

## RESEARCH ARTICLE

## Encoding edge type information in graphlets

Mingshan Jia<sup>1\*</sup>, Maité Van Alboom<sup>2</sup>, Liesbet Goubert<sup>2</sup>, Piet Bracke<sup>2</sup>, Bogdan Gabrys<sup>1</sup>, Katarzyna Musial<sup>1</sup><sup>1</sup> Complex Adaptive Systems Lab, School of Computer Science, University of Technology Sydney, Sydney, NSW, Australia, <sup>2</sup> Health Psychology Lab, Ghent University, Ghent, Belgium\* [mingshan.jia@student.uts.edu.au](mailto:mingshan.jia@student.uts.edu.au)

## Abstract

Graph embedding approaches have been attracting increasing attention in recent years mainly due to their universal applicability. They convert network data into a vector space in which the graph structural information and properties are maximumly preserved. Most existing approaches, however, ignore the rich information about interactions between nodes, i.e., edge attribute or edge type. Moreover, the learned embeddings suffer from a lack of explainability, and cannot be used to study the effects of typed structures in edge-attributed networks. In this paper, we introduce a framework to embed edge type information in graphlets and generate a Typed-Edge Graphlets Degree Vector (TyE-GDV). Additionally, we extend two combinatorial approaches, i.e., the colored graphlets and heterogeneous graphlets approaches to edge-attributed networks. Through applying the proposed method to a case study of chronic pain patients, we find that not only the network structure of a patient could indicate his/her perceived pain grade, but also certain social ties, such as those with friends, colleagues, and healthcare professionals, are more crucial in understanding the impact of chronic pain. Further, we demonstrate that in a node classification task, the edge-type encoded graphlets approaches outperform the traditional graphlet degree vector approach by a significant margin, and that TyE-GDV could achieve a competitive performance of the combinatorial approaches while being far more efficient in space requirements.

## OPEN ACCESS

**Citation:** Jia M, Van Alboom M, Goubert L, Bracke P, Gabrys B, Musial K (2022) Encoding edge type information in graphlets. PLoS ONE 17(8): e0273609. <https://doi.org/10.1371/journal.pone.0273609>

**Editor:** Lun Hu, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, CHINA

**Received:** June 2, 2022

**Accepted:** August 11, 2022

**Published:** August 26, 2022

**Peer Review History:** PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: <https://doi.org/10.1371/journal.pone.0273609>

**Copyright:** © 2022 Jia et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** The code, proposed approaches, and experiment results presented in the study are available from <https://github.com/MingshanJia/explore-local-structure>. Readers can

## 1 Introduction

Abstracting entities and their interactions as nodes and links, networks are a general model for studying complex systems [1]. Real-world complex networks contain not only topological information but also rich information about nodes and links [2]. Many previous works propose to exploit node attributes by jointly embedding them with topological structures, and the enhanced representation has been shown to be powerful for numerous applications, such as node classification [3–5], link prediction [6, 7], anomaly detection [8, 9], and network visualization [10].

These approaches, however, overlook rich information about interactions between nodes. Edge attribute or edge type information is indispensable when studying many networks. For instance, the label of each edge in a routing network reflects the cost of traffic via that edge and

also contact author Maité Van Alboom ([Maite.vanalboom@ugent.be](mailto:Maite.vanalboom@ugent.be)) in order to obtain access to the data.

**Funding:** This work was supported by the Australian Research Council, Grant No. DP190101087: "Dynamics and Control of Complex Social Networks". The data collection was supported by two grants from the Fund for Scientific Research-Flanders, Belgium (Grant No. G020118N awarded to Liesbet Goubert and Piet Bracke and grant No. 11K0421N awarded to Maité Van Alboom).

**Competing interests:** The authors have declared that no competing interests exist.

is used to determine the best possible routing scheme; in a user-object bipartite network, an edge is labelled with the user's rating for the product, based on which effective recommender systems can be built [11]; and in egocentric social networks, labels of edges illustrate different types of social relationships and are essential in analysing individuals' behaviours and characteristics [12].

To address this issue, we propose to incorporate edge type information into graphlets and form a Typed-Edge Graphlets Degree Vector (TyE-GDV) [13]. This is mainly inspired by the classic graphlets approach that generates a graphlet degree vector (GDV) [14]. Each coordinate in GDV has a clear meaning, i.e., representing a particular topological structure. Due to this excellent explainability, graphlets have gained considerable ground in a variety of domains. It is revealed that in molecular networks, proteins performing similar biological functions possess similar local structures depicted by GDV [15]. Graphlets are also used in computer vision and neuroscience, in order to capture the spatial structure of superpixels [16] or to detect structural and functional abnormalities in the brain [17]. Notably, in social science, egocentric graphlets are used to depict the social interaction patterns of individuals [18]. In the proposed TyE-GDV approach, we choose to add an extra dimension of edge type on top of GDV, that is to say, counting each type of edge touched by each graphlet. Therefore, each coordinate in the two-dimensional vector also has a clear meaning—the number of edges of a certain type in a certain graphlet. We also propose an egocentric version of TyE-GDV that is more succinct and space efficient when dealing with egocentric networks.

We then employ the proposed TyE-GDV and the classic graphlets degree vector [15] to evaluate and analyse a collection of egocentric social networks of chronic pain patients. The real-life data is gathered from two chronic pain leagues in Belgium [19]. Each patient creates an egocentric social network with edges denoted by the type of social relationships. The patients are divided into four groups based on their self-perceived pain grades. First, we find that graphlet patterns are indeed helpful in assessing the pain grade—patients with higher pain grades form more star-like structures (3-star graphlets), whereas patients with lower pain grades have more tightly connected structures (3-cliques, 4-chordal-cycles and 4-cliques). Second, the edge-type embedded graphlets depicted by TyE-GDV provide us with more insights into how particular social ties could affect the perceived pain. Specifically, we find that in patients of higher pain grades, friends and healthcare workers are the dominant social types in the poorly connected 3-stars; and that in patients of lower pain grades, friends and colleagues appear more often in the tightly connected graphlets such as 3-cliques and 4-cliques.

To compare with the proposed method, we further extend two recent graphlets-based approaches, i.e., the colored graphlets approach [20] and the heterogeneous graphlets approach [21], to edge-attributed networks and egocentric networks. We then apply TyE-GDV and the extended colored and heterogeneous graphlets approaches to a node classification task. Besides the egocentric social networks of chronic pain patients, the dataset also contains rich information about the patients' demographic attributes, pain scores and other physical/psychological well-being descriptors, which are used as baseline features in the experiment. We then set up to include features captured by the proposed method and other related approaches and aim to classify patients into different pain grade groups. The result shows that the edge-type augmented graphlet features are more distinctive than the traditional non-typed graphlet features provided by GDV in separating patients with different pain grades.

To summarise, the main contributions of this work are as follows:

- In order to effectively encode edge type information, we propose a novel framework to generate a Typed-Edge Graphlet Degree Vector;

- We further modify the TyE-GDV framework so that it is better suited for egocentric networks;
- We extend colored graphlets and heterogeneous graphlets approaches for edge-typed networks and egocentric networks.
- According to a case study on individuals with chronic pain, certain social ties are more crucial in understanding the effects of chronic pain and may result in more successful therapeutic interventions.
- We demonstrate that rich structural information enhanced by edge-type information leads to significant improvement in a typical machine learning task.

The remainder of this paper is organised as follows. Related works are discussed in Section 2. Preliminary knowledge is provided in Section 3. The proposed typed-edge graphlets, and the extended colored graphlets and heterogeneous graphlets are introduced in Section 4 and Section 5, respectively. Experiments, results and analysis are presented in Section 6. Finally, we conclude and discuss future directions in Section 7.

## 2 Related work

Compared to abundant approaches that take advantage of node attributes, fewer works have focused on leveraging edge attribute information in graph analysis. A straightforward approach is to construct an adjacency matrix containing edge attributes and then to factorise it [22]. This approach, however, involves the expensive matrix operation like the singular value decomposition and therefore lacks scalability. EdgeCentric focuses on the problem of anomaly detection and proposes to aggregate attribute values of edges incident to each node and defines an abnormality scoring function [23]. One limitation of EdgeCentric is that its topological scope is restricted within directly connected edges. The framework GERI proposes to first construct a heterogeneous graph by adding extra bridge nodes that represent node/edge attributes, then take a random walk to sample a node's neighbourhood, and learn its embedding [24]. However, converting attribute information into structural information will also make the attribute information lose its original meaning. Based on the approach of Poincaré embeddings [25], Chen and Quirk recently proposed an embedding method that simultaneously preserves the hierarchical property and edge attributes [26]. This approach is apparently limited in its exclusive focus on hierarchical relationships.

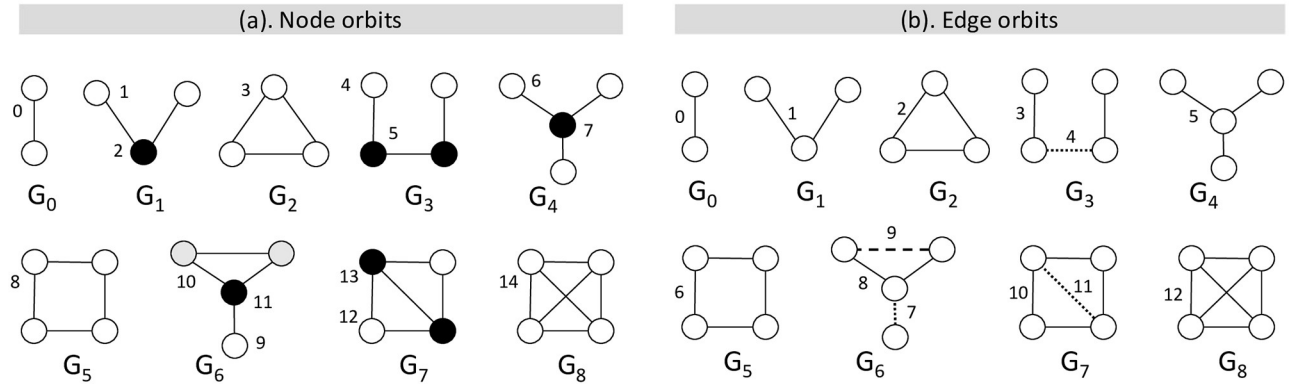
Although these approaches are shown to be effective in some downstream tasks, a common issue about them is that their learned embeddings lack explainability—we do not know what each element of the embedding vector means. They are, therefore, unable to reveal the deeper and, ideally, more easily explainable relationship between a local network structure and an edge attribute.

## 3 Preliminaries

In this section, we introduce the notions of graphlets and orbits, and discuss how they can be adapted in egocentric networks.

### 3.1 Graphlets and orbits

Node degree, being the most basic structural feature, counts the number of edges incident to a node. Graphlets or graphlets degree generalises the idea of node degree by counting the number of graphlets the node participates. Specifically, graphlets are a set of “small connected non-isomorphic induced subgraphs” [14]. Small is to say the size of subgraphs is small, usually no



**Fig 1. Graphlets of 2–4 nodes with the enumeration of orbits.** (a) Node orbits: there are in total 15 node orbits, different node colors indicating nonisomorphic node positions within a given graphlet. (b) Edge orbits: there are in total 13 edge orbits, different line types denoting nonisomorphic edge positions within a given graphlet.

<https://doi.org/10.1371/journal.pone.0273609.g001>

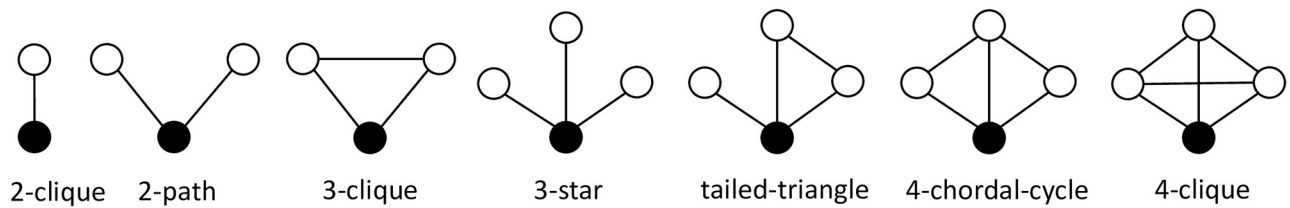
more than 4 or 5 nodes. Nonisomorphic means that those subgraphs are structurally distinct, and induced means that all the edges among the nodes in a subgraph need to be considered. The original work covers graphlets of sizes ranging from 2 to 5 nodes, resulting in a total number of 30 different graphlets. Besides, as a node-level structural measure, the non-symmetry of node position is also taken into account, leading to a total number of 73 different subgraph structures, termed automorphism orbits [15]. Briefly, orbits are graphlets that distinguish the position of a focal node (we use orbits and node-orbit graphlets interchangeably in this work). The Graphlet Degree Vector (GDV) of a particular node is thus defined as a vector of the frequencies of 73 orbits.

GDV, or sometimes normalised GDV, has been widely applied in various domains and has become a standard structural feature when measuring the similarities and differences between nodes [15–17]. We summarise node-orbit graphlets of 2 to 4 nodes in Fig 1(a). Taking one of the black nodes in  $G_7$ , for example, it touches orbit-0 three times (the degree of the node), orbit-2 once (the open triad), orbit-3 twice (the triangle), and orbit-13 once. Therefore, its graphlet degree vector has 3 at the 0<sup>th</sup> coordinate, 1s at the 2<sup>th</sup> and 13<sup>th</sup> coordinates, 2 at the 3<sup>rd</sup> coordinate, and 0s at the remaining coordinates.

The notion of orbits was originally established at a node level, distinguishing a node position when counting graphlets. Hočevar and Demšar later proposed to count graphlets at a link level and introduced the notion of edge orbits [27]. Fig 1(b) gives all edge orbits containing 2 to 4 nodes. Apparently, edge orbits are different from node orbits. For example, there is only one edge orbit in graphlet  $G_1$ , but two node orbits in it. We also refer to edge orbits as edge-orbit graphlets in this work. The concept of heterogeneous graphlets is built upon edge orbits, and we will discuss more about it in Section 5.

### 3.2 Egocentric graphlets

Graphlets is initially proposed for general networks or sociocentric networks. Although sociocentric networks appear to be more comprehensive modellings of complex systems, collecting sociocentric data via survey is also difficult because participants need to be identifiable to the researcher, and this lack of anonymity can result in unwillingness to participate or bias in responses [12]. Moreover, there are situations where we care more about individuals and their immediate environment. For example, we may want to understand why some people form densely connected ego networks while others don't.



**Fig 2. Egocentric graphlets of 2 to 4 nodes.** There are in total seven egocentric graphlets. The black node in a given graphlet is the ego node, other nodes are alter nodes.

<https://doi.org/10.1371/journal.pone.0273609.g002>

Being a node-level measure, graphlets are naturally suitable to be applied in egocentric networks, with two more restrictions. First, some graphlets that do not fit the definition of an egocentric network need to be eliminated. For example, in graphlets of 2 to 4 nodes (Fig 1(a)),  $G_3$  (3-path) and  $G_5$  (4-cycle) are excluded because any node in them acting as an ego cannot reach other nodes in a single hop. Second, since only one node can serve as the ego in an egocentric graphlet, it is unnecessary to discriminate between different orbits. Therefore, there are in total seven egocentric graphlets of 2 to 4 nodes, which are 2-clique, 2-path, 3-clique, 3-star, tailed-triangle, 4-chordal-cycle and 4-clique (Fig 2).

#### 4 Typed-edge graphlet degree vector

This section describes the framework for generating the typed-edge graphlet degree vector.

The classic graphlet degree vector manages to capture the structural patterns in homogeneous networks. However, many real-world networks also contain rich information on nodes and edges, making them node-attributed, edge-attributed or heterogeneous networks. Information about edge type is particularly important in social networks since it provides a detailed description of relationships among individuals. In the target dataset of this study, for instance, each patient with chronic pain specifies their egocentric social network, including up to ten actors, and each ego-to-alter edge is labelled with one of 13 different types of social ties. In order to analyse edge-attributed networks at a finer granularity and capture the rich edge-typed connectivity patterns, we propose to embed edge type information in graphlets. The original graphlet degree vector generates a one-dimensional vector by counting the instances of each type of graphlet. Here, we propose to build a two-dimensional vector by adding an extra dimension of edge type on top of GDV, that is to say, counting each type of edge contained in each type of graphlet.

We start by formally defining an edge-attributed network.

**Definition 1** An edge-attributed network  $G$  is a triple  $\langle V, E, \mathcal{T}_e \rangle$ , where  $V = \{v_1, v_2, \dots, v_n\}$  is the set of nodes,  $E = \{e_{ij}\} \subset V \times V$  is the set of edges where  $e_{ij}$  indicates an edge between nodes  $v_i$  and  $v_j$ , and  $\mathcal{T}_e$  is the set of edge types, where  $\tau_{e_{ij}}$  denotes the type of edge  $e_{ij}$ .

The initial step of the framework is a graph preprocessing, where the set of edge types is mapped to integers ranging from 0 to  $|\mathcal{T}_e| - 1$ . For example, the 13 different types of social ties in the target dataset are represented from 0 to 12. ( $\tau_e \in [0, 12]$ ). Additionally, the set of orbits  $\mathcal{O}$  is converted to integers from 0 to  $|\mathcal{O}| - 1$ . In this study, we take into account all the node-orbit graphlets within the size of 2 to 4 nodes (Fig 1(a)). Thus, there are 15 orbits coded from 0 to 14 ( $o \in [0, 14]$ ).

**Algorithm 1: Build Typed-Edge Graphlet Degree Vector.**

**input:** preprocessed graph  $G = \langle V, E, \mathcal{T}_e \rangle$ , set of node-orbits  $\mathcal{O}$ , node set  $V'$ .

**Output:** dictionary *dic* of vectors for all nodes  $\in V'$ .

1 initialise: *dic* = {}

2 **foreach**  $i \in V'$  **do**

3   initialise a 2d-vector *vec* of size  $|\mathcal{O}| \times |\mathcal{T}_e|$  with zeros

```

4  foreach  $o \in \mathcal{O}$  do
5     $L_e = \text{GETEDGE LIST}(o)$ ;
6     $\text{UPDATE}(vec, o, L_e)$ 
7     $dic[i] = vec$ ;

```

**Algorithm 2:** Update Vector.

```

1  Function UPDATE
   input 2d-vector  $vec$ , type of node orbit  $o$ , list of edges  $L_e$ .
2  foreach  $e \in L_e$  do
3     $\tau_e = \text{GETTYPE}(e)$ ;
   /*  $o$  and  $\tau_e$  are used as indices in  $vec$ . */
4     $vec[o][\tau_e]$  increase by 1;

```

**Algorithm 3:** Code Snippet for Orbit-6, 9 and 10.

```

1  foreach  $i \in V'$  do
2    initialise a 2d-vector  $vec$  of size  $|\mathcal{O}| \times |\mathcal{T}_e|$  with zeros;
3  foreach  $u \in N_i$  do
4    foreach  $v, w \in C(N_u, 2)$  do
   /*  $o_i$  denotes the corresponding index for orbit- $i$  */
5    if  $v \notin N_i \wedge w \notin N_i \wedge v \notin N_w$  then
6       $\text{UPDATE}(vec, o_6, [e_{iu}, e_{uv}, e_{uw}]);$            ▷ orbit-6
7    if  $v \notin N_i \wedge w \notin N_i \wedge v \in N_w$  then
8       $\text{UPDATE}(vec, o_9, [e_{iu}, e_{uv}, e_{uw}, e_{vw}]);$      ▷ orbit-9
9    if  $v \in N_i \wedge w \notin N_i \wedge v \notin N_w$  then
10      $\text{UPDATE}(vec, o_{10}, [e_{iu}, e_{uv}, e_{uw}, e_{iv}]);$ 
11     if  $v \notin N_i \wedge w \in N_i \wedge v \notin N_w$  then
12        $\text{UPDATE}(vec, o_{10}, [e_{iu}, e_{uv}, e_{uw}, e_{iw}]);$  }   ▷ orbit-10
13

```

Next, for any node of interest, the typed-edge graphlet degree vector (TyE-GDV), i.e., a two-dimensional vector of size  $|\mathcal{O}| \times |\mathcal{T}_e|$ , is generated using Algorithm 1. Concretely, after initialisation, for each node in a given node set  $V'$  and for each orbit in the set of node-orbit graphlets  $\mathcal{O}$ , the vector is updated through the UPDATE function (Algorithm 2). The calculation of each orbit in Algorithm 1 is omitted for a more concise expression. To demonstrate the detailed process, we give a program snippet for calculating orbit-6, orbit-9 and orbit-10 in Algorithm 3.  $C(N_u, 2)$  denotes all possible 2-combinations of the neighbours of node  $u$ . The use of combinations is to avoid repetitive calculation. In Algorithm 2,  $o$  and  $\tau_e$  are readily used as indices when updating the vector as a result of the preprocessing stage. Finally, at the end of Algorithm 1, a dictionary of nodes as keys and their corresponding TyE-GDV as values is returned. For example, if an orbit-9 is detected and its four edges are of type ‘0’, ‘1’, ‘2’ and ‘2’, vector elements at coordinates (9, 0), (9, 1), (9, 2) and (9, 2) will increase by 1. Obviously, the time complexity of generating TyE-GDV is the same as counting graphlets. Although the introduction and implementation of the typed-edge graphlets approach is aimed at dealing with edge attributed networks, it can be easily extended to node attributed networks by replacing an edge type with a node type, or to networks containing both different node and edge types by adding an extra dimension of a node type.

As discussed in Section 3.2, egocentric networks are sometimes of special interest, especially when edge type information is included (as in our case study dataset of chronic pain patients). With the restriction of being egocentric, there are fewer orbits in graphlets that need to be considered. Therefore, we propose a tailor-made version of the framework for egocentric networks, called TyE-EGDV (see Algorithm 4).  $C(N_i, 2)$  and  $C(N_i, 3)$  stand for all possible 2-combinations and 3-combinations of the neighbours of node  $i$ . Note that in TyE-EGDV, there are in total 7 orbits in  $\mathcal{O}$ , instead of 15 (see Fig 2). Therefore, the algorithm is more efficient in both time and space.

**Algorithm 4:** Build Typed-Edge Ego-Graphlet Degree Vector.

```

input : preprocessed graph  $G = \langle V, E, \mathcal{T}_e \rangle$ , set of egocentric node-orbits  $\mathcal{O}$ ,
        node set  $V'$ .
output : dictionary  $dic$  of vectors for all nodes  $\in V'$ .
1 initialise:  $dic = \{\}$ ;
2 foreach  $i \in V'$  do
3   initialise a 2d-vector  $vec$  of size  $|\mathcal{O}| \times |\mathcal{T}_e|$  with zeros;
4   foreach  $u \in N_i$  do
5     UPDATE( $vec, o_0, e_{iu}$ ); ▷ 2-clique
6   foreach  $u, v \in C(N_i, 2)$  do
7     if  $v \notin N_u$  then
8       UPDATE( $vec, o_1, [e_{iu}, e_{iv}]$ ); ▷ 2-path
9     else
10      UPDATE( $vec, o_2, [e_{iu}, e_{iv}, e_{uv}]$ ); ▷ 3-clique
11   foreach  $u, v, w \in C(N_i, 3)$  do
12     if  $u \notin N_v \wedge u \notin N_w \wedge v \notin N_w$  then
13       UPDATE( $vec, o_3, [e_{iu}, e_{iv}, e_{iw}]$ ); ▷ 3-star
14     else if  $v \in N_u \wedge w \notin N_u \wedge w \notin N_v$  then
15       UPDATE( $vec, o_4, [e_{iu}, e_{iv}, e_{iw}, e_{uv}]$ );
16     else if  $w \in N_u \wedge v \notin N_u \wedge v \notin N_w$  then
17       UPDATE( $vec, o_4, [e_{iu}, e_{iv}, e_{iw}, e_{uw}]$ ); ▷ tailed-tri
18     else if  $w \in N_v \wedge u \notin N_v \wedge u \notin N_w$  then
19       UPDATE( $vec, o_4, [e_{iu}, e_{iv}, e_{iw}, e_{vw}]$ );
20     else if  $u \in (N_v \cap N_w) \wedge w \notin N_v$  then
21       UPDATE( $vec, o_5, [e_{iu}, e_{iv}, e_{iw}, e_{uv}, e_{uw}]$ );
22     else if  $v \in (N_u \cap N_w) \wedge w \notin N_u$  then
23       UPDATE( $vec, o_5, [e_{iu}, e_{iv}, e_{iw}, e_{uv}, e_{vw}]$ ); ▷ 4-chord-cyc
24     else if  $w \in (N_u \cap N_v) \wedge v \notin N_u$  then
25       UPDATE( $vec, o_5, [e_{iu}, e_{iv}, e_{iw}, e_{uv}, e_{vw}]$ );
26     else
27       UPDATE( $vec, o_6, [e_{iu}, e_{iv}, e_{iw}, e_{uv}, e_{vw}, e_{uw}]$ ); ▷ 4-clique
28      $dic[i] = vec$ ;

```

**5 Typed-edge degree, colored graphlets and heterogeneous graphlets**

Since a node degree is the simplest network structural metric, a naive way of encoding edge type information in a network structure is first to have the notion of a typed-edge degree. Formally, the typed-edge degree of a node  $i$  with an edge type  $t$ , i.e.,  $d_i^t$ , is defined as the number of edges of type  $t$  that are connected to  $i$ . Then, a typed-edge degree vector (TyE-DV) can be defined as a vector containing typed-edge degrees of all types.

Some other approaches that also aim to take a node and/or an edge type into consideration include the colored motifs [28], colored graphlets [20] and heterogeneous graphlets [21]. Colored motifs, as the name suggests, extended G-Tries algorithm that counts motifs [29] by including the information of a node or edge type. This approach, however, is at the network level and is therefore not suitable for a node-level analysis.

Colored graphlets approach [20] is at the node level, and proposes to distinguish different graphlets according to all combinations of node types. The approach is said to be able to deal with typed edges, but without theoretical explanation or experimental demonstration. The article alleges that the total number of combinations equals  $2^T - 1$ , where  $T$  is the total number of possible node types. This is incorrect as it fails to take the size of the graphlet into account. When graphlet size is smaller than the number of node types, the total number of combinations will be smaller than  $2^T - 1$ . For example, when we consider the graphlet  $G_0$ , i.e., 2-clique, with

three possible node types, there are in total six combinations, instead of seven. The combination containing all three types cannot exist since there are only two nodes in this graphlet. Below, we give the amended equation for calculating the number of combinations in a given graphlet  $g$ :

$$C(g) = \sum_{n=1}^{\min(K(g),T)} \binom{T}{n}, \tag{1}$$

where  $K(g)$  is the number of nodes of the graphlet when  $T$  refers to a node type, or the number of edges of the graphlet when it refers to an edge type. Note that when  $K(g) \geq T$ , the equation becomes  $\sum_{n=1}^T \binom{T}{n}$ , which equals  $2^T - 1$ . We then develop a colored graphlets approach for edge-typed networks, named ColoredE-GDV, which is also applied to the case study in the next section.

The recently proposed heterogeneous graphlets approach [21] also considers a node type in graphlets. It is different from the colored graphlets approach in two ways. First, heterogeneous graphlets are computed at a link level. It distinguishes the position of a given edge, instead of a given node (please refer to the notion of edge-orbit graphlets in Section 3.1). The benefit of a link-based computation is that it is more time-efficient in sparse networks than node-based approaches. The downside, apparently, is that it is not suitable for a node-level analysis. Second, heterogeneous graphlets propose to use combinations with repetitions of node types, rather than just a combination, when distinguishing different graphlets. The total number of possible heterogeneous graphlets is calculated as:

$$H(g) = \sum_{n=1}^T \binom{T}{n} \cdot \binom{K(g)-1}{n-1} = \binom{T+K(g)-1}{K(g)}. \tag{2}$$

Similarly,  $K(g)$  is the number of nodes of the graphlet when  $T$  refers to a node type, and the number of edges when it refers to an edge type. Since type repetition is allowed in heterogeneous graphlets, the number of possible heterogeneous graphlets is larger than that of colored graphlets.

In order to extend the idea of heterogeneous graphlets to a node-level analysis and to deal with typed edges, we propose a node-based typed-edge heterogeneous graphlets approach, named HeteroE-GDV<sup>N</sup> (the original link-based typed-node approach is noted as HeteroN-GDV<sup>L</sup>). The approach of HeteroE-GDV<sup>N</sup> is demonstrated through Algorithm 5. We see clearly that its time complexity stays the same when counting untyped graphlets, but the space complexity grows fast with the number of edge types.

**Algorithm 5:** Node-based Heterogeneous Graphlets Degree Vector (Hetero-GDV<sup>N</sup>)  
**input:** preprocessed graph  $G = \langle V, E, T_e \rangle$ , set of node-orbits  $\mathcal{O}$ , node set  $V'$ .  
**output:** dictionary *dic* of vectors for all nodes  $\in V'$ .

```

1 initialise: dic = {};
2  $L^{T_e} = [0, 1, \dots, |T_e| - 1]$ ;
   /* range of edge number of graphlets of size 2–4 nodes */
3 for  $k \leftarrow 1$  to 6 do
4    $L_k = [\text{GETCOMBWITHREP}(L^{T_e}, k)]$ ;
5 foreach  $i \in V'$  do
6   for  $o \leftarrow 0$  to  $|\mathcal{O}| - 1$  do
7     initialise  $vec_o$ ;
8   foreach  $o \in \mathcal{O}$  do
9      $k = \text{GETNUMOFEDGE}(o)$ ;
10     $L_e = \text{GETEDGELIST}(o)$ ;
11     $tup = (\text{SORT}(L_e))$ ;
12     $vec_o[\text{GETINDEX}(L_k, tup)]$  increase by 1;
13  $vec = [vec_0, vec_1, \dots, vec_{|\mathcal{O}|-1}]$ ;
14  $dic[i] = vec$ ;
```



**Table 1. Time and space complexities of four approaches that deal with edge type information.** S is the maximum number of nodes in graphlets, K is the maximum number of edges in graphlets,  $|\mathcal{O}_e|$  is the number of edge-orbit graphlets.

| Approach                     | Time complexity               | Space complexity                                               |
|------------------------------|-------------------------------|----------------------------------------------------------------|
| Colored-GDV [20]             | $O( V  \cdot k_{\max}^{S-1})$ | $O( V  \cdot  \mathcal{O}  \cdot 2^{ \mathcal{T}_e })$         |
| Hetero-GDV <sup>L</sup> [21] | $O( E  \cdot k_{\max}^{S-2})$ | $O( E  \cdot  \mathcal{O}_e  \cdot K C_{ \mathcal{T}_e +K-1})$ |
| HeteroE-GDV <sup>N</sup>     | $O( V  \cdot k_{\max}^{S-1})$ | $O( V  \cdot  \mathcal{O}  \cdot K C_{ \mathcal{T}_e +K-1})$   |
| TyE-GDV                      | $O( V  \cdot k_{\max}^{S-1})$ | $O( V  \cdot  \mathcal{O}  \cdot  \mathcal{T}_e )$             |

<https://doi.org/10.1371/journal.pone.0273609.t001>

Although the above approaches seem powerful to capture all possible combinations (or combinations of repetitions) of different types of nodes or edges, their numbers of possible graphlets, which are also their space complexities, grow near-exponentially with the number of node or edge types. For example, with 9 node types, in the colored graphlets approach, there are 255 possible colored graphlets for a graphlet of 4 nodes; and in the heterogeneous graphlets approach, there are 495 possible graphlets. In comparison, the space complexity grows linearly with the number of edge types in the proposed TyE-GDV approach. Moreover, out of this large number of possible graphlets, only a tiny percentage of them actually exists in real networks. For example, in Cora citation network [30], only 19 heterogeneous graphlets exist out of 210 possible ones in a 4-clique graphlet.

In order to utilise the colored graphlets and the heterogeneous graphlets approaches in egocentric networks, we further develop their egocentric versions, and apply them in the chronic pain case study. With fewer node orbits to consider, egocentric colored graphlets and egocentric heterogeneous graphlets are faster and more space-saving than the original ones. The implementation of these algorithms is available at <https://github.com/MingshanJia/explore-local-structure>.

To conclude this section, we summarise the time and space complexities of the four main approaches in Table 1. Colored-GDV, HeteroE-GDV<sup>N</sup> and TyE-GDV share the same time complexity because they are all node-based algorithms. Hetero-GDV<sup>L</sup> as the only link-based algorithm, could be faster in sparse networks. When it comes to space complexity, the proposed TyE-GDV grows linearly with the number of edge types, while the other three methods grow near exponentially with it.

## 6 Experiments and analysis

In this section, we apply the proposed methods to analyse the egocentric social networks of chronic pain patients.

### 6.1 Dataset

The real-world dataset is collected from chronic pain patients of the League for Rheumatoid Arthritis, the League for Fibromyalgia and the Flemish Pain League [19]. Each patient creates their own egocentric social networks containing up to 10 alters using the graphical tool GENSI [31]. The types of social ties between the patient (the ego node) and his/her contacts (the alters) are explicitly given. There are in total 13 types of social relationships, including families, friends, colleagues, neighbours, etc. The full list of social ties and their total occurrences are listed in Table 2). The patients were also asked to fill out a questionnaire on pain-related and sociodemographic information. In addition to that, a daily diary consisting of items measuring pain intensity, and physical, psychological and social well-beings, was provided to participants for 14 consecutive days. After eliminating inconsistent and incomplete entries, the final dataset

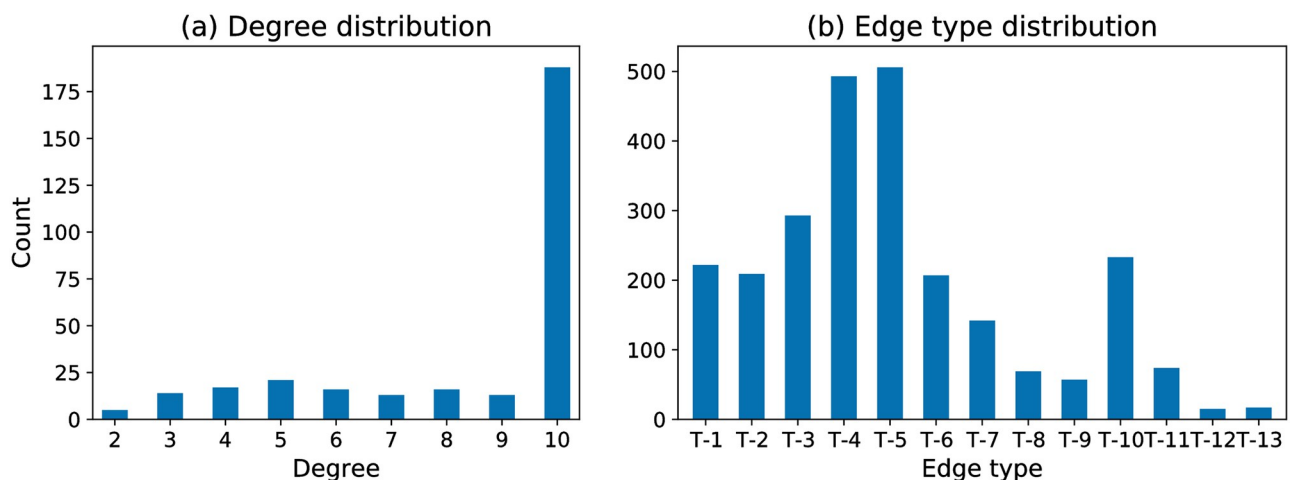
**Table 2. 13 types of social relationships and their total number of occurrences in 303 egocentric networks.**

| Social Relationship     | Type Code | Number of Occurrences |
|-------------------------|-----------|-----------------------|
| Partner                 | T-1       | 222                   |
| Father/Mother           | T-2       | 209                   |
| Brother/Sister          | T-3       | 293                   |
| Children/Grandchildren  | T-4       | 493                   |
| Friend                  | T-5       | 506                   |
| Family-in-law           | T-6       | 207                   |
| Other family            | T-7       | 142                   |
| Neighbour               | T-8       | 69                    |
| Colleague               | T-9       | 57                    |
| Healthcare worker       | T-10      | 233                   |
| Member of organisations | T-11      | 74                    |
| Acquaintance            | T-12      | 15                    |
| Other                   | T-13      | 17                    |

<https://doi.org/10.1371/journal.pone.0273609.t002>

consists of the egocentric social networks, sociodemographic and pain characteristics of 303 patients. The average age of all patients is  $53.5 \pm 12$  years, including 248 females and 55 males.

Some basic characteristics of the egocentric networks, such as the ego nodes' degree distribution and their edge-type distribution, are shown in Fig 3. The edge-type distribution is computed by summing over all ego nodes on each type of the edges, which is also displayed in the third column of Table 2. The degree distribution reveals that the majority of patients (62%) have ten social connections in their social networks (Fig 3a). However, we do not anticipate node degree to be a discriminative feature in the following analysis since ten contacts are the upper limit in the dataset. According to the edge-type distribution (Fig 3b), the most frequent types in these networks are T-5 "friend" and T-4 "children/grandchildren". In contrast, edge types T-8 "neighbour", T-9 "colleague" and T-11 "member of organisations" are underrepresented. T12 "acquaintance" and T-13 "other" are almost negligible because people would first list their strongest contacts with the limitation of ten connections, leaving little room for those weaker ties.

**Fig 3. Degree distribution and edge-type distribution of 303 egocentric social networks.**

<https://doi.org/10.1371/journal.pone.0273609.g003>

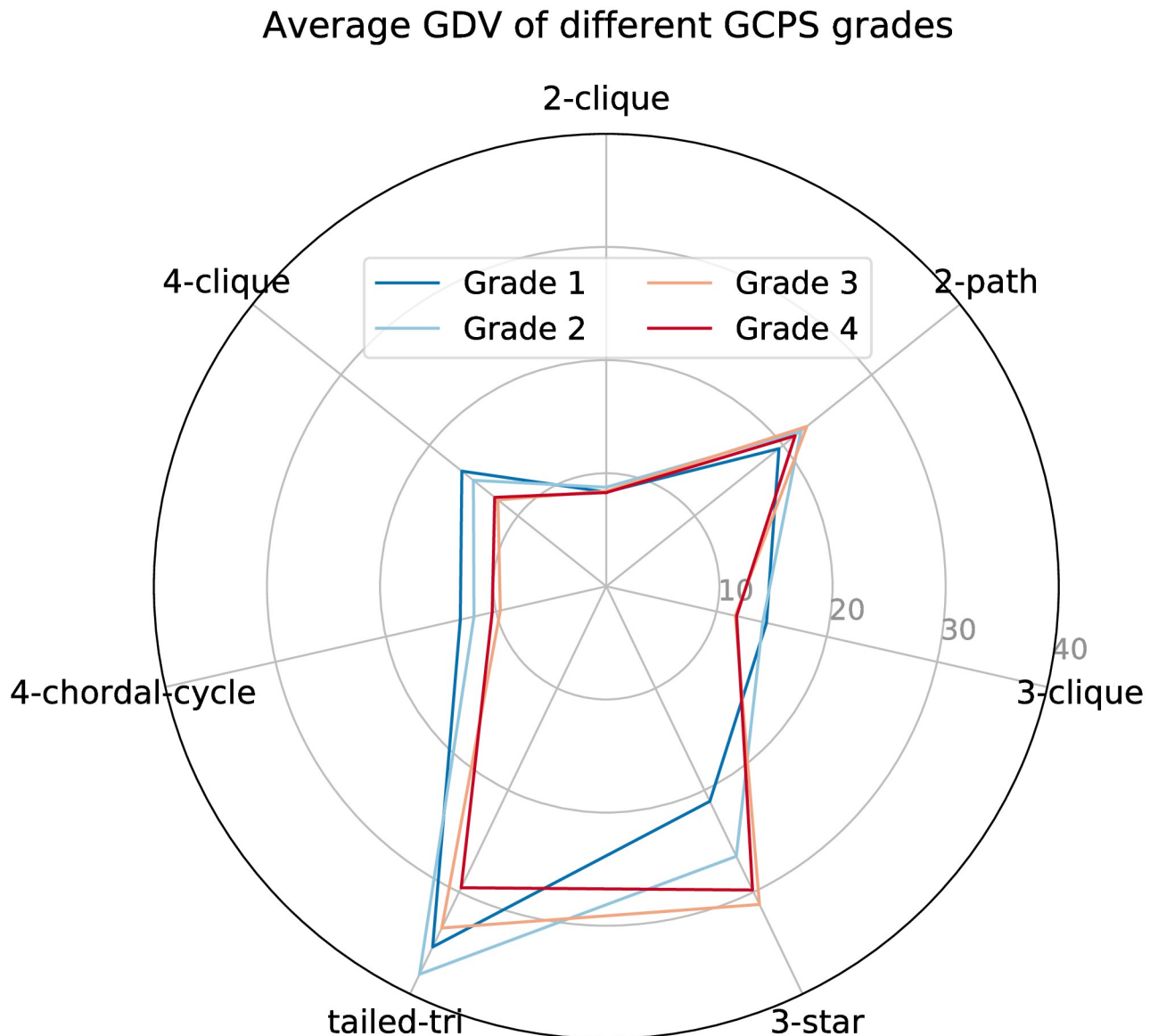
Moreover, the grades of chronic pain are calculated by means of the Graded Chronic Pain Scale (GCPS), which evaluates both pain disability and pain intensity [32]. Then, patients are divided into five grades based on their average intensity and disability scores: grade-0 for no pain; grade-1 for low intensity and low disability; grade-2 for high intensity and low disability; grade-3 for moderate disability irrespective of pain intensity; and grade-4 for high disability irrespective of pain intensity. Due to the fact that all participants have a certain degree of chronic pain, their GCPS grades vary from grade-1 to grade-4. Specifically, there are 21 patients in grade-1, 33 patients in grade-2, 67 patients in grade-3 and 182 patients in grade-4. In this study, we aim to investigate whether the structural feature, especially the edge type augmented structural feature captured by TyE-GDV, are helpful in understanding the patients' pain grades.

## 6.2 Analysing pain grades

Evidence within the fields of pain and rehabilitation science has shown that social interactions play an important role in the perception of pain [33]. Perceived social support and pain inference are found to be associated in individuals with chronic musculoskeletal pain [34]. Lower levels of social support and higher levels of pain intensity are observed in rheumatoid arthritis patients at the 3- and 5-year follow-ups [35]. It has also been demonstrated recently that reduced social isolation accounts for significant improvements in self-reported emotional and physical functioning [36]. Typically in these studies, the social milieu of a patient is assessed by the Social Support Satisfaction Scale (ESSS) [37] or the Patient Reported Outcome Measurement Information System (PROMIS<sup>®</sup>) [38]. However, as these measurements are not based on the real social networks of the patients, they are unable to shed light on the impact of network topologies, especially certain types of interactions, on the perception of pain. To address this issue, we choose to apply both the traditional graphlets approach and the proposed typed-edge graphlets approach to analyse the egocentric networks of chronic pain patients.

First, in order to investigate the impact of network structure on pain grade, we calculate the average egocentric graphlet degree vector for each GCPS grade. A radar chart shows the average values of the seven egocentric graphlets at each grade (Fig 4). We observe clearly that patients with higher pain grades (grade-3 and grade-4) possess more star-like structures (3-star graphlet) in their social networks, whereas patients with lower pain grades (grade-1 and grade-2) compose more clique-like or quasi-clique-like structures (3-clique, 4-clique and 4-chordal-cycle graphlets). A poorer-connected star-like structure denotes a more isolating social setting, whereas a better-connected structure, such as the 3-clique or 4-clique, may suggest stronger social support. These findings are in agreement with the aforementioned studies [33–36] and provide further evidence that a patient's social network may influence how much pain they perceive. Additionally, we discover that the number of immediate connections (2-cliques) is ineffective in differentiating pain grades, which may be partially caused by the limited number of contacts in the dataset. Nevertheless, Evers et al. [35] also discovered that changes in pain are not substantially correlated with the size of a patient's egocentric social network. Jia et al. revealed that the clustering coefficient and the quadrangle coefficient are useful topological features in assessing the perception of pain [39]. These findings further underline the need to consider more complex network topologies when examining patients' social networks.

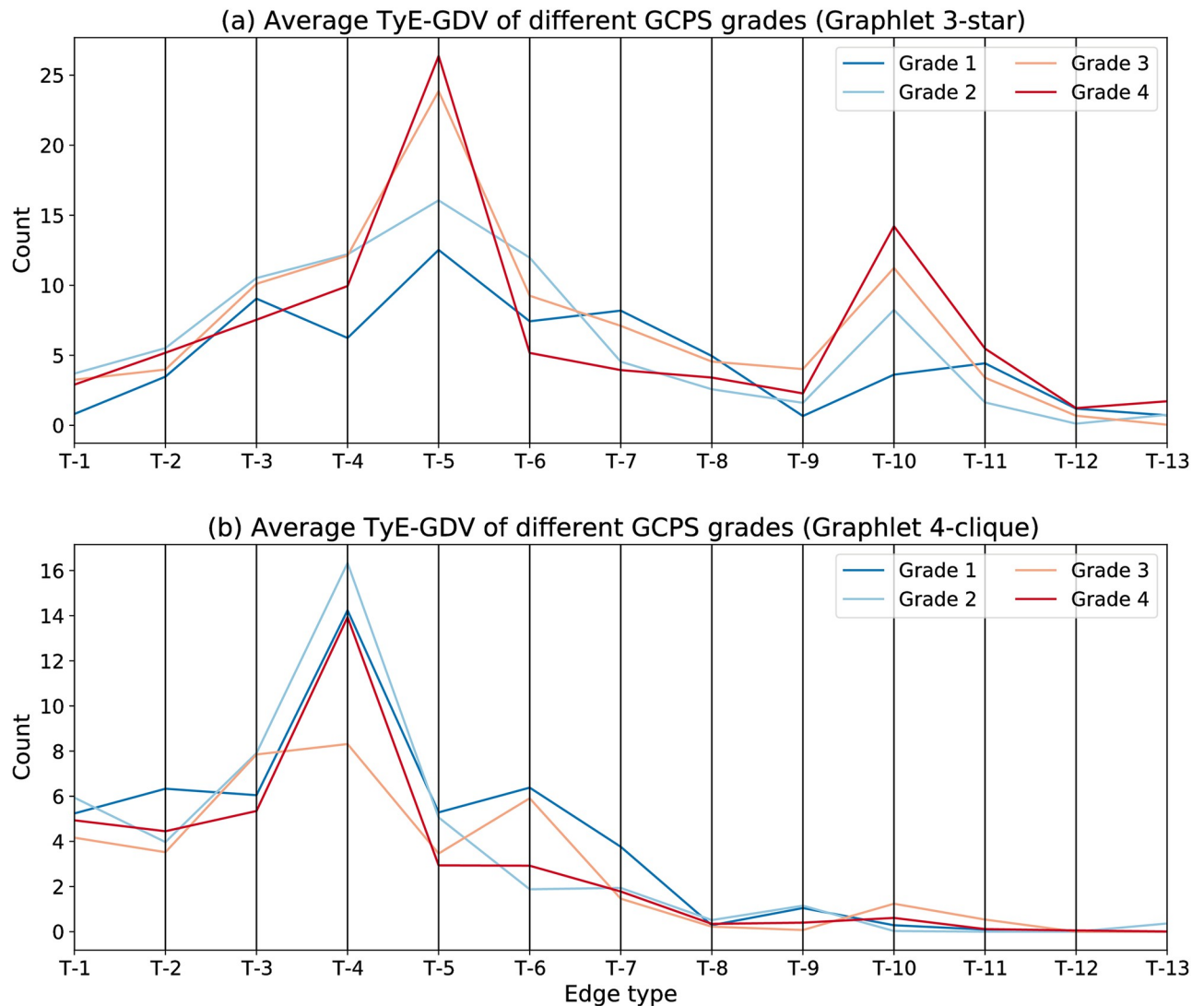
Furthermore, in order to analyse the association between the types of social ties and the perception of pain, we employ the typed-edge graphlet degree vector and focus on two specific graphlets, namely the weakly connected 3-star graphlet and the highly connected 4-clique graphlet. These two graphlets are selected not only because they represent two extremes of 4-node structures but also because distinct differences between patients with lower pain grades



**Fig 4. Radar chart of average GDV of different GCPS grades.** Each spoke represents the average number of graphlets belonging to that type.

<https://doi.org/10.1371/journal.pone.0273609.g004>

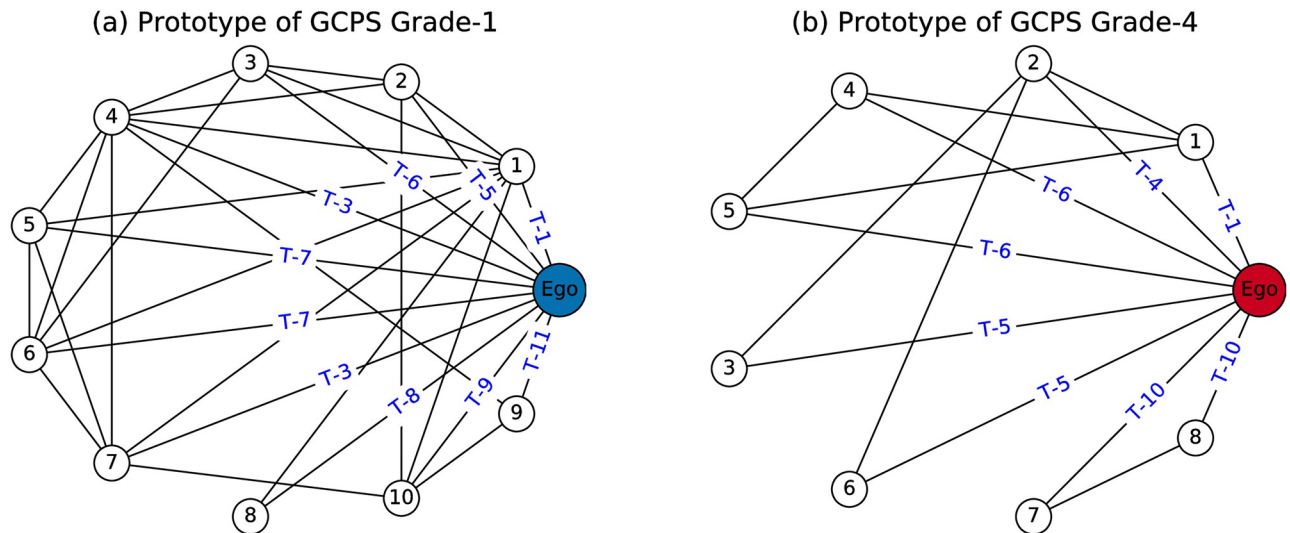
and patients with higher pain grades are observed in them. We first calculate the average counts of the 13 edge types at each pain grade for the 3-star graphlet, i.e., the 3<sup>rd</sup> row of the Typed-Edge Ego-Graphlet Degree Vector (see Algorithm 4), and draw a parallel coordinates plot (Fig 5(a)). We discover that in the poorly connected star-like structure, edges of type T-5 “friend” and T-10 “healthcare worker” are significantly more frequent in patients with higher pain grades than in patients of lower pain grades. That is to say, in the social networks of higher pain grade patients, friends and healthcare workers are in a rather isolated position—not well connected with other contacts of the patient. Thus, it provides the potential for treatments that boost a patient’s friends’ and healthcare professionals’ social involvement to improve chronic pain management.



**Fig 5. Two parallel coordinates plots revealing the association of edge type and pain grade.** (a). Average TyE-GDV of four GPCS grades for 3-star graphlet. (b). Average TyE-GDV of four GPCS grades for 4-clique graphlet.

<https://doi.org/10.1371/journal.pone.0273609.g005>

We then calculate the average counts of the 13 edge types at each pain grade for the 4-clique graphlet, i.e., the 6<sup>th</sup> row of the Typed-Edge Ego-Graphlet Degree Vector, and the corresponding parallel coordinates plot is given in Fig 5(b). We observe that, in this tightly-connected structure, patients with lower pain grades have more edges of type T-5 “friend” than patients with higher pain grades. In other words, friends are better involved in the social networks of patients who perceive lower level pain grades than those who perceive higher pain grades. The importance of friendship is revealed in both 3-star and 4-clique graphlets. As pointed out by other studies [40, 41], people with severe chronic pain may be more liable to a deterioration of their friend relationships and are in more need of supportive behaviours from friends. Another noticeable difference between patients of lower pain grades and patients of higher pain grades is found in edge T-9 “colleague”. In contrast to the lower pain grade group, where more than one colleague appears in the clique structures (1.1 on average), colleagues hardly exist in them among the higher pain grade group (0.24 on average). This could be a result of the negative



**Fig 6. Social network prototypes of patients with GPCS grade-1 and patients with GPCS grade-4.** (a) In the prototype network of patients with pain grade-1, contacts are tightly connected to each other with the appearance of T-5 friend and T-9 colleague; (b) In the prototype network of patients with pain grade-4, contacts are loosely connected with limited links incident to T-5 friend and T-10 healthcare workers.

<https://doi.org/10.1371/journal.pone.0273609.g006>

consequences that severe chronic pain has on patients' capacity for work [42]. To provide an intuitive grasp of the edge type encoded structural differences between the social networks of patients with different pain grades, we extract two real examples from the dataset as the network prototypes of patients of pain grade-1 and patients of pain grade-4, respectively (Fig 6).

This experiment demonstrates that the extra edge type information encoded in TyE-GDV provides us with more insights into the association between patients' perception of pain grade and the type of social ties in their egocentric networks. It thus has implications for improving therapeutic interventions through boosting particular types of social interactions.

### 6.3 Node classification

We now apply the proposed TyE-GDV, and the extended egocentric versions of colored graphlets (ColoredE-GDV) and heterogeneous graphlets (HeteroE-GDV<sup>N</sup>) approaches in a typical machine learning task.

Node classification, being one of the most popular and extensively explored tasks in network science [43], aims to predict the labelling of nodes based on a subset of nodes that have ground-truth labels. Here, our goal is to predict the GPCS grade of patients with chronic pain. In order to evaluate the effectiveness of the proposed approaches, we fit six sets of features into a random forest classifier. The first set comprises the patients' demographic attributes, pain-related descriptors and their physical and psychological well-being indicators. Since it contains no network-related information, we refer to it as the raw feature set. The second set and the third set add the typed-edge degree vector (TyE-DV) and the traditional graphlet degree vector (GDV), respectively, on top of the raw features. The fourth set combines the raw features with the proposed typed edge graphlet degree vector (TyE-GDV), and finally, the fifth set and the sixth set plus the colored graphlets degree vector (ColoredE-GDV) and the heterogeneous graphlets degree vector (HeteroE-GDV<sup>N</sup>), respectively, to the raw feature set.

Since the dataset is not big and the distribution of the four pain grades is not balanced (see Section 6.1), we adopt a stratified 5-fold cross-validation [44] to evaluate the classification

**Table 3. Result table of node classification, reported in the average macro-F1 score ( $\pm$  standard deviation), the average percentage gain over the raw feature set, and the total running time of 500 repetitions.**

|                                      | Macro F1 (Mean $\pm$ Std) | Gain over raw feat. (Mean) | Time in sec. (Sum) |
|--------------------------------------|---------------------------|----------------------------|--------------------|
| Stratified                           | 0.248 $\pm$ 0.024         | —                          | 3                  |
| Raw feat.                            | 0.578 $\pm$ 0.005         | —                          | 116                |
| Raw feat. + TyE-DV                   | 0.600 $\pm$ 0.005         | 3.8%                       | 130                |
| Raw feat. + GDV                      | 0.597 $\pm$ 0.008         | 3.3%                       | 138                |
| Raw feat. + ColoredE-GDV             | 0.608 $\pm$ 0.006         | 5.2%                       | 2091               |
| Raw feat. + HeteroE-GDV <sup>N</sup> | 0.638 $\pm$ 0.006         | 10.4%                      | 8230               |
| Raw feat. + TyE-GDV                  | 0.619 $\pm$ 0.004         | 7.1%                       | 252                |

<https://doi.org/10.1371/journal.pone.0273609.t003>

performance with different feature sets. Plus, we repeat the above step 500 times and report the mean metric score given the stochastic nature of decision tree-based models.

Table 3 lists the prediction results for six models. As this is a multi-class classification task, and the distribution of the four classes is imbalanced, the macro-F1 score is selected as the evaluation metric. A naive classifier named Stratified is also added to the table (the first row), which simply generates predictions by adhering to the class distribution in the training set. We see clearly that the bottom three approaches that encode type information in graphlets (raw features plus ColoredE-GDV, raw features plus HeteroE-GDV<sup>N</sup>, and raw features plus TyE-GDV) perform better than the set of raw features plus TyE-DV and the set of raw features plus GDV. Recall that TyE-DV captures edge type information but with very limited structural information, and GDV, on the other hand, captures the rich structural information but without edge type information. This evidently shows that combining edge type information and rich structural information could lead to more distinctive features in network learning tasks.

We also observe large differences in the running time of those methods. The running time of the set of raw features plus ColoredE-GDV, and especially the set of raw features plus HeteroE-GDV<sup>N</sup> are many times higher than other methods. This is because our dataset has 13 types of edges and the lengths of vectors generated from these two methods grow near exponentially with the number of edge types  $|\mathcal{T}_e|$ . Correspondingly, the speed of the machine learning algorithm will slow down as the feature vector becomes larger. Table 4 gives the vector lengths of all five approaches. Note that there is no edge type information between alter nodes in many egocentric networks, including this case study dataset. Thus, our implementations of ColoredE-GDV and HeteroE-GDV<sup>N</sup> have excluded all the impossible combinations. Overall speaking, the proposed TyE-GDV is able to achieve a competitive performance while maintaining a small vector length.

## 6.4 Limitations and future directions

Here, we describe some limitations of this work and outline how these might be overcome in future studies.

**Edge direction.** Our current work is limited to undirected networks. To encode edge type information in directed networks, a natural extension of our approach is to apply the notion of directed graphlets [45–47]. The potential approach would be more complex due to the larger

**Table 4. Comparison of vector length of different approaches.**

| Approach       | GDV | TyE-DV | TyE-GDV | ColoredE-GDV | HeteroE-GDV |
|----------------|-----|--------|---------|--------------|-------------|
| Len. of vector | 7   | 13     | 91      | 12367        | 38870       |

<https://doi.org/10.1371/journal.pone.0273609.t004>

number of directed node-orbit graphlets. For example, even without considering bidirectional edges, there are in total 40 directed graphlets and 128 directed node orbits for graphlets of 2 to 4 nodes [45].

**Temporal information.** The proposed approach is static or time-independent. To make it suitable for more real-world networks that have nodes and edges appearing and disappearing over time, a potential future work would be studying how to encode edge type or node type information in temporal graphlets [48]. With the extra dimension of time, the potential extension could be beneficial in predicting types of future links or nodes [49, 50].

**Potential applications.** Apart from social networks, the typed edge graphlets approach could be convenient in studying biological networks, especially molecular graphs, where link attributes or bond types are essential information. The proposed approach is promising to be applied in biological network alignment, which aims to find a node mapping between molecular networks that reveals similar network regions [20, 51]. Moreover, inspired by recent works that include subgraph counting in Graph Neural Networks [52, 53], an interesting avenue is to incorporate the edge type enhanced structural information in GNN's message passing scheme.

## 7 Conclusion

In this paper, we propose to encode edge type information in graphlets and introduce the framework for generating the Typed-Edge Graphlets Degree Vector for both sociocentric and egocentric networks. Moreover, we extended the colored graphlets approach and the heterogeneous graphlets approach to edge-typed networks and egocentric networks. Following the application of the traditional graphlet degree vector and the proposed TyE-GDV to the chronic pain patient dataset, we discover that 1) a patient's social network structure could inform their perceived pain; and 2) the extra edge type information encoded in TyE-GDV provides us with more insights into the association between specific social relationships and patients' perception of pain.

We also showed that the rich structural information combined with the edge type information results in a significant improvement of a typical machine learning task that predicts patients' pain grades. Due to the simplicity and excellent explainability, we anticipate that the typed edge graphlets approach would become a standard approach in studying edge-attributed networks and be applied in various tasks.

## Acknowledgments

The authors thank the editors and anonymous reviewers for their excellent comments and suggestions. The authors would also thank Volker Ahlers and Yu-Xuan Qiu for their helpful comments and discussions.

## Author Contributions

**Conceptualization:** Mingshan Jia, Maité Van Alboom, Katarzyna Musial.

**Data curation:** Maité Van Alboom, Liesbet Goubert, Piet Bracke.

**Formal analysis:** Mingshan Jia.

**Funding acquisition:** Maité Van Alboom, Liesbet Goubert, Piet Bracke, Bogdan Gabrys, Katarzyna Musial.

**Investigation:** Mingshan Jia, Maité Van Alboom.

**Methodology:** Mingshan Jia, Maité Van Alboom, Bogdan Gabrys, Katarzyna Musial.



**Project administration:** Mingshan Jia, Liesbet Goubert, Piet Bracke, Bogdan Gabrys, Katarzyna Musial.

**Resources:** Bogdan Gabrys, Katarzyna Musial.

**Software:** Mingshan Jia.

**Supervision:** Liesbet Goubert, Piet Bracke, Bogdan Gabrys, Katarzyna Musial.

**Validation:** Mingshan Jia.

**Visualization:** Mingshan Jia.

**Writing – original draft:** Mingshan Jia.

**Writing – review & editing:** Maité Van Alboom, Liesbet Goubert, Piet Bracke, Bogdan Gabrys, Katarzyna Musial.

## References

1. Barabási AL. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2013; 371(1987):20120375. <https://doi.org/10.1098/rsta.2012.0375> PMID: 23419844
2. Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU. Complex networks: Structure and dynamics. *Physics Reports*. 2006; 424(4-5):175–308. <https://doi.org/10.1016/j.physrep.2005.10.009>
3. Zhu S, Yu K, Chi Y, Gong Y. Combining content and link for classification using matrix factorization. In: *Proceedings of the 30th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*; 2007. p. 487–494.
4. Huang X, Li J, Hu X. Label informed attributed network embedding. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*; 2017. p. 731–739.
5. Huang X, Li J, Hu X. Accelerated attributed network embedding. In: *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM; 2017. p. 633–641.
6. Gao S, Denoyer L, Gallinari P. Temporal link prediction by integrating content and structure information. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*; 2011. p. 1169–1174.
7. Cui G, Zhou J, Yang C, Liu Z. Adaptive graph encoder for attributed graph embedding. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2020. p. 976–985.
8. Perozzi B, Akoglu L, Iglesias Sánchez P, Müller E. Focused clustering and outlier detection in large attributed graphs. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2014. p. 1346–1355.
9. Ding K, Li J, Bhanushali R, Liu H. Deep anomaly detection on attributed networks. In: *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM; 2019. p. 594–602.
10. Meng Z, Liang S, Zhang X, McCreddie R, Ounis I. Jointly learning representations of nodes and attributes for attributed networks. *ACM Transactions on Information Systems (TOIS)*. 2020; 38(2):1–32. <https://doi.org/10.1145/3377850>
11. Lü L, Medo M, Yeung CH, Zhang YC, Zhang ZK, Zhou T. Recommender systems. *Physics Reports*. 2012; 519(1):1–49. <https://doi.org/10.1016/j.physrep.2012.02.006>
12. Perry BL, Pescosolido BA, Borgatti SP. *Egocentric network analysis: Foundations, methods, and models*. Cambridge University Press; 2018.
13. Jia M, Alboom MV, Goubert L, Bracke P, Gabrys B, Musial K. Analysing Ego-Networks via Typed-Edge Graphlets: A Case Study of Chronic Pain Patients. In: *International Conference on Complex Networks and Their Applications*. Springer; 2021. p. 514–526.
14. Pržulj N, Corneil DG, Jurisica I. Modeling interactome: scale-free or geometric? *Bioinformatics*. 2004; 20(18):3508–3515. PMID: 15284103
15. Milenković T, Pržulj N. Uncovering biological network function via graphlet degree signatures. *Cancer Informatics*. 2008; 6:257–273. PMID: 19259413
16. Zhang L, Song M, Liu Z, Liu X, Bu J, Chen C. Probabilistic graphlet cut: Exploiting spatial structure cue for weakly supervised image segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2013. p. 1908–1915.

17. Ataei S, Attar N, Aliakbary S, Bakouie F. Graph theoretical approach for screening autism on brain complex networks. *SN Applied Sciences*. 2019; 1(9):1–4. <https://doi.org/10.1007/s42452-019-1079-y>
18. Teso S, Staiano J, Lepri B, Passerini A, Pianesi F. Ego-centric graphlets for personality and affective states recognition. In: *SocialCom*. IEEE; 2013. p. 874–877.
19. Van Alboom M, De Ruddere L, Kindt S, Loeys T, Van Ryckeghem D, Bracke P, et al. Well-being and Perceived Stigma in Individuals With Rheumatoid Arthritis and Fibromyalgia: A Daily Diary Study. *The Clinical Journal of Pain*. 2021; 37(5):349–358. <https://doi.org/10.1097/AJP.0000000000000929> PMID: [33734147](https://pubmed.ncbi.nlm.nih.gov/33734147/)
20. Gu S, Johnson J, Faisal FE, Milenković T. From homogeneous to heterogeneous network alignment via colored graphlets. *Scientific Reports*. 2018; 8(1):1–16. <https://doi.org/10.1038/s41598-018-30831-w> PMID: [30131590](https://pubmed.ncbi.nlm.nih.gov/30131590/)
21. Rossi RA, Ahmed NK, Carranza A, Arbour D, Rao A, Kim S, et al. Heterogeneous graphlets. *ACM Transactions on Knowledge Discovery from Data (TKDD)*. 2020; 15(1):1–43. <https://doi.org/10.1145/3418773>
22. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. *Computer*. 2009; 42(8):30–37. <https://doi.org/10.1109/MC.2009.263>
23. Shah N, Beutel A, Hooi B, Akoglu L, Gunnemann S, Makhija D, et al. Edgecentric: Anomaly detection in edge-attributed networks. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). IEEE; 2016. p. 327–334.
24. Sun G, Zhang X. A novel framework for node/edge attributed graph embedding. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer; 2019. p. 169–182.
25. Nickel M, Kiela D. Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems*. 2017; 30.
26. Chen M, Quirk C. Embedding edge-attributed relational hierarchies. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*; 2019. p. 873–876.
27. Hočevar T, Demšar J. Computation of graphlet orbits for nodes and edges in sparse graphs. *Journal of Statistical Software*. 2016; 71:1–24.
28. Ribeiro P, Silva F. Discovering colored network motifs. In: *Complex networks V*. Springer; 2014. p. 107–118.
29. Ribeiro P, Silva F. G-tries: an efficient data structure for discovering network motifs. In: *Proceedings of the 2010 ACM Symposium on Applied Computing*; 2010. p. 1559–1566.
30. Šubelj L, Bajec M. Model of complex networks based on citation dynamics. In: *Proceedings of the 22nd International Conference on World Wide Web*; 2013. p. 527–530.
31. Stark TH, Krosnick JA. GENSI: A new graphical tool to collect ego-centered network data. *Social Networks*. 2017; 48:36–45. <https://doi.org/10.1016/j.socnet.2016.07.007>
32. Von Korff M, Ormel J, Keefe FJ, Dworkin SF. Grading the severity of chronic pain. *Pain*. 1992; 50(2):133–149. [https://doi.org/10.1016/0304-3959\(92\)90154-4](https://doi.org/10.1016/0304-3959(92)90154-4) PMID: [1408309](https://pubmed.ncbi.nlm.nih.gov/1408309/)
33. Karayannis NV, Baumann I, Sturgeon JA, Melloh M, Mackey SC. The impact of social isolation on pain interference: a longitudinal study. *Annals of Behavioral Medicine*. 2019; 53(1):65–74. <https://doi.org/10.1093/abm/kay017> PMID: [29668841](https://pubmed.ncbi.nlm.nih.gov/29668841/)
34. Ferreira-Valente MA, Pais-Ribeiro JL, Jensen MP. Associations between psychosocial factors and pain intensity, physical functioning, and psychological functioning in patients with chronic pain: a cross-cultural comparison. *The Clinical Journal of Pain*. 2014; 30(8):713–723. <https://doi.org/10.1097/AJP.000000000000027> PMID: [24042349](https://pubmed.ncbi.nlm.nih.gov/24042349/)
35. Evers AW, Kraaimaat FW, Geenen R, Jacobs JW, Bijlsma JW. Pain coping and social support as predictors of long-term functional disability and pain in early rheumatoid arthritis. *Behaviour Research and Therapy*. 2003; 41(11):1295–1310. [https://doi.org/10.1016/S0005-7967\(03\)00036-6](https://doi.org/10.1016/S0005-7967(03)00036-6) PMID: [14527529](https://pubmed.ncbi.nlm.nih.gov/14527529/)
36. Bannon S, Greenberg J, Mace RA, Locascio JJ, Vranceanu AM. The role of social isolation in physical and emotional outcomes among patients with chronic pain. *General Hospital Psychiatry*. 2021; 69:50–54. <https://doi.org/10.1016/j.genhosppsych.2021.01.009> PMID: [33540223](https://pubmed.ncbi.nlm.nih.gov/33540223/)
37. Ribeiro JLP. Escala de satisfação com o suporte social (ESSS). *Análise Psicológica*. 1999; 3:547–558.
38. Hahn EA, DeVellis RF, Bode RK, Garcia SF, Castel LD, Eisen SV, et al. Measuring social health in the patient-reported outcomes measurement information system (PROMIS): item bank development and testing. *Quality of Life Research*. 2010; 19(7):1035–1044. <https://doi.org/10.1007/s11136-010-9654-0> PMID: [20419503](https://pubmed.ncbi.nlm.nih.gov/20419503/)

39. Jia M, Van Alboom M, Goubert L, Bracke P, Gabrys B, Musial K. Analysing Egocentric Networks via Local Structure and Centrality Measures: A Study on Chronic Pain Patients. In: 2022 International Conference on Information Networking (ICOIN). IEEE; 2022. p. 152–157.
40. Forgeron PA, McGrath P, Stevens B, Evans J, Dick B, Finley GA, et al. Social information processing in adolescents with chronic pain: My friends don't really understand me. *Pain*. 2011; 152(12):2773–2780. <https://doi.org/10.1016/j.pain.2011.09.001> PMID: 21963240
41. Yang Y, Grol-Prokopczyk H. Chronic pain and friendship among middle-aged and older us adults. *The Journals of Gerontology: Series B*. 2021; 76(10):2131–2142. <https://doi.org/10.1093/geronb/gbaa185>
42. Harris S, Morley S, Barton SB. Role loss and emotional adjustment in chronic pain. *Pain*. 2003; 105(1-2):363–370. [https://doi.org/10.1016/S0304-3959\(03\)00251-3](https://doi.org/10.1016/S0304-3959(03)00251-3) PMID: 14499455
43. Bhagat S, Cormode G, Muthukrishnan S. Node classification in social networks. In: *Social Network Data Analytics*. Springer; 2011. p. 115–148.
44. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*. 2011; 12:2825–2830.
45. Sarajlić A, Malod-Dognin N, Yaveroğlu ÖN, Pržulj N. Graphlet-based characterization of directed networks. *Scientific Reports*. 2016; 6(1):1–14. <https://doi.org/10.1038/srep35098> PMID: 27734973
46. Aparicio D, Ribeiro P, Silva F. Extending the applicability of graphlets to directed networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2016; 14(6):1302–1315. <https://doi.org/10.1109/TCBB.2016.2586046> PMID: 27362986
47. Trpevski I, Dimitrova T, Boshkovski T, Stikov N, Kocarev L. Graphlet characteristics in directed networks. *Scientific Reports*. 2016; 6(1):1–8. <https://doi.org/10.1038/srep37057> PMID: 27830769
48. Hulovatyy Y, Chen H, Milenković T. Exploring the structure and function of temporal networks with dynamic graphlets. *Bioinformatics*. 2015; 31(12):i171–i180. <https://doi.org/10.1093/bioinformatics/btv227> PMID: 26072480
49. Yin Z, Gupta M, Weninger T, Han J. Linkrec: a unified framework for link recommendation with user attributes and graph structure. In: *Proceedings of the 19th International Conference on World Wide Web*; 2010. p. 1211–1212.
50. Gong NZ, Talwalkar A, Mackey L, Huang L, Shin ECR, Stefanov E, et al. Jointly predicting links and inferring attributes using a social-attribute network (san). *CoRR*. 2011;abs/1112.3265.
51. Ma L, Shao Z, Li L, Huang J, Wang S, Lin Q, et al. Heuristics and Metaheuristics for Biological Network Alignment: A Review. *Neurocomputing*. 2022; 491:426–441. <https://doi.org/10.1016/j.neucom.2021.08.156>
52. Bouritsas G, Frasca F, Zafeiriou SP, Bronstein M. Improving graph neural network expressivity via sub-graph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2022. <https://doi.org/10.1109/TPAMI.2022.3154319> PMID: 35201983
53. Barceló P, Geerts F, Reutter J, Ryschkov M. Graph neural networks with local graph parameters. *Advances in Neural Information Processing Systems*. 2021; 34:25280–25293.