Friend Story Ranking with Edge-Contextual Local Graph Convolutions

Xianfeng Tang[†], Yozen Liu[‡], Xinran He[‡], Suhang Wang[§], Neil Shah[‡*]
Amazon[†], Snap Inc.[‡], The Pennsylvania State University[§],
xianft@amazon.com {yliu2, xhe2, nshah}@snap.com szw494@psu.edu

ABSTRACT

Social platforms have paved the way in creating new, modern ways for users to communicate with each other. In recent years, multiple platforms have introduced "Stories" features, which enable broadcasting of ephemeral multimedia content. Specifically, "Friend Stories," or Stories meant to be consumed by one's close friends, are a popular feature, promoting significant user-user interactions by allowing people to see (visually) what their friends and family are up to. A key challenge in surfacing Friend Stories for a given user, is in *ranking* over each viewing user's friends to efficiently prioritize and route limited user attention. In this work, we explore the novel problem of Friend Story Ranking from a graph representation learning perspective. More generally, our problem is a *link ranking* task, where inferences are made over *existing* links (relations), unlike common node or graph-based tasks, or link prediction tasks, where the goal is to make inferences about *non-existing* links. We propose ELR, an edge-contextual approach which carefully considers local graph structure, differences between local edge types and directionality, and rich edge attributes, building on the backbone of graph convolutions. ELR handles social sparsity challenges by considering and attending over neighboring nodes, and incorporating multiple edge types in local surrounding egonet structures. We validate ELR on two large country-level datasets with *millions* of users and *tens of millions* of links from Snapchat. ELR shows superior performance over alternatives by ≈ 8% and ≈ 5% error reduction measured by MSE and MAE correspondingly. Further generality, data efficiency and ablation experiments confirm the advantages of ELR.

CCS CONCEPTS

• **Information systems → Social networks**; • **Networks → Social media networks**; **Online social networks**; • **Computing methodologies → Neural networks**.

KEYWORDS

Social networks; graph neural networks; user modeling

ACM Reference Format:
Xianfeng Tang[†], Yozen Liu[‡], Xinran He[‡], Suhang Wang[§], Neil Shah[‡]. 2022. Friend Story Ranking with Edge-Contextual Local Graph Convolutions. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498398

[*]Work done when Xianfeng Tang is a PhD intern at Snap, before joining Amazon.

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498398

1 INTRODUCTION

In recent years, social platforms have become increasingly prevalent. Such platforms have revolutionized how users communicate with each other through new features which are enhanced by, and mutually benefit users in creating and maintaining friendships. "Stories" is one such extremely popular feature, initially introduced by Snapchat in 2013[1], which enables each user to broadcast user-generated multimedia (photos and short videos) to their friends and other viewers. Stories on Snapchat are ephemeral, and last for a short duration (24 hours), promoting frequent and ephemeral communication in the form of visual status updates. Since their inception, many social platforms such as Instagram, Twitter, Weibo, etc. have incorporated similar concepts into their apps, increasing their prevalence and widespread use. One of the most prevalent forms of Stories on Snapchat is Friend Stories, in which the target audience for posted Stories are friends and family. Users can see Friend Stories from their friends, each of which contains an ephemeral reel of content. Shown in Figure 1, these appear as circles with snapshots on each user's Discover page (left), each of which can be clicked and expanded to show the content reel (right). Since each user is exposed to content from possibly many friends, attention routing and prioritization becomes an important facet in promoting engagement. Our work focuses on this task of algorithmic Friend Story Ranking (FSR) with machine learning models.

Our problem is an instantiation of a *link ranking* task, of ranking relations for each user by affinity. Although typical modeling approaches for tabular data could be used (gradient boosting, feedforward networks) for this problem, we approach it here instead from the perspective of graph representation learning. Recently, graph neural networks (GNNs), which learn node representations via convolution over node features and graph topology, have emerged as a prevalent modeling paradigm for graph data. GNNs have shown promising results on several graph learning tasks, such as node classification [16, 24], link prediction [30, 61] and social network analysis [43, 49, 57]. However, their direct application for our problem is impeded by a few factors: (i) Most prior work in GNNs focuses on node and graph classification [16, 24, 54]; although a few works tackle prediction of *missing links* [60, 61], this is a different context than ours, which involves ranking over *existing links*. The link ranking task is central to our learning objective, and is mostly unexplored in prior work. (ii) Rich user-user interactions are common in social data, and key in relation measurement [22]; yet, most existing GNN frameworks are not proposed to handle edge-level attributes [16, 24, 52]. (iii) The exploration of GNNs for large-scale social graph learning problems

[1]https://snap.com/en-US/news/post/surprise

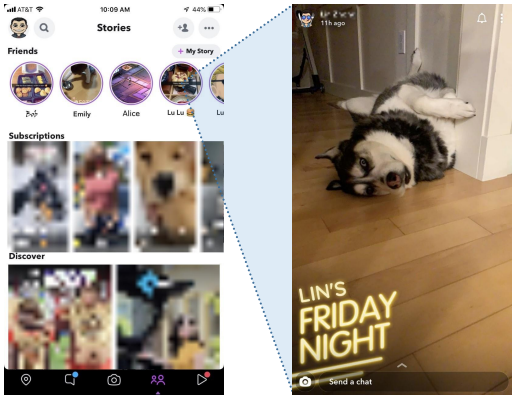

Figure 1: An example of Friend Stories on Snapchat. Friend Stories are displayed as a ranked list of circles (friends) on the "Stories" page (left). Tapping on a circle opens the Friend's Story, containing the friend's user-generated photos and videos about their activities (right).

is still quite limited [43, 49, 57], and subject to real-world challenges, including social graph structure and user interaction sparsity.

Our work discusses the use of GNNs in pursuit of solving the FSR problem. We first formulate our goal as pointwise prediction of (directed) Friend Story affinity between two users. In our model, we expand the perception of ranking a given link beyond the link in question, such that Friend Story affinity between two users is predicted as a function of not only their activities, but also based on interactions and influence from their local communities. To this end, we differentiate multiple types of relationships (pairwise, intra-ego, and inter-ego) in the union of the two users' egonets and consider each of their contributions, governed via self-attention over neighbors, improving learning over sparse user activity. We also consider edge attributes as a key component of message passing during aggregation, with the goal of learning rich representations for each user, link, and relation type, utilizing them to predict directed Friend Story affinity for ranking. Intuitively, our enhancements enable us to well-utilize local community information between the two user's friends and shared relations to overcome potential interaction sparsity and enhance inference quality. We propose a novel framework realizing these ideas, called ELR (Edge-contextual Local Graph Convolutions for Friend Story Ranking. We evaluate ELR using two large-scale datasets using country-level data from Snapchat. We show that ELR outperforms various tabular and graph-aware baselines across datasets and error metrics. We also study the generality of ELR, showing outperformance on multiple FSR-related objectives. Further ablation studies confirm the value of each of ELR' components. Lastly, we show some qualitative results demonstrating the importance of neighbor attention in our design.

In summary, our contributions are as follows:

- We study the Friend Story Ranking (FSR) problem, with application to different social platforms, using Graph Neural Networks (GNNs). FSR is a link ranking problem, differentiating it from most prior GNN-related work.
- We propose a novel model, ELR, which (i) utilizes local neighborhood information, (ii) heavily leverages edge attributes (ii) differentiates relation types, and (iv) attends over neighbors to learn predict Friend Story affinity.
- We experimentally show ELR's capacity for error reduction (up to 8%) against other approaches, generality and careful design on two large-scale graph datasets from Snapchat with *millions* of users and edges and *hundreds* of node and edge attributes.

We note that although ELR was designed for FSR, its intuition extends naturally to diverse link ranking problems, which the community may explore at large in future years.

2 RELATED WORK

We discuss related work in three areas: graph representation learning, edge-based inference, and social network analysis.

Graph representation learning. Graph representation learning (GRL) aims to learn representations suitable for graph-based tasks, mainly including node/graph classification [21, 48, 50, 62], and link prediction [17]. Early methods focused on non-attributed graphs, leveraging insights from language modeling [39] to learn embeddings which preserve node co-occurrence statistics on random walks [40]. Several future works explored variations including biased random walks [15], multiple orders of proximity [47, 53], attribute awareness [8], and more. [41] unified such methods as matrix factorization problems with nuanced differences. In recent years, graph neural networks (GNNs) have risen to prominence as a flexible modeling solution for attributed graphs. GNNs can be viewed from both spectral and spatial perspectives as convolutions of node features over graph topology [5], with components of feature transformation and aggregation [36]. Multiple architectures have been proposed in recent years, which adapt the aggregation scheme. Some notable architectural ideas include mean-pooling and first-order approximations [24], generalized aggregation, feature concatenation and inductive usage [16], adaptive aggregation via self-attention on node features [52], jump connections [54], and more. Several other works extend these ideas to graph-level representations via pooling [37, 58]. Most of these works focus on academic settings, with smaller datasets on well-defined, homophilic node classification tasks; only few works apply GNNs to large-scale industrial tasks [43, 49, 57], and these do not focus on link ranking as ours does.

Link-Based Inference. Several methods focus on inferences on edges, or links. The main line of edge-based inference is on link prediction (LP), which aims to predict missing (or future) links given existing graph information [34]. LP is a common problem in social networks, where it often manifests as a friend recommendation task. Early work in LP used unsupervised heuristics, such as common neighbors [56], centrality and random walks [31]. Later works used pair-based features to train traditional supervised models. In recent years, neural methods for modeling pairwise associations have become popular, such as neural collaborative filtering [18] and other feedforward architectures [6]. Many GRL methods [15, 40] also naturally produce edge probabilities between pairs of nodes, and GNN-based methods common often equip inner product decoders on top of node representations to train on LP tasks [16, 25]. A few works use more complex decoders; [1, 28] propose using asymmetric projections on learned node embeddings, while [12] suggests learning node embeddings conditioned on edge type. [60, 61] recently proposed methods for LP via enclosing subgraph classification on academic data, but suffers in scalability. Unlike these works, ours notably uses link attributes in learning, focuses on link ranking rather than LP, and uses a GNN encoder.

Social Network Analysis. Social network analysis literature coincides with graph mining literature on topics including graph clustering [4, 7, 46], community detection [3] anomaly detection [2, 9, 45], engagement modeling [44, 49, 55] and pattern or motif mining [27, 42]. Many works also focus on characterization and behavioral understanding in social networks. Different disciplines from sociology, computational social science and machine learning have yielded shared insights on the value of learning from "friends of friends" relationships. For example, [38, 59] observe homophilic tendencies between users across multiple modalities in social platforms. [19, 33] amongst others discuss the value of social triadic closure in link formation, and [26, 29] demonstrate evidence of supporting structural balance theory in signed networks. Other works in social sciences demonstrate the evident impact of users' local communities on success of relationships [23]. Several works have also shown the value of utilizing neighborhood information to enhance user-item recommendation [11, 20, 35], based on the idea that similar users have similar preferences. Our work incorporates these ideas of neighbor preference smoothness more directly to the task of affinity prediction via a GNN-based model.

3 PRELIMINARIES

Social Network: We consider a social network to be described by a graph $\mathcal{G}$, where nodes are defined over a set of registered users $\mathcal{V}$, and edges are defined over dyadic ties between users (commonly called edges or links) $\mathcal{E}$, reflecting friendships on the platform. We also consider that each node is endowed with d_u features, and each edge with d_e features. We denote the node features as a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d_u}$, and the edge features as a (sparse) tensor $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}| \times d_e}$. We write $\mathbf{x}_u$ to indicate the node features of a user u (a single row from $\mathbf{X}$), and $\mathbf{e}_{uv}$ to indicate the edge features on directed edge $u \rightarrow v$ (denoted as (u,v) as an ordered tuple, for brevity). Although all friendships are bidirectional in our case (if $u \rightarrow v$ exists, $v \rightarrow u$ also exists), we consider $\mathcal{G}$ to be a directed graph, since relationships are directed and two users may have asymmetric interactions. As such, $\mathbf{e}_{uv} \neq \mathbf{e}_{vu}$ in general.

Node features on users typically include information about how users use the app, such as login frequency, aggregate engagement habits, ad consumption, app tenure, etc. Edge features include information about social interactions, like sending and viewing direct messages (Chats, Snaps), watching each others' Friend Stories, etc. Although social network graphs are in general dynamic given users change engagement patterns, new users are added and removed, and new friendships formed and removed (i.e. $\mathcal{G}$,$\mathbf{X}$,$\mathbf{E}$ all change over time), we simplify the problem here by fixing the state of all three at time t. In practice, the model we will describe can be re-trained over time accordingly to maintain its accuracy and accommodate the evolving social network.

Ego Network: Let u denote a registered user. We use $\mathcal{N}(u)$ to represent the set of friends of u, or formally, $\{v \in \mathcal{V} || (u,v) \in \mathcal{E}\}$. The ego network (egonet) [10] of u is a subgraph of the whole social network graph $\mathcal{G}$. Nodes in the egonet include the focal node (ego) u and neighbors $\mathcal{N}(u)$ (alters). The egonet inherits all edges that are between any two nodes in $\{u\} \cup \mathcal{N}(u)$ Formally, the set of edges is $\{((u,v) \in \mathcal{E}) \cup ((v,u) \in \mathcal{E}) | v \in u \cup \mathcal{N}(u)\}$.

Friend Story Affinity: Friend Story affinity measures the possibility of a user engaging with a friend's Stories. To quantify the engagement, we define the *affinity score*, $s_{uv} \in (0,1)$ for directed relation (u,v), where larger s_{uv} implies a higher engagement. Note that s_{uv} and s_{vu} are different. A high engagement on v's stories by u does not indicate u's stories are also attractive to v. Solving the FSR problem of u can be simplified as sorting the Friend Story engagement scores for all $v \in \mathcal{N}(u)$. In particular, let v_1 and v_2 denote two friends of u, if $s_{uv_1} > s_{uv_2}$, then the story from v_1 should be ranked higher than the story from v_2. This is effectively a pointwise learning-to-rank task, where u is the query, and $\mathcal{N}(u)$ are the results. Various empirically observed metrics of interest can be selected to define s_{uv}, such as the click-through rate (CTR) for Friend Story (how many times did u view v's Friend Story normalized by the number of total opportunities or impressions), or the total view time (TVT) that u spends watching v's stories.

We define the Friend Story Ranking (FSR) problem as follows:

Problem (Friend Story Ranking). *Given graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, node features $\mathbf{X}$ and edge features $\mathbf{E}$, design a model to predict the future Friend Story affinity score s_{uv} for each $(u,v) \in \mathcal{E}$.*

More generally, we can consider a generalized link ranking problem by replacing the Friend Story specificity of the affinity score with a more general link-based affinity score. Note that we generally consider the observed edge features $\mathbf{e}_{uv}$ to contain information (explicitly or implicitly) about previously observed affinity s_{uv}; we clarify that observed $\mathbf{e}_{uv}$ are from past interactions, while our goal is to predict s_{uv} for a future time. Upon solving the problem, we can sort all relevant s_{uv} for $(u,v) \in \mathcal{E}$, keyed by u offline. In practice, we technically only need to make inferences for each $v \in \mathcal{N}(u)$ that actually has posted a Friend Story at the time when u views the Discover page; if v does not post a Friend Story, they cease to be a ranking candidate. Note that generating predictions for $(u,v) \in \mathcal{E}$ imposes a total ordering, which allows us to trivially select eligible, relevant candidates at the time of ranking.

4 OUR APPROACH: ELR

A straightforward way to approach the FSR problem would be to learn a tabular model over the edge features $\mathbf{e}_{uv}$. While conceptually simple, this approach falls short by missing the opportunity to use social information outside the relation of interest (u,v) to enhance our prediction. Thus, we approach the FSR problem from a graph representation learning perspective, which allows us to incorporate information from neighboring nodes, edges and their features into the inference. We reason that, intuitively, other social factors may govern the prediction of s_{uv}. For example, if u is a new user who has never watched v's Friend Story in the past, but several of u's friends enjoy v's Friend Stories, this improves our assessment of the future s_{uv}. Additionally, knowing that u's *best friend* enjoys v's Friend Story is especially informative, compared to a casual acquaintance u does not interact with. More generally, u and v may share some friends, and their relations could also mediate our estimate of s_{uv}. We utilize all these insights in designing our model, ELR.

Figure 2 illustrates ELR's design, which consists of three modules including local graph encoding, friendship categorization, and affinity score prediction. The local graph encoding module leverages a GNN-inspired neural network to learn representations for both users and relations in the local community graph; the friendship categorization module differentiates three types of relations including *pairwise relation*, *intra-ego relation* and *inter-ego relation* according to their relative locations and status in the local graph. Finally, the affinity

score prediction module jointly utilizes the three types of relations to predict the affinity score s_{uv}. Next, we introduce ELR in detail.

4.1 Edge-Contextual Graph Encoding

As aforementioned, predicting Friend Story affinity purely based on (u,v) neglects contextual information in the social network. We first introduce the concept of *edge-contextual graph* as an augmented input which expands our perception from pairwise relations to the context-providing surrounding edges.

4.1.1 Edge-Contextual Graph. The edge-contextual graph is a subgraph of the social network based on the egonets of the source and target nodes. We define it as

Definition 4.1 (Edge-Contextual Graph). *Given an edge $(u,v) \in \mathcal{G}$, we define its edge-contextual graph $\mathcal{G}_{uv}$ as the induced subgraph over nodes in the egonets of u and v. Formally, it contains the nodes $\mathcal{V}_{uv} = \mathcal{N}(u) \cup \mathcal{N}(v)$ and edges $\mathcal{E}_{uv} = \{((i,j) \in \mathcal{E}) \cup ((j,i) \in \mathcal{E}) | i \in (\mathcal{N}(u) \cup \mathcal{N}(v)) \wedge j \in (\mathcal{N}(u) \cup \mathcal{N}(v))\}$.*

Informally, the node set contains all users that are friends of u or v; this includes both u and v, since $u \in \mathcal{N}(v)$ and $v \in \mathcal{N}(u)$ by construction. The edge set includes all relations between u, v, and both their friends. While egonets focus on the local community centered around a single node, the edge-contextual graph, $\mathcal{G}_{uv}$ provides an expanded view of network relations with (u,v) as the focal point. Note that $\mathcal{G}_{uv}$ is not quite the same as the union of egonets for u and v, since it also includes all edges between u's friends and v's friends, but is similar in its intent to capture local community structure as a subgraph of $\mathcal{G}$. In the remainder of this section, as we build towards making inferences for s_{uv}, we utilize only the graph structure encompassed in edge-contextual graph $\mathcal{G}_{uv}$, with globally shared node and edge features $\mathbf{X}$,$\mathbf{E}$, rather than $\mathcal{G}$ as a whole. Specifically, ELR takes edge-contextual graph as an instance of input when making inferences for s_{uv}. We find this choice is advantageous, as it allows us to ignore spurious and irrelevant links too far from the edge (u,v) in question, and limits the accesses required when producing an inference for any edge.

In practice, we construct node features $\mathbf{X}$ and edge features $\mathbf{E}$ from historical user activities and user-user interactions respectively, using summary statistics such as number of Snaps/Chats sent, total session time, etc. A list of representative features are given in supplementary material. We note that $\mathcal{G}_{uv}$ contains rich edge features which characterize link behaviors. Since the affinity score s_{uv} is defined on links, modeling edge features properly is essential.

To capture complex associations between nodes, their features and interactions, our first step is to encode the edge-contextual graph into Euclidean space. To this end, we utilize a GNN-based encoding scheme. We adopt GraphSAGE [16] as our GNN model because of its scalability in industrial scenarios [57]. In particular, a GraphSAGE layer is defined as follows:

$$\mathbf{h}_u^l = \sigma(\mathbf{W} \cdot \texttt{AGG}(\{\mathbf{h}_u^{l-1}\} \cup \{\mathbf{h}_v^{l-1}, \forall v \in \mathcal{N}(u)\})), \tag{1}$$

where $\mathbf{h}_u^l$ is the node representation for user u from the l-th GraphSAGE layer ($\mathbf{h}_u^0 = \mathbf{X}$), and $\sigma(\cdot)$ represents a non-linear activation function, and $\mathbf{W}$ denotes neural network parameters (in our case, a single feedforward layer). $\texttt{AGG}(\cdot)$ is an aggregation function which pools information from u and u's neighbors to update the representation of u in the next layer. We choose the mean aggregator for $\texttt{AGG}(\cdot)$ because of its efficiency without sacrificing much performance. However, the original design of the mean aggregator is unable to handle edge features. Edge features characterize user interactions, and their utility has been shown in accurately predicting link-based affinity scores in deployed ML models internally. Therefore, it is imperative to preserve such information when learning representations using GNNs. Inspired by [13, 14], we adapt the aggregation function of GraphSAGE as follows:

$$\mathbf{h}_u^l = \sigma(\mathbf{W} \cdot [\mathbf{h}_u^{l-1} \oplus \texttt{MEAN}(\{[\mathbf{h}_v^{l-1} \oplus \mathbf{e}_{uv}], \forall v \in \mathcal{N}(u)\})]), \tag{2}$$

where $\oplus$ stands for vector concatenation. Specifically, we first concatenate node representation $\mathbf{h}_v^{l-1}$ with the corresponding user-to-user features (i.e., edge feature $\mathbf{e}_{uv}$), as shown in Figure 2. We call this custom layer Edge-GNN. The concatenated vector is then passed to the mean aggregator for the center node. We then combine the node representation of the center node and the mean aggregated vector to construct the node representation $\mathbf{h}_u^l$. Advantageously, Edge-GNN is able to well-capture historical activities, user interactions and topological friendship structures in the representation learning process.

4.1.2 User Representation. We can generally capture multiple orders of graph proximity using multiple Edge-GNN layers. In practice, we use two Edge-GNN layers on the edge-contextual graph input to ensure that both first-order and second-order proximity are considered. The output node representations, denoted by $\{\mathbf{h}_u^2\}$, are treated as the representation of users in the edge-contextual graph. To simplify the manuscript, we omit the superscript 2 and use h_u to denote the representation for user u.

4.1.3 Link Representation. Given a link (u,v), we use the corresponding pairwise edge features $\mathbf{e}_{uv}$ and the associated user representations from the edge-contextual graph, $\mathbf{h}_u$ and $\mathbf{h}_v$ to characterize the link between users. In particular, the direction sensitive link representation for (u,v) is constructed as follows:

$$\mathbf{f}_{uv} = \phi([\mathbf{h}_u \oplus \mathbf{h}_v \oplus \mathbf{e}_{uv}]), \tag{3}$$

where $\phi(\cdot)$ is a mapping function (e.g., neural network layers), and $\mathbf{e}_{uv}$ represents the directional edge feature for $u \rightarrow v$. We empirically find that reusing edge features for link representation directly can improve the overall performance. One likely reason is that edge features are highly correlated to the affinity score s_{uv}. Thus, adding edge features after Edge-GNN could re-emphasize individual user interactions after smoothing via the mean aggregator.

Given user representations and link representations for the edge-contextual graph, ELR next aims to categorize various types of friendship relations, and learn their representations separately to capture the influence of different facets of the edge-contextual graph.

4.2 Friendship Categorization

To predict the Friend Story affinity score, we capture the similarity of Friend Story affinities between different pair of users in the edge-contextual graph. Our assumption is that the Friend Story affinity score s_{uv} is correlated to (1) the **pairwise relation** between u and v. Frequent historical engagement between u and v indicates u's likeliness to watch v's stories; (2) the **intra-ego relations**: u shares similar affinity to v's stories as his/her first-order friends; (3) the **inter-ego relations**: the level of interest from u's friends towards stories of v's friends can imply u's interest in v's stories, due to properties of homophily and shared preferences between friends. Compared to state-of-the-art models in industry that mainly utilize two users' historical activities and interactions, ELR especially

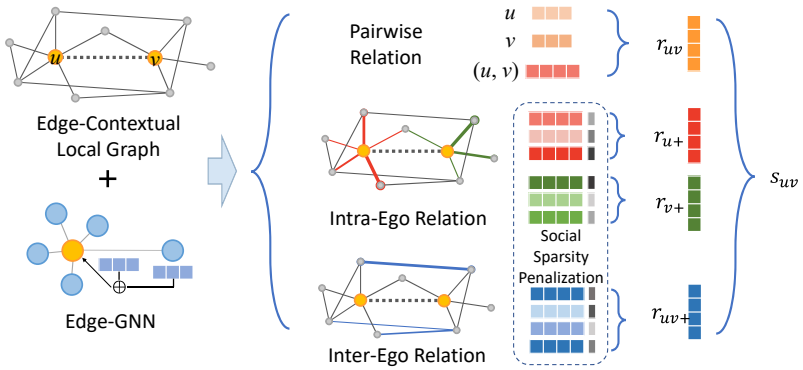

Figure 2: Overall framework of ELR: ELR considers an edge-contextual graph for each (u,v) of interest, and learns embeddings for each edge in the graph using an edge-attribute aware module, Edge-GNN. A friendship categorization module differentiates various relation types in the graph, and learns relation-specific embeddings for each: the target pairwise relation on (u,v) (r_{uv}), intra-ego relations between u and u's friends (r_{u+}) and v and v's friends (r_{v+}), and inter-ego relations across u's friends and v's friends (r_{uv+}), leveraging learned neighbor attention (grey) to counter inherent sparsity in social interactions. Finally, the affinity score prediction module jointly utilizes the relation-specific embeddings to predict the FSR affinity score, s_{uv}.

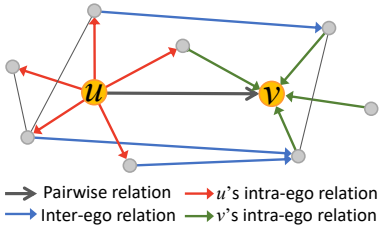

Figure 3: Examples of pairwise, inter-ego, and intra-ego relations in a edge-contextual graph. All relations are directed and aligned with respect to the pairwise relation, $u \rightarrow v$.

priorities local community information via edge-contextual graph signals, capturing transitive properties which help serve to regularize inferences via social structure. Figure 3 illustrates relations in a toy edge-contextual graph. Next, we discuss the detailed designs and intuitions for the above three category of relations.

4.2.1 Pairwise Relation. The pairwise relation of u and v characterizes the direct Friend Story engagement on $u \rightarrow v$. It consists of users' activities and user-to-user interactions. We leverage the user representations and the friendship representation from the encoded edge-contextual graph. As shown in Figure 2, we concatenate user representations $\mathbf{h}_u$, $\mathbf{h}_v$ and friendship representation $\mathbf{f}_{uv}$ and learn the pairwise relation from the link as:

$$\mathbf{r}_{uv} = \psi_{uv}(\mathbf{h}_u \oplus \mathbf{h}_v \oplus \mathbf{f}_{uv}), \tag{4}$$

where $\psi_{uv}(\cdot)$ is a fully-connected neural network layer.

Incorporating pairwise relations has been explored in state-of-the-art approaches. Predicting Friend Story affinity score based on pairwise relation is similar to feature-based link prediction [61]. However, in previous cases, the pairwise relation view is limited, and does not carefully consider the relationships between users and their ego network/local community.

4.2.2 Intra-Ego Relation. In addition to the "target" pairwise relation for (u,v), we also look at u's friendships and v's friendships. Intuitively, user behavior and interests are influenced by their friends because of their shared communications, and natural properties of homophily. Looking at immediate friends is beneficial and provide additional contextual information. For example, a user u may share similar affinity as his/her close friends to v's Stories. Therefore, we explicitly model interactions *within* each egonet. Specifically, the intra-ego relations consists of edges from u to u's friends ($u \rightarrow \mathcal{N}(u)$), and from v's friends to v ($\mathcal{N}(v) \rightarrow v$)), aligning directions with the target edge for inference (u,v). We then aim to define the intra-ego relation representation of u by aggregating edges between $u \rightarrow \mathcal{N}(u)$. A simple approach would be to apply mean aggregation on the corresponding link representations as follows:

$$\mathbf{r}_{u+} = \psi_{u+}\left(\frac{1}{|\mathcal{N}(u)|}\sum_{i\in\mathcal{N}(u)} \mathbf{f}_{ui}\right), \tag{5}$$

where $\psi_{u+}(\cdot)$ is a feedforward layer. However, this treats all of u's friends equally, and ignores the diversity of communication frequency and the closeness of friendships.

Despite having many friends, most users only interact frequently with several close friends and families. Figure 4 validates this claim: We randomly select $\approx$ 100K users from Snapchat, and analyze the percentage of interacted-with (via Chat) friends in past 3-day, 7-day, 14-day and 28-day time periods. If a user and their friend Chatted at least once during the time period, the friend is marked as "interacted". Evidently, most users only communicate with less than 20% of their on-platform friends in a two-week time period, and few users interact with more than 40% of their friends in a month (28 days), creating an interaction sparsity scenario. Therefore, simply averaging link representations of all friends could lead to noisy representations by considering many inactive friendships. Even though user-user interaction can be reflected by edge features, explicitly penalizing less communicated friends when aggregating has still been shown as beneficial [32, 49]. To appropriately characterize important friends by penalizing selected link representations when learning representations for intra-ego relations, we use a self-attention mechanism [51, 52] to assign friends among intra-ego relations different importance. Specifically, we compute the importance of friend i to u (and v to j) as:

$$\alpha_{ui} = \frac{\exp(\mathbf{h}_u^\top \mathbf{h}_i)}{\sum_{i'\in\mathcal{N}(u)}\exp(\mathbf{h}_u^\top \mathbf{h}_{i'})}, \ \alpha_{jv} = \frac{\exp(\mathbf{h}_j \mathbf{h}_v^\top)}{\sum_{j'\in\mathcal{N}(v)}\exp(\mathbf{h}_{j'} \mathbf{h}_v^\top)}. \tag{6}$$

where $\exp(\cdot)$ denotes the exponential function. The importance scores are further incorporated into the intra-ego relations for both u and v as:

$$\mathbf{r}_{u+} = \psi_{u+}\left(\sum_{i\in\mathcal{N}(u)} \alpha_{ui}\cdot\mathbf{f}_{ui}\right), \ \mathbf{r}_{v+} = \psi_{v+}\left(\sum_{j\in\mathcal{N}(v)} \alpha_{jv}\cdot\mathbf{f}_{jv}\right), \tag{7}$$

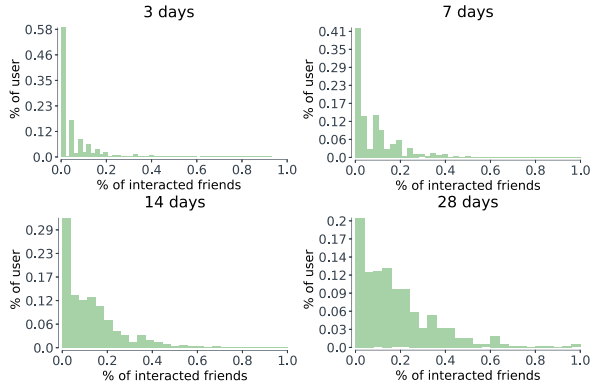

Figure 4: Percentage of interacted-with friends in the past 3, 7, 14 and 28 days. Most users only communicate with less than 20% of friends in the past 14 days. Few users interact with more than 40% of friends over the past 28 days.

where $\psi_{u+}(\cdot), \psi_{v+}(\cdot)$ are feedforward layers. As shown in Figure 2, two intra-ego relation representations highlight the immediate interactions along the direction of link (u,v), and are part of the input to the Friend Story engagement prediction module. Note that by considering directions of $\mathbf{r}_{u+}$ and $\mathbf{r}_{v+}$ asymmetrically, we are able to capture and attenuate information along the path $u \rightarrow f \rightarrow v$.

4.2.3 Inter-Ego Relations. Lastly, we model the friendships between friends of u and friends of v ($\mathcal{N}(u) \rightarrow \mathcal{N}(v)$), i.e., inter-ego network relations. The inter-ego aim to describe the engagement and interest of u's friends in v's friends. As friends share similarities, inter-ego relations are particularly helpful to discover underlying affinity. For example, if v is a less-active user whom neither u, u's friend nor v's friend have much interaction with recently, it may be challenging to predict the affinity score (u,v). However, if historical behaviors show strong engagements between u's friends and v's friends, our estimation of u's affinity to v would likely increase. More generally, we hypothesize that inter-ego friendships carry useful information of baseline engagement propensity between users in the two groups of u's friends and v's friends

Figure 5 bolsters our intuition between inter-ego relations and their association with (u,v): We randomly select 1 million edge-contextual graphs, and analyze the activeness of inter-ego relations for (u,v) with different Friend Story CTR as the measurement of engagement. We first compute the averaged Friend Story CTR value for all possible links/relations. If a friendship from inter-ego relations shows higher Friend Story CTR than average, we consider it an "active link", otherwise we consider it an "inactive link". The left plot shows that for (u,v) with low engagement (blue), the ratio of inactive inter-ego relations is markedly higher than (u,v) with high engagement, while the right plot shows clearly that high engagement (u,v) have considerably more active inter-ego relations. The active inter-ego relations, which frequently co-occur with high CTR for (u,v), is a valuable signal in solving the FSR problem.

Motivated by the above analysis, we construct inter-ego relation representations for all friendships between the alters of u and the alters of v ($\mathcal{N}(u) \rightarrow N(v)$) as follows:

$$\mathbf{r}_{uv+} = \psi_{uv+}\left(\sum_{\mathcal{E}^{\times}_{uv}} \alpha_{ui} \cdot \alpha_{jv} \cdot \mathbf{f}_{ij}\right), \tag{8}$$

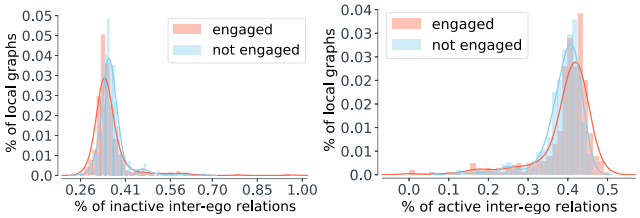

Figure 5: Percentage of active/inactive inter-ego relations according to the Friend Story engagement of (u,v). "engaged" means (u,v) with high engagement and "not engaged" denotes a low engagement for (u,v). We observe more active inter-ego relations for "engaged" (u,v); and more inactive inter-ego relations if (u,v) are "not engaged".

where $\psi_{uv+}(\cdot)$ is a feedforward layer, and $\mathcal{E}^{\times}_{uv} = \{(i,j) \in \mathcal{E}_{uv} | i \in \mathcal{N}(u) \wedge j \in \mathcal{N}(v) \wedge i \neq j\}$ denotes those cross-egonet edges in the edge-contextual graph which are between u's friends and v's friends. We reuse the importance of friends as introduced for intra-ego relations because of a similar intuition: when modeling friend-to-friend relations, the closeness of friends should be considered. Since the friendship representation $\mathbf{f}_{ij}$ is weighted by both users, the closeness of i to u and the closeness of v to j are incorporated. Specifically, $\mathbf{f}_{ij}$ is more reliable if i is a close friend to u, and also if j is a close friend to v. Using attention scores α_{ui} and α_{jv} to weight the link representation can reduce noise and learn a concentrated inter-ego representation for Friend Story affinity prediction.

4.3 Affinity Score Prediction

Lastly, we predict the Friend Story affinity score by jointly modeling the pairwise relation, intra-ego relations and inter-ego relations. To construct a complete view of the edge-contextual graph, We first concatenate the three types of representations, then use a neural network $\varphi(\cdot)$ to generate the predicted score:

$$\hat{s}_{uv} = \varphi(\mathbf{r}_{uv} \oplus \mathbf{r}_{u+} \oplus \mathbf{r}_{v+} \oplus \mathbf{r}_{uv+}). \tag{9}$$

We use mean square error as the loss function:

$$\mathcal{L} = \frac{1}{|\mathcal{E}|} \sum_{(u,v)} \|\hat{s}_{uv} - s_{uv}\|^2. \tag{10}$$

In practice, we utilize minibatching and stochastic gradient descent to train the model efficiently on large-scale graph data.

5 EXPERIMENTS

In this section, we conduct extensive experiments to answer the following experimental questions (EQs):

- **EQ1**: Can ELR accurately predict affinity scores for FSR, outperforming alternatives?
- **EQ2**: How do different components of ELR contribute to the resulting predictive performance?
- **EQ3**: Are ELR learned attention weights intuitive assessments of friend importance?
- **EQ4**: How does ELR perform with limited training data?

5.1 Setup

We construct two large-scale datasets from Snapchat, each spanning one country (i.e., Region 1 and Region 2). For each dataset, we take a snapshot of the social network structure on a specific date, and select all recently users active in each country. Detailed construction

of datasets are introduced in supplementary materials. We collect historical user activities and user-user interactions in the past four weeks from the specified date. Frequencies of each type of activity and interaction type are aggregated with the past 3/7/14/28 day intervals as node features and edge features. In all, we have $d_u = 376$ (node features) and $d_e = 188$ (edge features). Feature examples are further provided in supplementary materials. We then preprocess our data with feature-wise log-normalization and z-score normalization.

We primarily define the target Friend Story affinity score s_{uv} using click-through rate (CTR) on (u,v). CTR is formulated as the number of times u view v's stories divided by the total number of times v's stories were shown to u, where CTR $\in [0,1]$. In addition, to demonstrate the generality of ELR, we consider a second prediction task, where s_{uv} is defined using the total Friend Story view time (TVT). We also apply log-normalization and z-score normalization on TVT for preprocessing. We discuss further data statistics, data pre-processing, model implementation details and model training in supplementary materials.

5.2 Baselines

We compare ELR with the following baselines to validate its performance. In particular, we choose three pairwise feature based methods and two GNN methods.

- **Linear Regression** (LR): since the linear regression model cannot handle graph-structured data, we concatenate original node features $\mathbf{x}_u$,$\mathbf{x}_v$ and edge features $\mathbf{e}_{uv}$ as input to the model.
- **XGBoost** (XGB): We construct input similarly to LR, using a boosted trees model instead.
- **Multi-Layer Perceptron** (MLP): we build a two-layer MLP model, using the same input features to LR and XGB models.
- **GraphSAGE** (GS): we train a two-layer GraphSAGE model with mean aggregation on the entire network $\mathcal{G}$ to predict s_{uv} using an inner product decoder on final node embeddings. We only use node features with graph structure since [16] does not use them.
- **Edge-GraphSAGE** (E-GS): The same as GS, but we replace the message-passing in the original GraphSAGE with Equation 2 so that edge features are incorporated explicitly in the aggregation.

We note that the above GNN approaches treat FSR akin to link prediction. As discussed in Sections 1-2, our task involves link ranking, which is different in its focus on inference for existing links and not missing ones. However, we train the baseline approaches using ideas posited for link prediction for best comparison. Comparing ELR and these GNN approaches inherently shows differences in the inherent problem settings, and their design considerations.

We use mean square error (MSE) and mean absolute error (MAE) as the evaluation metrics for all compared methods. We treat the ranking task as a pointwise prediction one. In general, lower MSE/MAE also strongly correlate with traditional ranking metrics given our observations. We note that we could also adapt our task to a listwise or pairwise ranking setting, but do not discuss these aspects in our work for simplicity. Note that we do not directly predict/optimize mean reciprocal rank (MRR) and/or normalized discounted cumulative gain (NDCG) of the ranked friend stories. The production ranking model utilizes a multi-task setting where a value model is applied to combine different predictions like CTR and TVT to generate final ranking to satisfy business goals. As a result, we focus on the prediction accuracy of each individual task instead of overall ranking metrics. The accuracy improvement of each individual prediction task naturally results in improvement of the final recommendation quality (e.g., MRR, NDCG).

Table 1: Dataset summmary statistics.

Name	Region 1	Region 2
# of users ($\lvert\mathcal{V}\rvert$)	$\approx 1.2M$	$\approx 3.3M$
# of friendships ($\lvert\mathcal{E}\rvert$)	$\approx 20.4M$	$\approx 25.0M$
Node ftr dim (d_u)	376	376
Edge ftr dim (d_e)	188	188

Table 2: ELR outperforms baselines on the Friend Story CTR prediction task across datasets and metrics.

	Region 1		Region 2	
	MSE	**MAE**	**MSE**	**MAE**
LR	.1510±.0140	.4078±.0101	.1411±.0095	.3832±.0079
XGBoost	.0991±.0032	.2747±.0035	.0908±.0018	.2611±.0020
MLP	.1014±.0015	.2595±.0077	.0986±.0037	.2541±.0040
GS	.1115±.0027	.3038±.0043	.0965±.0033	.2866±.0032
E-GS	.0914±.0009	.2511±.0021	.0823±.0012	.2474±.0011
ELR	**.0865±.0011***	**.2389±.0019***	**.0773±.0020***	**.2404±.0032***

* significant with Student's T-test $p < 0.001$ compared with E-GS

Table 3: ELR is general, showing similar outperformance for an alternate Friend Story TVT prediction task.

	Region 1		Region 2	
	MSE	**MAE**	**MSE**	**MAE**
LR	.0730±.0060	.2492±.0032	.0683±.0044	.2357±.0050
XGBoost	.0434±.0014	.1830±.0010	.0496±.0025	.1943±.0022
MLP	.0414±.0008	.1852±.0009	.0435±.0012	.1823±.0015
GS	.0405±.0003	.1755±.0013	.0443±.0011	.1840±.0015
E-GS	.0396±.0004	.1707±.0017	.0428±.0009	.1817±.0006
ELR	**.0383±.0003***	**.1680±.0010***	**.0410±.0011***	**.1789±.0012***

* significant with Student's T-test $p < 0.001$ compared with E-GS

5.3 Prediction Performance

We study **EQ1** by comparing ELR against all baselines on predicting Friend Story CTR. The results are reported in Table 2, measured by mean square error (MSE) and mean absolute error (MAE). Clearly, ELR achieves the best performance, with both MSE and MAE markedly lower than baseline methods on both datasets. ELR significantly outperforms traditional tabular methods like LR, XGB and MLP on both datasets. Primarily because tabular methods fail to model the rich social network information outside of pairwise features. Although GS and E-GS are trained on the entire social network and provide more information than pairwise feature based approaches, their errors are still considerably higher than ELR. In particular, both E-GS and ELR leverage edge features to model user interactions for friend story ranking. However, ELR surpasses E-GS by almost 8%, suggesting that modeling intra-ego and inter-ego relations explicitly is key in an accurate prediction.

To validate the generality of the ideas adopted by ELR, we extend the task to TVT prediction on both datasets. The results are shown in Table 3, where ELR achieves the best performance against other baselines. In particular, ELR improves over E-GS by around $\approx 5\%$ error reduction. The extended experiment further confirms our intuition that our edge-focused design choices can benefit other link-based inference tasks than our original CTR target metric.

Table 4: An ablation study reveals that the neighbor attention, inter-ego relation and intra-ego relation components of ELR all contribute to predictive power.

	Region 1		Region 2	
	MSE	**MAE**	**MSE**	**MAE**
ELR_{mean}	.0871	.2441	.0794	.2425
ELR_{no_inter}	.0894	.2482	.0819	.2435
ELR_{no_intra}	.0908	.2505	.0810	.2460
ELR	**.0865**	**.2389**	**.0773**	**.2404**

5.4 Ablation Study

We next aim to answer **EQ2** by conducting an ablation study to evaluate the relative impact of attention-based friendship importance, incorporating intra-ego relations, and inter-ego relations to ELR. We consider several variations of ELR including (i) ELR_{mean}, which drops the friendship importance and uses mean pooling over link representations for both intra-ego relations and inter-ego relations; (ii) ELR_{no_inter}, which removes the inter-ego relation representation r_{uv+} in the affinity score prediction module, and (iii) ELR_{no_intra}, which excludes the intra-ego relations (r_{u+},r_{v+} from ELR). Table 4 shows the performance of predicting friend story CTR using these ablation variants. From the results we conclude that (i) the performance of ELR_{mean} is worse than ELR, showing that the friendship importance filters unrelated links and reduces noise in ELR; (2) ELR has lower error than ELR_{no_inter} on both datasets because ELR_{no_inter} drops the inter-ego relations and fails to model the dependencies between u's friends and v's friends; (3) ELR_{no_intra} has higher error than both ELR and ELR_{no_inter}, suggesting that intra-ego relations (which were dropped) are especially important in achieving good performance, by encouraging a smoothed representation over u and u's friends, and v and v's friends respectively. In summary, all components contribute to ELR's improved performance. All cases suggest inherent advantages of our incorporation of edge features.

5.5 Friend Importance Analysis

We next evaluate how the learned friendship importances (α_{ui} from Eq.6) are correlated to raw, exhibited edge features $\mathbf{e}_{ui}$ to answer **EQ3**. We randomly select ≈ 100K users from the testing samples in Region 1. For each user, we compute the Pearson's correlation coefficient between the importance scores and four representative interaction signals: number of sent/received Chats, number of viewed Snaps, number of Friend Story views, and total Friend Story view time. We plot the cumulative distribution function (CDF) across the correlation statistics for the 100K users in Figure 6. We observe that the majority of users show positive correlations between friendship importance scores and each representative interaction signals; more than 60% of users report positive Pearson's correlation coefficient in each interaction signal. Importantly, this shows that α_{ui} well-aligns learned attention across multiple raw signals, only one or few of which may be relevant for any given user pair (e.g. some user pairs may Snap a lot, while others only Chat, etc.) Together with our ablation experiment ELR_{mean}, this builds confidence that the learned scores well-characterize close friendships and reduce noise from sparsity in representation learning.

5.6 Data Efficiency

One may think that given ELR's modeling choices, that it may require massive amounts of user data to generalize well. We thus analyze

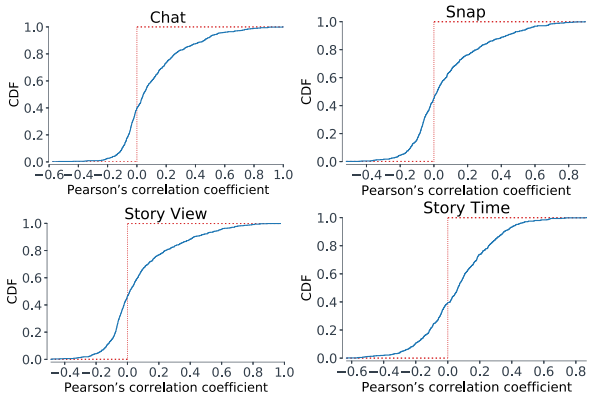

Figure 6: Learned friend attentions are strongly correlated to raw edge features. From CDFs of Pearson's correlation coefficients across users, we observe that representative user interaction signals are positively correlated to friend attention scores for the majority of users.

Table 5: Training on small datasets: ELR_{lite} achieves competitive performance with much fewer (≈1%) training samples.

	Region 1		Region 2	
	MSE	**MAE**	**MSE**	**MAE**
ELR_{lite}	.0880	.2408	.0791	.2422
ELR	.0865	.2389	.0773	.2404

ELR's efficiency with respect to the number of training samples to answer **EQ4**. We compile two small datasets from Region 1 and Region 2. Specifically, we select ≈ 1% of training links (u,v) for each region. We train ELR_{lite} on the two small datasets, and report the CTR task performance as shown in Table 5. We can observe that even with a comparatively small amount of training samples, ELR_{lite}) achieves comparable performance to ELR. Moreover, this performance is still improved over baselines using the whole datasets (ref. Table 2), This is highly advantageous for recurrent training, as per industry use cases.

6 CONCLUSION

In this work, we study the problem of Friend Story Ranking (FSR). FSR is an instantiation of a more general link ranking problem, tasked with inference over existing network links, rather than inferring over missing or future ones. We tackle the problem from a graph representation learning perspective, building upon recent advances in GNNs in the ELR model. We incorporate socially-motivated intuitions into our model design, focusing our inference to a localized edge-contextual graph for each target link for inference, emphasizing importance of edge attributes in user-user relationships in message passing, while also learning distinguished representations across different types of pairwise, intra-ego, and inter-ego relations in local graph structure, and carefully attending over nodes in deriving these representations to overcome social interaction sparsity. Through extensive experiments, we show ELR outperforms baselines with an impressive 8% error reduction in Friend Story CTR prediction. We further show ELR's generality, the relative importance of modeling components, and qualitative findings.

ACKNOWLEDGMENTS

Suhang Wang is partially supported by National Science Foundation (NSF) under grant IIS #1955851.

REFERENCES

[1] Sami Abu-El-Haija, Bryan Perozzi, and Rami Al-Rfou. 2017. Learning edge representations via low-rank asymmetric projections. In *KDD*. 1787–1796.

[2] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. Oddball: Spotting anomalies in weighted graphs. In *PAKDD*. Springer, 410–421.

[3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *JSM* 2008, 10 (2008), P10008.

[4] Cécile Bothorel, Juan David Cruz, Matteo Magnani, and Barbora Micenkova. 2015. Clustering attributed graphs: models, measures and methods. *arXiv preprint arXiv:1501.01676* (2015).

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).

[6] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. In *WSDM*. 177–185.

[7] Inderjit S Dhillon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. 269–274.

[8] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*. 135–144.

[9] Dhivya Eswaran, Christos Faloutsos, Sudipto Guha, and Nina Mishra. 2018. Spotlight: Detecting anomalies in streaming graphs. In *KDD*. 1378–1386.

[10] Martin Everett and Stephen P Borgatti. 2005. Ego network betweenness. *Social networks* 27, 1 (2005), 31–38.

[11] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *TheWebConf*.

[12] Zheng Gao, Gang Fu, Chunping Ouyang, Satoshi Tsutsui, Xiaozhong Liu, and Ying Ding. 2019. *edge2vec: Learning node representation using edge semantics*. Technical Report.

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *ICML*. PMLR.

[14] Liyu Gong and Qiang Cheng. 2019. Exploiting edge features for graph neural networks. In *CVPR*. 9211–9219.

[15] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. 855–864.

[16] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*.

[17] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[19] Hong Huang, Jie Tang, Sen Wu, Lu Liu, and Xiaoming Fu. 2014. Mining triadic closure patterns in social networks. In *WWW*. 499–504.

[20] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2012. Social contextual recommendation. In *CIKM*. 45–54.

[21] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. 2020. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 66–74.

[22] Parisa Kaghazgaran, Maarten Bos, Leonardo Neves, and Neil Shah. 2020. Social Factors in Closed-Network Content Consumption. In *CIKM*. 595–604.

[23] Hyun J Kim and James B Stiff. 1991. Social networks and the development of close relationships. *Human Communication Research* 18, 1 (1991), 70–91.

[24] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[25] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[26] Alec Kirkley, George T Cantwell, and MEJ Newman. 2019. Balance in signed networks. *Physical Review E* 99, 1 (2019), 012320.

[27] Danai Koutra, U Kang, Jilles Vreeken, and Christos Faloutsos. 2014. Vog: Summarizing and understanding large graphs. In *SDM*. SIAM, 91–99.

[28] Yi-Yu Lai and Jennifer Neville. 2020. MERL: Multi-View Edge Representation Learning in Social Networks. In *CIKM*. 675–684.

[29] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Signed networks in social media. In *CHI*. 1361–1370.

[30] Zhao Li, Zhanlin Liu, Jiaming Huang, Geyu Tang, Yucong Duan, Zhiqiang Zhang, and Yifan Yang. 2019. MV-GCN: multi-view graph convolutional networks for link prediction. *IEEE Access* 7 (2019), 176317–176328.

[31] Weiping Liu and Linyuan Lü. 2010. Link prediction based on local random walk. *EPL* 89, 5 (2010), 58007.

[32] Yozen Liu, Xiaolin Shi, Lucas Pierce, and Xiang Ren. 2019. Characterizing and forecasting user engagement with in-app action graph: A case study of snapchat. In *KDD*. 2023–2031.

[33] Tiancheng Lou, Jie Tang, John Hopcroft, Zhanpeng Fang, and Xiaowen Ding. 2013. Learning to predict reciprocity and triadic closure in social networks. *TKDD* 7, 2 (2013), 1–25.

[34] Linyuan Lü and Tao Zhou. 2011. Link prediction in complex networks: A survey. *Physica A* 390, 6 (2011), 1150–1170.

[35] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*. 931–940.

[36] Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. 2020. A unified view on graph neural networks as graph signal denoising. *arXiv preprint arXiv:2010.01777* (2020).

[37] Yao Ma, Suhang Wang, Charu C Aggarwal, and Jiliang Tang. 2019. Graph convolutional networks with eigenpooling. In *KDD*. 723–731.

[38] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001).

[39] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546* (2013).

[40] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*. 701–710.

[41] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM*. 459–467.

[42] Ryan A Rossi, Nesreen K Ahmed, Aldo Carranza, David Arbour, Anup Rao, Sungchul Kim, and Eunyee Koh. 2019. Heterogeneous network motifs. *arXiv preprint arXiv:1901.10026* (2019).

[43] Aravind Sankar, Yozen Liu, Jun Yu, and Neil Shah. 2021. Graph Neural Networks for Friend Ranking in Large-scale Social Platforms. In *TheWebConf*.

[44] Aravind Sankar, Yozen Liu, Jun Yu, and Neil Shah. 2021. Graph Neural Networks for Friend Ranking in Large-scale Social Platforms. In *Proceedings of the Web Conference 2021*. 2535–2546.

[45] Neil Shah, Alex Beutel, Bryan Hooi, Leman Akoglu, Stephan Gunnemann, Disha Makhija, Mohit Kumar, and Christos Faloutsos. 2016. Edgecentric: Anomaly detection in edge-attributed networks. In *ICDMW*. 327–334.

[46] Guolei Sun and Xiangliang Zhang. 2019. A novel framework for node/edge attributed graph embedding. In *PAKDD*. Springer, 169–182.

[47] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. 1067–1077.

[48] Xianfeng Tang, Yandong Li, Yiwei Sun, Huaxiu Yao, Prasenjit Mitra, and Suhang Wang. 2020. Transferring robustness for graph neural network against poisoning attacks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 600–608.

[49] Xianfeng Tang, Yozen Liu, Neil Shah, Xiaolin Shi, Prasenjit Mitra, and Suhang Wang. 2020. Knowing your FATE: Friendship, Action and Temporal Explanations for User Engagement Prediction on Social Apps. In *KDD*. 2269–2279.

[50] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. 2020. Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1435–1444.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[52] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

[53] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *KDD*. 1225–1234.

[54] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[55] Carl Yang, Xiaolin Shi, Luo Jie, and Jiawei Han. 2018. I know you'll be back: Interpretable new user clustering and churn prediction on a mobile social application. In *KDD*. 914–922.

[56] Lin Yao, Luning Wang, Lv Pan, and Kai Yao. 2016. Link prediction based on common-neighbors for dynamic social network. *PCS* 83 (2016).

[57] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*. 974–983.

[58] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *arXiv preprint arXiv:1806.08804* (2018).

[59] Ke Zhang and Konstantinos Pelechrinis. 2014. Understanding spatial homophily: the case of peer influence and social selection. In *WWW*. 271–282.

[60] Muhan Zhang and Yixin Chen. 2017. Weisfeiler-lehman neural machine for link prediction. In *KDD*. 575–583.

[61] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *arXiv preprint arXiv:1802.09691* (2018).

[62] Tianxiang Zhao, Xianfeng Tang, Xiang Zhang, and Suhang Wang. 2020. Semi-Supervised Graph-to-Graph Translation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1863–1872.