The Right to Hide:
Masking Community Affiliation via Minimal Graph Rewiring

Matteo Silvestri
silvestri.m@di.uniroma1.it
Sapienza University of Rome
Rome, Italy

Edoardo Gabrielli
edoardo.gabrielli@uniroma1.it
Sapienza University of Rome
Rome, Italy

Fabrizio Silvestri
Sapienza University of Rome
Rome, Italy

Gabriele Tolomei
Sapienza University of Rome
Rome, Italy

Abstract

Protecting privacy in social graphs may require obscuring nodes' membership in sensitive communities. However, doing so without significantly disrupting the underlying graph topology remains a key challenge. In this work, we address the *community membership hiding* problem, which involves strategically modifying the graph structure to conceal a target node's affiliation with a community, regardless of the detection algorithm used. We reformulate the original discrete, counterfactual graph search objective as a differentiable constrained optimisation task. To this end, we introduce ∇- CMH, a new gradient-based method that operates within a feasible modification budget to minimise structural changes while effectively hiding a node's community membership. Extensive experiments on multiple datasets and community detection methods demonstrate that our technique outperforms existing baselines, achieving the best balance between node hiding effectiveness and graph rewiring cost, while preserving computational efficiency.

Keywords

Community membership hiding, community detection, social graph privacy, counterfactual graph, gradient-based optimisation

1 Introduction

Community detection is a fundamental tool for analysing complex graph structures such as social networks, biological systems, and communication networks [19]. It is typically performed using *community detection algorithms* [6], which aim to uncover groups of tightly connected nodes – called *communities* – that exhibit similar characteristics, interactions, or structural patterns.

The ability to identify such communities has enabled a wide range of applications across diverse domains [23, 26], including targeted advertising [32], recommendation systems [1], and network security [14]. However, despite its utility, community detection also raises significant privacy concerns, especially in the context of social graphs, where it can inadvertently reveal sensitive affiliations or personal traits. For example, community assignments may expose users' political or religious beliefs, age, gender [52], or ties to controversial or conspiratorial groups [44].

While one option for protecting privacy is to leave the platform entirely, such an action is often too drastic. A more flexible approach would enable individuals to control their visibility within detected communities while continuing to participate. This strikes a better balance between preserving privacy and maintaining the utility of community detection. Furthermore, it better aligns with modern data protection initiatives such as the European GDPR [51] and AI Act [2], which includes provisions like the "*right to be forgotten*" [9].

Motivated by this challenge, we address the problem of *community membership hiding*, first introduced by Bernini et al. [4]. This task draws inspiration from *counterfactual reasoning* [31, 47, 48], and involves strategically modifying a graph's topology to prevent a target node from being identified as part of a specific community by detection algorithms.

In this work, we build upon the method proposed by Bernini et al. [4], formulating the community membership hiding task as a counterfactual graph objective. Specifically, we cast it as a constrained optimisation problem, where the goal is to perturb the structure surrounding the target node to obscure its community affiliation. However, unlike the original approach that treats the problem as a discrete objective and solves it via deep reinforcement learning, our method is inspired by adversarial attacks on graph neural networks [50]. We reformulate the task as a *differentiable* objective, enabling efficient solution through gradient-based optimisation techniques. This continuous formulation allows for minimal structural modifications while achieving effective concealment. Our approach offers three key advantages over the original technique [4]: (*i*) a higher success rate in the community membership hiding task, (*ii*) meticulous allocation of the available budget to achieve the goal, and (*iii*) improved computational efficiency.

Our main contributions are summarised below.

(1) We reformulate the original community membership hiding task as a differentiable counterfactual graph objective;
(2) We propose ∇- CMH, a gradient-based method that strategically perturbs the target node's neighbourhood to hide;
(3) We evaluate our approach on real-world graph datasets and show its superiority over existing baselines. The source code of our method is available at: https://anonymous.4open.science/r/community-membership-hiding-B188/.

The remainder of the paper is organised as follows. Section 2 reviews related work. In Section 3, we present background and preliminaries. Section 4 reformulates the problem setting. Our proposed method is detailed in Section 5, followed by an extensive empirical evaluation in Section 6. We discuss the current limitations of our method in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

The body of related work primarily falls into two key areas: *community detection* and *community membership hiding*. The latter also shares connections with *adversarial attacks on graphs*. Below, we review the most relevant contributions in each of these domains.

Community Detection. Community detection algorithms play a crucial role in analysing graph structures by identifying and grouping nodes into *communities*. These communities are clusters of nodes that exhibit a higher density of connections within the group compared to their connections with the rest of the graph. Existing approaches to identify non-overlapping communities include Modularity Optimisation [7, 8, 49], Spectrum Optimisation [43], Random Walk [37], Label Propagation [39], or Statistical Inference [3]. In contrast, overlapping community detection algorithms frequently use methods such as Matrix Factorisation [56], Neighbourhood-Inflated Seed Expansion [55], or techniques based on minimising the Hamiltonian of the Potts model [41]. For a comprehensive overview of these methods, see the extensive summary by Jin et al. [24]. Furthermore, deep learning models have been increasingly employed to tackle the complex problem of community detection [38, 46], with DGCLUSTER [5] representing a notable approach.

Community Membership Hiding. Community membership hiding addresses the problem of concealing a single node's affiliation with a particular community. The seminal work in this area is by Bernini et al. [4], who introduce *DRL-Agent*, a method that strategically modifies a node's local neighbourhood to obscure its community membership. They formulate the task as a counterfactual graph objective and leverage a graph neural network (GNN) to capture the input graph's structural complexity. Their approach uses deep reinforcement learning (DRL) within a Markov decision process to determine which edges to modify, under a fixed budget to limit the number of changes for efficiency and realism.

Unlike DRL-Agent, our method reformulates the problem as a *differentiable* counterfactual objective, enabling the use of well-known gradient-based optimisation techniques. This brings three key benefits: (*i*) improved node hiding effectiveness, (*ii*) a more efficient use of the available budget, avoiding its full exhaustion, and (*iii*) lower computational overhead by avoiding costly DRL training.

Adversarial Attacks on Graphs. The task of hiding a node's community affiliation can also be interpreted as a targeted objective within the broader framework of adversarial attacks on graphs. Although this domain has been extensively explored [15, 25], most efforts focus on evading link prediction [12, 30, 50], or disrupting node and graph classification [16, 57], leaving attacks against non-parametric graph clustering unexplored [13]. In addition, many methods target specifically GNNs, limiting their applicability.

Some works tailored for community detection aim to hide groups of nodes by dispersing them across communities [18, 29, 54]. These approaches focus on group-level obfuscation and rely on global modifications, which are ill-suited for scenarios requiring localised changes. Other works, in contrast, aim to disrupt community detection performance by minimising modularity [11] or altering node-level features, such as centrality [54]. However, we tackle a more fine-grained problem: obscuring the community membership of a single target node by modifying *only* its local neighbourhood – namely, by altering edges that the target node itself can control.

3 Background and Preliminaries

In this section, we first introduce the notation used throughout the paper. We then provide a brief overview of the well-known community detection problem, which forms the foundation for defining the community membership hiding problem.

Let $G = (V, E)$ be an arbitrary (undirected[1]) graph, where V is the set of nodes with $|V| = n$, and $E \subseteq V \times V$ is the set of edges with $|E| = m$. The structure of G is represented by a binary adjacency matrix denoted by $A = (A_{u,v})_{u,v \in V}$, where $A_{u,v} = 1$ if there is an edge between nodes u and v, i.e., $(u, v) \in E$, and $A_{u,v} = 0$ otherwise. The neighbourhood of a node u, defined as the set of nodes directly connected to u, corresponds to the u-th row of A. We denote this row as A_u and refer to it as the *adjacency vector* of u.

The *community detection* problem aims to partition the nodes of a graph into clusters, referred to as *communities*. Intuitively, communities are groups of nodes with strong intra-cluster connections compared to their links with nodes outside the cluster. In this work, we focus on detecting non-overlapping communities based solely on the graph's edge structure, as in [4], leaving the exploration of methods that consider node features to future research. Formally, a community detection algorithm is a function $f(\cdot)$ that partitions the graph G into a set of non-empty, disjoint communities $f(G) = \{C_1, C_2, \ldots, C_k\}$, where each node u is assigned to *exactly one* community, and the number of communities k is typically unknown. Note that the actual input to f can be *any* suitable representation of G, ranging from the simple adjacency matrix A to more complex forms such as $g = (A, X)$, where X is the node feature matrix and g is a graph neural network. Without loss of generality, hereinafter we denote the input simply as G.

Community detection algorithms often aim to maximise a score that quantifies intra-community cohesiveness. A widely used metric for this purpose is Modularity [35]. However, optimising Modularity is generally NP-hard. To address this challenge, numerous practical approximation methods have been developed. Notable examples include Greedy [8], Louvain [7], Leiden [49], WalkTrap [37], InfoMap [10], Label Propagation [39], Leading Eigenvectors [34], Edge-Betweenness [21], and SpinGlass [40].

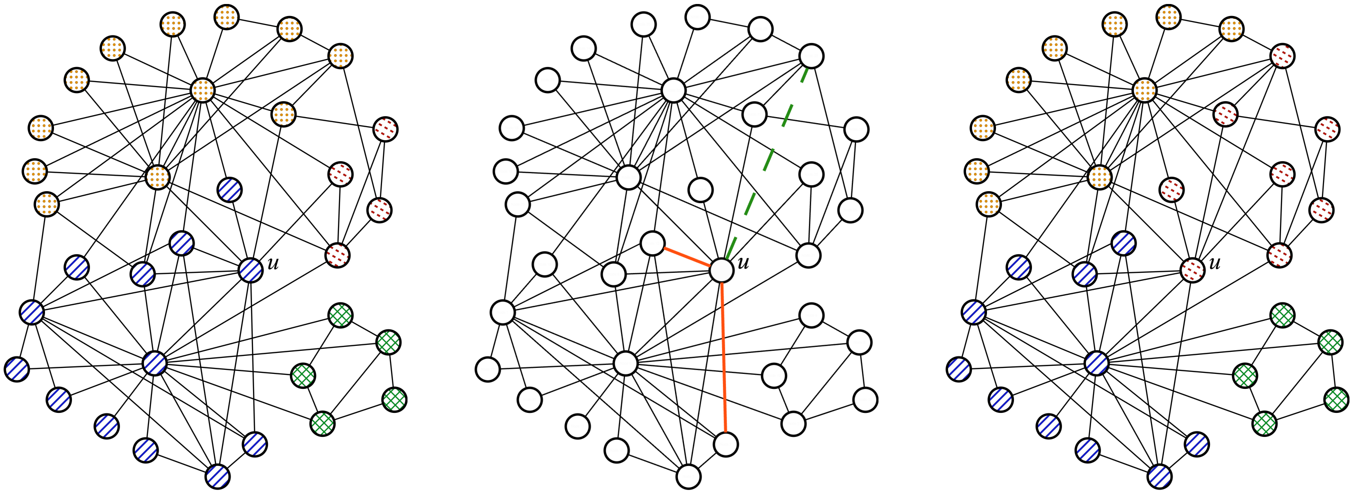
4 Community Membership Hiding (CMH)

In its most general form, *community membership hiding* aims to prevent a specific node from being identified as part of a designated cluster by a link-based community detection algorithm. This is achieved by strategically modifying the node's connections – i.e., its local neighbourhood – which corresponds to altering its row in the adjacency matrix.

This definition is domain-agnostic and applies to *any* graph, as reflected in the problem formulation in Section 4.1. However, the most typical scenario arises in the context of social networks. In such settings, the motivating use case involves a user who is either aware or suspects that they belong to a particular community, yet wishes to obscure this membership for privacy reasons.

Depending on who performs the hiding, we distinguish between a *platform-mediated* and a *user-initiated* model. In the former, the network owner offers a hiding service, with full access to the graph and knowledge of the community detection algorithm $f(\cdot)$. In the

[1] The same reasoning easily extends to the case where G is directed.



(a) Target node u and the communities $f(G)$ detected by the *Louvain* algorithm.

(b) Counterfactual graph G', where bold (red) edges represent deletions and dashed (green) edges indicate additions.

(c) New communities $f(G')$ detected on the new graph G'. The hiding objective is achieved if $sim(C_i \setminus \{u\}, C_i' \setminus \{u\}) \leq \tau$.

Figure 1: Example of the Community Membership Hiding Problem on the Karate Club graph.

latter, the user acts independently, relying solely on access to their local neighbourhood and without knowledge of $f(\cdot)$.

In this work, we focus on a *restricted* platform-mediated setting, following the assumptions made by Bernini et al. [4]. Specifically, we treat $f(\cdot)$ as a black box, leveraging only on its input–output behaviour without requiring insight into its internal mechanics. Note that this is a conservative assumption, since in platform-mediated scenarios $f(\cdot)$ may in fact be fully known. We also assume full access to the graph topology, consistent with the case where the platform executes the hiding on behalf of the user, though our approach can be extended to partial-knowledge settings.

4.1 Problem Definition

Let $G = (V, E)$ be a graph and $f(G) = \{C_1 \ldots C_k\}$ denote the community partitioning obtained by applying a community detection algorithm $f(\cdot)$. Suppose that $f(\cdot)$ identifies the target node u as a member of the community C_i, denoted as $u \in C_i$.

The goal of the community membership hiding problem, as introduced by Bernini et al. [4], is to learn a *perturbation function* $h_\theta(\cdot)$, parameterised by θ, that transforms the original graph $G = (V, E)$ into a perturbed version $G' = h_\theta(G) = (V, E')$. The objective is to find a perturbation that ensures the target node u is no longer assigned to its original community C_i when the community detection algorithm $f(\cdot)$ is applied to G'.

The definition of community membership hiding may vary. For instance, if a target node is reassigned to a new community C_i' in the perturbed graph G', one goal may be to minimise the similarity between the original community C_i and the new community C_i'. Alternatively, the hiding objective might focus on ensuring that specific nodes from C_i do not belong to C_i'. For example, if $C_i = \{u, v, w, y, z\}$, we might aim to reassign u to C_i' such that $y, z \notin C_i'$.

In this work, we adopt the first definition of membership hiding, leaving the exploration of alternative definitions for future research. Specifically, given a similarity function $sim(\cdot, \cdot)$ and a non-negative threshold τ, we consider the hiding task successful if the condition $sim(C_i \setminus \{u\}, C_i' \setminus \{u\}) \leq \tau$ is satisfied. We assume that $sim(\cdot, \cdot)$ outputs values in the range $[0, 1]$ and $\tau \in [0, 1)$. When $\tau = 0$, the condition is most restrictive, requiring no overlap between C_i and C_i' (excluding the node u). As τ increases, the condition becomes less stringent, making it easier to achieve the hiding goal.

More formally, the community membership hiding task resorts to solving the following constrained optimisation problem:

$$\theta^* = \arg\min_{\theta} \left\{ \mathcal{L}(h_\theta; G, f, u) \right\} \quad (1)$$
$$\text{subject to: } \|\boldsymbol{b}_u\|_0 \leq \beta,$$

where $\mathcal{L}(\cdot)$ is a loss function defined as follows:

$$\mathcal{L}(h_\theta; G, f, u) = \ell_{hide}(h_\theta, G; f, u) + \lambda\, \ell_{dist}(h_\theta(G), G; f). \quad (2)$$

The first term (ℓ_{hide}) incentivises the target node u to detach from its original community, i.e., to reach the specific hiding goal. The second term (ℓ_{dist}) quantifies the impact of modifications – e.g., by computing the distance between the original graph G and the modified graph G', the distance between their corresponding partitions $f(G)$ and $f(G')$, or a convex combination of both, as done by Bernini et al. [4]. The distance between partitions captures how local perturbations can alter the global community structure, potentially reshaping the memberships of nodes beyond the target, as illustrated in Fig. 1. By penalising significant alterations via the weighting factor λ, the loss function encourages minimal modifications to the graph while still attaining the desired hiding effect.

Concretely, given a fixed budget $\beta > 0$ for modifying the target node u's neighbourhood, the problem reduces to identifying the optimal function $h^* = h_{\theta^*}$, where the parameters θ^* are determined by solving the constrained objective defined in Eq. (1). Here, $\boldsymbol{b}_u$ is a $|V|$-dimensional binary vector such that $\boldsymbol{b}_u[v] = 1$ if and only if the edge (u, v) is modified by h_θ, and $\boldsymbol{b}_u[v] = 0$ otherwise. Thus, Eq. (1) can be interpreted as identifying the *counterfactual graph*

$G^* = h^*(G)$, i.e., a modified version of the original graph that masks the community membership of the target node u when input back to the community detection algorithm $f(\cdot)$.

However, in contrast to the original method [4], we adopt a more flexible strategy by not limiting the set of edges eligible for modification, i.e., any $\boldsymbol{b}_u \in \{0,1\}^{|V|}$ is theoretically admissible. Specifically, we consider all possible edges between the target node u and every other node in the graph, enabling both the addition and removal of connections. This unrestricted approach allows the optimisation process to be fully guided by the loss landscape, empowering it to identify the most impactful modifications.

Note that the constraint on $||\boldsymbol{b}_u||_0$ makes the objective non-convex, due to the inherent non-convexity of the L^0-norm itself. To overcome this challenge, rather than adopting a reinforcement learning framework as in Bernini et al. [4], we propose a novel differentiable loss function. This formulation enables the use of efficient gradient-based optimisation methods for solving the community membership hiding task. A detailed discussion on this matter follows in the next section.

5 Continuous Relaxation of the CMH Problem

The community membership hiding problem outlined in Section 4 is inherently discrete, rendering it unsuitable for direct optimisation using gradient-based techniques. To overcome this, we adopt a strategy inspired by Trappolini et al. [50] by introducing a *perturbation vector p* that is applied to the adjacency vector A_u:

$$A'_u = \text{clamp}(A_u + p), \tag{3}$$

where $p \in \{-1,0,1\}^{|V|}$. Intuitively, a value of -1 in p corresponds to removing an existing edge or leaving a non-existent edge unaltered, 0 preserves the current edge state, and 1 either adds a new edge or retains an existing one. The function $\text{clamp}(x) = \max(0, \min(x, 1))$ ensures that the elements of new adjacency vector A'_u are contained to $\{0,1\}$, mapping the set $\{-1,0,1,2\}$ to binary values.

However, since the values in p remain discrete, we first introduce a real-valued vector $\hat{p}$, whose entries are constrained to the range $[-1,1]$ using a tanh transformation. These values are then thresholded to obtain the discrete perturbation vector p, defined as:

$$p_i = \begin{cases} +1 & \text{if } \hat{p}_i \geq t^+, \\ -1 & \text{if } \hat{p}_i \leq t^-, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

A straightforward choice for the thresholds is $t^+ = 0.5$ and $t^- = -0.5$, which cleanly separates positive, negative, and neutral perturbations. Consequently, $\hat{p}$ becomes the *only* set of parameters subject to optimisation, fully governing the perturbation process.

5.1 Designing a Differentiable Loss

The optimisation operates on a vector $\hat{p}$ initialised uniformly within $[-0.5, 0.5]^{|V|}$, which corresponds to starting the process from a null perturbation state. Since we assume no internal knowledge of $f(\cdot)$, its outcomes cannot be directly incorporated into the loss to guide optimisation. To address this, we introduce a vector $\tilde{A}_u$, representing what we refer to as *promising actions* – that is, edge modifications that node u should prioritise to escape its current community. In principle, as discussed in Section 4.1, our framework is general enough to treat *any* edge between the target node u and any other node as a valid candidate for addition or removal – i.e., any $\boldsymbol{b}_u \in \{0,1\}^{|V|}$ is theoretically admissible. However, prior work has shown that certain edge modifications are more influential than others in disrupting community assignments [4, 18, 54]. Our promising actions aim to capture this intuition.

The simplest form of $\tilde{A}_u$ is given by $\neg A_u$, the complement of node u's adjacency vector. This suggests disconnecting from all current neighbours while linking to previously unconnected nodes. A more refined heuristic, which accounts for the structural properties of nodes in the graph, is presented in Section 5.2. Moreover, the flexibility of our framework allows for the integration of more sophisticated approaches, such as *learning* promising actions dynamically rather than statically defining them a priori. We leave the exploration of such adaptive strategies to future work.

Therefore, we define the first term of the loss (ℓ_{hide}) as:

$$\ell_{hide}(\hat{p}; A_u, \tilde{A}_u, q) = ||\tilde{A}_u - (A_u + \hat{p})||_q, \tag{5}$$

where $q \geq 1$. In contrast, the second component of the loss (ℓ_{dist}) is designed to discourage large perturbations, aiming to identify the minimal counterfactual graph that causes u to belong to a different community, according to $f(\cdot)$. To this end, we assess the distance between the original and intermediate adjacency vectors:

$$\ell_{dist}(\hat{p}; A_u, q) = ||A_u - (A_u + \hat{p})||_q = ||\hat{p}||_q, \tag{6}$$

where $q \geq 1$. Eventually, the objective is to determine the optimal perturbation vector p^* that balances hiding effectiveness with minimal graph modifications. This formulation leads to the following constrained optimisation problem, which can now be solved via standard gradient-based methods:

$$\begin{gathered} p^* = \arg\min_{\hat{p}} \left\{ \mathcal{L}(\hat{p}; A_u, \tilde{A}_u, q) \right\} \\ \text{subject to: } ||\boldsymbol{b}_u||_0 \leq \beta. \end{gathered} \tag{7}$$

5.2 Promising Actions

As discussed in Section 5.1, we employ $\tilde{A}_u$ as a surrogate for the output of $f(\cdot)$, which cannot be directly incorporated into the loss. Since $\tilde{A}_u$ prioritises certain promising actions, there are various ways to define it. In this work, we adopt the following heuristic. Specifically, we introduce the notion of a node's significance by assigning each node v a real-valued score $S_v \in [0,1]$. Accordingly, each entry of $\tilde{A}_u$ is defined as:

$$\tilde{A}_{u,v} = \begin{cases} (1 - S_v)/2 & \text{if } v \in C_i, \\ (1 + S_v)/2 & \text{if } v \notin C_i. \end{cases} \tag{8}$$

If a node v belongs to the same community as u ($v \in C_i$) and has a high score ($S_v \approx 1$), then $\tilde{A}_{u,v} \approx 0$, which encourages the algorithm to disconnect from it if an edge exists. Conversely, if the node lies outside C_i and also has a high score, $\tilde{A}_{u,v} \approx 1$, the algorithm is more likely to establish a connection if none exists. This reflects the aim of reducing cohesiveness within the community while strengthening connections outside of it. On the other hand, when a node has a low score ($S_v \approx 0$), $\tilde{A}_{u,v} \approx \frac{1}{2}$ in both scenarios, indicating no preference for adding or removing that connection, thus favouring no changes.

For each node, we calculate the values of K structural properties, denoted by $\Omega = \{\omega_1 \ldots, \omega_K\}$. Then, we compute a $|V|$-dimensional ranking vector $\boldsymbol{r}_i$ for each property $\omega_i \in \Omega$. Each element of this vector indicates the position of a node in the list of values for ω_i, sorted in non-decreasing order. For example, consider a set of nodes $V = \{v_1, v_2, v_3\}$ and a property ω_i, with values [42,120,5], i.e., $\omega_i(v_1) = 42$, $\omega_i(v_2) = 120$, and $\omega_i(v_3) = 5$. Sorting these values in non-decreasing order yields [5,42,120]. The ranking vector $\boldsymbol{r}_i$ assigns to each node the index (starting from 1) of its property value ω_i in the sorted list, resulting in $\boldsymbol{r}_i$ =[2,3,1], i.e., $\boldsymbol{r}_i[v_1] = 2$, as $\omega_i(v_1) = 42$ is the second element of the sorted list, and so on. Thus, we normalise the rankings as follows:

$$S_v^i = \frac{\boldsymbol{r}_i[v] - 1}{|V| - 1} \quad \forall v \in V, \forall i = 1 \dots K. \tag{9}$$

The final scores are obtained by aggregating the individual scores associated with each property, for example through a linear combination: $S_v = \sum_{i=1}^{K} a_i \, S_v^i \; \forall v \in V$, where $a_i \in [0, 1]$ and $\sum_{i=1}^{K} a_i = 1$.

In this work, we consider the following structural properties of a node: $\Omega = \{$*degree*, *betweenness centrality*, *intra/inter-community degree*$\}$. These properties are chosen to ensure consistency with the baselines methods, which also rely on them, as detailed in Section 6.1. We will explore alternative structural metrics and more advanced aggregation strategies to compute S_v in future work.

5.3 ∇- CMH

In this section, we describe our proposed method, referred to as ∇- CMH. The method is outlined in Algorithm 1, which provides an overview of its operational mechanics.

Our approach distinguishes itself from prior work [4], which performs the maximum permitted actions without verifying the outcome of $f(\cdot)$. In contrast, our technique adopts a more efficient utilisation of the available budget by dynamically recalculating the community structure after each modification to the graph. Specifically, every time A'_u is altered, we reevaluate the community structure. This recalibration allows us to potentially achieve the hiding objective *before* fully exhausting the allocated budget. Furthermore, if the method depletes the budget without achieving the hiding objective, the optimisation process is restarted to explore alternative counterfactuals. This iterative restart mechanism is regulated by a predefined maximum iteration limit, denoted as T, which ensures computational feasibility by preventing infinite loops.

Convergence Guarantee. The convergence of ∇- CMH is rooted in the principles of gradient-based optimisation. The objective function, as defined in Eq. (7), comprises two components, ℓ_{hide} and ℓ_{dist}, both involving the L^q-norm, with $q \geq 1$ (see Eqs. (5) and (6)). Therefore, both terms are convex with respect to $\hat{p}$, and their combination ensures that the first part of the objective is convex. However, the constraint on the number of modified edges for node u, expressed as $||\boldsymbol{b}_u||_0 \leq \beta$, introduces non-convexity due to the L^0-norm. This makes the overall optimisation problem NP-hard [33], necessitating a numerical approximation via stochastic gradient-based methods. From a theoretical standpoint, our method leverages the guarantees of gradient-based optimisation in non-convex settings. By using a sufficiently small learning rate (η), we can ensure convergence to a stationary point, which may correspond to either a local minimum or a saddle point. However, achieving global optimality is not guaranteed in non-convex problems. To mitigate this limitation, we employ a strategy of running the method multiple times with different random initialisations, a widely adopted approach.

Algorithm 1 ∇- CMH

```
Input: Graph $G = (V, E)$; target node $u$; community detection algorithm $f(\cdot)$; max iterations $T$; learning rate $\eta$; similarity function $sim(\cdot)$; budget $\beta$; similarity threshold $\tau$.
Output: Counterfactual graph $G'$
$\hat{p} \sim \mathcal{U}([-0.5, 0.5])^{|V|}$
$f(G) = \{C_0, \dots, C_k\}$, with $u \in C_i$
Compute $\tilde{A}_u$ as defined in Eq. (8)
$t \leftarrow 1, C'_i \leftarrow C_i, A'_u \leftarrow A_u$ , $G' \leftarrow G, \boldsymbol{b}_u^{(0)} \leftarrow \boldsymbol{0}$
while $sim(C_i \setminus \{u\}, C'_i \setminus \{u\}) > \tau$ and $t \leq T$ do
    $\hat{p} \leftarrow \tanh(\hat{p} - \eta \nabla_{\hat{p}} \mathcal{L}(\hat{p}; A_u, \tilde{A}_u, q))$
    $p \leftarrow$ threshold$(\hat{p})$
    $A'_u \leftarrow$ clamp$(A_u + p)$
    $\boldsymbol{b}_u^{(t)} \leftarrow$ changes in $A'_u$ up to step $t$
    if $||\boldsymbol{b}_u^{(t)} - \boldsymbol{b}_u^{(t-1)}||_0 > 0$ then
        Update $G'$ based on $\boldsymbol{b}_u^{(t)}$
        $f(G') \leftarrow \{C'_0, \dots, C'_r\}$, with $u \in C'_i$
    end if
    if $||\boldsymbol{b}_u^{(t)}||_0 > \beta$ then
        $\hat{p} \sim \mathcal{U}([-0.5, 0.5])^{|V|}$
    end if
    $t \leftarrow t + 1$
end while
return $G'$
```

Computational Complexity. We examine the computational complexity of our approach to determine its feasibility for deployment in large-scale production environments. In this analysis, we consider a graph with $|V| = n$ nodes and $|E| = m$ edges. The computational cost of our method primarily hinges on two key operations outside the optimisation process, as other operations reduce to simple $O(n)$ vector computations. Let $F(n, m)$ represent the cost of applying the detection algorithm $f(\cdot)$, and $\tilde{F}(n, m)$ denote the cost of constructing the vector $\tilde{A}_u$. With a maximum of T iterations for the optimisation process, the total computational complexity is:

$$O\left[n + \tilde{F}(n, m) + F(n, m) + T(n + F(n, m))\right]. \tag{10}$$

In our implementation, the construction of $\tilde{A}_u$ is dominated by betweenness centrality calculations, leading to a complexity of $\tilde{F}(n, m) = O(mn)$. For the detection algorithm $f(\cdot)$, the complexity depends on the specific community detection method employed.

5.4 ∇- CMH-Projected

The optimisation loop in Algorithm 1 does not guarantee that the counterfactual graph fully exhausts the available modification budget. In contrast, state-of-the-art methods typically enforce $||\boldsymbol{b}_u||_0 = \beta$ by applying changes until the budget is exactly met. To ensure that ∇- CMH is not overly penalised in such comparisons, we introduce a *projection step* performed after the optimisation loop, which adjusts the final perturbation to exactly match the prescribed budget. Let T^* be the iteration at which ∇- CMH terminates. We define the used budget as $\beta_{used} = ||\boldsymbol{b}_u^{(T^*)}||_0$, and the remaining budget as $\beta_{rem} = \beta - \beta_{used}$. Starting from the final perturbation parameters $\hat{p}$ and the latest discrete adjacency vector A'_u, we derive a momentum-smoothed descent direction by computing an exponentially weighted average of the stored gradients $\{g^{(t)}\}_{t=1}^{T^*}$:

$$\bar{g} = (1-\gamma)\sum_{t=1}^{T^*} \gamma^{T^*-t} g^{(t)}, \tag{11}$$

where γ is the decay factor controlling the weighting of past gradients, set to 0.9 in accordance with the literature [27]. As long as $\beta_{rem} > 0$, we update the parameters as $\hat{p} = \hat{p} - \eta\,\bar{g}$, then discretise and clamp the result to obtain a new binary adjacency vector A'_u.

Let δ denote the number of new modifications introduced in one projection step. When $0 < \delta \leq \beta_{rem}$, we apply all proposed changes to the counterfactual graph G', and update the remaining budget as $\beta_{rem} = \beta_{rem} - \delta$. Otherwise, if $\delta > \beta_{rem}$, we score each candidate modification involving node v (the node to be attached to or detached from u) by $s_v = |\hat{p}_v| \cdot |\bar{g}_v|$, favouring changes with strong gradient support. We then sort candidates in descending order of s_v and apply the top β_{rem} changes. This procedure guarantees the entire budget is exhausted ($\|\boldsymbol{b}_u\|_0 = \beta$), while keeping the final perturbation aligned with the optimiser's search direction. We refer to this variant of our method as ∇- CMH-P.

6 Experiments

6.1 Experimental Setup

Datasets. We evaluate our method on a diverse collection of real-world undirected graphs, encompassing a range of domains and data types. These include social and human interaction networks (`kar`,[2] Wikipedia's `vote`,[3] and Facebook `fb-75`[3]), information networks (`words`[2] and `arxiv`[4] Condensed Matter), and infrastructure networks (US Power Grid `pow`[2]).

Community Detection Algorithms. We consider four community detection algorithms: two modularity-based approaches – *greedy* [8] and *leiden* [49]; the *walktrap* algorithm [37], which relies on random walks; and *dgcluster* [5], a deep learning-based method that performs clustering using node attributes – in our case, embeddings are generated with *node2vec* [22].
Table 1 summarises the datasets used in our evaluation, including their size and the number of communities per algorithm.

Similarity Metric. To determine the success of obscuring community memberships, we use Sørensen-Dice coefficient [17] as the similarity function $sim(\cdot,\cdot)$ in Algorithm 1. This metric measures similarity between two sets, ranging from 0 (no similarity) to 1 (high similarity). The objective is achieved if $sim(C_i \setminus \{u\}, C'_i \setminus \{u\}) \leq \tau$.

Baselines. We compare the hiding assessment of our method, namely ∇- CMH, against six baseline approaches:

(1) *DRL-Agent*. A deep reinforcement learning method [4].
(2) *DICE*. This heuristic, originally proposed for hiding communities, is based on the principle of *Disconnect Internally, Connect Externally* [54]. We adapt it to our setting by removing one edge of u within its current community, specifically targeting the highest-degree node, and then creating $\beta - 1$ external connection from u with the same logic.
(3) *ROAM*. This approach builds on the Roam heuristic [54], which is originally developed to reduce a node's centrality in a graph. It works by removing the connection to the highest-degree neighbour v_0, and then adding up to $\beta - 1$ new edges from v_0 to neighbours of u that are not already connected to v_0, selecting them based on high-degree.
(4) *Random-based*. This method picks a random node from V, and either adds or remove the connection to it.
(5) *Degree-based*. It modifies an edge that involves the highest-degree node, either by removing or adding the link.
(6) *Centrality-based*. This approach alters the edge connected to the node having the highest betweenness centrality [20].

Table 1: Properties of the graph datasets considered in this work, including the number of communities identified by *greedy, leiden, walktrap,* and *dgcluster.*

Dataset	$\|V\|$	$\|E\|$	Number of Communities			
			greedy	*leiden*	*walktrap*	*dgcluster*
`kar`	34	78	3	4	5	2
`words`	112	425	7	7	25	4
`vote`	889	2,900	12	8	42	5
`pow`	4,941	6,594	41	38	364	664
`fb-75`	6,386	217,662	16	13	349	9
`arxiv`	23,133	93,497	270	56	2306	1322

Evaluation Metrics. We evaluate the effectiveness of each method in achieving the node hiding goal using the following metrics.

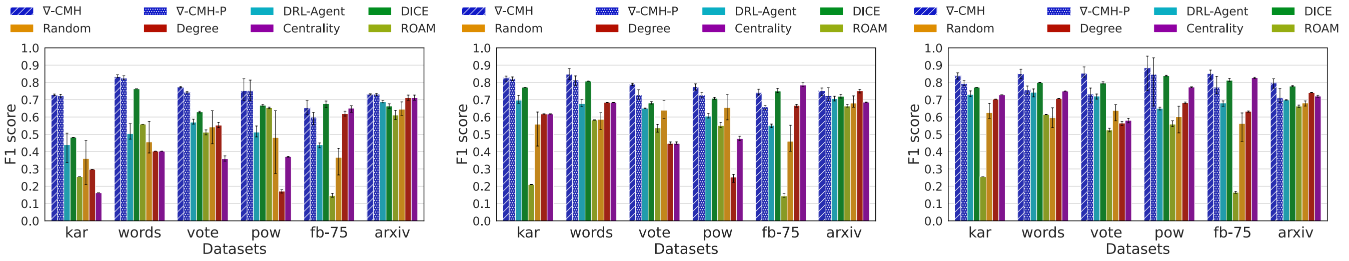
(1) *Success Rate* (SR). It measures the percentage of cases in which the target node u is successfully hidden from its original community, i.e., when $sim(C_i \setminus \{u\}, C'_i \setminus \{u\}) \leq \tau$. Higher values indicate better performance.
(2) *Normalised Mutual Information* (NMI). To assess the impact of the counterfactual graph G' on the resulting community structure $f(G')$, we compute the NMI score [28, 45] between $f(G')$ and the original structure $f(G)$. Higher values indicate greater similarity, and thus a lower cost.

In general, SR and NMI are inherently contrasting metrics – higher SR often corresponds to lower NMI, and vice versa. To evaluate the trade-off between them, we compute their harmonic mean (i.e., F1 score) using the following formula: $\frac{2\times\text{SR}\times\text{NMI}}{\text{SR}+\text{NMI}}$.

∇- CMH/∇- CMH-P. We use Adam optimiser to solve the objective of Eq. (7), setting $q = 2$ and leaving the exploration of alternative norms to future work. We perform a Bayesian hyperparameter search for λ, T, η, and $\{a_i\}_{i=1}^{K}$ to maximise the F1 score between SR and NMI. Further details are reported in Section 6.3.

Evaluation Protocol. We analyse ∇- CMH/∇- CMH-P under various parameter configurations. Specifically, we vary the similarity constraint τ with values in $\{0.3, 0.5, 0.8\}$, and test across different fixed budget values $\beta \in \{\mu/2, \mu, 2\mu\}$, where $\mu = |E|/|V|$ represents the average node degree.[5] To ensure diversity in the evaluation,

[2] http://konect.cc/
[3] https://networkrepository.com
[4] https://snap.stanford.edu/data/
[5] For `kar` and `pow`, we specifically use $\mu = \frac{|E|}{|V|} + 1$.



(a) $\beta = \mu/2$.

(b) $\beta = \mu$.

(c) $\beta = 2\mu$.

Figure 2: F1 score between SR and NMI in the symmetric setting ($f(\cdot)$: *greedy*) when $\tau = 0.5$ across different budgets β.

we select 3 communities of varying sizes – approximately 30%, 50%, and 80% of the largest community. Within each, we randomly sample up to 100 nodes as targets. This sampling strategy enables broad coverage across different community types. Experiments are conducted under two distinct setups: *symmetric* and *asymmetric*. In the symmetric setup, the same community detection algorithm $f(\cdot)$ is used for both the optimisation and evaluation. In contrast, the asymmetric setup evaluates our method using a detection algorithm $g(\cdot)$, which differs from the optimisation algorithm $f(\cdot)$. This design allows us to assess the *transferability* of our method – that is, its ability to generalise to detection algorithms it was not explicitly optimised against. Specifically, we use *greedy* as $f(\cdot)$ to ensure consistency with DRL-Agent's training setup. In the asymmetric setting, we use one of *leiden*, *walktrap*, or *dgcluster* as $g(\cdot)$. To account for the inherent stochasticity in the detection algorithms, hiding methods, and sampling process, results are presented as the mean ± standard deviation computed over three independent runs.

6.2 Results and Key Findings

We evaluate ∇- CMH/∇- CMH-P along three key axes. First, we assess its effectiveness in hiding target nodes from their communities, comparing against all baselines. Second, we evaluate its transferability in an asymmetric setting, where $f(\cdot)$ differs from the actual community detection algorithm in use. Third, we analyse its computational efficiency relative to its main competitor, DRL-Agent.

Hiding Assessment (Symmetric). In Fig. 2, we compare the performance of ∇- CMH with selected baselines in the symmetric setting. Specifically, we report the aggregated quality score – computed as the F1 score between SR and NMI – across varying budgets.

Our method consistently outperforms all baseline approaches, except for a specific case on the `fb-75` dataset when $\beta = \mu/2$. We attribute this to the structural characteristics of the graph, where heuristic methods such as Centrality or DICE tend to perform particularly well. Indeed, these approaches prioritise modifying edges connected to highly influential nodes, as discussed in Section 6.4.

Moreover, it is worth noting that, unlike baseline methods that fully exhaust their allocated budgets, ∇- CMH employs its resources more strategically and efficiently. As shown in Table 2, which summarises budget usage across all datasets, our method achieves strong performance without utilising the entire budget. In fact, when baseline methods are constrained to the same budget used by ∇- CMH, the performance gap becomes even more pronounced. This highlights the ability of our method to balance effectiveness with resource efficiency, achieving robust results at a lower cost. A similar trend holds across different values of τ.

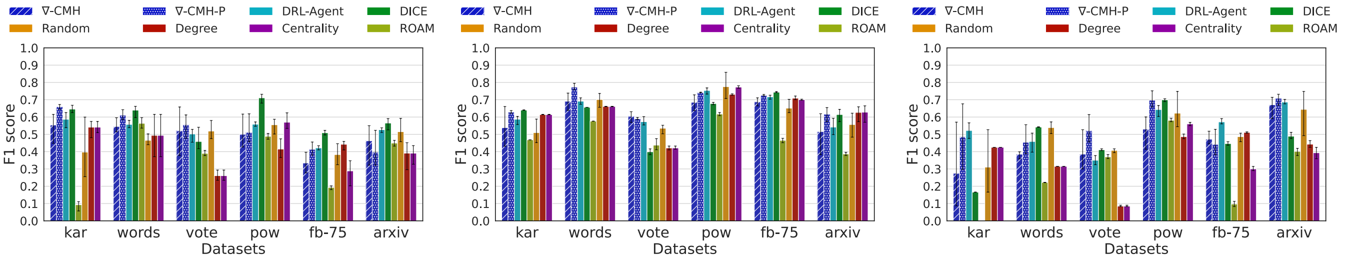
Table 2: Average budget usage (absolute/%) by ∇- CMH across all datasets, fixed budget settings ($\beta \in \{\mu/2, \mu, 2\mu\}$), and $\tau = 0.5$.

β	Dataset					
	`kar`	`words`	`vote`	`pow`	`fb-75`	`arxiv`
$\mu/2$	1.0 / 100	1.0 / 100	1.0 / 100	1.0 / 100	11.3 / 66.3	1.7 / 87.2
μ	2.3 / 76.7	1.9 / 63.0	2.2 / 71.7	1.8 / 90.8	21.5 / 63.3	3.1 / 78.2
2μ	3.8 / 63.5	3.4 / 56.2	3.6 / 60.8	2.9 / 71.5	39.9 / 58.7	5.2 / 65.4

Interestingly, the variant of our method that is forced to fully exhaust the available budget (∇- CMH-P) offers no performance gain. In fact, its additional modifications tend to degrade results, likely because these extra graph rewiring operations are redundant. Their associated cost outweighs any marginal benefit beyond what is already achieved by the more selective modifications of ∇- CMH.

Transferability (Asymmetric). In Fig. 3, we present the results for the asymmetric setting, which assesses each method's ability to generalise across different community detection algorithms. Our ∇- CMH demonstrates transferability comparable to DRL-Agent, but generally lower than that of heuristic-based methods. This limitation is particularly evident on graphs such as `fb-75`, and remains consistent across different budget configurations.

We conjecture that this gap arises from a tendency of ∇- CMH to "overfit" the specific community detection algorithm used during optimisation – a limitation that heuristic-based methods inherently avoid due to their algorithm-agnostic nature. Notably, the variant ∇- CMH-P, which is forced to exhaust the entire budget, suffers less from this overfitting tendency. It generally outperforms ∇- CMH and DRL-Agent and, in some cases, even surpasses heuristic-based methods. We attribute this to its ability to explore a broader range of graph rewiring operations, which may help it uncover patterns that transfer more effectively to unseen algorithms, thereby improving generalisation. Addressing this challenge opens a path for future improvements. In particular, reducing overfitting in asymmetric scenarios could involve optimising over multiple community detection algorithms rather than just a single one.



(a) $g(\cdot)$: ***leiden.*** **(b)** $g(\cdot)$: ***walktrap.*** **(c)** $g(\cdot)$: ***dgcluster.***

Figure 3: F1 score between SR and NMI in the asymmetric settings ($f(\cdot)$: *greedy*) when $\tau = 0.5$ and $\beta = \mu$.

Computational Efficiency.[6] Lastly, our method exhibits notable computational efficiency, operating faster than DRL-Agent. This speed advantage further enhances its practical applicability in real-world scenarios. Table 3 illustrates the average hiding time across datasets in the symmetric setting with $\tau = 0.5$ and $\beta = \mu$, clearly showing that ∇- CMH consistently outperforms the agent.

Table 3: Average hiding time (secs.) of ∇- CMH compared to DRL-Agent using $f(\cdot)$: *greedy*, with $\tau = 0.5$ and $\beta = \mu$.

Dataset	Algorithm		Time Speed-up
	∇- CMH (ours)	DRL-Agent	
kar	0.013	0.027	×2.050 ▲
words	0.016	0.047	×2.932 ▲
vote	0.063	0.166	×2.643 ▲
pow	0.115	0.213	×1.851 ▲
fb-75	22.169	110.463	×4.983 ▲
arxiv	2.980	5.227	×1.754 ▲

To summarise, our experiments yield three key insights. First, in the symmetric setting, ∇- CMH consistently outperforms all baselines. Fully exhausting the budget with ∇- CMH-P offers no additional benefit, as ∇- CMH already identifies the minimal set of graph rewiring operations required to achieve the hiding goal. Second, learning-based methods such as ∇- CMH and DRL-Agent exhibit lower transferability compared to heuristic-based techniques, which are inherently agnostic to the specific community detection algorithm. Notably, ∇- CMH-P alleviates this limitation by exploring a broader range of graph modifications. Third, ∇- CMH is significantly more efficient, achieving much faster runtimes than the other learning-based competitor, DRL-Agent.

6.3 Parameter and Hyperparameter Analysis

Our method depends on two distinct sets of critical factors:

- *method parameters*, which define the operational constraints of the CMH problem: the similarity threshold τ and the modification budget β.
- *hyperparameters*, which control the optimisation process of ∇- CMH: the learning rate η, the regularisation strength λ, the number of maximum iterations T, and the coefficients associated with the $K = 4$ promising actions, denoted as $\{a_i\}_{i=1}^{K}$. Specifically, ω_1 (betweenness centrality) is associated with a_1, ω_2 (degree) with a_2, ω_3 (intra-community degree) with a_3, and ω_4 (inter-community degree) with a_4.

Hyperparameters are tuned to maximise performance on a validation set, while method parameters are set according to the target use-case and later analysed to assess their effect on behaviour. For clarity, we first report the optimisation of hyperparameters and then examine sensitivity to the method parameters.

Hyperparameter Optimisation. Table 4 lists the hyperparameters selected via Bayesian optimisation for the setting $\beta = \mu$, with the objective of maximising the F1 score between SR and NMI across various datasets.

Table 4: Hyperparameters of our method across all datasets for $\beta = \mu$ (a_1: betweenness centrality; a_2: degree; a_3: intra-community degree; a_4: inter-community degree).

Dataset	Hyperparameters						
	η	λ	T	a_1	a_2	a_3	a_4
kar	0.079	1.71	120	0.33	0.20	0.21	0.24
words	0.006	0.04	110	0.16	0.26	0.34	0.22
vote	0.017	0.37	140	0.48	0.25	0.01	0.24
pow	0.008	18.1	130	0.05	0.17	0.41	0.35
fb-75	0.004	0.15	140	0.29	0.59	0.09	0.01
arxiv	0.001	17.2	140	0.40	0.21	0.05	0.32

Sensitivity to Method Parameters. In Table 5, we illustrate how variations in τ and β affect the F1 score of SR and NMI in the community membership hiding task, comparing our method against the three best baselines. The reported results correspond to the symmetric setting on the vote dataset. As expected, increasing the threshold simplifies the achievement of the concealment goal, raising the F1 score for a fixed budget by imposing less strict requirements on membership hiding. In contrast, increasing the budget for a fixed threshold does not necessarily improve performance, as it permits more substantial modifications to the neighbourhood, which can further reduce the NMI score.

6.4 The Importance of Modified Edges

So far, we have treated every edge modification uniformly, namely, adding or removing an edge has the same "cost" regardless of the

[6] All experiments were conducted on a GPU NVIDIA GeForce RTX 4090 and an AMD Ryzen 9 7900 12-Core CPU.

Table 5: Impact of τ and β on the F1 score between SR and NMI in the symmetric setting ($f(\cdot)$: *greedy*) on the vote dataset for ∇- CMH, DRL-Agent, DICE, and Random. Bold indicates the best (highest) value; underlined values are second best.

τ	β	Algorithm			
		∇- CMH (ours)	DRL-Agent	DICE	Random
0.3	$\mu/2$	$\mathbf{0.72 \pm 0.01}$	0.45 ± 0.01	$\underline{0.62 \pm 0.01}$	0.42 ± 0.02
	μ	$\mathbf{0.77 \pm 0.02}$	0.56 ± 0.01	$\underline{0.65 \pm 0.01}$	0.57 ± 0.05
	2μ	$\mathbf{0.85 \pm 0.03}$	0.64 ± 0.01	$\underline{0.71 \pm 0.01}$	0.53 ± 0.01
0.5	$\mu/2$	$\mathbf{0.77 \pm 0.00}$	0.57 ± 0.01	$\underline{0.63 \pm 0.00}$	0.54 ± 0.08
	μ	$\mathbf{0.79 \pm 0.01}$	0.65 ± 0.00	$\underline{0.68 \pm 0.01}$	0.64 ± 0.04
	2μ	$\mathbf{0.85 \pm 0.03}$	0.72 ± 0.01	$\underline{0.80 \pm 0.01}$	0.64 ± 0.04
0.8	$\mu/2$	$\mathbf{0.83 \pm 0.01}$	$\underline{0.75 \pm 0.02}$	$\underline{0.75 \pm 0.01}$	0.71 ± 0.04
	μ	$\mathbf{0.86 \pm 0.01}$	0.80 ± 0.01	$\underline{0.82 \pm 0.00}$	0.80 ± 0.00
	2μ	$\mathbf{0.90 \pm 0.04}$	0.78 ± 0.01	$\underline{0.82 \pm 0.00}$	0.76 ± 0.01

other node involved in the modification. However, this assumption often does not hold in practice. For example, a user attempting to hide from a community may hesitate to remove or add a connection with a highly influential individual, while they may have no issue modifying connections with a less prominent one.

Heuristic-based methods like Centrality or DICE, though demonstrating strong transferability performance, achieve this by systematically prioritising modifications that affect edges connected to high-degree (i.e., important) nodes, by design. This intrinsic bias can limit their practicality in real-world settings, where achieving the hiding goal should discourage from altering connections to key nodes. In contrast, ∇- CMH avoids this limitation by not restricting its modifications to edges involving high-importance nodes, making it more suitable for realistic scenarios.

To further validate this claim, we compute the average PageRank score [36] of the nodes involved in the edge modifications performed by each method, across all datasets and target nodes considered. In Table 6, we report the result of this experiment. We may observe that ∇- CMH modifies connections with nodes having the smallest average PageRank, i.e., less important nodes. On the other hand, methods like DICE explicitly focus on influential nodes to achieve the hiding goal. This inherently biased behaviour highlights a fundamental limitation of heuristic-based techniques in practical applications, which ∇- CMH overcomes.

Table 6: Average PageRank of the nodes involved in the perturbation, as computed by our method (∇- CMH) and the baselines (DICE and Centrality), for $\beta = \mu$. Bold indicates the best (lowest) value; underlined values are second best. Differences are all statistically significant ($\alpha = 0.01$).

Algorithm	Dataset					
	kar	words	vote	pow	fb-75	arxiv
∇- CMH	$\mathbf{4.5 \cdot 10^{-2}}$	$\mathbf{9.2 \cdot 10^{-3}}$	$\mathbf{1.2 \cdot 10^{-3}}$	$\mathbf{2.9 \cdot 10^{-4}}$	$\mathbf{1.6 \cdot 10^{-4}}$	$\mathbf{1.4 \cdot 10^{-4}}$
DICE	$\underline{7.8 \cdot 10^{-2}}$	$\underline{3.1 \cdot 10^{-2}}$	$\underline{9.3 \cdot 10^{-3}}$	$7.3 \cdot 10^{-4}$	$7.8 \cdot 10^{-4}$	$\underline{7.1 \cdot 10^{-4}}$
Centrality	$8.9 \cdot 10^{-2}$	$3.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$\underline{3.9 \cdot 10^{-4}}$	$\underline{7.5 \cdot 10^{-4}}$	$7.9 \cdot 10^{-4}$

7 Current Limitations and Future Work

Like any approach tackling a new problem, ∇- CMH has certain limitations that open avenues for future work.

Scalability. While we have evaluated the scalability of ∇- CMH on relatively large graphs such as fb-75 and arxiv, and demonstrated its efficiency compared to its closest competitor (DRL-Agent), its applicability to truly large-scale graphs – i.e., on the order of millions or billions of nodes – though promising, needs further validation.

Hiding Goal. The notion of being "masked" from a community is operationalised in this work as a binary event, determined by whether the similarity between the original community and the one assigned after perturbation falls below a given threshold τ. This abstraction, while practical, may not fully capture more nuanced or context-dependent definitions of community affiliation.

Side Effects. Solving the community membership hiding task for a single node by modifying its local neighbourhood can unintentionally impact other nodes. For example, obscuring a node from its original community could reassign neighbouring nodes or alter the structure of nearby communities. In practice, we recommend post hoc validation to identify such side effects *before* deployment. Nonetheless, we consider this assumption reasonable, since ∇- CMH is primarily intended for use by entities controlling the graph network (e.g., online platforms), rather than individual users.

Multiple Nodes/Multiple Communities. Our current formulation assumes a single target node requesting to be masked from a single community. In more realistic settings, multiple nodes may simultaneously seek to obscure their affiliations, possibly across several different communities. This scenario introduces new challenges, such as coordinating perturbations and mitigating compounded side effects, and remains an important direction for future work.

8 Conclusion

We presented ∇- CMH, a counterfactual graph generator designed to solve the community membership hiding task through gradient-based optimisation. This method employs a perturbation vector, added element-wise to the adjacency vector of the target node to mask. To ensure differentiability, we define an intermediate real-valued perturbation vector and a loss function that encourages minimal changes to the graph under consideration. Graph modifications are further guided by a vector of *promising actions*, namely the set of allowed operations to successfully escape the community.

Experimental results demonstrate the superiority of ∇- CMH, which consistently outperforms existing approaches in concealing target nodes. Moreover, our method efficiently identifies solutions that prioritise less disruptive graph rewiring operations, achieving strong performance without exhausting the allocated budget. Finally, we introduced a variant, ∇- CMH-P, which is forced to consume the entire budget. This variant improves transferability across different community detection algorithms, highlighting the potential for further enhancements in asymmetric scenarios.

Ethical Considerations

The *community membership hiding* problem arises whenever a node in a generic graph must be concealed from a specific cluster of nodes. While broadly applicable in several domains, its most immediate and impactful use lies in safeguarding user privacy on online platforms, such as social networks.

In this section, we examine the potential ethical implications of community membership hiding techniques, using our method (∇- CMH) as a representative example. We assume that the platform owner (e.g., Meta, X, TikTok) offers ∇- CMH as a service and executes it on behalf of users seeking to hide from a specific community (*platform-mediated*). We do not address scenarios in which the end user performs this hiding independently (*user-initiated*), as such cases involve a relaxed assumption about graph knowledge (i.e., shifting from global to local).

Although primarily designed to enhance individual privacy, ∇- CMH possesses an inherent dual-use nature, making it essential to carefully weigh its potential benefits against its associated risks.

Promoting Privacy and Autonomy. At its core, community membership hiding aligns with fundamental privacy principles, including the "right to be forgotten" [9, 42], as recognised in regulations such as the European Union's GDPR [51] and AI Act [2]. These frameworks emphasise individuals' control over their digital footprint and the prevention of unwanted inferences about sensitive personal information.

As community detection algorithms can inadvertently reveal political beliefs, health conditions, or affiliations with vulnerable groups, ∇- CMH offers a crucial mechanism for digital autonomy. It provides a flexible and scalable alternative to the drastic measure of leaving online platforms entirely, allowing users to manage their visibility while retaining their online presence. For individuals in sensitive professions, such as journalists or human rights activists operating in authoritarian regimes, the ability to obscure their network affiliations could be vital for their safety and the integrity of their work. Similarly, it could combat online criminal activities by disrupting communication patterns among malicious users, though this application also carries its own set of ethical issues.

Facing the Risks of Misuse. While designed to enhance privacy, ∇- CMH also has significant misuse potential. In the hands of malicious actors, this technique could obstruct legitimate graph analysis tools or security efforts by masking illicit or criminal activities within the network. Possible scenarios include:

- *Organised Crime*: Criminal groups obscuring their leadership structures and communication hubs to hinder law enforcement detection.
- *Disinformation Campaigns*: State-sponsored or coordinated actors masking bot networks or inauthentic communities to evade platform moderation.
- *Terrorist Recruitment*: Extremist organisations hiding recruitment pathways and coordination channels from intelligence agencies.

This dual-use reality creates an inherent tension: while legitimate platforms might offer ∇- CMH as a privacy-enhancing feature, malicious actors could exploit tools like that for illicit purposes. If widely available without safeguards, such tools could trigger an "arms race" in network obfuscation.

Comprehensive Stakeholder Analysis. A truly responsible ethical framework requires considering the diverse impacts on all affected stakeholders:

- *Target Users*: Gain privacy and control over their digital identity, safeguarding them from discrimination or harm.
- *Non-Target Users/Communities*: May experience unintended "side effects," where the modifications made to hide a target node inadvertently alter the community assignments or perceived affiliations of other, non-target individuals. This could lead to privacy breaches or mischaracterisations.
- *Online Platforms*: Bear the responsibility of balancing user privacy with platform integrity and public safety. They must navigate complex legal and ethical landscapes, ensuring compliance with regulations while preventing misuse.
- *Law Enforcement and Intelligence Agencies*: Face significant challenges if ∇- CMH impedes their ability to investigate crimes, counter terrorism, or maintain public safety. This raises ethical dilemmas regarding the boundaries of individual privacy versus collective security.
- *Policy Makers and Regulators*: Are tasked with developing legal and ethical frameworks that balance competing rights and interests in the digital space, potentially requiring new legislation to address the complexities of network privacy.
- *Researchers and Developers*: Hold an ethical responsibility to consider the societal impact of their innovations. This includes responsible disclosure of potential vulnerabilities and contributing to the development of ethical guidelines for research in this sensitive area.
- *Victims of Illicit Activities*: Could suffer direct and indirect harms if ∇- CMH facilitates criminal or harmful online behavior, underscoring the societal cost of unchecked misuse.

Mitigation Strategies and Safeguards. Given these complex considerations, platforms offering community membership hiding capabilities must implement robust mitigation strategies and safeguards:

- *Access Controls and Permissions*: Hiding capabilities should not be granted indiscriminately. Platforms could implement tiered access based on factors such as user verification, content type, or compliance with strict terms of service. Each request to apply ∇- CMH should be evaluated for its potential impact before being satisfied, possibly denying those associated with high-risk activities or likely misuse.
- *Transparency Mechanisms*: While avoiding the revelation of exploitable vulnerabilities, platforms should be transparent about the general principles of ∇- CMH implementation and usage. This could involve publishing aggregate statistics on hiding requests or clear, accessible user policies.
- *Auditing and Monitoring*: Robust technical and organisational mechanisms are essential to detect and prevent the misuse of ∇- CMH. This might include anomaly detection systems that flag unusual patterns of graph modifications or post-hoc analysis to identify potential illicit activities.
- *Ethical Guidelines and Policies*: Platforms should develop internal ethical guidelines and comprehensive policies for the responsible deployment and governance of ∇- CMH features, involving interdisciplinary teams (e.g., ethicists, legal experts, engineers).

- *Collaboration with Law Enforcement*: Establishing clear, ethically sound protocols for collaboration with law enforcement agencies is crucial. This involves defining the conditions under which information might be shared, respecting legal due process, and protecting user rights.

The "Right to Hide" vs. "Right to Know" Trade-off. The core ethical tension inherent in ∇-CMH lies in the conflict between an individual's "right to hide" their affiliations for privacy and society's "right to know" for public safety, law enforcement, and even legitimate academic research into social dynamics. This is not merely a problem of "misuse" but a fundamental clash of legitimate interests. When these two rights conflict, society must engage in a dialogue to establish clear boundaries, mechanisms for arbitration, and potentially new legal frameworks that address situations where individual privacy and collective security needs diverge.

Alignment with Ethical AI Principles. ∇-CMH's design also aligns with broader ethical AI principles:

- *Fairness*: Ensuring that the ability to hide community membership is equitably accessible and does not inadvertently create unfair advantages or disadvantages for certain users or groups.
- *Accountability*: Establishing clear lines of responsibility for the design, deployment, and monitoring of ∇-CMH systems, ensuring that mechanisms are in place for redress in cases of misuse or unintended harm.
- *Transparency*: The gradient-based nature of ∇-CMH offers a degree of interpretability, allowing for insights into why specific modifications are chosen. This intrinsic transparency is crucial for building user trust, enabling auditing, and distinguishing legitimate privacy-preserving actions from malicious obfuscation.
- *Privacy by Design*: ∇-CMH embodies the principle of "privacy by design," integrating privacy considerations into the system's architecture from the outset, rather than as an afterthought.

In conclusion, while ∇-CMH represents a significant technical advancement in social graph privacy, its deployment requires a profound and ongoing ethical commitment. Future research must not only advance the technical capabilities of such methods but also actively engage with the complex societal implications, fostering interdisciplinary dialogue to ensure responsible innovation that truly serves the public good.

References

[1] Sabrine Ben Abdrabbah, Raouia Ayachi, and Nahla Ben Amor. 2014. Collaborative filtering based on dynamic community detection. In *Proceedings of the 2nd International Conference on Dynamic Networks and Knowledge Discovery - Volume 1229* (Nancy, France) *(DYNAK'14)*. CEUR-WS.org, Aachen, DEU, 85–106.

[2] EU Artificial Intelligence Act. 2024. The EU Artificial Intelligence Act. https://artificialintelligenceact.eu/the-act/

[3] Edo M Airoldi, David Blei, Stephen Fienberg, and Eric Xing. 2008. Mixed Membership Stochastic Blockmodels. In *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.), Vol. 21. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2008/file/8613985ec49eb8f757ae6439e879bb2a-Paper.pdf

[4] Andrea Bernini, Fabrizio Silvestri, and Gabriele Tolomei. 2024. Evading Community Detection via Counterfactual Neighborhood Search. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 131–140. doi:10.1145/3637528.3671896

[5] Aritra Bhowmick, Mert Kosan, Zexi Huang, Ambuj Singh, and Sourav Medya. 2024. DGCLUSTER: A Neural Framework for Attributed Graph Clustering via Modularity Maximization. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 10 (2024), 11069–11077. doi:10.1609/aaai.v38i10.28983

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008. http://stacks.iop.org/1742-5468/2008/i=10/a=P10008

[7] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (Oct 2008), P10008. doi:10.1088/1742-5468/2008/10/P10008

[8] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Görke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. 2008. On Modularity Clustering. *IEEE Transactions on Knowledge and Data Engineering* 20 (2008), 172–188. https://api.semanticscholar.org/CorpusID:150684

[9] De Terwangne C. 2013. *The Right to be Forgotten and the Informational Autonomy in the Digital Environment.* Scientific analysis or review LB-NA-26434-EN-N. Luxembourg (Luxembourg). doi:10.2788/54562

[10] Alina Campan, Yasmeen Alufaisan, and Traian Marius Truta. 2015. Preserving Communities in Anonymized Social Networks. *Transactions on Data Privacy* 8, 1 (Dec 2015), 55–87.

[11] Jinyin Chen, Lihong Chen, Yixian Chen, Minghao Zhao, Shanqing Yu, Qi Xuan, and Xiaoniu Yang. 2019. GA-based Q-attack on community detection. *IEEE Transactions on Computational Social Systems* 6, 3 (2019), 491–503.

[12] Jinyin Chen, Xiang Lin, Ziqiang Shi, and Yi Liu. 2020. Link Prediction Adversarial Attack Via Iterative Gradient Attack. *IEEE Transactions on Computational Social Systems* 7, 4 (2020), 1081–1094. doi:10.1109/TCSS.2020.3004059

[13] Yizheng Chen, Yacin Nadji, Athanasios Kountouras, Fabian Monrose, Roberto Perdisci, Manos Antonakakis, and Nikolaos Vasiloglou. 2017. Practical Attacks Against Graph-based Clustering. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, 1125–1142. doi:10.1145/3133956.3134083

[14] Carlos Garcia Cordero, Emmanouil Vasilomanolakis, Max Mühlhäuser, and Mathias Fischer. 2015. Community-Based Collaborative Intrusion Detection. In *Security and Privacy in Communication Networks*, Bhavani Thuraisingham, XiaoFeng Wang, and Vinod Yegneswaran (Eds.). Springer International Publishing, Cham, 665–681.

[15] Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. 2024. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. *Machine Intelligence Research* 21, 6 (2024), 1011–1061. doi:10.1007/s11633-024-1510-8

[16] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial Attack on Graph Structured Data. arXiv:1806.02371 [cs.LG] https://arxiv.org/abs/1806.02371

[17] Lee R. Dice. 1945. Measures of the Amount of Ecologic Association Between Species. *Ecology* 26, 3 (1945), 297–302. doi:10.2307/1932409 arXiv:https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.2307/1932409

[18] Valeria Fionda and Giuseppe Pirrò. 2018. Community Deception or: How to Stop Fearing Community Detection Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 30, 4 (2018), 660–673. doi:10.1109/TKDE.2017.2776133

[19] Santo Fortunato. 2010. Community Detection in Graphs. *Physics Reports* 486, 3 (2010), 75–174. doi:10.1016/j.physrep.2009.11.002

[20] Linton C. Freeman. 1977. A Set of Measures of Centrality Based on Betweenness. *Sociometry* 40, 1 (1977), 35–41. http://www.jstor.org/stable/3033543

[21] M. Girvan and M. E. J. Newman. 2002. Community Structure in Social and Biological Networks. *Proceedings of the National Academy of Sciences* 99, 12 (2002), 7821–7826. doi:10.1073/pnas.122653799 arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.122653799

[22] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[23] Muhammad Aqib Javed, Muhammad Shahzad Younis, Siddique Latif, Junaid Qadir, and Adeel Baig. 2018. Community detection in networks: A multidisciplinary review. *J. Netw. Comput. Appl.* 108, C (April 2018), 87–111. doi:10.1016/j.jnca.2018.02.011

[24] Di Jin, Zhizhi Yu, Pengfei Jiao, Shirui Pan, Dongxiao He, Jia Wu, Philip S. Yu, and Weixiong Zhang. 2021. A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning. arXiv:2101.01669 [cs.SI]

[25] Wei Jin, Yaxing Li, Han Xu, Yiqi Wang, Shuiwang Ji, Charu Aggarwal, and Jiliang Tang. 2021. Adversarial Attacks and Defenses on Graphs. *SIGKDD Explor. Newsl.* 22, 2 (Jan. 2021), 19–34. doi:10.1145/3447556.3447566

[26] Arzum Karataş and Serap Şahin. 2018. Application Areas of Community Detection: A Review. In *Proceedings of the International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*. 65–70. doi:10.1109/IBIGDELFT.2018.8625349

[27] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization.. In *ICLR (Poster)*, Yoshua Bengio and Yann LeCun (Eds.). http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14

[28] J. Kreer. 1957. A Question of Terminology. *IRE Transactions on Information Theory* 3, 3 (1957), 208–208. doi:10.1109/TIT.1957.1057418

[29] Jia Li, Honglei Zhang, Zhichao Han, Yu Rong, Hong Cheng, and Junzhou Huang. 2020. Adversarial Attack on Community Detection by Hiding Individuals. In *Proceedings of The Web Conference 2020 (WWW '20)*. ACM. doi:10.1145/3366423.3380171

[30] Wanyu Lin, Shengxiang Ji, and Baochun Li. 2020. Adversarial Attacks on Link Prediction Algorithms Based on Graph Neural Networks. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security* (Taipei, Taiwan) *(ASIA CCS '20)*. Association for Computing Machinery, New York, NY, USA, 370–380. doi:10.1145/3320269.3384750

[31] Ana Lucic, Maartje A. ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. 2022. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event (Proceedings of Machine Learning Research, Vol. 151)*, Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (Eds.). PMLR, 4499–4511. https://proceedings.mlr.press/v151/lucic22a.html

[32] Mohammad Javad Mosadegh and Mehdi Behboudi. 2011. Using Social Network Paradigm for Developing a Conceptual Framework in CRM. *Australian Journal of Business and Management Research* 1, 4 (2011), 63.

[33] A. S. Nemirovskiĭ and D. B. Yudin. 1983. *Problem Complexity and Method Efficiency in Optimization*. Wiley. https://books.google.it/books?id=6ULvAAAAMAAJ

[34] Mark E. J. Newman. 2006. Finding Community Structure in Networks Using the Eigenvectors of Matrices. *Physical Review E* 74 (2006), 036104. doi:10.1103/PhysRevE.74.036104

[35] Mark E. J. Newman. 2006. Modularity and Community Structure in Networks. *Proceedings of the National Academy of Sciences* 103, 23 (2006), 8577–8582. doi:10.1073/pnas.0601602103

[36] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab. http://ilpubs.stanford.edu:8090/422/ Previous number = SIDL-WP-1999-0120.

[37] Pascal Pons and Matthieu Latapy. 2005. Computing Communities in Large Networks Using Random Walks. (2005), 284–293.

[38] Meng Qin, Chaorui Zhang, Yu Gao, Weixi Zhang, and Dit-Yan Yeung. 2024. Pre-train and refine: Towards higher efficiency in k-agnostic community detection without quality degradation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2467–2478.

[39] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks. *Physical Review E* 76 (Sep 2007), 036106. doi:10.1103/PhysRevE.76.036106

[40] Jörg Reichardt and Stefan Bornholdt. 2006. Statistical Mechanics of Community Detection. *Physical Review E* 74 (2006), 016110. doi:10.1103/PhysRevE.74.016110

[41] Peter Ronhovde and Zohar Nussinov. 2009. Multiresolution Community Detection for Megascale Networks by Information-Based Replica Correlations. *Physical Review E* 80, 1 (Jul 2009). doi:10.1103/physreve.80.016109

[42] Jeffrey Rosen. 2011. The Right to Be Forgotten. *Stan. L. Rev. Online* 64 (2011), 88.

[43] XingMao Ruan, YueHeng Sun, Bo Wang, and Shuo Zhang. 2012. The Community Detection of Complex Networks Based on Markov Matrix Spectrum Optimization. In *2012 International Conference on Control Engineering and Communication Technology*. 608–611. doi:10.1109/ICCECT.2012.192

[44] Mattia Samory and Tanushree Mitra. 2018. Conspiracies Online: User Discussions in a Conspiracy Community Following Dramatic Events. *Proceedings of the International AAAI Conference on Web and Social Media* 12, 1 (Jun. 2018). doi:10.1609/icwsm.v12i1.15039

[45] C. E. Shannon. 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal* 27, 3 (1948), 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x

[46] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Di Jin, Quan Z. Sheng, and Philip S. Yu. 2024. A Comprehensive Survey on Community Detection With Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems* 35, 4 (2024), 4682–4702. doi:10.1109/TNNLS.2021.3137396

[47] Gabriele Tolomei and Fabrizio Silvestri. 2021. Generating Actionable Interpretations from Ensembles of Decision Trees. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (2021), 1540–1553. doi:10.1109/TKDE.2019.2945326

[48] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. 2017. Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 465–474. doi:10.1145/3097983.3098039

[49] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports* 9, 1 (2019).

[50] Giovanni Trappolini, Valentino Maiorca, Silvio Severino, Emanuele Rodola, Fabrizio Silvestri, and Gabriele Tolomei. 2023. Sparse Vicious Attacks on Graph Neural Networks. *IEEE Transactions on Artificial Intelligence* (2023).

[51] European Union. 2016. General Data Protection Regulation (GDPR). https://eur-lex.europa.eu/eli/reg/2016/679/oj Accessed: 2025-01-31.

[52] Isaac Waller and Ashton Anderson. 2021. Quantifying Social Organization and Political Polarization in Online Platforms. *Nature* 600, 7888 (2021), 264–268.

[53] Po-Wei Wang and J Zico Kolter. 2020. Community detection using fast low-cardinality semidefinite programming. *Advances in neural information processing systems* 33 (2020), 3374–3385.

[54] Marcin Waniek, Tomasz P. Michalak, Michael J. Wooldridge, and Talal Rahwan. 2018. Hiding Individuals and Communities in a Social Network. *Nature Human Behaviour* 2, 2 (Jan 2018), 139–147. doi:10.1038/s41562-017-0290-3

[55] Joyce Jiyoung Whang, David F. Gleich, and Inderjit S. Dhillon. 2015. Overlapping Community Detection Using Neighborhood-Inflated Seed Expansion. arXiv:1503.07439 [cs.SI]

[56] Jaewon Yang and Jure Leskovec. 2013. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining* (Rome, Italy) *(WSDM '13)*. Association for Computing Machinery, New York, NY, USA, 587–596. doi:10.1145/2433396.2433471

[57] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial Attacks on Neural Networks for Graph Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, 2847–2856. doi:10.1145/3219819.3220078