraph model, as specified in Section 4.1. Specifically, we define a *performance gap* as the performance discrepancy between the full subgraph model and budget-constrained models following different querying approaches.

We visualize the performance gap measured by accuracy, fidelity, and F1 score across all 6 datasets of interest in Figure 2. Further results on the performance gap are presented in Table 4 and discussed in Appendix B.3. We summarize the key observations below: (1) From the perspective of measurement, CEGA consistently exhibits a lower performance gap across all the metrics tested (accuracy, fidelity, F1 score) compared to the baselines, indicating its superior ability to recover as much information as possible under stringent budget constraints. (2) From the perspective of adaptability, CEGA maintains a consistently lower gap across all the datasets we tested by a notifiable margin (e.g., 1-2%) despite variations in node attributes and graph structures. This reveals that CEGA is more robust and effective than baselines in conducting model extraction and acquisition on graph data with different levels of intrinsic complexity.

4.4. Ablation Study

To answer **RQ3**, we perform an ablation study by systematically removing the contribution of each one out of the 3 evaluation modules of CEGA. We then compare the performance of the resulting models with only the two remaining modules involved with that of the full CEGA model. We evaluate budget-constrained models with a query budget of $20C$ across all 6 datasets.

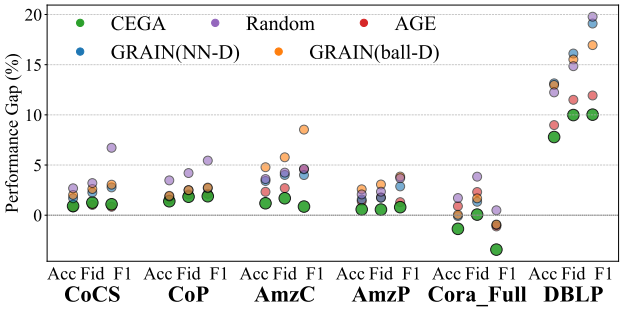

Figure 2. Model performance gaps between budget-constrained and full subgraph models, measured by accuracy, fidelity, and F1 score, across datasets. The gap indicates the negative impact of the budget constraints on the model performances. Therefore, lower gaps (i.e., less negative impact) are preferred.

Table 2. Ablation study results on fidelity for CEGA and variants with one evaluation module removed. *Cen* stands for Centrality, *UnC* stands for Uncertainty, *Div* stands for Diversity. The best results are in **bold**.

	CEGA	No Cen	No UnC	No Div
CoCS	**93.4 ± 0.6**	93.2 ± 0.2	91.9 ± 0.5	93.4 ± 0.6
CoP	**95.8 ± 0.5**	94.9 ± 0.4	90.2 ± 3.3	95.7 ± 0.5
AmzC	**90.8 ± 0.4**	90.0 ± 1.2	87.1 ± 2.2	90.7 ± 0.7
AmzP	**95.3 ± 0.5**	95.1 ± 0.3	93.7 ± 0.9	95.3 ± 0.7
Cora_Full	77.9 ± 0.9	75.3 ± 0.6	74.9 ± 0.9	**78.3 ± 1.1**
DBLP	78.5 ± 0.9	74.2 ± 2.4	65.1 ± 5.5	**78.6 ± 1.4**

Table 2 compares the mean fidelity and its variance between the original CEGA and the three ablation models in the 6 datasets of interest, while Figure 3 shows a similar pattern for accuracy and F1 score is available in Appendix B.4. We summarize the key observations below: (1) From the perspective of model performance, CEGA demonstrates comparable to significantly better average performance than ablated models across different test datasets and metrics, particularly outperforming models where centrality or uncertainty is ablated by a large margin. This highlights the pivotal rule of these two components in identifying informative nodes for querying in early cycles. (2) From the perspective of consistency of estimates, CEGA provides more stable estimates across all metrics compared to the model with diversity ablated, especially for the Amazon-Computer and DBLP datasets. This reveals that the diversity component of CEGA plays a crucial role in later querying cycles by supporting performance stability.

5. Related Work

Query-Efficient Model Extraction Attack. (Tramèr et al., 2016) pioneered the study of MEA with high-fidelity extraction towards target black-box MLaaS models. (Pal et al., 2020) applies active learning techniques that dynamically adjust query selection based on self-feedback to extract deep classifiers in the domain of image and text, showing that querying 10% to 30% of samples from the dataset can yield a high-fidelity extraction model. Recent work shows that budget-sensitive MEA is feasible even for the data-free setting, where the attacks are performed without any in-distribution data (Lin et al., 2023b). Another proposed way to obtain a better query efficiency in MEA is to simultaneously train two clone models with the same samples and force them to learn from mismatching samples (Rosenthal et al., 2023). (Dai et al., 2023) employs clustering-based data reduction to minimize information redundancy in the query pool and realize query efficiency for the task of MEA in NLP. However, all these works on query-efficient MEA fail to consider graph-based model extraction.

Model Extraction Attack in Graph Learning. (Oliynyk et al., 2023) systematizes MEA on multiple model types, summarizes defense strategies based on available resources, and highlights an upward trend of popularity of literature on the attacks towards and defense for ML models. Recently, the research interest has pivoted towards the application of MEA on graph-related models. (DeFazio & Ramesh, 2019) first consider an adversarial model extraction approach for a graph-structured dataset. (Wu et al., 2022) provides a comprehensive analysis of GNN-based MEAs under various categories based on the attacker's knowledge of target graph structure and node attributes. (Shen et al., 2022) claims a significant contribution by proposing an inductive model structure that allows the attack graph to add new nodes to the existing model without the necessity of retraining. This development overcomes a major limitation of the prevalent GCN-based transductive approach (Kipf & Welling, 2017) that requires model retraining with the introduction of a new attacking or testing node. Generally, the GNN-based MEAs can be divided into two primary categories, distinguished by the attacker's knowledge of the target model's graph structure (Oliynyk et al., 2023). Go beyond existing publications, our work addresses a serious concern regarding the efficiency and budget limitation in model extraction on graph learning by subsequently improving the practicality.

6. Conclusion

In this paper, we introduce CEGA, a cost-effective framework specialized in node querying for graph-based model extraction. In particular, we formulate and study the problem of budget-constrained model extraction on graphs, where the objective is to maximize the extracted model's performance and resemblance to the target model with a minimized query budget. To overcome this challenge, we develop CEGA by designing an adaptive node selection strategy that effectively queries the most informative nodes based on incremental history information accumulated in the training progress. We present a theoretical guarantee on the feasibility and efficiency of our approach in measuring the uncertainty of history-based interim predictions for candidate nodes. Extensive experiments on real-world graph datasets demonstrate CEGA's superiority over state-of-the-art baselines across multiple key aspects. Looking ahead, two future directions warrant further exploration. First, our current framework is based on a *transductive* assumption, and we aim to extend the CEGA framework to *inductive* GNNs, following previous investigation by (Shen et al., 2022). Second, as suggested by (Guan et al., 2024), refining our approach by leveraging edge information, especially in the early query cycles when the number of selected nodes is small, could further improve CEGA's performance.

Acknowledgements

Y.D. acknowledges funding in part by Start-Up Grant and FYAP (First Year Assistant Professor) Grant from Florida State University, Tallahassee, FL, USA.

Impact Statement

We introduce CEGA (Cost-Efficient Graph Acquisition), a framework to deploy model extraction and acquisition on Graph Neural Networks (GNNs) under realistic constraints of limited query budgets and structural complexity.

For practitioners in high-stakes fields, CEGA formalizes the problem of *GNN Model Extraction With Limited Budgets*, laying a foundation for the development of practical defenses against GNN-based model extraction attacks (MEAs) against Machine Learning as a Service (MLaaS).

For researchers, CEGA reveals its non-adversarial potential of GNN extraction and acquisition in domains where expert labeling is prohibitively expensive and large-scale training is impractical. Ethical model acquisition offers a viable path to democratize high-performance GNNs and adapt them to specialized downstream tasks.

We emphasize the responsible use of CEGA, as its insights should be used to strengthen MLaaS security and advocate ethical research under limited resources.

References

Abbahaddou, Y., Ennadir, S., Lutzeyer, J. F., Vazirgiannis, M., and Boström, H. Bounding the expected robustness of graph neural networks subject to node feature attacks.

In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DfPtC8uSot.

Ahmedt-Aristizabal, D., Armin, M. A., Denman, S., Fookes, C., and Petersson, L. Graph-based deep learning for medical diagnosis and analysis: Past, present and future. *Sensors (Basel)*, 21(14):47–58, 2021. doi: doi:10.3390/s21144758.

Armstrong, M. Trade secret protection in the pharma industry. *Pharmaceutical Patent Analyst*, 5(5):285–288, 2016. doi: https://doi.org/10.4155/ppa-2016-0022.

Bessen, J. E. and Meurer, M. J. The private costs of patent litigation. *Boston University School of Law Working Paper No 07-08, 2nd Annual Conference on Empirical Legal Studies Paper*, pp. 1–43, 2008.

Blakely, D., Lanchantin, J., and Qi, Y. Time and space complexity of graph convolutional networks, 2021. Accessed: Dec 31, 2021.

Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., Anderson, H., Roff, H., Allen, G. C., Steinhardt, J., Flynn, C., hÉigeartaigh, S. Ó., Beard, S., Belfield, H., Farquhar, S., Lyle, C., Crootof, R., Evans, O., Page, M., Bryson, J., Yampolskiy, R., and Amodei, D. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228*, 2018.

Cai, H., Zheng, V. W., and Chang, K. C.-C. Active learning for graph embedding. *arXiv preprint arXiv:1705.05085*, 2017.

Cheng, Y., Guo, J., Long, S., Wu, Y., Sun, M., and Zhang, R. Advanced financial fraud detection using gnn-cl model. In *2024 International Conference on Computers, Information Processing and Advanced Education (CIPAE)*, pp. 453–460, 2024. doi: 10.1109/CIPAE64326.2024.00088.

Cheng, Z., Shen, B., Sha, T., Gao, Y., Li, S., and Dong, Y. Atom: A framework of detecting query-based model extraction attacks for graph neural networks. *arXiv preprint arXiv:2503.16693*, 2025.

Dai, C., Lv, M., Li, K., and Zhou, W. Meaeq: Mount model extraction attacks with efficient queries. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=D97Zfgv4em.

DeFazio, D. and Ramesh, A. Adversarial model extraction on graph neural networks. *arXiv preprint arXiv:1912.07721*, 2019.

Dong, Y., Wang, S., Ma, J., Liu, N., and Li, J. Interpreting unfairness in graph neural networks via training node attribution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37:7441–7449, 06 2023. doi: 10.1609/aaai.v37i6.25905.

Dong, Y., Zhang, B., Lei, Z., Zou, N., and Li, J. Idea: A flexible framework of certified unlearning for graph neural networks. In *KDD '24: Proceeding of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 621–630, 2024. doi: https://doi.org/10.1145/3637528.3671744.

Dong, Y., Soga, P., He, Y., Wang, S., and Li, J. Graph neural networks are more than filters: Revisiting and benchmarking from a spectral perspective. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=nWdQX5hOL9.

Ennadir, S., Abbahaddou, Y., Vazirgiannis, M., and Boström, H. A simple and yet fairly effective defense for graph neural networks. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023. URL https://openreview.net/forum?id=CJgBMut3nC.

Gan, Z., Zhou, D., Rush, E., Panickan, V. A., Ho, Y.-L., Ostrouchovm, G., Xu, Z., Shen, S., Xiong, X., Greco, K. F., Hong, C., Bonzel, C.-L., Wen, J., Costa, L., Cai, T., Begoli, E., Xia, Z., Gaziano, J. M., Liao, Katherine, P., Cho, K., Cai, T., and Lu, J. Arch: Large-scale knowledge graph via aggregated narrative codified health records analysis. *Journal of Biomedical Informatics*, 162:104761, 2025. doi: https://doi.org/10.1016/j.jbi.2024.104761.

Gong, X., Wang, Q., Chen, Y., Yang, W., and Jiang, X. Model extraction attacks and defenses on cloud-based machine learning models. *IEEE Communications Magazine*, 58(12):83–89, 2020. doi: 10.1109/MCOM.001.2000196.

Guan, F., Zhu, T., Tong, H., and Zhou, W. A realistic model extraction attack against graph neural networks. *Knowledge-Based Systems*, 300:112144, 2024. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2024.112144.

Hong, C., Rush, E., Liu, M., Zhou, D., Sun, J., Sonabend, A., Castro, V. M., Schubert, P., Panickan, V. A., Cai, T., Costa, L., He, Z., Link, N., Hauster, R., Gaziano, J. M., Murphy, S. N., Ostrouchov, G., Ho, Y.-L., Begoli, E., Lu, J., Cho, K., Liao, K. P., Cai, T., and Program, V. M. V. Clinical knowledge extraction via sparse embedding regression (keser) with multi-center large scale electronic health record data. *npj Digital Medicine*, 4(151):1–11, 2021. doi: https://doi.org/10.1038/s41746-021-00519-z.

Hou, J., Qian, J., Wang, Y., Li, X.-Y., Du, H., and Chen, L. Ml defense: Against prediction api threats in cloud-based machine learning service. In *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2019. doi: 10.1145/3326285.3329042.

Huo, C., Jin, D., Li, Y., He, D., Yang, Y.-B., and Wu, L. T2-gnn: graph neural networks for graphs with incomplete features and structure via teacher-student distillation. In *AAAI'23/IAAI'23/EAAI'23: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, pp. 4339–4346, 2023. doi: https://doi.org/10.1609/aaai.v37i4.25553.

Ju, W., Mao, Z., Yi, S., Qin, Y., Gu, Y., Xiao, Z., Shen, J., Qiao, Z., and Zhang, M. Cluster-guided contrastive class-imbalanced graph classification. *arXiv preprint arXiv:2412.12984*, 2024.

Juuti, M., Szyller, S., Marchal, S., and Asokan, N. Prada: Protecting against dnn model stealing attacks. *In 2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 512–527, 2019.

Kahn, A. E., Szymula, K., Loman, S., Haggerty, E. B., Nyema, N., Aguirre, G. K., and Bassett, D. S. Network structure influences the strength of learned neural representations. *Nature Communications*, 16(994), 2025.

Karmakar, P. and Basu, D. Marich, a query-efficient distributionally equivalent model extraction attack. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum.id=bAI21VEMvM.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Kose, H. T., Nunez-Yanez, J., Piechocki, R., and Pope, J. A survey of computationally efficient graph neural networks for reconfigurable systems. *Information*, 15(7):377, 2024. doi: https://doi.org/10.3390/info15070377.

Li, X., Fan, Z., Huang, F., Hu, X., Deng, Y., Wang, L., and Zhao, X. Graph neural network with curriculum learning for imbalanced node classification. *Neurocomputing*, 574:127229, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.127229. URL https://www.sciencedirect.com/science/article/pii/S0925231223013528.

Lin, J., Wan, Y., Xu, j., and Qi, X. Semantic graph neural network with multi-measure learning for semi-supervised classification. In *Engineering Applications of Artificial Intelligence*, volume 140, pp. 109647, 2025. doi: https://doi.org/10.1016/j.engappai.2024.109647.

Lin, Y., Lu, K., Yu, S., Cai, T., and Marinka, Z. Multimodal learning on graphs for disease relation extraction. *Journal of Biomedical Informatics*, 143:104415, 2023a. doi: https://doi.org/10.1016/j.jbi.2023.104415.

Lin, Z., Xu, K., Fang, C., Zheng, H., Ahmed Jaheezuddin, A., and Shi, J. Quda: Query-limited data-free model extraction. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, ASIA CCS '23, pp. 913–924, New York, NY, USA, 2023b. Association for Computing Machinery. ISBN 9798400700989. doi: 10.1145/3579856.3590336.

Liu, S., Cai, T., and Li, X. Representation-enhanced neural knowledge integration with application to large-scale medical ontology learning. *arXiv preprint arXiv:2410.07454*, 2024.

Liu, X., Wu, B., Yuan, X., and Yi, X. Leia: A lightweight cryptographic neural network inference system at the edge. *IEEE Transactions on Information Forensics and Security*, 17:237–252, 2022. doi: 10.1109/TIFS.2021.3138611.

Liu, Y., Luo, J., Yang, Y., Wang, X., Gheisari, M., and Luo, F. Shrewdattack: Low cost high accuracy model extraction. *Entropy (Basel)*, 25:282, 2023.

Lou, C., Wang, Y., Li, J., Qian, Y., and Li, X. Graph neural network for fraud detection via context encoding and adaptive aggregation. *Expert Systems with Applications*, 261:125473, 2025. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2024.125473. URL https://www.sciencedirect.com/science/article/pii/S0957417424023406.

Lu, H. and Uddin, S. A weighted patient network-based framework for predicting chronic diseases using graph neural networks. *Scientific Reports*, 11(22607):1–12, 2021.

Luan, S., Hua, C., Xu, M., Lu, Q., Zhu, J., Chang, X.-W., Fu, J., Leskovec, J., and Precup, D. When do graph neural networks help with node classification? investigating the homophily principle on node distinguishability. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=kJmYu3Ti2z.

Motie, S. and Raahemi, B. Financial fraud detection using graph neural networks: A systematic review. *Expert*

Systems with Applications, 240(C):122–156, 2024. doi: https://doi.org/10.1016/j.eswa.2023.122156.

Mujkanovic, F., Geisler, S., Günnemann, S., and Bojchevski, A. Are defenses for graph neural networks robust? In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=yCJVkELVT9d.

Nealey, T., Daignault, R. M., and Cai, Y. Trade secrets in life science and pharmaceutical companies. *Cold Spring Harbor Perspectives in Medicine*, 5(4):a020982, 2015. doi: https://doi.org/10.1101/cshperspect.a020982.

Oliynyk, D., Mayer, R., and Rauber, A. I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Computing Surveys*, 55: 1–41, 2023.

Oloulade, B. M., Gao, J., Chen, J., Al-Sabri, R., and Wu, Z. Cancer drug response prediction with surrogate modeling-based graph neural architecture search. *Bioinformatics*, 39(8):btad478, 2023. doi: https://doi.org/10.1093/bioinformatics/btad478.

Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120.

Pal, S., Gupta, Y., Shukla, A., Kanade, A., Shevade, S., and Ganapathy, V. Activethief: Model extraction using active learning and unannotated public data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01): 865–872, 2020. doi: 10.1609/aaai.v34i01.5432.

Paul, S. G., Saha, A., Hasan, M. Z., Noori, S. R. H., and Moustafa, A. A systematic review of graph neural network in healthcare-based applications: Recent advances, trends, and future directions. *IEEE Access*, 12:15145–15170, 2024. doi: 10.1109/ACCESS.2024.3354809.

Qin, Z., Liu, Y., He, Q., and Ao, X. Explainable graph-based fraud detection via neural meta-graph search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pp. 4414–4418, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557598. URL https://doi.org/10.1145/3511808.3557598.

Rosenthal, J., Enouen, E., Pham, H. V., and Tan, L. Disguide: Disagreement-guided data-free model extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9614–9622, Jun. 2023. doi: 10.1609/aaai.v37i8.26150.

Shen, Y., He, X., Han, Y., and Zhang, Y. Model stealing attacks against inductive graph neural networks. *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1175–1192, 2022.

Shi, Y., Sagduyu, Y., and Grushin, A. How to steal a machine learning classifier with deep learning. In *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, pp. 1–5, 2017. doi: 10.1109/THS.2017.7943475.

Sun, R., Dai, H., and Yu, A. W. Does GNN pretraining help molecular representation? In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=uytgM9N0vlR.

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. Stealing machine learning models via prediction apis. *Proceedings of the 25th USENIX Security Symposium, August 10–12, 2016 in Austin, TX*, 2016.

Tran, H. N. T., Thomas, J. J., and Ahamed Hassain Malim, N. H. Deepnc: a framework for drug-target interaction prediction with graph neural networks. *PeerJ.*, 10:e13163, 2022.

Vora, L. K., Gholap, A. D., Jetha, K., Thakur, R. R. S., Solanki, H. K., and Chavda, V. P. Artificial intelligence in pharmaceutical technology and drug delivery design. *Pharmaceutics*, 15(7):1916, 2023.

Wang, Y., Li, Z., and Barati Farimani, A. *Graph Neural Networks for Molecules*, pp. 21–66. Springer International Publishing, 2023. ISBN 978-3-031-37196-7. doi: https://doi.org/10.1007/978-3-031-37196-7_2.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. Knowledge graph embedding by translating on hyperplanes. *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 28(1): 1112–1119, 2014. doi: https://doi.org/10.1609/aaai.v28i1.8870.

Wieder, O., Kohlbacher, S., Kuenemann, M., Garon, A., Ducrot, P., Seidel, T., and Langer, T. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

Wu, B., Yang, X., Pan, S., and Yuan, X. Model extraction attacks on graph neural networks: Taxonomy and realisation. *ASIA CCS '22: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pp. 337–350, 2022.

Wu, B., Zhang, H., Yang, X., Wang, S., Xue, M., Pan, S., and Yuan, X. Graphguard: Detecting and counteracting training data misuse in graph neural networks. *arXiv preprint arXiv:2312.07861*, 2023a.

Wu, B., Yuan, X., Wang, S., Li, Q., Xue, M., and Pan, S. Securing Graph Neural Networks in MLaaS: A Comprehensive Realization of Query-based Integrity Verification . In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 2534–2552, Los Alamitos, CA, USA, 2024. IEEE Computer Society. doi: 10.1109/SP54263.2024.00110. URL https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00110.

Wu, W., Zhang, J., Wei, V. J., Chen, X., Zheng, Z., King, I., and Lyu, M. R. Practical and efficient model extraction of sentiment analysis apis. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 524–536, 2023b. doi: 10.1109/ICSE48619.2023.00054.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021. doi: doi:10.1109/TNNLS.2020.2978386.

Xu, Z., Gan, Z., Zhou, D., Shen, S., Lu, J., and Cai, T. Inference of dependency knowledge graph for electronic health records. *arXiv preprint arXiv:2312.15611*, 2023.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings. In *ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning*, volume 48 of *ICML'16*, pp. 40–48, 2016. doi: https://doi.org/10.5555/3045390.3045396.

Zafeiropoulos, N., Bitilis, P., Tsekouras, G. E., and Kotis, K. Graph neural networks for parkinson's disease monitoring and alerting. *Sensors (Basel)*, 23(23):8936, 2023. doi: 10.3390/s23218936.

Zang, X., Zhao, X., and Tang, B. Hierarchical molecular graph self-supervised learning for property prediction. *Communications Chemistry*, 6(34), 2023.

Zhang, W., Yang, Z., Wang, Y., Shen, Y., Li, Y., Wang, L., and Cui, B. Grain: improving data efficiency of graph neural networks via diversified in fluence maximization. *Proceedings of the VLDB Endowment*, 14(11):2473–2482, 2021.

Zhang, X. and Zitnik, M. Gnnguard: Defending graph neural networks against adversarial attacks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9263–9275. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/690d83983a63aa1818423fd6edd3bfdb-Paper.pdf.

Zhang, Y., Xu, Y., and Zhang, Y. A graph neural network node classification application model with enhanced node association. *Applied Sciences*, 13(12):7150, 2023. doi: https://doi.org/10.3390/app13127150.

Zhang, Z., Chen, L., Zhong, F., Wang, D., Jiang, J., Zhang, S., Jiang, H., Zheng, M., and Li, X. Graph neural network approaches for drug-target interactions. *Current Opinion in Structural Biology*, 73:102327, 2022. doi: https://doi.org/10.1016/j.sbi.2021.102327.

Zhao, B., Xu, W., Guan, J., and Zhou, S. Molecular property prediction based on graph structure learning. *Bioinformatics*, 40(5):btae304, 2024. doi: https://doi.org/10.1093/bioinformatics/btae304.

Zhao, K., Li, L., Ding, K., Gong, N. Z., Zhao, Y., and Dong, Y. A survey of model extraction attacks and defenses in distributed computing environments. *arXiv preprint arXiv:2502.16065*, 2025.

Zhou, D., Gan, Z., Shi, X., Patwari, A., Rush, E., Bonzel, C.-L., Panickan, V. A., Hong, C., Ho, Y.-L., Cai, T., Costa, L., Li, X., Castro, V. M., Murphy, S. N., Brat, G., Weber, G., Avillach, P., Gaziano, J. M., Cho, K., Liao, K. P., Lu, J., and Cai, T. Multiview incomplete knowledge graph integration with application to cross-institutional ehr data harmonization. *Journal of Biomedical Informatics*, 133:104147, 2022. doi: https://doi.org/10.1016/j.jbi.2022.104147.

Zhou, D., Tong, H., Wang, L., Liu, S., Xiong, X., Gan, Z., Griffier, R., Hejblum, B., Liu, Y.-C., Hong, C., Bonzel, C.-L., Cai, t., Pan, K., Ho, Y.-L., Costa, L., Panickan, V. A., Gaziano, J. M., Mandl, K., Jouhet, V., Thiebaut, R., Xia, Z., Cho, K., Liao, K., and Cai, T. Representation learning to advance multi-institutional studies with electronic health record data. *arXiv preprint arXiv:2502.08547*, 2025.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. doi: https://doi.org/10.1016/j.aiopen.2021.01.001.

A. Proofs to Theoretical Analysis

In this section, we provide the proof to the theoretical contribution of CEGA.

Proof to Proposition 3.1: Before we calculate the complexity of the γth cycle of CEGA, we need to conduct the forward step of $f_{\gamma-1}$ to obtain the softmax scores for each nodes of interest that are used in the subsequent procedures of CEGA. (Blakely et al., 2021) shows that the forward step for an L-layer GCN has time complexity $O(LN^2d + LNd^2)$ and space complexity $O(N^2 + Ld^2 + LNd)$. The output takes $O(CN + Nh)$ space and is shared among the following tasks. Here, h is the dimension of the embeddings.

For CEGA's entropy-based approach to evaluate the uncertainty based on historical information, the space is required for $O(Cn_{\gamma-1})$ softmax scores. Our task is to calculate the entropy of the $O(n_{\gamma-1})$ nodes, which involves the computation of $O(Cn_{\gamma-1})$. Sorting the nodes involves $O(n_{\gamma-1}\log n_{\gamma-1})$ time complexity and $O(n_{\gamma-1})$ space usage.

For the more resource-consuming perturbation-based alternative, we first consider the complexity involved each time we redo the perturbation. For each time we take the perturbation, we prepare the perturbed attributes for the nodes in $\mathcal{G}_{\mathrm{a}}$, which takes $O(Nd)$ space and has $O(Nd)$ time consumption. As we pass the perturbed attributes forward through the GNN model and calculate the softmax scores for the perturbed scores, the time complexity is $O(LN^2d + LNd^2)$. For each time we redo the perturbation, the output takes $O(CN)$ space, and a time complexity of $O(Cn_{\gamma-1})$ is required to compare the labels derived from the perturbed softmax score and the original score. The output is stored in a vector with $O(n_{\gamma-1})$ dimensions, where the space for temporarily perturbed attributes and softmax scores can be released after each perturbation. We redo the perturbation procedure S times and sort the nodes based on $\mathcal{L}_2^{\gamma}$ with $O(n_{\gamma-1}\log n_{\gamma-1})$ time complexity.

Taking summations on all the procedures implies that the additional time complexity is $O(Cn_{\gamma-1} + n_{\gamma-1}\log n_{\gamma-1})$ for CEGA's entropy-based approach. For the perturbation-based alternative, the additional time complexity is $O(SNd + SLN^2d + SLNd^2 + SCn_{\gamma-1} + n_{\gamma-1}\log n_{\gamma-1})$. Given all the assumptions of Proposition 3.1, we summarize that the time complexity are $O(CN + N\log N)$ and $O(SLN^2d + SLNd^2)$, respectively. Under a similar step of calculation, we have that the space complexity of CEGA's entropy-based approach and the perturbation-based alternative are $O(CN)$ and $O(N^2 + Ld^2 + LNd)$, respectively.

Proof to Theorem 3.2: In the proof, we consider a generic GNN network. Taking the GCN model (Kipf & Welling, 2017) as an example, we have that

$$\mathbf{G}^{(1)} = \sigma\big(f(\mathbf{A})\mathbf{X}\mathbf{W}^{(1)}\big);\;\; \mathbf{G}^{(\ell+1)} = \sigma\big(f(\mathbf{A})\mathbf{G}^{(\ell)}\mathbf{W}^{(\ell+1)}\big).$$

Here $f(\mathbf{A}) = \widehat{\mathbf{D}}^{-1/2}\widehat{\mathbf{A}}\widehat{\mathbf{D}}^{-1/2}$. The last layer before the output is conducted by the softmax procedure. For one specific node τ, we consider a generic two-layer GNN model

$$\mathbf{g}_{\tau}^{(1)} = \sigma\Big(\mathbf{W}_{sa}^{(1)}\mathbf{x}_{\tau} + \mathbf{W}_{n}^{(1)}\sum_{\mu\in\mathcal{N}_{\tau}}\mathbf{x}_{\mu}\Big);\;\; \mathbf{g}_{\tau}^{(2)} = \mathbf{W}_{out}\mathbf{g}_{\tau}^{(1)} + \mathbf{b};\;\; \widehat{\mathbf{y}}_{\tau} = \mathrm{softmax}(\mathbf{g}_{\tau}^{(2)}). \tag{5}$$

Here $\mathbf{W}_{sa}^{(1)}$ and $\mathbf{W}_{n}^{(1)}$ represents the weights assigned to the self-attention term and the neighborhood for the node τ, specifically. $\mathcal{N}_{\tau}$ represents the neighborhood of the node τ. Substituting the node attributes $\mathbf{x}_{\tau}$ to $\widetilde{\mathbf{x}}_{\tau}$ indicates that

$$\widetilde{\mathbf{g}}_{\tau}^{(1)} = \sigma\Big(\mathbf{W}_{sa}^{(1)}\widetilde{\mathbf{x}}_{\tau} + \mathbf{W}_{n}^{(1)}\sum_{\mu\in\mathcal{N}_{\tau}}\widetilde{\mathbf{x}}_{\mu}\Big);\;\; \widetilde{\mathbf{g}}_{\tau}^{(2)} = \mathbf{W}_{out}\widetilde{\mathbf{g}}_{\tau}^{(1)} + \mathbf{b};\;\; \widehat{\mathbf{y}}_{\tau}^{p} = \mathrm{softmax}(\widetilde{\mathbf{g}}_{\tau}^{(2)}). \tag{6}$$

Aggregating (5) and (6) indicates that

$$\begin{aligned}
&\big\|\widehat{\mathbf{y}}_{\tau} - \widehat{\mathbf{y}}_{\tau}^{p}\big\|_2 \le \mathcal{C}_{soft}\big\|\mathbf{g}_{\tau}^{(2)} - \widetilde{\mathbf{g}}_{\tau}^{(2)}\big\|_2 \le \mathcal{C}_{soft}\big\|\mathbf{W}_{out}\big\|_2\big\|\mathbf{g}_{\tau}^{(1)} - \widetilde{\mathbf{g}}_{\tau}^{(1)}\big\|_2 \\
&\le \mathcal{C}_{soft}\,\mathcal{C}_{\sigma}\big\|\mathbf{W}_{out}\big\|_2\big\|\mathbf{W}_{sa}^{(1)}\big\|_2\big\|\mathbf{x}_{\tau} - \widetilde{\mathbf{x}}_{\tau}\big\|_2 + \mathcal{C}_{soft}\,\mathcal{C}_{\sigma}\big\|\mathbf{W}_{out}\big\|_2\big\|\mathbf{W}_{n}^{(1)}\big\|_2\Big\|\sum_{\mu\in\mathcal{N}_{\tau}}\Big(\mathbf{x}_{\mu} - \widetilde{\mathbf{x}}_{\mu}\Big)\Big\|_2.
\end{aligned}$$

Here $\mathcal{C}_{soft}$ and $\mathcal{C}_{\sigma}$ denotes the Lipschitz constant for softmax function and the activation function $\sigma(\cdot)$, respectively. The norms $\big\|\mathbf{W}_{out}\big\|_2$, $\big\|\mathbf{W}_{sa}^{(1)}\big\|_2$, and $\big\|\mathbf{W}_{n}^{(1)}\big\|_2$ are bounded from above given that the estimation function is bounded after one step of model fitting. For simplicity, we re-arrange the terms and form the inequality such that

$$\big\|\widehat{\mathbf{y}}_{\tau} - \widehat{\mathbf{y}}_{\tau}^{p}\big\|_2 \le \eta_1\big\|\mathbf{x}_{\tau} - \widetilde{\mathbf{x}}_{\tau}\big\|_2 + \eta_2\Big\|\sum_{\mu\in\mathcal{N}_{\tau}}\Big(\mathbf{x}_{\mu} - \widetilde{\mathbf{x}}_{\mu}\Big)\Big\|_2,$$

for some positive constants $\eta_1, \eta_2 < \infty$. Given that $\mathbf{x}_\mu - \widetilde{\mathbf{x}}_\mu \sim \mathcal{N}(0, \epsilon^2)$ for any $\mu \in \mathcal{N}_\tau$, we can apply Hoeffding's inequality, which implies that

$$\mathbb{P}\Big(\sum_{\mu \in \mathcal{N}_\tau} \Big(\mathbf{x}_\mu - \widetilde{\mathbf{x}}_\mu \Big) \geq t \Big) \leq \exp\Big(- \frac{t^2}{2\,|\mathcal{N}_\tau| \epsilon^2} \Big).$$

We then select

$$\epsilon = \min \left\{ \frac{\zeta}{\eta_1 \sqrt{2 \log(1/\delta)}}, \frac{\zeta}{\eta_2 \sqrt{2 |\mathcal{N}_\tau| \log(1/\delta)}} \right\},$$

where we guarantee that the difference of the outcome label probability has an upper bound with a large probability.

B. Supplementary Results and Discussion

In this section, we elaborate the discussion of the additional results of the experiment based on our implementation of CEGA, which is available at `https://github.com/LabRAI/CEGA`, to a series of widely studied benchmark graph datasets.

B.1. Datasets

Dataset	#Nodes	#Edges	#Features	#Classes
AmzComputer	13,752	491,722	767	10
AmzPhoto	7,650	238,163	745	8
CoauthorCS	18,333	163,788	6,805	15
CoauthorPhysics	34,493	495,924	8,415	5
Cora Full	19,793	126,842	8,710	70
DBLP	17,716	105,734	1,639	5

Table 3. Dataset statistics

Table 3 presents the statistics of six benchmark datasets used in our study, covering a range of node, edge, feature, and class distributions. Amazon-Computer and Amazon-Photo are e-commerce co-purchase networks characterized by dense connectivity. Coauthor-CS and Coauthor-Physics represent academic collaboration graphs with a larger number of features. Cora_Full and DBLP are citation network datasets where nodes represent academic papers and edges denote citation relationships. Cora Full spans diverse machine learning subfields with 70 classes, while DBLP focuses on computer science publications with five broad research categories. These datasets provide diverse graph structures and feature distributions for evaluating model performance.

B.2. Setup of Hyperparameters

Setup of GNN Model Extraction We follow the *Attack 0* framework of (Wu et al., 2022) to perform GNN model extraction. Initially, we train a target model, f_{T}, for 1000 epochs with a learning rate of 1e-3, which provides predictions for surrogate model training. If training and test sets are not provided, we randomly select 60% of the nodes for training and use the remaining 40% for testing. This serves as the initial setup; however, following the *Attack 0* framework, these masks are later adjusted based on whether the nodes are subject to queries for extraction.

Setup of Node Selection Models For our experiments, we randomly set the candidate node pool $\mathcal{V}_{\mathrm{a}}$ comprising 10% of the nodes in graphs with fewer classes, including Amazon-Computer, Amazon-Photo, Coauthor-CS, Coauthor-Physics, and DBLP. For graph with significantly higher number of classes (e.g., Cora-Full, which has 70 classes), the pool includes 25% of the nodes. Our setup is inspired by widely accepted works, such as (Wu et al., 2022; Shen et al., 2022). In the initialization step, we randomly select 2 nodes from each class across all the tested datasets, resulting in a total of $2C$ nodes, where C is the number of classes. In practice, this procedure remains feasible as the extractors can attain partial knowledge of the class distribution through domain expertise or external sources, especially when such knowledge offers strategic advantages in building a high-fidelity extracted model. For the remaining budget, we employ different node selection methods, with the total budget capped at $20C$.

For the baseline node selection methods, hyperparameters are set as follows. In GRAIN (Ball-D), the radius r is fixed at 0.005 for all datasets, while in GRAIN (NN-D), γ is set to 1. For AGE, we adopt the time-sensitive parameter setting, where $\gamma_t \sim \text{Beta}(1, n_t)$, with n_t increasing as iterations progress, defined as $n_t = 1.05 - 0.95^t$. Here, t denotes the number of iterations. The parameters α_t and β_t are set as $\alpha_t = \beta_t = \frac{1-\gamma_t}{2}$.

For our proposed method, in cycle γ, CEGA queries $\kappa = 1$ node and trains a 2-layer GCN model with $\{\mathcal{V}_\gamma, \mathcal{G}_\text{a}\}$ for $E = 1$ epoch. In the analysis for node diversity, we set the weight $\rho = 0.8$ to ensure that the order $\mathcal{R}_3^\gamma$ is designed to prioritize the nodes associated with underrepresented prediction labels. For the weighted average ranking mechanism, we set

$$\omega_1(\gamma) = \alpha_1 + \Delta e^{-\lambda\gamma};\ \ \omega_2(\gamma) = \alpha_2 + \Delta\big(1 - e^{-\lambda\gamma}\big);\ \ \omega_3(\gamma) = \alpha_3(1 - e^{-\gamma}). \tag{7}$$

We design this weighting approach under the heuristics such that the subgraph structure is the most important information when the information gathered in the history is not accurate enough to guide further queries. As the number of queries increases, the contribution from history information becomes more prominent, and diversity concerns need to be considered more seriously. In practice, we set the initial weight coefficients as $\alpha_1 = \alpha_2 = \alpha_3 = 0.2$, the measurement of the initial weight gap between $\mathcal{R}_1^\gamma$ and $\mathcal{R}_2^\gamma$ as $\Delta = 0.6$, the measurement of the curvature for the weight changes as $\lambda = 0.3$. The CEGA hyperparameters $(\alpha_1, \alpha_2, \alpha_3, \Delta, \lambda)$ are applied uniformly across all the tested graph datasets to mitigate potential concerns of tuning bias.

After the node selection process, we train a 2-layer GCN with a hidden dimension of 16. The model is optimized with a learning rate of 1e-3 and trained for 1000 epochs. For AGE, we apply a warm-up period of 400 epochs. All experiments are conducted on two NVIDIA RTX 6000 Ada GPUs. Model performance is evaluated for node selections ranging from $2C$ to $20C$, with evaluations performed at every C. Selected nodes are trained for 1000 epochs using a learning rate of 1e-3.

B.3. Model Performance Gap

Table 4. Performance gaps between budget-constrained models and subgraph models, measured by Accuracy, Fidelity, and F1, across various datasets. The best results are in **bold**.

		CoCS	CoP	AmzC	AmzP	Cora_Full	DBLP
Accuracy	**Random**	2.68 ± 0.6	3.47 ± 1.0	3.60 ± 1.1	2.07 ± 1.6	1.71 ± 0.5	12.24 ± 1.3
	GRAIN(NN-D)	1.69 ± 0.6	1.87 ± 0.8	3.41 ± 1.1	1.56 ± 0.4	-0.09 ± 1.1	13.14 ± 1.0
	GRAIN(ball-D)	2.02 ± 0.6	1.93 ± 1.0	4.78 ± 1.3	2.58 ± 1.2	0.04 ± 1.0	12.98 ± 1.2
	AGE	**0.78 ± 0.4**	1.56 ± 0.3	2.33 ± 0.9	1.39 ± 1.8	0.90 ± 0.4	8.98 ± 2.0
	CEGA	0.91 ± 0.4	**1.39 ± 0.4**	**1.19 ± 0.8**	**0.58 ± 0.3**	**-1.36 ± 0.2**	**7.79 ± 1.1**
Fidelity	**Random**	3.20 ± 0.7	4.20 ± 1.0	4.24 ± 1.2	2.34 ± 1.6	3.84 ± 0.5	14.85 ± 1.7
	GRAIN(NN-D)	2.26 ± 0.7	2.49 ± 1.0	4.01 ± 1.4	1.74 ± 0.6	1.34 ± 1.7	16.11 ± 1.2
	GRAIN(ball-D)	2.61 ± 0.7	2.50 ± 1.2	5.77 ± 1.7	3.06 ± 1.4	1.68 ± 1.2	15.52 ± 1.4
	AGE	**1.02 ± 0.4**	2.07 ± 0.4	2.70 ± 1.0	1.77 ± 2.4	2.31 ± 0.7	11.50 ± 2.2
	CEGA	1.25 ± 0.5	**1.84 ± 0.5**	**1.69 ± 0.8**	**0.56 ± 0.5**	**0.06 ± 0.4**	**9.98 ± 1.2**
F1	**Random**	6.72 ± 1.6	5.44 ± 1.8	4.61 ± 2.8	3.69 ± 3.5	0.49 ± 0.3	19.78 ± 5.4
	GRAIN(NN-D)	2.77 ± 1.5	2.69 ± 1.1	4.00 ± 1.2	2.87 ± 0.8	-0.97 ± 1.0	19.10 ± 4.0
	GRAIN(ball-D)	3.04 ± 1.3	2.75 ± 1.5	8.53 ± 5.3	3.85 ± 1.7	-0.93 ± 0.3	16.95 ± 3.7
	AGE	**0.84 ± 0.6**	2.13 ± 0.5	4.57 ± 5.1	1.30 ± 1.6	-1.12 ± 0.3	11.93 ± 2.9
	CEGA	1.09 ± 0.7	**1.89 ± 0.6**	**0.86 ± 2.8**	**0.80 ± 0.7**	**-3.43 ± 0.5**	**10.02 ± 1.0**

Table 4 quantifies the performance gap between budget-constrained models and subgraph models across various datasets, using Accuracy, Fidelity, and F1 as evaluation metrics. A smaller gap indicates a more effective node selection strategy, with negative values suggesting cases where the budget-constrained model outperforms the subgraph model. Notably,

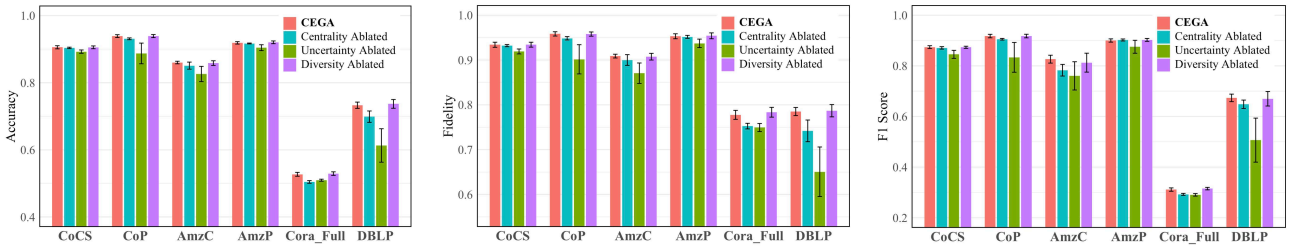

Figure 3. Ablation study results on accuracy (left), fidelity (middle), and F1 score (right) for CEGA and variants with one of the three node evaluation modules removed.

CEGA almost outperforms the benchmark models across all three metrics, demonstrating its effectiveness in maintaining model performance under stringent budget constraints. Additionally, this table provides detailed numerical insights that complement the trends illustrated in Figure 2.

B.4. Ablation Study

To address **RQ3**, an ablation study is conducted where we implement two of the three analyses proposed in Section 3.2. Specifically, we compare the original CEGA model against three variants: (1) *CEGA with Centrality Module Ablated*: A variant removing the centrality-based selection mechanism, which we expect to evaluate the contribution of the subgraph model structure to the selection of nodes to be queried; (2) *CEGA with Uncertainty Module Ablated*: A variant removing the contribution of prediction uncertainty under the guidance of history information, which we expect to evaluate the contribution of history information extracted from previous queries; (3) *CEGA with Diversity Module Ablated*: A variant removing the contribution that enhances the diversity of the selected nodes, which we expect to evaluate the contribution of node diversity in providing a more stable estimation with a smaller variation across different random initialization setups. The setup of our ablation study follows the standard of the most recent works on GNN node classification tasks, such as (Lin et al., 2025). In practice, we set the respective cycle-specific weight $\omega_i(\gamma) = 0$, as specified in (7), among all $\gamma \in \{1, 2, ..., \Gamma\}$ for the specific index $i \in \{1, 2, 3\}$.

The results of the ablation study, as shown in Figure 3, are consistent across all three performance metrics, namely accuracy, fidelity, and F1 score. This indicates that the ablation study yields stable findings regardless of the evaluation criterion. This alignment reinforces the robustness of our proposed approach and suggests that each module contributes meaningfully to the overall performance of the model.