Measuring Legislature-Aligned Privacy Risks in Synthetic Graphs

Abele Mălan
University of Neuchâtel
Neuchâtel, Switzerland
abele.malan@unine.ch

Ahmad Al Kurdi
RPTU University Kaiserslautern-Landau
Kaiserslautern, Germany
a.kurdi@cs.rptu.de

Stefanie Roos
RPTU University Kaiserslautern-Landau
Kaiserslautern, Germany
stefanie.roos@cs.rptu.de

Lydia Chen
University of Neuchâtel
Neuchâtel, Switzerland
Delft University of Technology
Delft, Netherlands
lydiaychen@ieee.org

Abstract

Graphs are a ubiquitous form of structured data, with applications in many privacy-sensitive domains, such as social and healthcare. As for other modalities, modern graph synthesizers enable the creation of realistic synthetic samples, facilitating privacy-preserving data sharing while maintaining high utility. Unfortunately, unlike such other modalities, there is no relevant work on evaluating the privacy risk associated with synthetic graphs. The fact that graphs, unlike, e.g., tables, naturally capture relationships between individuals means that existing approaches are not easily transferable.

To allow quantifying these privacy risks, we introduce *SyntheGrAnon*, a framework for evaluating **synthe**tic **gr**aph **anon**ymity. *SyntheGrAnon* primarily targets the singling out, linkability, and inference risks outlined in the EU GDPR at the node and community levels, while also including edge-level attacks as an extension of the node-level setting. We design attacks tailored to synthetic graphs and, in addition, extend the existing methodology by leveraging multiple synthetic samples for our black-box attacks.

In our evaluation, spanning datasets from social and financial domains and five generative graph models, including three modern diffusion-based options, we find that our attacks are mostly effective, achieving risks close to the maximum of 1 in some cases. However, they struggle with large-scale, attribute-scarce graphs.

Keywords

graphs, synthetic data, privacy

1 Introduction

Both at the individual [4] and legislative levels (the EU's GDPR [33], Brazil's LGPD [8], California's CCPA/CPRA [31], and Switzerland's FADP [32]), there has been a push to protect the ever-increasing quantity of shared personal data from misuse. As the modern capabilities of many essential services are highly reliant on collecting and processing data from individuals (e.g., medical research, anti-fraud financial models, urban planning), tension exists between improving utility in the form of better service or research and protecting privacy. The issue of protecting private data is a long-standing and well-studied one: classic anonymization techniques involve redacting sensitive data or decreasing the specificity of retained information [11, 38]. Unfortunately, such approaches often struggle to strike the optimal balance between maximizing privacy protection and maintaining utility, and introducing differential privacy into the data analysis process comes at significant challenges for practitioners [36]. Synthetic data generated by modern deep learning generative models represents an appealing alternative. Such models can faithfully learn to approximate complex data distributions from a set of examples. They can sample new synthetic examples belonging to the distribution without being tied to any specific real example [12]. Nevertheless, in practice, many such models still exhibit some degree of memorization or related issues, which risk leaking sensitive information from their training data [42]. Recent works [14, 20] investigate privacy risks in synthetic data, though they primarily focus on the tabular domain.

In contrast, graph data is essential for representing complex relationships between a set of entities. They are the standard way to represent networks in privacy-sensitive domains such as social [30], financial [5], and epidemiological modeling [13], as well as computer [43] or transportation [9] networks. Consider the real-world example of using synthetic contact graphs derived from census information to model disease outbreaks [13]. In these networks, nodes represent individuals, their attributes encode demographic or occupational information, and edges capture contacts through which infection may spread. While such a method helps study public health intervention strategies without directly releasing underlying population data, it also illustrates why privacy leaks from synthetic graphs can be serious, as they may involve sensitive data such as medical information. Moreover, when augmented with additional information in the form of attributes, at the node, community, or edge level, graphs encode substantial amounts of information. For instance, in a social network graph, nodes could represent individuals and include personal information such as names, ages, or occupations. Edges could represent friendship relations and include information about the frequency of communication or the duration of the friendship. The attributes of different nodes are like records in a table, with the structural connectivity between them representing an additional layer of complexity that current algorithms for evaluating privacy risks cannot capture.

Proceedings on Privacy Enhancing Technologies 2026(4), 181–197

https://doi.org/10.56553/popets-2026-0115

When it comes to privacy in graph representations, nodes are a primary concern, as these generally represent individuals or concepts associated with them (e.g., persons, patients, or bank accounts). In the contact tracing setting above, a synthetic contact graph could, for instance, enable an attacker to single out a person from the training population using only coarse properties, such as being a 60+ year-old plumber with at most ten contacts; to link partial records, like occupation and number of contacts, as belonging to the same person; or to infer a sensitive attribute, such as an occupation, health status, or high-risk contact pattern, from otherwise public information. Another example comes from the Netflix Prize dataset, where researchers were able to deanonymize a significant portion of users by cross-referencing their movie ratings with publicly available ones from IMDb [28]. Moreover, some leaks primarily affect groups of users or specific communities rather than individuals. For the contact tracing example, such risks target entire subpopulations, such as neighborhoods, workplaces, or marginalized areas, whose aggregate structure or attributes can be exposed even when no single individual is explicitly named. An additional example of such violations is the inadvertent leak of secret army bases by US military personnel through the heatmap feature of the workout-tracking platform Strava [18].

Modern graph generation models employ the iterative denoising paradigm, a widely adopted approach in various modalities, to produce high-quality samples [2, 21, 37]. In non-generative graph learning scenarios, a vast number of works investigate privacy attacks and defenses, but much less insight exists in the generative case [23]. In one of the few works tackling privacy defenses, Yang et al. [44] apply a graph-specific adaptation of differential privacy to early graph models based on VAE and GAN paradigms. However, these techniques have severe limitations in the types and, especially, the quality of graphs they can generate compared to current denoising models, and are hence no longer used.

To address the lack of insight into **synthe**tic **gr**aph **anon**ymity evaluation, we introduce the *SyntheGrAnon* framework. We include in *SyntheGrAnon* realizations of the three main types of attacks with legislative relevance, namely singling out, linkability, and inference [33], at the node, community, and edge levels. Informally, singling out refers to re-identifying a node, community, or edge from the training data using the synthetic data; linkability enables matching data about an individual or community across different sources; and inference refers to the ability to infer (sensitive) private information from public information.

We adopt the risk assessment framework previously designed for tabular data to quantify the risk of the three types of attacks [14]. However, since tabular data lacks the relationship information that graphs provide, we design, as our main focus, six novel attacks: three to infer individual information and three to infer community information, leveraging graph structure. Concretely, we determine key structural properties of nodes, such as degree or centrality, and of communities, such as transitivity. We combine the resulting structural attributes with tabular node attributes to effectively single out nodes, link their partial information, or infer their secret attributes from synthetic data. Similarly, we can learn information about communities. We additionally formulate edge-level attacks as an extension of the node-level setting, representing each edge by its endpoint nodes and edge-specific structural attributes.

In contrast to previous work, our attacks also account for the case where an attacker has access to multiple datasets generated by the same model, though the attack remains black-box. Hence, we use multiple synthetic graphs in the singling out attack, forcing properties that potentially identify an individual or community to be checked against multiple synthetic graphs before they are considered likely to single out a party. In this manner, the certainty of having found a pattern that exists in the real data increases.

We evaluate our six attacks across three datasets of attributed graphs from the social and financial domains, using five graph generation models: three diffusion-based (EDGE [7], GGSD [26], and GruM [19]) and two non-diffusion-based (SPECTRE [24] and GRAN [22]). Node-level singling out and inference can reach very high risks on ego-network datasets for some models and attack parameterizations, whereas node-level linkability remains comparatively low. Community-level singling out is consistently weaker, while community-level linkability and inference can be substantial. Edge-level results show a similar pattern: inference can be highly effective, but singling out and linkability are much weaker. In terms of high-level graph properties, large, attribute-scarce graphs, such as Elliptic, generally reduce attack success, though they do not eliminate risk. Finally, higher utility or fidelity often coincides with higher privacy risk, but the relationship is not monotonic and depends on the dataset, attack, and synthesizer family.

In summary, our contributions are:

- We propose *SyntheGrAnon*, the first analysis of privacy risks in synthetic graphs, focusing on node- and community-level attacks, while further formulating edge-level attacks as an extension of the node-level setting.
- By leveraging multiple synthetic graphs in an attack, we can increase the certainty of identifying individuals in the unknown training data, an approach that might also be of interest for other data modalities.
- We extensively evaluate performance over multiple modern synthesizers and datasets in privacy-sensitive domains, finding that (i) high-risk attacks are possible at all granularity levels; (ii) smaller, attribute-rich graphs tend to be easier to attack; and (iii) while samples from high fidelity models yield higher risk on average, the increase in risk is non-uniform across datasets and attacks.
- We open-source our framework to encourage further study of graph generation models from a privacy risk perspective by making our code available at https://github.com/AbeleMM/SyntheGrAnon.

2 Related Work

Graph Generative Models. Deep learning models for data generation have made significant strides, thanks to advances in architectures such as transformers and modeling techniques like diffusion [2]. State-of-the-art models for data generation in modalities including image [21], video [39], time series [45], and tabular [37], as well as graphs [34] employ such stochastic interpolants. Notably, graph generation faces specific challenges: working with the discrete space of the adjacency matrix, scaling generation as the graph size (node and edge counts) grows, and integrating graph structure and node attribute generation. Modern diffusion-based graph

generation models propose different approaches to balancing or prioritizing these challenges. We highlight the representative state-of-the-art graph synthesizers.[1] EDGE [7] is a highly scalable model with a discrete-state diffusion formulation that helps preserve the natural sparsity of graph structures. It explicitly incorporates node degrees into its architecture to improve the statistical similarity between the generated and real graphs. GGSD [26] uses diffusion to synthesize eigenvalues and corresponding eigenvectors from the Laplacian matrix, from which it then reconstructs the adjacency. While operating in the continuous spectral space is advantageous for scalability, it unfortunately limits generative capabilities, particularly for modeling fine-grained changes in topology. GruM [19] uses a mixture of multiple diffusion processes conditioned on samples from the training data, achieving very high generation quality at the expense of scalability compared to the abovementioned models. It also provides more effective support for jointly generating node attributes with the structure. For evaluation, we also consider two established non-diffusion graph generators as baselines: GRAN, an autoregressive recurrent-attention model [22], and SPECTRE, a spectrally-conditioned generative adversarial network (GAN) [24].

Privacy Risks for Synthetic Data. A vast amount of work exists on privacy-related attacks and defenses for sensitive data in tasks such as data mining and release [27]. In deep learning scenarios, whether discriminative or generative, attacks are commonly grouped based on the level of access they assume about a model: black-box attacks require only the ability to run the model on some inputs, while white-box attacks involve unrestricted access to the model's weights or intermediate outputs [35].

In the following, we treat the training data as sensitive rather than the model or its parameters. For discriminative models, membership inference is a common attack that aims to determine whether a given sample was part of a model's training set [10]. Generative models enable a broader class of attacks, as the synthetic data they produce resides in the same space as the original training data. For instance, models have been shown to memorize information about their training samples [6]. From a defense standpoint, differential privacy is a popular strategy even for generative models [1, 20]. However, a comprehensive evaluation of privacy risk requires a standardized way to quantify the potential for harm directly.

As the most closely related work to ours, Anonymeter [14] outlines a legally aligned framework for quantifying privacy risk in synthetic tabular data. We adopt Anonymeter's risk semantics and estimation protocol, but not its tabular-optimized data representation. In the tabular case, attacked items are always table rows. In graphs, they may be nodes, edges, or even entire communities, each posing different challenges. Nodes in a graph relate to each other via the edges connecting them, instead of being drawn independently from some prior distribution. Meanwhile, graphs in a dataset vary in size and contain substantial information, making it challenging to efficiently create meaningful, compact embeddings. Moreover, explicit attributes, which are the primary attack vector for tables, are optional in graphs, while structural properties are always present, albeit implicitly. In *SyntheGrAnon*, we address these challenges via graph-specific embeddings and attack attributes derived from structural properties. We also explore new opportunities enabled by considering information from all available synthetic graphs when formulating individual node-level attacks.

Privacy Attacks in Graph Learning. In the graph domain, privacy-related attacks and defenses are classifiable based on their scope as: node- [15], edge- [17], or (sub)graph-level [41]. Regardless of the attack level, existing works focus on similar attack types to other modalities, including membership inference, often in a discriminative context [23, 46]. In a membership inference context, a node-level formulation determines whether a specific node in a graph is in the model's training split. Similarly, an edge-based attack determines whether a link exists between two nodes in the training data. The community-level variant most closely mimics the classical scenario, as it determines whether a graph is part of the training dataset. Such a variety of targets within the same data structure increases the attack surface. Attacks on nodes are naturally high-stakes targets for privacy breaches and defenses, as nodes directly represent entities within an interconnected structure. Attacks on subgraphs or entire graphs are also relevant, as they target an entire group of entities at once, even if in a coarser manner. The severity of edge attacks tends to be inherently lower as long as the privacy of the nodes forming the endpoints of a link is maintained. In a generative context, Yang et al. [44] propose a differential privacy formulation at the level of edges. More recently, Wang et al. [40] investigate attacks specifically on diffusion graph models, including inferring membership, recovering graphs similar to those in training data, and inferring global properties of the training data (e.g., edge density distribution). Despite the growing use of synthetic graphs, few studies tackle privacy risks specific to generative graph models.

3 Preliminaries

We first introduce the key elements of graph generation models, followed by the threat model and risk estimation procedure, which are partly based on Giomi et al. [14]. Appendix A contains a summary of the notation used throughout the paper.

3.1 Synthetic Graphs

We begin with the conventional definition of (synthetic) graphs. Let a (synthetic) graph be defined as $\mathbf{G} = (\mathbf{X}^{\text{tab}}, \mathbf{A})$, with tabular node attributes $\mathbf{X}^{\text{tab}} \in \mathbb{R}^{N \times D_t}$ and adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, where N is the number of nodes, and D_t the number of tabular attributes. Thus, $\mathbf{X}^{\text{tab}}[i, a]$ denotes the i^{th} node's a^{th} attribute, while $\mathbf{A}[i, j] = 1$ iff an edge exists between the i^{th} and j^{th} nodes. We specifically consider undirected graphs, in which $\mathbf{A}[i, j] = \mathbf{A}[j, i]\ \forall\ 1 \leq i, j \leq N$. We consider a generalized setting in which nodes can have an arbitrary number of numerical and categorical attributes. However, we note that existing graph synthesizers mainly focus on graph datasets in which nodes have at most a single categorical attribute [26, 34].

3.2 Threat Model

Our attacks operate in a black-box setting, as the attacker has no access to the architecture, weights, gradients, training procedure, or internal states of the trained synthesizer $\mathcal{G}$. The attacker observes only a released synthetic dataset $\mathcal{D}_{\text{syn}} = \{\mathbf{G}_1^{\text{syn}}, \mathbf{G}_2^{\text{syn}}, \dots\}$ sampled from $\mathcal{G}$, and, where applicable, some form of auxiliary information

[1] We use the term graph synthesizer interchangeably with graph generation model.

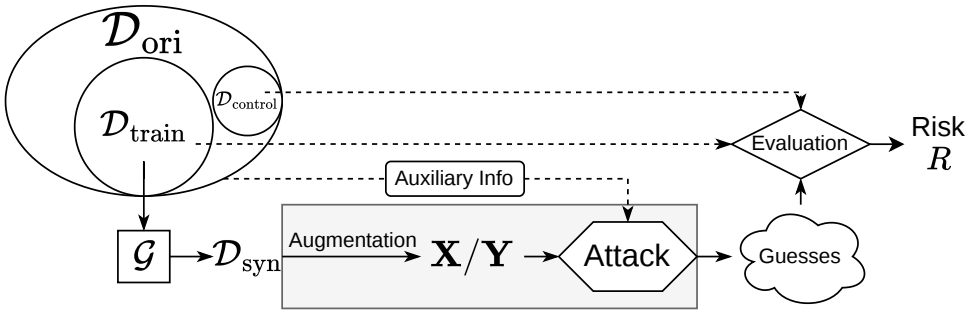


Figure 1: Overview of *SyntheGrAnon*, from executing attacks through $\mathcal{D}_{\text{syn}}$ to computing the risk by validating the guesses on $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{control}}$.

describing part of the real data. We focus on node- and community-level attacks, with edge-level as an extension to the node case.

Employed auxiliary information varies per attack. Singling out attacks assume no such information; they derive candidate predicates from $\mathcal{D}_{\text{syn}}$ and test whether these predicates uniquely identify records in the real data during evaluation. Linkability attacks assume two partial views of the same set of nodes or communities, each containing disjoint subsets of attributes, and try to match records across the two views. Inference attacks assume that some attributes of a target node or community are known and attempt to infer a withheld attribute. Auxiliary information may contain a mix of tabular and structural attributes. A greater amount of auxiliary data provides the attacker with more information, but obtaining large amounts of it may be hard or infeasible in a real-world setting.

3.3 Risk Evaluation Protocol

We follow the risk evaluation protocol presented in Giomi et al. [14], visualized for our specific setting in Figure 1. To assess attack success rate using synthetic data, we leverage the original dataset, which is further partitioned into a training and a control set.

Specifically, we assume the original data set of real graphs is $\mathcal{D}_{\text{ori}} = \{\mathbf{G}_1^{\text{ori}}, \mathbf{G}_2^{\text{ori}}, \dots\}$ drawn from the data distribution of target graphs. From $\mathcal{D}_{\text{ori}}$, two disjoint subsets are drawn $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{control}}$. The $\mathcal{D}_{\text{train}} = \{\mathbf{G}_1^{\text{train}}, \mathbf{G}_2^{\text{train}}, \dots\}$ is used to train the synthesizer $\mathcal{G}$. The $\mathcal{D}_{\text{control}}$ is used only during privacy risk analysis to help distinguish between the effects of merely accurately modeling the true distribution and the risk specific to the samples $\mathcal{D}_{\text{train}}$ used to train the $\mathcal{G}$. We prioritize splits where $|\mathcal{D}_{\text{train}}| > |\mathcal{D}_{\text{control}}|$, leaving more data for training.

The protocol comprises three phases: attack, evaluation, and risk estimation. The **attack phase** takes as input the attack parameterization and $\mathcal{D}_{\text{syn}}$, and outputs a set of N_A guesses $\mathbf{g} = \{g_1, g_2, \dots, g_{N_A}\}$ to later check against the training and control data. The **evaluation phase** takes the guesses $\mathbf{g}$ and checks them to obtain $\mathbf{o} = \{o_1, o_2, \dots, o_{N_A}\}$, where $o_i = 1$ iff the i^{th} guess is correct, and $o_i = 0$ otherwise. $\mathbf{g}$ is checked against $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{control}}$, yielding $\mathbf{o}_{\text{train}}$ and $\mathbf{o}_{\text{control}}$. In the **risk estimation phase**, we compute the train and control statistical risk from which we derive the final privacy risk. The statistical risk r and its corresponding error margin δ_α for a confidence level $\alpha \in [0, 1]$ are given by the Wilson Confidence Interval as:

$$r = \frac{N_S + \frac{z_\alpha^2}{2}}{N_A + z_\alpha^2}, \quad \delta_\alpha = \frac{z_\alpha}{N_A + z_\alpha^2}\sqrt{\frac{N_S(N_A - N_S)}{N_A} + \frac{z_\alpha^2}{4}}$$

where $N_S = \sum_{i=1}^{N_A} o_i$ is the number of correct guesses, and z_α is the inverse of the normal distribution's cumulative distribution function at α. From the train and control statistical risks r_{train} and r_{control}, we get the final privacy risk $R = \frac{r_{\text{train}} - r_{\text{control}}}{1 - r_{\text{control}}}$. A higher $R \in [0, 1]$ corresponds to a greater risk.

4 *SyntheGrAnon*

We propose *SyntheGrAnon* as a framework to empirically evaluate the privacy risk of synthetic graph data in a legislature-aligned manner. We formulate both node-level and community-level variants for attacks targeting singling out, linkability, and inference risk. We also present edge-level attacks as an extension of the node-level counterparts, since they share a similar level of granularity. While our risk scoring follows the tabular evaluation protocol of Giomi et al. [14], directly applying tabular attacks to graph data is insufficient: nodes are not independent records, communities lack a native tabular representation, and connectivity encodes privacy-relevant information rather than (only) explicit attributes.

A key challenge is therefore to construct attackable record representations that preserve graph-specific information while remaining compatible with the risk evaluation protocol. For node-level attacks, we augment node attributes with local structural attributes and pool records across multiple synthetic graphs. For community-level attacks, we encode each graph/community through global structural attributes. A further challenge is keeping embeddings computationally tractable while still expressive enough to support effective attacks across datasets. Finally, in the node-level setting, we consider all available synthetic graphs from the same synthesizer to formulate attacks, and, within singling out, also to filter out weaker attacks, a novel idea not present in previous works.

Objectives of attacks. The high-level goal of *singling out* attacks is to find filters that match exactly one node/graph in the training data, using synthetic data as a proxy. It quantifies the risk of deriving (simple) rules that describe the presence of a node (e.g., a person) or a graph (e.g., a neighborhood) in a model's training set. The *linkability* attacks attempt to match records across two subsets $\mathcal{A}$ and $\mathcal{B}$ that span the same nodes but provide disjoint

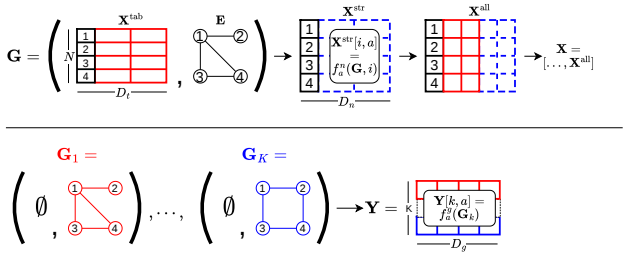


Figure 2: Augmented structural attributes at the node (top) and the community (bottom) level: (i) computing the i^{th} node's a^{th} structural attribute using the property function f_a^n for $\mathbf{X}^{\text{str}}$, and then combining it with $\mathbf{X}^{\text{tab}}$ to form $\mathbf{X}^{\text{all}}$; and (ii) each element in $\mathbf{Y}$ represents the k^{th} graph's a^{th} structural attribute computed using the property function f_a^g.

attributes about the nodes, i.e., for a set of attributes in $\mathcal{A}$, the attack aims to identify the set of attributes in $\mathcal{B}$ belonging to the same node. The known attributes on each side serve as auxiliary data for the attack. In practice, the attack measures the risk of successfully cross-referencing information from two partial datasets (e.g., of two different governmental agencies) with a shared set of nodes or graphs. In *inference* attacks, the aim is to predict a specific node attribute/graph from a partial record of its attributes. It thus follows the setup of a standard machine learning classification (or regression) problem, repurposed to test the effectiveness of using synthetic data to predict a secret attribute from a set of public ones. Figures 3 and 4 visualize the attacks for both covered levels.

4.1 Augmented Structural Attributes

While the attributes of nodes have the same format as records in a tabular database, they are not drawn independently from some target population. Nodes are strongly influenced by their structural connections to other nodes in the same graph (or the lack thereof). For instance, homophily, the effect that nodes with certain similar traits are more likely to be connected, is commonly observed in social graphs [16]. Connectivity patterns in a graph therefore give rise to structural signals that tabular-only attacks cannot capture.

We include well-known properties [3, 29] that capture complementary local and global connectivity patterns while remaining computationally tractable on networks with several thousand nodes. Chosen properties balance being inexpensive enough to compute repeatedly for the target graphs while remaining expressive enough to expose privacy risks that would be invisible from tabular attributes alone. Specifically, we denote such additional structural attributes for (i) individual nodes $\mathbf{X}^{\text{str}} \in \mathbb{R}^{N \times D_n}$ (for an N-node graph with D_n structural attributes) and, (ii) whole graphs $\mathbf{Y} \in \mathbb{R}^{K \times D_g}$ (for K graphs with D_g structural attributes). Figure 2 provides the illustration of augmenting structural attributes at the node and graph levels. At the node level, we concatenate each graph's chosen tabular and structural attributes into $\mathbf{X}^{\text{all}} = [\mathbf{X}^{\text{tab}}, \mathbf{X}^{\text{str}}] \in \mathbb{R}^{N \times (D_t + D_n)}$. Then, we further pool the representations for all K available synthetic graphs into $\mathbf{X} = \{\mathbf{X}_1^{\text{all}}, \mathbf{X}_2^{\text{all}}, \ldots, \mathbf{X}_K^{\text{all}}\}$. We indicate the function that defines the a^{th} structural property for the i^{th} node in graph $\mathbf{G}$ as $f_a^n : (\mathbf{G}, i) \to \mathbb{R}$. Similarly, for structural properties defined in

Algorithm 1 Node-level augmentation

Require: $\mathcal{D}, [f_1^n, f_2^n, \ldots, f_{D_n}^n]$
$\mathbf{X} \leftarrow [\ ]$
for all $\mathbf{G} \in \mathcal{D}$ **do**
  $(\mathbf{X}^{\text{tab}}, \mathbf{A}) \leftarrow \mathbf{G}$
  $\mathbf{X}^{\text{str}} \leftarrow \mathbf{0}^{N \times D_n}$ ▷ N-node graph
  for all $(i, a) \in (\{1, \ldots, N\} \times \{1, \ldots, D_n\})$ **do**
    $\mathbf{X}^{\text{str}}[i, a] \leftarrow f_a^n(\mathbf{G}, i)$
  $\mathbf{X}^{\text{all}} \leftarrow [\mathbf{X}^{\text{tab}} \mid \mathbf{X}^{\text{str}}]$ ▷ Concat tab and str attrs
  $\mathbf{X} \leftarrow \mathbf{X} + [\mathbf{X}^{\text{all}}]$ ▷ Append $\mathbf{X}^{\text{all}}$ to $\mathbf{X}$
return $\mathbf{X}$

Algorithm 2 Community-level augmentation

Require: $\mathcal{D}, [f_1^g, f_2^g, \ldots, f_{D_g}^g]$
$\mathbf{Y} \leftarrow \mathbf{0}^{K \times D_g}$ ▷ $K = |\mathcal{D}|$
for all $(k, a) \in (\{1, \ldots, K\} \times \{1, \ldots, D_g\})$ **do**
  $\mathbf{Y}[k, a] \leftarrow f_a^g(\mathbf{G})$
return $\mathbf{Y}$

the entire graph, we use $f_a^g : \mathbf{G} \to \mathbb{R}$. The details of f^n and f^g are explained below. We note that the augmented structural attributes are extracted for both synthetic and real graphs, where the former is used to conduct attacks and the latter to evaluate attack success.

Node structural properties (f^n): We consider 6 node-level properties. *Average neighbor degree* provides an overview of how well connected each node's immediate neighbors are. The *clustering coefficient* measures the fraction of triangle structures the node is part of relative to the maximum number possible for its degree. *Degree centrality* is a fast measurement for node centrality in a network, denoting the degree divided by the total number of nodes in the network. Similarly, *square clustering* measures the fraction of square structures (i.e., four-node subgraphs where each node has degree two) that the node is part of. *Eigenvector centrality* measures a node's importance by assigning higher scores to nodes connected to other highly central nodes. *Betweenness centrality* measures how often a node lies on the shortest paths between other pairs of nodes, indicating its role as a bridge within the network.

Graph structural properties (f^g): We consider 8 community-level properties. We naturally include the *number of nodes* as a basic statistic. The *average node degree* is the community-level counterpart to the degree centrality (and neighbor degree) measurements for single nodes. *Average node clustering coefficient* averages the clustering coefficient of all nodes into a single value. *Average shortest path length* measures how well-connected the graph is by averaging the path containing the minimum number of edges for interconnecting any pair of nodes. The *diameter* is a related property, denoting the maximum shortest-path distance between any pair of nodes, instead of the average. The *radius* measures the minimum node eccentricity, where a node's eccentricity is the longest shortest path from it to any other node. *Transitivity* is the community-level counterpart to the clustering coefficient, computing the fraction of all triangles present in the graph relative to the maximum possible number. Finally, the *degree assortativity coefficient* is defined as the Pearson correlation coefficient of the node degrees for all

pairs of neighboring nodes in the graph. It is positive for graphs in which nodes tend to connect with other nodes of similar degree, and negative for graphs in which neighboring nodes tend to have significantly different degrees; a higher magnitude indicates a stronger correlation. Finally, *average shortest path length* gives the mean length of shortest paths between every pair of nodes. We include more community-level structural properties than node-level to compensate for the lack of tabular attributes.

Edge structural properties (f^m): We consider 5 edge-level properties. Some of them can be computed more generally for any pair of nodes, even if they are not adjacent. However, we only consider nodes that share an edge in our computation. *Edge betweenness centrality* measures how often an edge lies on the shortest paths between pairs of nodes, indicating its importance as a bridge between different parts of the network. For the *edge clustering coefficient*, we compute the fraction of the smaller endpoint neighborhood that is shared by both endpoints, excluding the endpoints themselves. The *Adamic–Adar index* measures node similarity by weighting shared neighbors more strongly when those neighbors themselves have low degree. The *Jaccard coefficient* measures the similarity between two nodes by comparing the size of their shared neighborhood to the size of their combined neighborhood. The *preferential attachment score* estimates the likelihood of a connection between two nodes based on the product of their degrees, reflecting the tendency of highly connected nodes to attract more links.

Algorithm 1 shows how we compute the node structural attributes $\mathbf{X}^{\text{str}}$ using f^n and construct the final representation of nodes $\mathbf{X}^{\text{all}}$ from a graph $\mathbf{G}$ and its $\mathbf{X}^{\text{tab}}$. Moreover, we group the $\mathbf{X}^{\text{all}}$ outputs of multiple graphs in $\mathbf{X}$. Figure 2 illustrates an example of constructing $\mathbf{X}^{\text{all}}$ and $\mathbf{X}$. Algorithm 2 outlines the augmentation of structural attributes at the graph level, $\mathbf{Y}$, using f^g. For each synthetic graph and a given f^g, we construct a single value in $\mathbf{Y}$. The number of available graphs K thus gives the number of rows in $\mathbf{Y}$. Figure 2 also contains an example of constructing $\mathbf{Y}$. The process for edge-level attacks is partly analogous to the node setting, as we augment any available explicit tabular attributes by computing relevant structural attributes for each edge, and then join the resulting edge representations across multiple graphs. While we use the tabular attributes of both nodes linked by an edge, direct edge-specific tabular attributes may trivially replace or be added to these, if present. Appendix B.1 includes the relevant pseudocode.

4.2 Node-level Attacks

As already mentioned, in the context of working with synthetic graph data that models connectivity among entities, node-level threats pose a particularly high risk, as nodes represent individuals, such as companies or persons, in the graph.

When using the privacy evaluation protocol explained in Section 3.3, we further assume the following for each attack. The $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{control}}$ datasets each consist of a single graph from the same population, but with different individuals (i.e., nodes). As such, we have $|\mathcal{D}_{\text{train}}| = |\mathcal{D}_{\text{control}}| = 1$. In graph learning and generation works [7, 26], having a single, large-enough graph serving as the training dataset for a generator $\mathcal{G}$ is a common setting.

4.2.1 Singling out. Since we represent each node as a row in a table, with tabular and structural attributes corresponding to columns,

Algorithm 3 Node-level Singling Out Predicates Generation

```
function NodeSinglingOut(D_syn, N_A, N_V, L)
    X ← DatasetToNodeRecords(D_syn, ...)                ▷ Algorithm 1
    if N_V = 1 then                                     ▷ Univariate attack
        preds ← [ ]
        for all (X_i^all, a) ∈ (X × {1, ..., D_t + D_n}) do
            if X_i^all[:, a] is categorical then
                cands ← UniqueValPreds(X_i^all[:, a])
            else                                        ▷ X_i^all[:, a] is numerical
                cands ← MinMaxValPreds(X_i^all[:, a])
            for all P ∈ cands do                        ▷ Check candidate preds
                if NrSingledOutGraphs(X, P) ≥ L then
                    preds ← preds + [P]
        preds ← N_A samples from preds
    else                                                ▷ Multivariate attack
        while |preds| < N_A do
            X_i^all, P ← sample from X, ⊤
            for all a ∈ (N_V samples from {1, ..., D_t + D_n}) do
                if X_i^all[:, a] is categorical then
                    P ← P ∧ RandValVar(X_i^all[:, a])
                else
                    P ← P ∧ MedianValVar(X_i^all[:, a])
            if NrSingledOutGraphs(X, P) ≥ L then
                preds ← preds + [P]
    return preds
```

we follow previous work in using filters based on predicates [14]. Specifically, a predicate $P = \bigwedge_{i=1}^{N_V} V_i$ is a conjunction of variable assignments V representing (in)equality checks over $N_V \geq 1$ different columns/attributes. In the example of a social network graph, a practical predicate could filter for nodes (persons) with *degree greater than 10 and married*, where the degree effectively denotes the number of friends a person has. Shown in Figure 3, two predicates are used to single out a real node in the training data set: a tabular attribute on the age greater than 18 and a structural attribute of node degree equal to 2.

Algorithm 3 describes the attack procedure. For each guess and its corresponding predicate P, we first randomly select one of the graphs in $\mathcal{D}_{\text{syn}}$ to serve as the base for P. We then construct a predicate based on the selected graph, with the concrete construction strategy depending on the number of variables per predicate selected for the attack. For categorical univariate predicates, we cycle through all categorical attributes and create variables testing equality with values (categories) that appear only once across the selected graph's nodes. The function *UniqueValPreds* yields all such predicates for a given graph and attribute. Meanwhile, for numerical counterparts, we choose variables checking for values lower than or equal to the minimum in the synthetic dataset and greater than or equal to the maximum. In the pseudocode, *MinMaxValPreds* creates the min and max predicates for some attribute in a graph. To create a multivariate predicate, we randomly sample the desired number of columns, create a variable for each, and then chain them together. Sampling is required as the total number of possible attribute combinations grows from $D_t + D_n$ (trivially) in

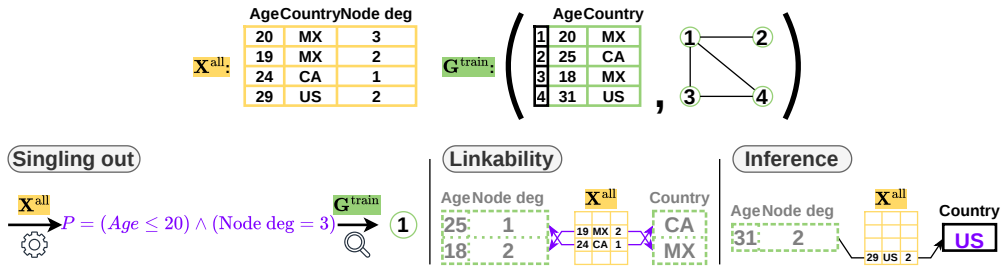


Figure 3: Node-level attacks based on synthetic graphs and evaluated against real graphs in the training data. The singling out attack tries to identify node 3 using two predicates. The linkability attack tries to link two records using synthetic graphs. The inference attack tries to infer the country based on the age and node degree using synthetic graphs.

the univariate case to $\binom{D_t+D_n}{N_V}$ for predicates with N_V variables. Since multivariate predicates chain multiple conditions together, individual variables should be less specific than in the univariate case, as otherwise the full predicate risks being overly specific (i.e., matching zero nodes instead of precisely one). Thus, for categorical columns, we select, via *RandValVar*, distinct categories (regardless of whether they appear exactly once in the chosen graph). For continuous columns, we employ *MedianValVar* to create variables that filter for values either below or above the median.

After constructing a predicate, we check not only whether it singles out a node in the synthetic graph used to construct it, but also in other available synthetic graphs. In our implementation, we denote the number of graphs in $\mathbf{X}$ singled-out by P with *NrSingledOutGraphs*. We only add a predicate to the list of guesses when it successfully singles out a minimum number of synthetic graphs based on a selectable threshold $L \geq 1$. Doing so allows us to leverage all available synthetic graphs, which are effectively variants of the population modeled during training, to increase confidence in a predicate's suitability when later evaluated on the original data. Increasing the minimum number of singled-out synthetic graphs initially leads to attacks that are more likely to single out the original data, but requiring too many synthetic graphs to fit overtrains on artifacts only present in the synthetic data. Moreover, the total number of possible guesses (creatable within a given search time budget) decreases as L increases, since many are discarded due to insufficient matches. Maximizing attack success rate while keeping the total number of attacks high requires balancing the parameter's value. The approach of using multiple instances of synthetic data is a strong contrast to Anonymeter, which only relies on one table at a time. It addresses a graph-specific challenge absent from the tabular setting: the synthesizer may expose several synthetic views of the same modeled population, and a predicate that appears in only one sample may reflect sampling noise or a model artifact rather than a stable property of the training graph.

Node-level Singling Out Figure 3 Example: In the synthetic $\mathbf{X}^{\text{all}}$, we find a single node with an age of at most 20 and a degree of 3, giving the multivariate predicate (age $\leq$ 20) $\wedge$ (degree = 3). Applying the corresponding multivariate predicate on $\mathbf{G}^{\text{train}}$, we successfully single out the node with identifier 1, resulting in a successful guess.

4.2.2 Linkability. In the linkability attack we have as auxiliary data: $\mathcal{A}, \mathcal{B} \subset \{1, 2, \ldots, D_t + D_n\}$, two disjoint ($\mathcal{A} \cap \mathcal{B} = \emptyset$) subsets of all possible (tabular and structural) attribute identifiers, and $\mathbf{X}[\cdot, \mathcal{A}], \mathbf{X}[\cdot, \mathcal{B}]$ two sets of partial node records from dataset $\mathbf{X}$, such that $\mathbf{X}[i, \mathcal{A}]$ and $\mathbf{X}[i, \mathcal{B}]$ are both parts of the same $\mathbf{X}_i$. Each record on one side is assumed to have a matching one on the opposite side. The goal is to match records between $\mathbf{X}[i, \mathcal{A}]$ and its counterpart $\mathbf{X}[i, \mathcal{B}]$ based on a common link from $\mathcal{D}_{\text{syn}}$. A successful attack involves correctly matching a row in $\mathbf{X}[\cdot, \mathcal{A}]$ with its counterpart in $\mathbf{X}[\cdot, \mathcal{B}]$. For each $i \in \{1, 2, \ldots, N_A\}$, the attack finds $\mathbf{I}_i^{\mathcal{A}}$ and $\mathbf{I}_i^{\mathcal{B}}$, indices of k records from $\mathcal{D}_{\text{syn}}$ closest to $\mathbf{X}[:, \mathcal{A}]$ and $\mathbf{X}[:, \mathcal{B}]$, respectively. In line with existing work on tabular data, we take the Gower coefficient as a distance measure supporting mixed-type columns [14]. k is a tunable attack parameter. If $\mathbf{I}_i^{\mathcal{A}} \cap \mathbf{I}_i^{\mathcal{B}} \neq \emptyset$, the guess is considered successful. Increasing k allows a more lenient definition of success, while still yielding a meaningful risk estimate, as long as $k \ll N_A$.

In addition to arbitrary (random) splits for $\mathcal{A}$ and $\mathcal{B}$, we specifically consider ones specific to the tabular and structural nature of node records. Two such cases are those when both sets contain strictly tabular or structural attributes, ignoring the other group entirely. Considering only tabular attributes emulates the tabular data setting, where records come from the same overall distribution but otherwise get treated independently. Considering only structural attributes emulates access only to graph connectivity patterns, not to any additional tabular attributes. Finally, there is the case in which $\mathcal{A}$ contains only tabular attributes, while $\mathcal{B}$ contains only structural information (or vice versa). It implicitly tests the strength of the correlation between the two types of attributes.

Node-level Linkability Figure 3 Example: We have two sets of partial information for the nodes with identifiers 2 and 3 from $\mathbf{G}^{\text{train}}$. One set ($\mathcal{A}$) contains age and node degree data, while the other ($\mathcal{B}$) contains only country data. For the record in $\mathcal{A}$ with age 25 and node degree 1 and the record in $\mathcal{B}$ with country "CA" we find the shared closest match from $\mathbf{X}^{\text{all}}$ to be a synthetic node with age

24, degree 1, and country "CA", and successfully link them together. A similar process happens for the other pair of records in $\mathcal{A}$ and $\mathcal{B}$.

4.2.3 Inference. The attack is given $\mathbf{X}[:, \mathcal{A}]$, a subset of N_A partial node records from $\mathbf{X}$ as auxiliary information, where $\mathcal{A} \subset \{1, 2, \ldots, D_t + D_n\}$ is a subset of attribute identifiers obeying $1 \leq |\mathcal{A}| < D_t + D_n$. It has to predict a secret attribute $s \in \{1, 2, \ldots, D_t + D_n\} \setminus \mathcal{A}$. To predict the secret attribute of $\mathbf{X}[i, \mathcal{A}]$, the attack looks for the nearest neighbor from $\mathbf{X}_{\text{syn}}$, taking its value as the guess for s. As with the linkability attack, we measure the distance between records via the Gower coefficient. For categorical attributes, a guess is correct if the selected category matches the one in the real data, whereas for numerical values, we allow a small margin of error.

In our implementation, we focus specifically on secret values drawn from the set of tabular attributes, while the auxiliary information can contain structural attributes as well. As in purely table-based data, specific tabular attributes of nodes can disclose particularly sensitive information (e.g., sexual or religious identity). In contrast, structural attributes, despite their usefulness in augmenting attacks, are often less directly interpretable and less damaging, as they naturally aggregate information about a node's connectivity without generally disclosing or allowing one to infer the exact links of a node.

Node-level Inference Figure 3 Example: We have to infer the country (secret column) for a partial record from $\mathbf{G}^{\text{train}}$ with age 31 and node degree 2. We find the closest record from the synthetic ones in $\mathbf{X}^{\text{all}}$ to be the one with age 29 and the same degree of two. From the country value of our found synthetic record, we correctly infer that the partial record's country should be "US".

4.3 Community-level Attacks

Our community-level attacks rely on representing each graph as a row in a table, with structural attributes as columns. Since such a scenario aligns more closely with synthetic data generation approaches from other modalities, where $|\mathcal{D}_{\text{train}}| > 1$ and all graphs belong to a shared family (e.g., social network friendship groups). We visually showcase all community-level attacks in Figure 4.

For the singling out attack, we build predicates to single out graphs rather than nodes. Note that, unlike nodes, which can be grouped based on the graph to which they belong, graphs cannot be grouped based on some higher-order entity, other than the dataset as a whole, to which all graphs belong. As such, there is no equivalent to the minimum number of singled-out synthetic graphs from the attack on nodes. We search for predicates that match exactly one of the graphs in $\mathcal{D}_{\text{syn}}$, in the hopes that they also uniquely describe individual networks included in $\mathcal{D}_{\text{train}}$. Since we only have structural attributes, the linkability attack at the community-level is most closely related to the variant of its node-level sibling in which both $\mathcal{A}$ and $\mathcal{B}$ contain strictly structural node attributes. However, in the community-level case, the goal shifts from cross-referencing members within one population using local connectivity patterns to cross-referencing entire populations via global connectivity patterns. For the community-level inference attack, unlike in the node-level case, we exclusively use structural attributes for both the auxiliary data and the secret. The task effectively entails deriving new structural attributes of a graph based on a given partial set. However, we note that, for domains/tasks in which whole graphs also contain one or more attributes, such data can again be included in the auxiliary data or set as the secret.

Figure 4 Examples:
Singling Out – In the synthetic $\mathbf{Y}$, there is a single graph with 3 nodes, so we construct the corresponding predicate. Upon testing it on the train $\mathbf{Y}$, we find there is also exactly one entry denoting a graph with three nodes, corresponding to $\mathbf{G}_1$, so our attack is successful.
Linkability – We have partial information for $\mathbf{G}_2$ and $\mathbf{G}_3$ from the real $\mathbf{Y}$, split into two disjoint sets. On one side, we have information on the number of nodes and the average degree, whereas on the other, we have the diameter. One of our entries in the synthetic $\mathbf{Y}$, describing a graph output by the generator with 4 nodes, average node degree of 2, and diameter 2, exactly matches the first partial record from either set, thus naturally being the closest neighbor of both. As such, we can successfully link the two partial training data records using our synthetic data. We then link the remaining two records on either side using available information about another synthetic graph.
Inference – Given a real graph from the target distribution with 4 nodes and an average rounded degree of 1, we must infer its diameter. The closest synthetic graph in $\mathbf{Y}$ has diameter 2, which we take as our guess.

4.4 Edge-level Attacks

In the edge-level case, after constructing $\mathbf{E}^{\text{all}}$, the counterpart to the node-level $\mathbf{X}^{\text{all}}$, attacks proceed analogously to the node setting. As such, the formulation also allows increasing the minimum number of singled-out synthetic graphs, and, in general, attacks benefit from access to multiple synthetic graphs. Note that, although we only consider true edges in our representation instead of every possible node pair in the given graphs, the search space for edge-level attacks is considerably larger than when dealing with nodes. Leveraging the tabular attributes of two nodes per edge, along with structural attributes, grows the search space even further.

5 Evaluation

We analyze the effectiveness of our six attacks across three datasets and five synthesizers, varying the attack parameters. The rest of the section focuses on node- and community-level attacks. Results and discussion for edge-level scenarios are in Appendix B.2.

5.1 Evaluation Setup

Below, we include details on the setup of our evaluation, including datasets, synthesizers, and experiment configuration.

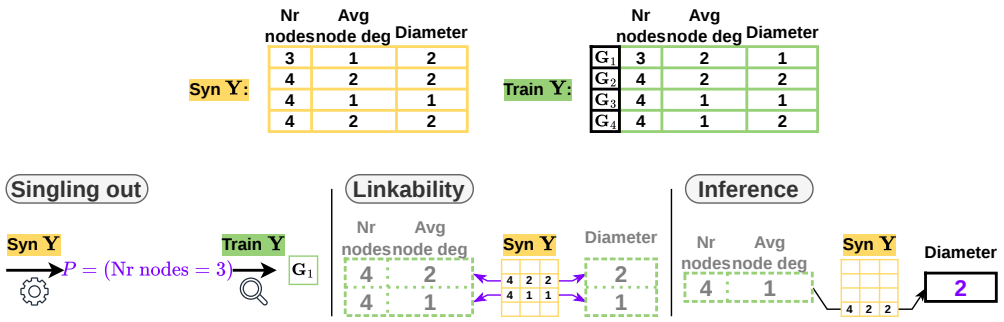


Figure 4: Community-level attacks using augmented structural properties of synthetic graphs. The singling out attack aims to identify the unique graph using the node predicate. The likability attack tries to link the two graphs and their attributes using synthetic graphs. The inference attack aims to infer a graph's diameter from its number of nodes and the average node degree.

Dataset	Node-level (one attr. graph)			Community-level (many unattr. graphs)		Domain
	#Train nodes	#Control nodes	Node attributes	#Train graphs	#Control graphs	
Ego-Facebook	1046	171	28 labels	8	2	Social
Ego-Twitter	251	250	22 labels	80	20	Social
Elliptic	7880	7140	3 classes	39	10	Finance

Table 1: Dataset overview. Node-level setting: For *Ego-Twitter* and *Ego-Facebook*, we select the two largest disjoint ego networks as the training and control graphs, and consider their common node attributes. For *Elliptic*, we pick the two largest connected components, with node class as the only attribute. Community-level setting: For each, we choose the largest up to 100 ego-nets (or, in the case of *Elliptic*, connected components) without node attributes, then apply a random 80/20% train/control split.

5.1.1 Datasets. Table 1 outlines the graph datasets our evaluation is based on. *Ego-Facebook* and *Ego-Twitter* [25] are neighborhood networks of Facebook and Twitter users, respectively. For Facebook, there are 10 ego networks, whereas for Twitter, we selected the 100 largest from 973 total.

For node-level attacks, the largest ego network (in terms of number of nodes) was chosen as the training graph, which resulted in a graph of more than 1,000 nodes for *Ego-Facebook* and slightly more than 250 for *Ego-Twitter*, while the control graph is chosen as the next-largest ego network disjoint to the training graph, which resulted in a network of 171 nodes for *Ego-Facebook* and 250 nodes for *Ego-Twitter*. In this manner, we also obtained two distinct relations between training and control data: For *Ego-Facebook*, the two datasets differ substantially in size, as is common for the relation between training and control data, while the two graphs for *Ego-Twitter* have almost the same number of nodes. Both datasets have binary attributes, though the attributes vary between ego networks, with many attributes appearing in multiple ego networks. Enabling attacks that can be executed on both the training and the control graphs means we are restricted to the attributes they share, which are 28 for *Ego-Facebook* and 22 for *Ego-Twitter*. For community-level attacks on the two social graphs, we randomly split the ego networks into training and control sets, with 20% of the graphs serving as the control data.

Our third dataset, *Elliptic*, is a financial dataset with nodes representing Bitcoin transactions and edges indicating that outputs of one transaction correspond to inputs of another. The dataset consists of a single network with multiple disconnected components.

	Singling out	Linkability	Inference
Node-level	$5(2D_t + 3D_n)$	$5(3 + D_n)$	$D_t(3 + D_n)$
Community-level	$5D_g$	$5(D_g - 1)$	$D_g(D_g - 1)$

Table 2: Total number of experiments per attack. D_t is the number of tabular node attributes, while D_n and D_g are the structural node- and community-level counterparts.

For node-level attacks, we chose the largest connected component, with almost 8,000 nodes, as the training and the second-largest connected component, with more than 7,000 nodes, as the control graph. Nodes are classified into one of three classes (licit, illicit, or undefined) so the graph has only one node attribute with three possible values. For community-level attacks on *Elliptic*, we treated each connected component as a graph and split them into training and control data using an approximate 80/20 split of the 49 graphs.

Note that *Ego-Facebook* is data-scarce, with only 10 graphs in total, which is primarily an issue for community-level attacks, as we have few samples. Furthermore, *Elliptic* is attribute-poor, and the graphs chosen for the node-level attacks are quite large, which is likely to make it challenging to identify individual patterns, especially in comparison to the other two datasets, which have fewer nodes and more potentially identifying information per node. Overall, these structural differences make cross-data comparisons challenging: for a fixed model, risk on a smaller, attribute-rich network may differ from that on a larger, attribute-poor network.

5.1.2 Graph Synthesizers. To generate synthetic graphs for our evaluation we use three diffusion models (EDGE [7], GGSD [26], and GruM [19]), alongside two further options based on autoregressive (GRAN [22]) and GAN (SPECTRE [24]) modeling paradigms. We train each synthesizer separately for each attack level and dataset pair. We base our training configuration for each run on the most similar config (in terms of the number of graphs in the dataset and the range of nodes within individual graphs) among those used in the original authors' evaluation of the model. We only enable attribute generation when training the node-level versions of the datasets. However, we note that the five synthesizers differ in their support for generating node attributes. While EDGE's base formulation only covers structure generation, we leverage support in the model's codebase for sampling categorical node attributes alongside edges. For GGSD, we modify the codebase to allow the generation of multiple discrete node attributes. Note that we omit GGSD from the unattributed comparisons because we could not reliably sample valid (connected) graphs for two of the three datasets we tested. For SPECTRE, we condition on real spectra to maximize output quality and forgo testing the Elliptic dataset variants due to overly high memory requirements. GRAN has no native support for node attribute generation that we can modify to fit our workloads, so we consider only the community-level (unattributed) versions of our datasets, and further drop Ego-Facebook as we could not sample valid graphs from any examined training checkpoints.

We measure utility using a downstream link-prediction task. Specifically, we train a small graph neural network on synthetic data and report the ROC AUC score on a test set of real graphs. We concatenate the training and control datasets, then perform a validation-test split on the resulting dataset and use the validation set to determine when to end training the link predictor, based on a patience parameter. For unattributed graphs, we initialize node attributes to one-hot-encoded node degrees. We report the test-set score for the checkpoint with the highest validation ROC AUC. As another utility metric, we measure maximum mean discrepancy (MMD) for various structural properties in Appendix C.

5.1.3 Parameter Selection. The three attacks (singling out, linkability, and inference) can be instantiated with different parametrization, such as the choice of the two disjoint attribute sets of a linkability attack. We consider an experiment to be a set of attacks using the same parametrization. For node-level linkability and inference attacks, the number of attacks in the experiment is equal to the total number of nodes in the training and control graphs, i.e., each node is attacked once. Analogously, for community-level linkability and inference attacks, there is one attack for each graph. In contrast, for singling out attacks, the number of potential attacks depends on the number of predicates, which increases exponentially with the number of variables. Hence, to keep the evaluation feasible, we always considered 500 predicates per experiment.

The number of experiments considered depended on the type of attack. For a node-level singling out attack, we varied the minimum number L of synthetic graphs that have to contain exactly one node matching a predicate between 1 and 5. For each value of L, we considered four types of predicates: i) tabular attributes only, where we varied the number of variables between 1 and D_t, resulting in D_t experiments, ii) structural attributes only, where we varied the number of variables between 1 and D_n, resulting in D_n experiments, iii) all attributes, resulting in a total of $D_t + D_n$ experiments, iv) all tabular attributes and one structural attribute varying the structural attribute chosen, so D_n experiments, which is an ablation study to show the impact of individual structural attributes. For community-level singling out, we only have structural attributes to consider. As for the node-level attacks, we varied L from 1 to 5 and the number of variables in a predicate from 1 to D_g, the number of structural attributes at the graph level.

For node-level linkability, we varied the number k of nearest neighbors considered between 1 and 5. For each value of k, we ran three experiments, splitting into equally sized random subsets $\mathcal{A}$ and $\mathcal{B}$ either: i) only the tabular attributes, ii) only the structural attributes, or iii) all attributes. To study the impact of different structural attributes, we also split all tabular attributes and one of the D_n structural attributes at a time. For community-level attacks, we also vary k between 1 and 5. We enumerated the graph structural attributes, then we varied i from 1 to $D_g - 1$, assigning the first i attributes to set $\mathcal{A}$ and the others to set $\mathcal{B}$.

For node-level inference, we assigned each tabular attribute as the secret. For each secret, we considered the following four possible sets of auxiliary information: i) all tabular attributes except the secret, ii) all structural attributes, iii) all attributes except the secret, and iv) all tabular attributes and one structural attribute, varying the latter to try all D_n such attributes. For community-level inference, we vary the secret attribute within the set of D_g structural attributes. Then we enumerate the remaining attributes and include the first i in the auxiliary data for $i = 1, \ldots, D_g - 1$.

As detailed above, we do not consider every possible attack parametrization due to computational intractability. For instance, the number of ways the attribute set can be divided into subsets $\mathcal{A}$ and $\mathcal{B}$ scales exponentially in the number of attributes. Hence, we consider random splits with specific properties, e.g., only tabular or structural attributes, to limit the number of experiments needed. We still cover a wide range of attacks, with Table 2 showing the number of experiments per attack type, and each experiment comprising hundreds to thousands of attack instances.

When studying the risk of our six attacks, we report both the mean and the maximum risks across all experiments. Here, the mean represents an attacker who just tries a random variant of the attack, whereas the max represents an attacker who has learned a suitable attack parametrization. The performance of a realistic attack is likely somewhere between the two cases. As the attacker does not know the original data, they cannot fine-tune their attack to that exact data; however, they can experiment with attack parameterizations on other data and identify promising settings.

5.2 Node-level Attacks

We first present the main results on the mean and maximum attack risks in the attributed, node-focused scenarios, followed by a study of the individual attack components.

5.2.1 Overall Results. Node-level privacy risks are severe across datasets, but vary substantially by attack type, model, and dataset scale. Table 3 reports the mean and maximum risk of node-level attacks for each attack type alongside model utility. GruM achieves the highest utility on *Ego-Facebook* (87.1%), while on *Ego-Twitter*,

Dataset	Model	Attack (Mean + SD) ↑			Attack (Max + CI) ↑			Utility ↑
		Singling out	Linkability	Inference	Singling out	Linkability	Inference	
Ego-Facebook	EDGE	0.46 ± 0.41	0.00 ± 0.00	0.58 ± 0.31	1.00 ± 0.00	0.01 ± 0.01	0.98 ± 0.02	66.2%
	GGSD	0.05 ± 0.04	0.00 ± 0.00	0.36 ± 0.34	0.22 ± 0.05	0.00 ± 0.01	0.93 ± 0.03	60.7%
	GruM	0.14 ± 0.09	0.00 ± 0.00	0.42 ± 0.33	0.35 ± 0.04	0.01 ± 0.01	0.95 ± 0.02	87.1%
	SPECTRE	0.04 ± 0.04	0.00 ± 0.00	0.22 ± 0.27	0.25 ± 0.04	0.02 ± 0.02	0.83 ± 0.16	55.0%
Ego-Twitter	EDGE	0.42 ± 0.42	0.02 ± 0.05	0.28 ± 0.27	1.00 ± 0.00	0.24 ± 0.06	0.89 ± 0.10	77.9%
	GGSD	0.02 ± 0.03	0.00 ± 0.00	0.11 ± 0.23	0.18 ± 0.32	0.02 ± 0.04	0.83 ± 0.16	62.6%
	GruM	0.18 ± 0.19	0.02 ± 0.03	0.20 ± 0.25	0.88 ± 0.10	0.11 ± 0.04	0.87 ± 0.13	77.1%
	SPECTRE	0.07 ± 0.05	0.00 ± 0.01	0.17 ± 0.23	0.27 ± 0.07	0.05 ± 0.03	0.80 ± 0.17	69.8%
Elliptic	EDGE	0.09 ± 0.17	0.00 ± 0.00	0.05 ± 0.06	0.72 ± 0.16	0.00 ± 0.00	0.18 ± 0.04	69.9%
	GGSD	0.05 ± 0.08	0.00 ± 0.00	0.11 ± 0.07	0.38 ± 0.04	0.00 ± 0.00	0.27 ± 0.03	66.0%

Table 3: Mean and max risk for the node-level setting.

EDGE has a marginal advantage over GruM (77.9% vs. 77.1%). SPECTRE has lower utility on both ego datasets (55.0% and 69.8%), where it is most comparable to GGSD, the lowest-utility diffusion-based model (60.7% and 62.6%). On *Elliptic*, the two evaluated models, EDGE and GGSD, achieve 69.9% and 66.0% utility, respectively.

For singling out, EDGE represents the most vulnerable model, reaching a maximum risk of 1.00 on both ego datasets and 0.72 on *Elliptic*. GruM also shows high singling out risk on *Ego-Twitter* (0.88), but is lower on *Ego-Facebook* (0.35). SPECTRE and GGSD are substantially less vulnerable on the ego datasets, with maximum risks of 0.27 and 0.22, respectively. On *Elliptic*, GGSD reaches 0.38, higher than on the ego datasets, suggesting that its weaker support for many node attributes suppresses risk on the attribute-rich ego graphs more than on the attribute-poor financial graph.

Linkability risk is generally lower across node-level results. On *Ego-Facebook*, all models have maximum linkability risk at or below 0.02; on *Ego-Twitter*, EDGE is highest at 0.24, followed by GruM at 0.11, SPECTRE at 0.05, and GGSD at 0.02. On *Elliptic*, both evaluated models yield zero linkability risk. Thus, in contrast to singling out and inference, node-level linkability is less of a threat in the proposed experiments.

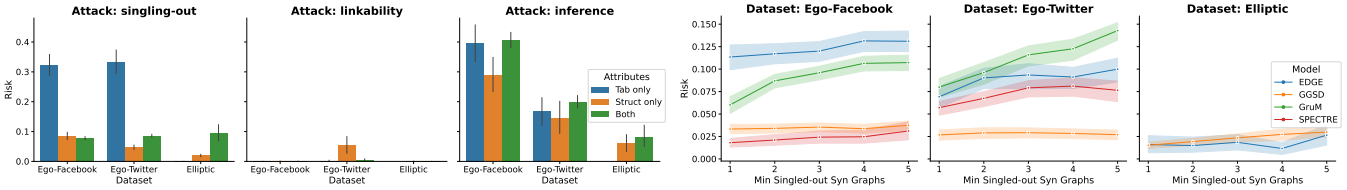
Inference risk is severe on the ego datasets. Maximum inference risks range from 0.83 to 0.98 on *Ego-Facebook* and from 0.80 to 0.89 on *Ego-Twitter*. EDGE attains the highest maximum inference risk across both ego datasets, while SPECTRE is consistently the lowest, though still non-trivial. On *Elliptic*, inference risk is much lower, with maxima of 0.18 for EDGE and 0.27 for GGSD.

Overall, utility and privacy risk are related, albeit not monotonically coupled. Lower-utility models such as SPECTRE generally exhibit lower node-level risk, but EDGE has higher singling out and inference risks than GruM on *Ego-Facebook*, despite substantially lower utility. Our results suggest that the utility–privacy relationship depends on the attack type and on which aspects of the graph distribution a synthesizer preserves.

5.2.2 Effect of tabular versus structural node attributes. The most effective attribute type depends on the attack, as shown in Figure 5a. For singling out, tabular attributes dominate both ego datasets, suggesting that native node labels provide more discriminative predicates than structural attributes alone. On *Elliptic*, where the tabular space contains only three classes, all variants remain low, with the combined setting yielding the highest risk by a modest margin. For inference, tabular or combined attributes are generally strongest on the ego datasets, while structural-only information remains useful but weaker. For linkability, the overall risks are small, but structural-only attributes provide the largest non-zero signal, especially on *Ego-Twitter*. Such behavior is consistent with the fact that certain subsets of structural variables are often (but not always) mutually correlated and that graph generators are explicitly optimized to preserve structural statistics. On *Elliptic*, the large number of nodes and limited attribute space leave both tabular and structural views with limited discriminative power.

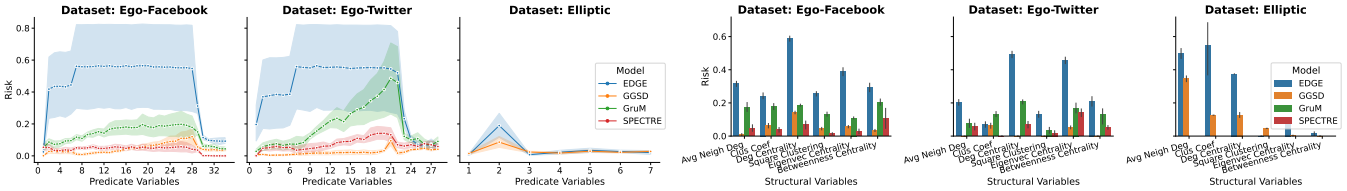
5.2.3 Effect of minimum number of matching synthetic graphs on singling out. Requiring a singling out predicate to match multiple synthetic graphs often strengthens the attack. However, the effect is moderate in the structural-only setting shown in Figure 5b. On the ego datasets, the privacy risk of EDGE and GruM generally increases with the number of required matching synthetic graphs, while the risk for SPECTRE grows more mildly, and GGSD is an outlier, remaining nearly flat. The consistently low risk for GGSD may indicate that maximal risk is relatively low and is already reachable with the standard version of the attack. On *Elliptic*, the risk is low and changes little across thresholds for the evaluated models. Overall, access to multiple synthetic graphs remains an exploitable advantage, but it is most useful for small, attribute-rich datasets and models that reproduce structural patterns well.

5.2.4 Effect of number of predicate variables on singling out. As displayed in Figure 5c, singling out risk rises quickly with predicate complexity on the ego datasets, but can collapse once predicates become over-specified. EDGE reaches the highest plateau on both ego datasets, while GruM grows more gradually and peaks more strongly on *Ego-Twitter* than on *Ego-Facebook*. SPECTRE remains below EDGE and usually below GruM, while GGSD stays low across variable counts. On *Elliptic*, among the tested EDGE and GGSD, risk is much lower overall, aside from a spike for bivariate predicates, reflecting the limited discriminative power of a three-class node attribute space at large scale.



(a) Effect of tabular and structural node attributes in isolation, as well as jointly, across all attack types.

(b) Effect of increasing the minimum number of synthetic graphs singled out by each predicate.



(c) Effect of increasing the number of predicate variables for singling out attacks.

(d) Effect of allowing only one structural property at a time in singling out predicates.

Figure 5: Ablations for node-level attacks.

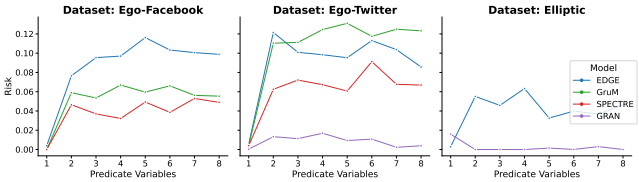


Figure 6: Effect of increasing the number of predicate variables for community-level singling out attacks.

5.2.5 Effect of individual structural variable choice on singling out. Figure 5d examines how individual structural variable choices affect singling out risk. The most informative structural variable is dataset-dependent: degree-based properties dominate on the ego datasets, while clustering-related properties are more informative on *Elliptic*. On the ego datasets, degree and degree centrality produce the highest risks for EDGE, with GruM and SPECTRE generally yielding lower but still visible risks, and GGSD remaining low. On *Elliptic*, EDGE achieves its highest single-variable risk on clustering-related structural properties, although this remains below the maximum risk reached when all predicate variables are available. The results suggest that an attacker benefits from structural variables tailored to the target graph family rather than a fixed choice across domains.

5.3 Community-level Attacks

For the unattributed, community-focused scenarios, we again report the mean and maximum attack risks across the tested configurations, followed by a brief ablation discussion.

5.3.1 Overall Results. Per Table 4, GruM achieves the highest utility on *Ego-Facebook* (90.2%), then EDGE (85.2%) and SPECTRE (83.1%). On *Ego-Twitter*, GRAN has the highest utility (81.2%), followed closely by GruM (80.0%), SPECTRE (78.6%), and EDGE (78.2%). On *Elliptic*, EDGE has higher utility than GRAN (92.2% vs. 81.3%).

Singling out is consistently low across the board, with maximum risks at or below 0.13, supporting the conclusion that, for the considered structural properties, constructing predicates that isolate an entire graph is harder than isolating individual nodes. GRAN's risk is especially low for singling out, reaching a maximum value of only 0.02 across both evaluated datasets.

Linkability and inference are more substantial but strongly model- and dataset-dependent. On *Ego-Facebook*, GruM has the highest linkability risk (0.76), while SPECTRE is close behind (0.68) and also has the highest inference risk (0.68); EDGE is lower for both attacks (0.38 and 0.26). However, as the aforementioned dataset has only eight training and two control graphs, values should be interpreted with caution. On *Ego-Twitter*, GRAN has the highest linkability risk (0.57), while GruM, SPECTRE, and GRAN all reach the same maximum inference risk of 0.72; EDGE is lower for linkability (0.17) and inference (0.07). On *Elliptic*, inference risk is minimal for both evaluated models, but GRAN shows non-negligible linkability risk (0.29), whereas EDGE remains near zero (0.03).

Overall, the community-level results again show that higher utility does not imply uniformly higher privacy risk. For example, SPECTRE has lower utility than EDGE on *Ego-Facebook* but higher linkability and inference risk. In contrast, EDGE has the highest utility on *Elliptic* but lower risk than GRAN.

5.3.2 Effect of number of predicate variables on community-level singling out. All variants of community-level singling out have risk below 0.15 across all datasets, as shown in Figure 6. On *Ego-Facebook*, EDGE reaches the highest risk, while GruM and SPECTRE remain lower. On *Ego-Twitter*, EDGE and GruM rise quickly to roughly 0.10–0.13, SPECTRE remains somewhat lower, and GRAN stays close to zero. On *Elliptic*, EDGE has a modest non-zero risk with no clear monotonic trend, while GRAN remains near zero. All models except GRAN show sharp jumps in risk when moving from one to two variables in all datasets, after which risk tends to fluctuate without any consistent upward trend, indicating that adding more variables beyond this point does not reliably increase

Dataset	Model	Attack (Mean + SD) ↑			Attack (Max + CI) ↑			Utility ↑
		Singling out	Linkability	Inference	Singling out	Linkability	Inference	
Ego-Facebook	EDGE	0.09 ± 0.04	0.09 ± 0.14	0.02 ± 0.06	0.12 ± 0.03	0.38 ± 0.51	0.26 ± 0.56	85.2%
	GruM	0.05 ± 0.02	0.21 ± 0.21	0.19 ± 0.26	0.07 ± 0.02	0.76 ± 0.27	0.63 ± 0.38	90.2%
	SPECTRE	0.04 ± 0.02	0.29 ± 0.20	0.12 ± 0.23	0.05 ± 0.04	0.68 ± 0.42	0.68 ± 0.42	83.1%
Ego-Twitter	EDGE	0.09 ± 0.04	0.04 ± 0.05	0.00 ± 0.01	0.12 ± 0.03	0.17 ± 0.20	0.07 ± 0.11	78.2%
	GruM	0.11 ± 0.04	0.04 ± 0.05	0.13 ± 0.27	0.13 ± 0.04	0.25 ± 0.31	0.72 ± 0.40	80.0%
	SPECTRE	0.06 ± 0.03	0.00 ± 0.01	0.06 ± 0.18	0.09 ± 0.04	0.08 ± 0.39	0.72 ± 0.40	78.6%
	GRAN	0.01 ± 0.01	0.04 ± 0.11	0.02 ± 0.10	0.02 ± 0.01	0.57 ± 0.28	0.72 ± 0.40	81.2%
Elliptic	EDGE	0.04 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.06 ± 0.03	0.03 ± 0.29	0.00 ± 0.08	92.2%
	GRAN	0.00 ± 0.01	0.04 ± 0.08	0.00 ± 0.00	0.02 ± 0.01	0.29 ± 0.26	0.00 ± 0.17	81.3%

Table 4: Mean and max risk for the community-level setting.

attack strength. Nevertheless, together with Table 4, these results confirm that singling out is not the primary threat in community-level synthesis; linkability and inference risks are more relevant when they occur.

5.4 Summary

Our results indicate a clear difference between node-level and community-level attacks. For node-level attacks, singling out and inference can be highly effective, while linkability remains comparatively low in our experiments, with maximum risk at most 0.24. For community-level attacks, singling out is consistently weak, whereas linkability and inference can be substantially higher, especially on the ego datasets and for GruM, SPECTRE, or GRAN.

This difference suggests that individual nodes can stand out within a single graph more readily than whole graphs stand out within a dataset of related graphs. The datasets we use contain graphs from the same domain, so whole-graph structural outliers are comparatively rare. Simultaneously, dependencies among structural graph attributes can enhance community-level linkability and inference. However, the relatively small number of community-level training graphs, especially for *Ego-Facebook*, can inflate variance.

There is a large gap between mean and maximum risk across many settings, so attack parameterization matters. Even without access to the training data, an attacker can use general heuristics, such as favoring tabular attributes for node-level singling out and inference on attribute-rich graphs, structural attributes for linkability, or model-specific predicate complexity, to improve on the average attack performance.

Datasets and synthesizers both play a major role in attack success. *Elliptic*, which is large and poor in node attributes, generally yields lower risk, supporting the view that scale and lack of distinguishing attributes provide some natural protection. Our conclusion, though, is naturally conditional on the overall set of tested models and datasets, as well as their feasible combinations.

The added models reinforce that the utility–privacy relationship is complex. SPECTRE usually has lower node-level risks than EDGE and GruM, but can still show substantial community-level linkability and inference risk. GRAN, only evaluated at the community-level, has very low singling out risk, but non-trivial linkability on *Ego-Twitter* and *Elliptic* and high inference risk on *Ego-Twitter*. Thus, high utility often coincides with higher risk, but the relationship is not universal and depends on the attack, dataset, and model family.

Edge-level attacks follow the same broad pattern: inference is the dominant risk, reaching near-maximal values on the ego datasets, while singling out remains modest, and linkability is close to negligible. Compared to node-level attacks, the edge-based formulations preserve the main inference-risk concern but provide weaker evidence of reliable singling out or linking; Elliptic again shows substantially lower risk, reinforcing that larger, attribute-poor graphs are harder to attack. Further details are in Appendix B.2.

6 Conclusion

We are the first to consider the privacy risks of graph synthesis, finding that risks exist at both the community and, more importantly, node level, allowing the recognition of sensitive attributes of individuals based on publicly available graph data. Our attacks do not require the attacker to have access to the training data or the model to be effective.

Our evaluation study covering six main attack types, three datasets from two domains, and five graph synthesizers indicates that singling out and inference risk is generally high for node-level attacks. Meanwhile, linkability and inference are the most effective community-level attacks.

We find that while there is a correlation between utility and privacy risk, it is not always necessary that a high-fidelity synthesizer carries the highest risk. The result offers hope that future research can identify methods to enable the design of novel graph synthesizers that generate realistic data at a low privacy risk. Up to now, existing models have not considered privacy as a design goal, suggesting that while there is a risk in current designs, future research may succeed in eliminating such risks.

Acknowledgments

This research is partly funded by the Priv-GSyn (200021E_229204/550302673) and Tracer (2000-1-242997) projects of the Swiss National Science Foundation and the German Science Foundation, as well as the DEPMAT project (P20-22/N21022) of the Dutch Research Council, and ASM International NV.

The manuscript includes input from generative artificial intelligence tools for editing tasks such as rephrasing and typographical fixes. The accompanying code also includes input from generative artificial intelligence tools for tasks such as implementing standard methods and refactoring.

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM, 308–318. https://doi.org/10.1145/2976749.2978318

[2] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. 2023. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. *CoRR* abs/2303.08797 (2023).

[3] Albert-László Barabási. 2013. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371, 1987 (2013).

[4] European Data Protection Board. 2024. EDPB annual report 2024: protecting personal data in a changing landscape. https://www.edpb.europa.eu/system/files/2025-04/edpb-annual-report-2024_en.pdf

[5] Vladimir Boginski, Sergiy Butenko, and Panos M. Pardalos. 2005. Statistical analysis of financial networks. *Comput. Stat. Data Anal.* 48, 2 (2005), 431–443. https://doi.org/10.1016/J.CSDA.2004.02.004

[6] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting Training Data from Diffusion Models. In *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, Joseph A. Calandrino and Carmela Troncoso (Eds.). USENIX Association, 5253–5270. https://www.usenix.org/conference/usenixsecurity23/presentation/carlini

[7] Xiaohui Chen, Jiaxing He, Xu Han, and Liping Liu. 2023. Efficient and Degree-Guided Graph Generation via Discrete Diffusion Modeling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 4585–4610. https://proceedings.mlr.press/v202/chen23k.html

[8] Ministério da Justiça e Segurança Pública. 2018. Lei Geral de Proteção de Dados Pessoais. https://www.gov.br/anpd/pt-br/centrais-de-conteudo/brazilian-data-protection-law.pdf

[9] Sybil Derrible and Christopher Kennedy. 2011. Applications of graph theory and network science to transit network design. *Transport reviews* 31, 4 (2011), 495–519.

[10] Josep Domingo-Ferrer. 2025. How Worrying Are Privacy Attacks Against Machine Learning? *CoRR* abs/2511.10516 (2025).

[11] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 4052)*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.). Springer, 1–12. https://doi.org/10.1007/11787006_1

[12] Peter Eigenschink, Thomas Reutterer, Stefan Vamosi, Ralf Vamosi, Chang Sun, and Klaudius Kalcher. 2023. Deep Generative Models for Synthetic Data: A Survey. *IEEE Access* 11 (2023), 47304–47320.

[13] Stephen Eubank, Hasan Guclu, VS Anil Kumar, Madhav V Marathe, Aravind Srinivasan, Zoltan Toroczkai, and Nan Wang. 2004. Modelling disease outbreaks in realistic urban social networks. *Nature* 429, 6988 (2004), 180–184.

[14] Matteo Giomi, Franziska Boenisch, Christoph Wehmeyer, and Borbála Tasnádi. 2023. A Unified Framework for Quantifying Privacy Risk in Synthetic Data. *Proc. Priv. Enhancing Technol.* 2023, 2 (2023), 312–328. https://doi.org/10.56553/POPETS-2023-0055

[15] Faqian Guan, Tianqing Zhu, Wanlei Zhou, and Philip S. Yu. 2025. Topology-Based Node-Level Membership Inference Attacks on Graph Neural Networks. *IEEE Trans. Big Data* 11, 5 (2025), 2809–2826. https://doi.org/10.1109/TBDATA.2025.3558855

[16] Shion Guha and Stephen B. Wicker. 2015. Do Birds of a Feather Watch Each Other?: Homophily and Social Surveillance in Location Based Social Networks. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, CSCW 2015, Vancouver, BC, Canada, March 14 - 18, 2015*, Dan Cosley, Andrea Forte, Luigina Ciolfi, and David McDonald (Eds.). ACM, 1010–1020. https://doi.org/10.1145/2675133.2675179

[17] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2021. Stealing Links from Graph Neural Networks. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, Michael D. Bailey and Rachel Greenstadt (Eds.). USENIX Association, 2669–2686. https://www.usenix.org/conference/usenixsecurity21/presentation/he-xinlei

[18] Alex Hern. 2018. Fitness tracking app Strava gives away location of secret US army bases. *The Guardian* (2018). https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases

[19] Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. 2024. Graph Generation with Diffusion Mixture. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. https://openreview.net/forum?id=cZTFxktg23

[20] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. 2019. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=S1zk9iRqF7

[21] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. 2025. FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space. *CoRR* abs/2506.15742 (2025). https://doi.org/10.48550/ARXIV.2506.15742

[22] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard S. Zemel. 2019. Efficient Graph Generation with Graph Recurrent Attention Networks. In *NeurIPS*. 4257–4267.

[23] Lanhua Luo, Wang Ren, Huasheng Huang, and Fengling Wang. 2024. A Survey on Privacy Attacks and Defenses in Graph Neural Networks. *Inf. Technol. Control.* 53, 4 (2024), 1251–1278. https://doi.org/10.5755/J01.ITC.53.4.37737

[24] Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. 2022. SPECTRE: Spectral Conditioning Helps to Overcome the Expressivity Limits of One-shot Graph Generators. In *ICML (Proceedings of Machine Learning Research)*. PMLR, 15159–15179.

[25] Julian J. McAuley and Jure Leskovec. 2012. Learning to Discover Social Circles in Ego Networks. In *NIPS*. 548–556.

[26] Giorgia Minello, Alessandro Bicciato, Luca Rossi, Andrea Torsello, and Luca Cosmo. 2025. Generating Graphs via Spectral Diffusion. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. https://openreview.net/forum?id=AAXBfJNHDt

[27] Noman Mohammed, Rui Chen, Benjamin C. M. Fung, and Philip S. Yu. 2011. Differentially private data release for data mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, Chid Apté, Joydeep Ghosh, and Padhraic Smyth (Eds.). ACM, 493–501. https://doi.org/10.1145/2020408.2020487

[28] Arvind Narayanan and Vitaly Shmatikov. 2006. How To Break Anonymity of the Netflix Prize Dataset. *CoRR* abs/cs/0610105 (2006). arXiv:cs/0610105 http://arxiv.org/abs/cs/0610105

[29] Mark Newman. 2018. *Networks*. Oxford university press.

[30] Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. 2002. Random graph models of social networks. *Proceedings of the national academy of sciences* 99, suppl_1 (2002), 2566–2572.

[31] State of California Department of Justice. 2018. California Consumer Privacy Act (CCPA). https://oag.ca.gov/privacy/ccpa

[32] The Federal Assembly of the Swiss Confederation. 2020. Federal Act on Data Protection. https://www.fedlex.admin.ch/eli/cc/2022/491/en

[33] Article 29 Data Protection Working Party. 2014. Opinion 05/2014 on anonymisation techniques. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf

[34] Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. 2024. DeFoG: Discrete Flow Matching for Graph Generation. *CoRR* abs/2410.04263 (2024). https://doi.org/10.48550/ARXIV.2410.04263 arXiv:2410.04263

[35] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 5558–5567. http://proceedings.mlr.press/v97/sablayrolles19a.html

[36] Jayshree Sarathy, Sophia Song, Audrey Haque, Tania Schlatter, and Salil Vadhan. 2023. Don't look at the data! how differential privacy reconfigures the practices of data science. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[37] Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. 2025. TabDiff: a Mixed-type Diffusion Model for Tabular Data Generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. https://openreview.net/forum?id=swvURjrt8z

[38] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems* 10, 05 (2002), 557–570.

[39] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua

Huang, Xiaofeng Meng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. 2025. Wan: Open and Advanced Large-Scale Video Generative Models. *CoRR* abs/2503.20314 (2025). https://doi.org/10.48550/ARXIV.2503.20314 arXiv:2503.20314

[40] Xiuling Wang, Xin Huang, Guibo Luo, and Jianliang Xu. 2026. Inference Attacks Against Graph Generative Diffusion Models. *CoRR* abs/2601.03701 (2026).

[41] Xiuling Wang and Wendy Hui Wang. 2024. Subgraph Structure Membership Inference Attacks against Graph Neural Networks. *Proc. Priv. Enhancing Technol.* 2024, 4 (2024), 268–290. https://doi.org/10.56553/POPETS-2024-0116

[42] Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. 2024. Detecting, Explaining, and Mitigating Memorization in Diffusion Models. In *ICLR*. OpenReview.net.

[43] Robert S. Wilkov. 1972. Analysis and Design of Reliable Computer Networks. *IEEE Trans. Commun.* 20, 3 (1972), 660–678. https://doi.org/10.1109/TCOM.1972.1091214

[44] Carl Yang, Haonan Wang, Ke Zhang, Liang Chen, and Lichao Sun. 2021. Secure Deep Graph Generation with Link Differential Privacy. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Zhi-Hua Zhou (Ed.). ijcai.org, 3271–3278. https://doi.org/10.24963/IJCAI.2021/450

[45] Xinyu Yuan and Yan Qiao. 2024. Diffusion-TS: Interpretable Diffusion for General Time Series Generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=4h1apFjO99

[46] Yi Zhang, Yuying Zhao, Zhaoqing Li, Xueqi Cheng, Yu Wang, Olivera Kotevska, Philip S. Yu, and Tyler Derr. 2024. A Survey on Privacy in Graph Neural Networks: Attacks, Preservation, and Applications. *IEEE Trans. Knowl. Data Eng.* 36, 12 (2024), 7497–7515. https://doi.org/10.1109/TKDE.2024.3454328

A Notation Summary

Table 5 summarizes the main notation used throughout the paper.

Symbol	Description
$\mathbf{G}$	graph
$\mathbf{A}$	adjacency matrix of a graph
$\mathbf{X}^{\text{tab}}$	tabular node attrs. of a graph
$\mathbf{X}^{\text{str}}$	structural node attrs. of a graph
$\mathbf{X}^{\text{all}}$	tabular and structural node attrs. of a graph
$\mathbf{X}$	tabular and structural node attrs. of multiple graphs
$\mathbf{Y}$	structural community attrs. of multiple graphs
$\mathbf{E}^{\text{str}}$	structural edge attrs. of a graph
$\mathbf{E}^{\text{all}}$	all edge attrs. of a graph
$\mathbf{E}$	all edge attrs. of multiple graphs
N	number of nodes in a graph
M	number of edges in a graph
K	number of graphs in a dataset
f^n	func. to compute a structural node attr. for a graph
f^g	func. to compute a structural community attr. for a graph
f^m	func. to compute a structural edge attr. for a graph
D_t	number of tabular node attrs.
D_n	number of structural node attrs.
D_g	number of structural graph attrs.
D_m	number of structural edge attrs.
$\mathcal{D}_{\text{ori}}$	original graph dataset
$\mathcal{D}_{\text{train}}$	train graph dataset
$\mathcal{D}_{\text{control}}$	control graph dataset
$\mathcal{D}_{\text{syn}}$	synthetic graph dataset
R	privacy risk
N_A	number of attacks
N_V	number of singling out variables
L	minimum number of syn. singled-out graphs
P	singling out predicate
$\mathcal{A}$, $\mathcal{B}$	linkability column subsets

Table 5: Main notation overview.

B Details on Edge-level Attacks

B.1 Pseudocode

Algorithm 4 describes the edge-level augmentation based on tabular node attributes $\mathbf{X}^{\text{tab}}$ and edge structural properties f^m. For a graph $\mathbf{G}$, $\mathbf{E}^{\text{str}}$ denotes the structural properties of all its edges. $\mathbf{E}^{\text{all}}$ is the final representation, which also includes explicit attributes (in our implementation, the tabular attributes of edge endpoints) for each edge, and joins together all edges from available graphs.

Algorithm 4 Edge-level augmentation

Require: $\mathcal{D}, [f_1^m, f_2^m, \ldots, f_{D_m}^m]$
$\mathbf{E} \leftarrow [\quad]$
for all $\mathbf{G} \in \mathcal{D}$ **do**
  $(\mathbf{X}^{\text{tab}}, \mathbf{A}) \leftarrow \mathbf{G}$
  $\mathbf{E}^{\text{str}} \leftarrow \mathbf{0}^{M \times D_m}$ ▷ M-edge graph
  $\mathbf{E}^{\text{inds}} \leftarrow$ edge list for $\mathbf{A}$ ▷ $M \times 2$ entries for edge endpoints
  for all $(i, a) \in (\{1, \ldots, M\} \times \{1, \ldots, D_m\})$ **do**
    $\mathbf{E}^{\text{str}}[i, a] \leftarrow f_a^m(\mathbf{G}, \mathbf{E}^{\text{inds}}[i, 0], \mathbf{E}^{\text{inds}}[i, 1])$
  $\mathbf{E}^{\text{all}} \leftarrow [\mathbf{X}^{\text{tab}}[\mathbf{E}^{\text{inds}}[:, 0]] \mid \mathbf{X}^{\text{tab}}[\mathbf{E}^{\text{inds}}[:, 1]] \mid \mathbf{E}^{\text{str}}]$
  $\mathbf{E} \leftarrow \mathbf{E} + [\mathbf{E}^{\text{all}}]$ ▷ Append $\mathbf{E}^{\text{all}}$ to $\mathbf{E}$
return $\mathbf{E}$

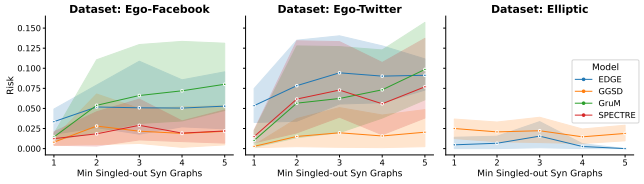
B.2 Experiments

Throughout evaluated edge-level scenarios, we allow all structural and tabular attributes. For singling out, we independently sweep L and N_V from 1 to 5, yielding 25 attacks in total; we run 500 attacks, as in the main evaluation. In terms of linkability, we sweep k from 1 to 5, giving 5 total attacks. For inference, we test each tabular attribute of an edge's endpoint as the target property, leading to $2D_t$ attacks. Unlike in the main evaluation, where we try all possible attacks, due to the larger search space, we limit linkability and inference tests to 2,000 attacks each.

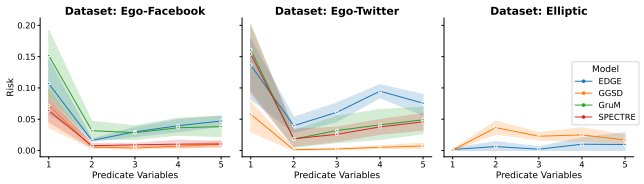
Table 6 shows that inference attacks dominate edge-level risk. Across the two ego datasets, the maximum inference risk is consistently high across models, reaching 0.98–0.99 for EDGE and GruM and remaining above 0.86 even in the lowest-performing settings. Mean inference risk is also substantial, especially for EDGE on ego datasets, while Elliptic again has a much lower ceiling, with maximum inference risk only 0.18 for EDGE and 0.23 for GGSD. By contrast, singling out remains modest across all datasets and models, with maxima of at most 0.21, and linkability is effectively negligible, with maximum risks at or near zero except for small non-zero values on Ego-Twitter. Compared to the node-level results of Table 3, edge-level inference is the only attack that reaches similarly severe risks: on the ego datasets, its maxima are close to the node-level inference maxima, and its means are often comparable. The main difference is singling out: node-level singling out can

Dataset	Model	Attack (Mean + SD) ↑			Attack (Max + CI) ↑		
		Singling out	Linkability	Inference	Singling out	Linkability	Inference
Ego-Facebook	EDGE	0.05 ± 0.04	0.01 ± 0.01	0.61 ± 0.28	0.17 ± 0.03	0.02 ± 0.01	0.98 ± 0.01
	GGSD	0.02 ± 0.03	0.00 ± 0.00	0.33 ± 0.34	0.11 ± 0.03	0.00 ± 0.00	0.96 ± 0.01
	GruM	0.06 ± 0.06	0.00 ± 0.00	0.38 ± 0.35	0.20 ± 0.04	0.00 ± 0.00	0.97 ± 0.01
	SPECTRE	0.02 ± 0.03	0.00 ± 0.00	0.23 ± 0.31	0.09 ± 0.03	0.01 ± 0.00	0.92 ± 0.03
Ego-Twitter	EDGE	0.08 ± 0.05	0.04 ± 0.02	0.49 ± 0.36	0.19 ± 0.04	0.07 ± 0.01	0.98 ± 0.01
	GGSD	0.01 ± 0.03	0.00 ± 0.00	0.29 ± 0.35	0.08 ± 0.04	0.00 ± 0.00	0.97 ± 0.02
	GruM	0.06 ± 0.06	0.02 ± 0.02	0.31 ± 0.37	0.21 ± 0.04	0.04 ± 0.01	0.99 ± 0.01
	SPECTRE	0.06 ± 0.06	0.01 ± 0.01	0.27 ± 0.30	0.20 ± 0.04	0.01 ± 0.01	0.86 ± 0.10
Elliptic	EDGE	0.01 ± 0.01	0.00 ± 0.00	0.16 ± 0.04	0.05 ± 0.05	0.00 ± 0.00	0.18 ± 0.05
	GGSD	0.02 ± 0.02	0.00 ± 0.00	0.20 ± 0.04	0.05 ± 0.02	0.00 ± 0.00	0.23 ± 0.06

Table 6: Mean and max risk for the edge-level setting.



(a) Effect of increasing the minimum number of synthetic graphs singled out by each predicate.



(b) Effect of increasing the number of predicate variables for singling out attacks.

Figure 7: Ablations for edge-level attacks.

Dataset (& Level)	Model	Degree	Wavelet	Spectre	Cluster	Motif	Orbit
Ego-Facebook (Node/Edge)	EDGE	0.0661	0.1489	0.0324	0.9612	**0.5043**	**0.5100**
	GGSD	0.3235	0.2441	0.0986	1.4824	0.5377	0.5159
	GruM	**0.0458**	**0.0338**	**0.0179**	0.7773	0.5105	**0.5100**
	SPECTRE	0.1421	0.1804	0.4023	**0.6476**	0.7273	**0.5100**
Ego-Twitter (Node/Edge)	EDGE	**0.0420**	0.0727	0.0365	0.5938	0.5140	**0.5100**
	GGSD	0.4192	0.2502	0.0776	1.1035	0.5475	0.5308
	GruM	0.0437	**0.0289**	**0.0159**	**0.5157**	0.5115	**0.5100**
	SPECTRE	0.0556	0.0478	0.0357	0.5237	**0.5114**	**0.5100**
Elliptic (Node/Edge)	EDGE	**0.0755**	**0.0838**	**0.0812**	**0.0284**	**0.2933**	1.0278
	GGSD	0.1908	0.1596	0.1813	0.0805	0.3764	**0.5065**
Ego-Facebook (Community)	EDGE	**0.0135**	0.0354	0.0287	0.1219	0.1355	0.1135
	GruM	0.0245	**0.0096**	**0.0151**	**0.1196**	0.1441	**0.1078**
	SPECTRE	0.0337	0.0276	0.0152	0.2297	**0.1307**	0.1100
Ego-Twitter (Community)	EDGE	**0.0025**	0.0091	0.0040	0.0250	**0.0192**	**0.0200**
	GruM	0.0114	**0.0059**	0.0048	**0.0226**	0.0196	**0.0200**
	SPECTRE	0.0065	0.0079	**0.0018**	**0.0226**	0.0209	**0.0200**
	GRAN	0.1141	0.0970	0.1074	0.0257	0.0202	**0.0200**
Elliptic (Community)	EDGE	**0.0435**	**0.0617**	**0.0457**	**0.0134**	**0.0172**	**0.0619**
	GRAN	0.3305	0.3367	0.2468	0.9615	0.6735	0.0647

Table 7: MMD measurements between synthetic and control graphs for the tested datasets and models. Lower is better.

be highly effective, reaching 1.0 on both ego datasets for EDGE, whereas edge-level singling out remains far lower throughout. Linkability is low at both levels, but it is even lower for edges. Overall, the edge-level setting preserves the main inference-risk concern from the node-level case, while offering much less evidence that individual edges can be singled out or linked reliably; the Elliptic results remain consistent with the broader pattern that large, attribute-poor graphs are harder to attack.

Figure 7 ablates over the minimum number of matching synthetic graphs and the number of variables in predicates for edge-level singling out attacks. Note that for edge-level inference and linkability, we run only 2000 attacks per scenario, rather than all possible ones, so the confidence intervals are less precise. Moreover, per Table 2, the number of scenarios for each attack is lower than the main experiments, especially relative to the node-level part. Nevertheless, when validating the singling out attacks against two or more synthetic graphs, Figure 7a still shows a trend of increasing risk for the ego datasets. Elliptic deviates from the pattern with little variation in risk across the board, but a considerably lower risk than in the other two datasets for all models. The phenomenon suggests a reduced risk ceiling in the dataset, which simpler attacks can already reach. For the number of predicate variables, Figure 7b shows that, apart from GGSD on Elliptic, univariate predicates tend to carry the highest risk, and two-variable ones have the lowest, even if increasing the number of variables from two onward partly helps recover the drop.

C MMD Results

Table 7 contains MMD (Maximum Mean Discrepancy) measurements for different structural properties across tested dataset and model combinations. We examine properties included in previous works [7, 19, 22, 24, 26]: node **degree**, eigenspace **wavelet** transform, eigenvalues of the normalized Laplacian **spectrum**, node **cluster**ing coefficient, graph **motif**s, and four-node **orbit** counts.

Node/edge levels correspond to the test setting with one attributed training graph, while the community level corresponds to the setting with multiple unattributed training graphs.

While no single model dominates across all datasets and structural statistics, GruM is consistently competitive on the ego datasets in node/edge settings, achieving the best or near-best MMD values for degree, wavelet, spectrum, and clustering. The behavior matches the presumption that, while not the most scalable, GruM is the most capable tested model. SPECTRE performs well on several community-level metrics, particularly on Ego-Twitter, where it matches or exceeds the best clustering, spectrum, and orbit scores. EDGE is strongest on the Elliptic datasets, substantially outperforming the alternatives on nearly all node-, community- and edge-level statistics except orbit in the node/edge setting, where GGSD is better. As a highly scalable model, EDGE's good performance on the Elliptic-based datasets, which are the largest ones tested, is fitting. In contrast, GGSD and GRAN are generally less competitive. GGSD still stands out in the Elliptic node/edge setting, achieving the best orbit performance. Unfortunately, GRAN's poor performance is particularly acute in the Elliptic community dataset, where it lags significantly across all metrics but orbit.