STLSP: Integrating Structure and Text with Large Language Models for Link Sign Prediction of Networks

Lijia Ma[1] , **Haoyang Fu**[1] , **Zhijie Cao**[1] , **Xiongnan Jin**[2,3,*] , **Qiuzhen Lin**[1] and **Jianqiang Li**[2,3]
[1]College of Computer Science and Software Engineering, Shenzhen University
[2]School of Artificial Intelligence, Shenzhen University
[3]National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University
omegamalj@gmail.com, 2410105018@mails.szu.edu.cn, caozj2003@gmail.com,
xiongnanjin@szu.edu.cn, qiuzhlin@szu.edu.cn, lijq@szu.edu.cn

Abstract

Link Sign Prediction (LSP) in signed networks is a critical task with applications in recommendation systems, community detection, and social network analysis. Existing methods primarily rely on graph neural networks to exploit structural information, often neglecting the valuable insights from edge-level textual data. Furthermore, utilizing large language models (LLMs) for LSP faces challenges in reliability and interpreting graph structures. To address these issues, we propose a novel STLSP framework that integrates signed networks' Structural and Textual information with LLMs for the LSP task. STLSP leverages structural balance theory to generate node embeddings that capture positive and negative relationships. These embeddings are transformed into natural language representations through clustering techniques, allowing LLMs to utilize the structural context fully. By integrating these representations with edge text, STLSP improves the accuracy and reliability of the LSP task. Extensive experiments conducted on five real-world datasets demonstrate that STLSP outperformed state-of-the-art baselines, achieving an 8.7% improvement in terms of accuracy. Moreover, STLSP shows robust performance across various LLMs, making it adaptable to different computational environments. The code and data are publically available at https://github.com/sss483/STLSP.

1 Introduction

Signed networks are graph structures consisting of nodes representing various entities and edges assigned with positive or negative signs to denote different relationships or interactions between nodes. For instance, in an online social platform, user entities can have "like" relationships (positive edges) with each other, while "dislike" actions can be represented as negative edges. Due to the distinctive feature of accurately reflecting the relationships in real-world complex systems, signed networks are widely employed in various applications [Xie *et al.*, 2022]. Link Sign Prediction (LSP) is a crucial downstream task for signed networks. LSP determines whether a relationship is positive or negative, directly empowering applications such as community detection [Wang *et al.*, 2023] and recommendation systems [Zhou *et al.*, 2023].

Currently, LSP research can be broadly divided into two main categories: feature-based methods and network embedding-based methods [Fang *et al.*, 2024]. Feature-based approaches gather specific features from the graph structure and identify the sign of a target link based on those features. In [Leskovec *et al.*, 2010], logistic regression is adopted to predict link signs using node degrees and triad types features. HOC [Chiang *et al.*, 2011] presents a supervised approach that extracts features from longer cycles to predict link signs. [Beigi *et al.*, 2020] leverage Emotional Information, Diffusion of Innovations, and Individual Personality theories to guide feature engineering to address the data sparsity problem in link sign prediction.

On the other hand, network embedding-based methods transform the signed graph into a low-dimensional vector space and then use the node representations for LSP. SGCN [Derr *et al.*, 2018] applies graph convolution to signed networks, effectively capturing both positive and negative edges for accurate LSP. RSGNN [Zhang *et al.*, 2023] analyzes the impact of noisy edges on signed graph neural networks using an extended Weisfeiler-Lehman test and presents a dual architecture to denoise the graph and learn node representations simultaneously. S-GNN [Lin and Li, 2024] separates pairwise interactions into receptive and generative types and uses two-component status convolutional layers for status aggregation and propagation.

However, although existing methods effectively utilize graph structures, the textual information on edges, which could provide valuable evidence for LSP, is often overlooked. As shown in the first motivating scenario of Figure 1, the textual description of the target edge clearly points to the negative sign. Regardless, a conventional GNN-based LSP method produces a wrong answer by analyzing the graph structure without considering edge text. Here, only 1-hop neighbors of target nodes are illustrated for simplicity.

With the emergence of large language models (LLMs), LLM-based LSP has become a promising direction for leveraging edge text in signed networks. However, LLMs

* Corresponding author

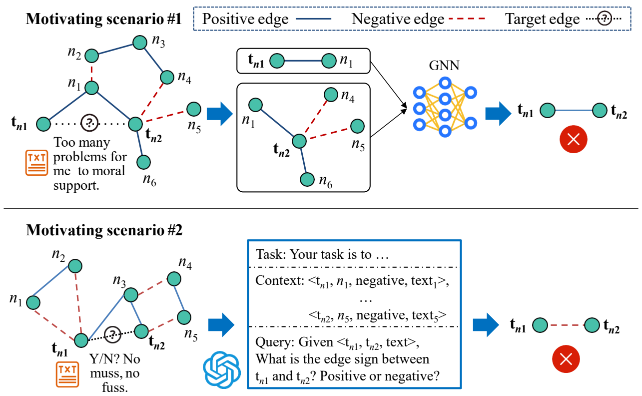


Figure 1: Motivating scenarios, where t_n indicates target node and n denotes node.

face inherent limitations in understanding topological graph data [Huang *et al.*, 2024]. A straightforward way is to convert the graph structure into text and inject it into prompts for decision-making by LLMs. As shown in the incorrect prediction result given by an LLM in the second motivating scenario in Figure 1, there remain research challenges for LLM-based LSP. Firstly, LLMs cannot fully interpret raw structural information in a simple text format such as quadruple. Secondly, relying solely on LLMs to perform downstream tasks like LSP could result in unreliable outputs [Majeed and Hwang, 2024]. It stems from the fact that LLMs generate results through a softmax layer in the final stage, where the one with the highest probability is selected as the output.

To address the challenges mentioned above, in this paper, we propose an approach that integrates Structural and Textual information in signed networks with LLMs for Link Sign Prediction, namely STLSP. STLSP combines the structure learning capabilities of graph neural networks (GNNs) with the natural language understanding of LLMs. Specifically, we adopt the structural balance theory for structural learning to guide the generation of node embeddings that capture both positive and negative relationships between nodes. Then, a clustering technique is adopted to identify different balance groups. Natural language labels are generated depending on whether the nodes belong to the same balance group to enable LLMs to comprehend the graph structure deeply. Finally, LLMs are employed to perform the LSP based on the created contextual information to predict the signs of edges.

The main contributions are summarized as follows:

- We propose STLSP, which combines structural information from signed networks, processed through structural balance theory, with textual data from edges. This integration leverages the strengths of graph neural networks (GNNs) and large language models (LLMs) to enhance the accuracy and reliability of link sign prediction (LSP).
- We transform graph structural information into natural language expressions using network embedding and clustering techniques. This enables LLMs to interpret and utilize graph topological data effectively, addressing their limitations in directly understanding raw graphs.
- Experimental results on five real-world datasets show that STLSP outperforms state-of-the-art LSP methods with an average margin of 8.7% in terms of accuracy. STLSP is independent of the parameter size of LLMs and works well with relatively small-scale LLMs such as LLama3-7B. Besides, STLSP demonstrates significantly higher reliability than a text-only approach, highlighting the effectiveness of the proposed integration mechanism of graph structure and edge text.

2 Related Work

2.1 Link Sign Prediction

Feature-based and network embedding-based methods are two mainstream methods in LSP research. Feature-based LSP methods focus on extracting specific features from the graph structure to predict the sign of a target link. HOC [Chiang *et al.*, 2011] leverages features derived from longer cycles in signed networks to predict link signs. Extending beyond local structures like triangles, it incorporates higher-order patterns to capture the structural imbalances in the graph better. In SLF [Xu *et al.*, 2019], latent factor decomposition models positive and negative relationships between nodes, focusing on topological features extracted from the signed graph. In [Beigi *et al.*, 2020], features inspired by social science theories, such as emotional information, diffusion of innovations, and individual personality, are extracted to address the signed link prediction problem. These features leverage nonstructural information from user behavior and interactions to compensate for the sparsity of signed links.

Network embedding-based LSP approaches learn low-dimensional representations of signed graphs to capture structural and signed characteristics for link sign prediction. Among these methods, SGCN [Derr *et al.*, 2018] applies graph convolution to signed networks, leveraging structural balance theory to model local triangular structures and distinguish between positive and negative edges. Extending this, SDGNN [Huang *et al.*, 2021] employs a signed directional aggregator to capture edge directionality better, improving upon SGCN while maintaining its capability to model signed features. RSGNN [Zhang *et al.*, 2023] leverages an extended Weisfeiler-Lehman test to analyze the effects of noisy edges on signed graph neural networks. Integrating a dual architecture that simultaneously denoises the graph and learns robust node representations addresses challenges in modeling complex signed structures and improves learning.

2.2 Large Language Models for Predicton Tasks

LLMs have proven highly effective in graph-related tasks, particularly in datasets with rich textual information. By leveraging their advanced natural language understanding capabilities, LLMs consistently achieve decent performance in natural language processing (NLP) tasks [Mekrache *et al.*, 2024]. However, LLMs fall short of capturing structural information, such as node relationships, graph topology, and neighborhood aggregation, which are the strengths of GNNs. Considering this, many recent methods aim to integrate LLMs with GNNs, creating novel frameworks that effectively combine structural and textual information [Jin *et* *al.*, 2024]. Within this paradigm, link prediction has garnered significant attention due to the enhanced ability to capture richer relational semantics, as the integration of textual and structural data enables models to infer better and contextualize connections between nodes. Specifically, in the paradigm where LLMs assist GNNs, SimTeG [Duan *et al.*, 2023] improves link prediction by combining LLM-generated node embeddings with graph structure. The embeddings capture rich textual features, which enhance the GNN's ability to model relationships between nodes effectively.

On the other hand, in the paradigm where GNNs assist LLMs, LPNL [Bi *et al.*, 2024] enhances the link prediction capability of LLMs by transforming graph structural information into natural language prompts, leveraging a two-stage sampling pipeline and a divide-and-conquer strategy to effectively handle large-scale heterogeneous graphs, enabling the model to infer better connections between nodes. In this paper, we focus on signed networks, extending the application of LLMs from traditional link prediction to link sign prediction. Implementation details are provided in Section 4.

3 Preliminaries

Signed Networks. Signed networks are defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{T})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges, $\mathbf{X}$ represents the node properties, and $\mathbf{T}$ indicates the textual information associated with the edges. Specifically, $\mathcal{E}$ can be further expressed as $\mathcal{E} = (\mathcal{E}^+, \mathcal{E}^-)$, where $\mathcal{E}^+$ denotes edges with positive values and $\mathcal{E}^-$ stands for edges with negative values. For simplicity, we re-formalize $\mathcal{E}$ using an adjacency matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$, where $a_{ij} = 1$ indicates a positive edge between nodes i and j, and $a_{ij} = -1$ denotes a negative edge. Additionally, each a_{ij} is associated with text $t_{ij} \in \mathbf{T}$, representing the raw textual data linked to a_{ij}.

Link Sign Prediction (LSP). The LSP problem is formally defined as follows. Given a signed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{T})$, the task of link sign prediction aims to predict the sign $a_{ij} \in \{-1, 1\}$ of an edge e_{ij} based on the structural properties of the network, the attributes of nodes $\mathbf{X}$, and the textual information $t_{ij} \in \mathbf{T}$. The goal is to determine whether the relationship between nodes i and j is positive ($a_{ij} = 1$) or negative ($a_{ij} = -1$).

Structural Balance Theory. Structural balance theory can be used to understand the stability of relationships within social networks [Heider, 1946]. According to the theory, the stability of networks depends on the balanced states of their relational triads. A triad is considered "balanced" under two conditions: 1) all three edges are positive, representing full cooperation and reflecting the principle that "the friend of my friend is my friend"; 2) one edge is positive while the other two edges are negative, reflecting the principle that "the enemy of my enemy is my friend". From the clustering view, the positive and negative links are considered balanced if they can be divided into the same and different clusters, respectively. Moreover, clustering under structural balance divides a signed network into a number (K) of clusters to minimize the number of imbalanced links.

4 Proposed STLSP

In this section, we introduce the proposed approach STLSP that predicts edge signs using both trained node features and edge-related textual information. STLSP mainly comprises three modules: Graph Divider, Embeddings Trainer, and LLM Predictor. The overall architecture of the proposed STLSP is shown in Figure 2. For training, we adopted a node training strategy incorporating the structural balance properties unique to signed networks, ensuring that node embeddings effectively reflect these characteristics, as detailed in section 4.2. After obtaining the embeddings, we preprocess them to make the structural information interpretable by LLMs through a clustering-based partitioning module that converts structural data into natural language representations. Finally, we combine the structural and textual information using a carefully designed prompt, enabling the LLM-based prediction module to generate the final sign prediction results.

4.1 Graph Divider

The graph divider splits the graph data into the train and test sets, as shown in Figure 2(a). The embeddings trainer utilizes the train set to learn graph representations, and the test set is delivered to the LLM predictor to infer LSP results.

Specifically, an edge-masking strategy is adopted to partition the signed network constructed from the dataset into a train set and a test set. Due to the network topology in the test set not being a continuous graph, it is not feasible to directly apply the trained node embedding generation module to produce structure-balanced node embeddings for the test set. Therefore, during the processing stage, we ensure that every node in the test set has been included in the train set and has obtained its embedding through training. In other words, during the edge-masking process, we continuously check for the emergence of isolated nodes (nodes with a degree of 0). The corresponding edges are deemed unsuitable for masking if such nodes are detected.

Formally, the signed network is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{T})$. When a candidate edge $e = (u, v) \in \mathcal{E}$ is masked, the edge set $\mathcal{E}$ updates to $\mathcal{E}'$, and the signed network becomes $\mathcal{G}' = (\mathcal{V}, \mathcal{E}', \mathbf{X}, \mathbf{T})$. Then we define:

$$d'(u) = \deg_{\mathcal{G}'}(u) \tag{1}$$

where $d'(u)$ is the degree of node u in graph $\mathcal{G}'$. Finally, the overall edge-masking strategy can be summarized as follows:

$$\mathrm{H}(e) = \begin{cases} 1, & \text{if } d'(u) > 0 \text{ and } d'(v) > 0 \\ 0, & \text{otherwise} \end{cases}, u, v \in e \tag{2}$$

where $\mathrm{H}(e) = 1$ indicates that the edge e can be masked, while $\mathrm{H}(e) = 0$ denotes that the edge e must be retained.

4.2 Embedding Trainer

In the embedding trainer module, inspired by SiNE [Wang *et al.*, 2017], we propose a novel loss function to adapt sparse networks better. Due to the sparsity of edges, relying on traids to analyze the structure between nodes is computationally expensive and suboptimal. Therefore, we directly constrain independent edges instead of constructing triads. Guided by the

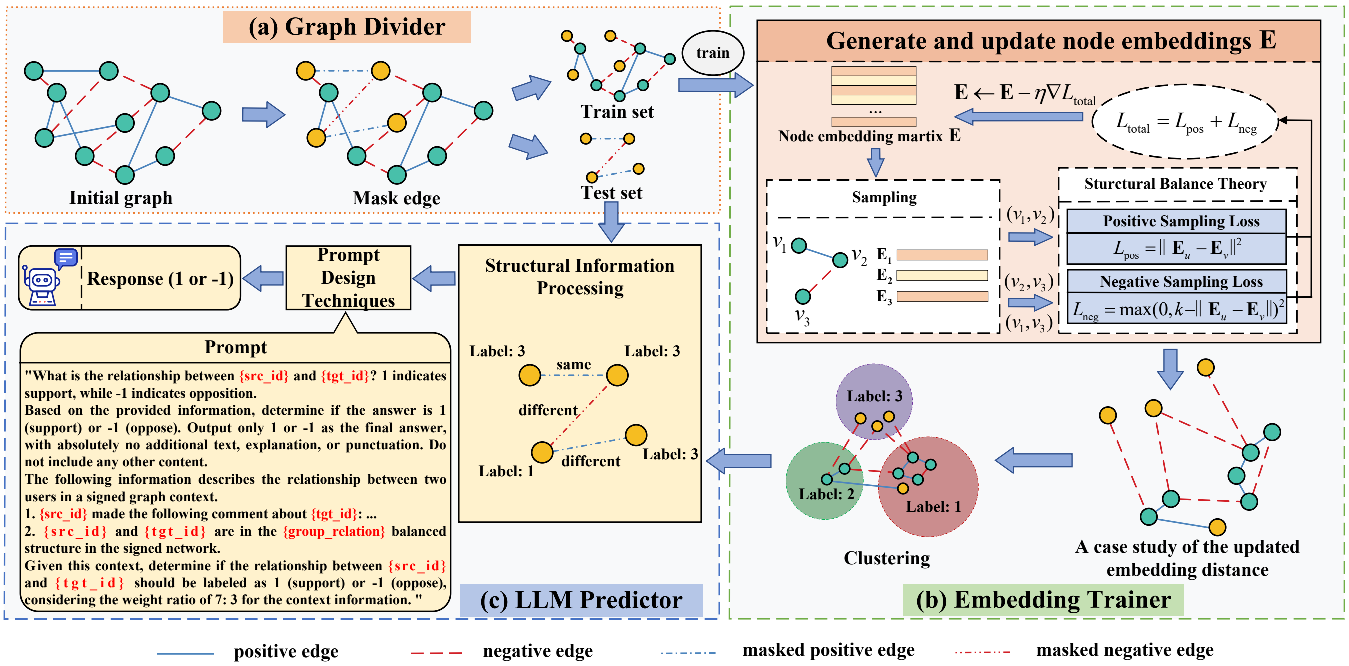


Figure 2: Overall architecture of the proposed STLSP approach.

structural balance theory, which posits that a triad is balanced if "the friend of my friend is my friend" and "the enemy of my enemy is my friend", we enforce cohesion for positively linked nodes and separation for negatively linked nodes.

For positive edges, inspired by the structural balance theory, which emphasizes that positively linked nodes should exhibit strong cohesion, we minimize the Euclidean distance between their embeddings. It ensures that nodes connected by positive links are drawn closer in the embedding space, reflecting their harmonious relationship. The loss function for positive edges is defined as follows:

$$L_{\text{pos}} = \|\mathbf{E}_u - \mathbf{E}_v\|^2 \tag{3}$$

where $\mathbf{E}_u$ represents the embedding vector of node u, and v denotes the adjacent node of u.

For negative edges, inspired by the structural balance theory, which emphasizes that negatively linked nodes should exhibit clear separation, we enforce a threshold-based separation by penalizing distances shorter than a predefined margin between the node embedding of u and v. The loss function for negative edges is defined as follows:

$$L_{\text{neg}} = \max(0, k - \|\mathbf{E}_u - \mathbf{E}_v\|)^2 \tag{4}$$

We use the ReLU function [Kouvaros and Lomuscio, 2021] to constrain the distance for negative edges to a threshold of k. When the distance exceeds k, the loss becomes zero, avoiding unnecessary constraints on distant negative edges. This soft constraint effectively reduces the influence of noisy edges, ensuring that the optimization process focuses on closer negative edges with insufficient distance, thus improving the robustness of the model. Therefore, the total loss function is designed as follows:

$$L_{\text{total}} = L_{\text{pos}} + L_{\text{neg}} \tag{5}$$

The trained node embeddings are inherently designed to reflect the principles of structural balance theory, where nodes connected by negative edges are represented as farther apart in the embedding space, and nodes connected by positive edges are closer. This property ensures that the embeddings naturally align with the K-means clustering assumption, which groups points that are close in the feature space while separating those farther apart. Thus, we applied the K-means [MacQueen, 1967] to partition the embeddings simply but effectively without requiring more complex approaches.

$$\underset{\mathbf{C},\mathbf{L}}{\operatorname{argmin}} \sum_{i=1}^{n} \|\mathbf{E}_i - \mathbf{C}_{l_i}\|^2 \tag{6}$$

Eq. (6) aims to minimize the intra-cluster distance, where $\mathbf{C}$ is the set of cluster centers, and $\mathbf{C}_{l_i}$ represents the cluster center of node i. $\mathbf{E} \in \mathbb{R}^{n \times d}$, where n is the number of nodes, d is the embedding dimension of the nodes, and $\mathbf{E}_i$ is the embedding vector of node i. $\mathbf{L} = \{l_1, l_2, \ldots, l_n\}$ is the set of cluster labels for the nodes.

4.3 LLM Predictor

Here, we will introduce two components of LLM Predictor. The first focuses on processing structural information into natural language, and the second addresses prompt design.

Structural Information Processing. When we have the node embeddings, transforming them into a format that LLMs can effectively perceive is a significant challenge. To address this, we propose a simple yet effective module. Specifically, by comparing the balanced population labels of each pair of

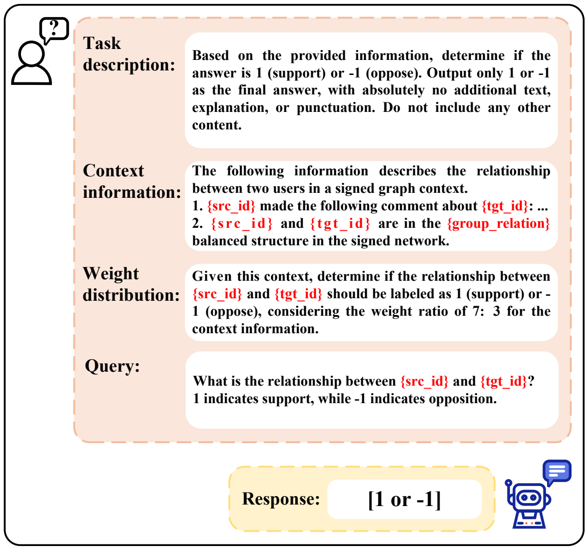


Figure 3: The general structure of the prompt.

nodes, we assign a special label to the edge connecting them based on whether their labels match. During the prompt design phase, this structural information is carefully integrated with the textual information and then input into the LLMs. Experimental results demonstrate that this structural information can be fully utilized by LLMs, significantly enhancing the performance of the tasks. The label generation process is defined as follows. For any pair of nodes (u, v), the label of the edge $\mathcal{E}_{uv}$ can be defined as:

$$g(u,v) = \begin{cases} \text{same}, & \text{if } l_u = l_v \\ \text{different}, & \text{if } l_u \neq l_v \end{cases} \tag{7}$$

where the label $g(u, v)$ indicates whether u and v belong to the same balance group.

Prompt Design. Building on the representation of structural information $g(u, v)$, we design a carefully tailored prompt to guide LLMs leveraging structural and textual information for prediction tasks. The prompt comprises two key elements: explicit task instructions and the seamless integration of multimodal data. The task instructions are designed to constrain the output format, ensuring that the model outputs only a prediction of 1 or -1 without irrelevant text, thereby maintaining clarity and consistency in the output.

Moreover, the prompt integrates the structural balance labels $g(u, v)$ with the text attributes of the edges, providing the model with a comprehensive context that fuses graph-theoretic and linguistic perspectives. By specifying a weight ratio for structural and textual information, the prompt prioritizes the structural cues while allowing the textual data to complement the decision-making process. Experimental results demonstrate that the design enhances the interpretability of predictions and significantly improves the performance of prediction tasks. It highlights the importance of integrating multimodal information into Prompt Design Techniques for signed network reasoning tasks. The general structure of the prompt is shown in Figure 3. Because a 6:4 weight ratio between structural information and textual information causes excessive interference from structural information, while an 8:2 ratio renders structural information underpowered, a 7:3 weight ratio is chosen. This ratio ensures that structural information remains useful without interfering with reasoning when clear semantic information is present.

Dataset	# nodes	# pos links	# neg links	% pos
wiki-rfa	11,381	144,451	41,176	77.81%
amazon_software	368,461	285,989	133,987	68.09%
amazon_office	3,518,540	4,368,219	829,225	84.04%
amazon_video	1,521,407	1,898,218	453,114	80.72%
amazon_crafts	1,780,025	2,344,185	331,077	87.62%

Table 1: Statistic of the datasets.

5 Experiment

Extensive experiments were conducted to verify the effectiveness of the proposed STLSP, focusing on answering the following research questions (RQs):

- **RQ1**: Does STLSP achieve superior performance in the LSP task comparing to the state-of-the-art baselines?
- **RQ2**: Is the performance of STLSP dependent to the parameter size of LLMs?
- **RQ3**: Does STLSP effectively incorporate textual information for the LSP task?
- **RQ4**: Does STLSP enhance the reliability of LLM inference in the context of LSP?

5.1 Experimental Settings

Datasets. Five datasets were employed in our experiments, which are wiki-rfa [West *et al.*, 2014] dataset and four subsets from the Amazon Review Data [Ni *et al.*, 2019]. For dataset splitting, we set the ratio of edges between the train and test sets as 8:2, following common configuration in the LSP task [Huang *et al.*, 2021]. We also ensured that all nodes in the test set were included in the train set during sampling. The dataset statistics are shown in Table 1. The proportion of positive edges was nearly 80%.

Baselines and LLM models. We compare our STLSP with six LSP baselines divided into two categories. The first category is the feature-based method, like SLF [Xu *et al.*, 2019], which relies on explicit feature extraction to model signed relationships. The second category includes network embedding-based methods, including SGCN [Derr *et al.*, 2018], BESIDE [Chen *et al.*, 2018], SDGNN [Huang *et al.*, 2021], SiGAT [Huang *et al.*, 2019], and RSGNN [Zhang *et al.*, 2023]. Network embedding-based methods leverage graph neural networks or embedding techniques to capture the latent representations of signed graphs, commonly guided by balance theory, for improving the effectiveness of LSP.

For LLMs, we adopted close-sourced large model GPT-4o [Cheng *et al.*, 2024] and open-sourced relatively small models such as Gemma-7B [Team *et al.*, 2024], LLama3-7B [Dubey *et al.*, 2024] and Mistral-7B [Liu *et al.*, 2023]. Unless otherwise specified, all experiments were conducted using the STLSP equipped with GPT-4o.

Category	Method	wiki-rfa		amazon_software		amazon_office		amazon_video		amazon_crafts		Average	
		Acc.	Binary-F1	Acc.	Binary-F1	Acc.	Binary-F1	Acc.	Binary-F1	Acc.	Binary-F1	Acc.	Binary-F1
Feature	SLF	74.1%	83.4%	66.1%	77.7%	74.0%	84.8%	63.9%	77.2%	74.2%	84.9%	70.5%	81.6%
Network embedding	SiGAT	76.0%	85.1%	73.8%	82.6%	85.9%	92.3%	82.2%	89.9%	90.3%	94.8%	81.6%	88.9%
	SGCN	76.2%	86.4%	67.4%	80.0%	81.8%	89.9%	76.7%	86.7%	79.3%	88.3%	76.3%	86.3%
	BESIDE	75.5%	84.4%	71.1%	81.2%	83.8%	91.1%	81.6%	89.7%	88.5%	93.9%	80.1%	88.1%
	SDGNN	76.0%	84.9%	78.3%	85.7%	86.4%	92.6%	82.9%	90.3%	90.4%	94.9%	82.8%	89.7%
	RSGNN	76.2%	86.5%	69.6%	82.1%	85.8%	92.4%	82.5%	90.4%	88.8%	94.1%	80.6%	89.1%
	STLSP (ours)	**90.5%**	**90.2%**	**90.7%**	**90.8%**	**91.7%**	**92.7%**	**89.9%**	**90.8%**	**94.9%**	**95.5%**	**91.5%**	**92.0%**
	- improvement	+14.3%	+3.7%	+12.4%	+5.1%	+5.3%	+0.1%	+7.0%	+0.4%	+4.5%	+0.6%	+8.7%	+2.3%

Table 2: Evaluation results of STLSP and LSP baselines on five datasets. Acc. is short for accuracy. The best scores in each metric are highlighted in **bold**, and the follow-ups are marked with underline.

Evaluation Metrics. Accuracy and Binary-F1 were used as evaluation metrics. Accuracy reflects the overall correctness of the model in predicting both positive and negative links, formally $\frac{TruePositives+TrueNegatives}{TotalLinks}$. Binary-F1 provides a balanced evaluation of predicting positive and negative links, defined as $2 \times \frac{Precision \times Recall}{Precision+Recall}$.

Experimental Environment. The experiments were conducted using a workstation running on Ubuntu 20.04. It had an AMD EPYC 7763 64-core CPU and an NVIDIA GeForce RTX 4090 GPU. Our STLSP was implemented using Python (version 3.9.20) with a Pytorch backbone (version 2.4.0).

5.2 Overall Comparison with Baselines (RQ1)

We compared the proposed STLSP method with baseline approaches across five different datasets, and the results are presented in Table 2. In the first group, SLF represents a feature-based approach that primarily relies on explicit feature extraction methods, such as triadic structures or node attributes, to model pairwise relationships. However, this method overlooks the deeper integration of global topological features and the complex interactions within signed networks, which limits its overall performance. For example, on the amazon_software dataset, SLF achieves an accuracy of only 66.1%, significantly lower than the best-performing methods. Feature-based approaches struggle to capture the rich structural information in signed networks effectively.

Based on the experimental results, embedding-based methods generally outperformed SLF, demonstrating their superiority caused by learning latent representations from signed networks. The SGCN, as a foundational embedding-based method, shows significant improvements over SLF. For example, on the amazon_video dataset, SGCN achieved an accuracy that was 12.8% higher than that of SLF. However, SGCN's dependence on local features and simple aggregation techniques limits its ability to model global relationships.

SiGAT addresses this issue by introducing a sign-aware graph attention and achieved an accuracy of 73.8% on the amazon_software dataset. Despite this achievement, it has difficulties modeling complex interactions, which constrains its effectiveness. Building on SGCN, RSGNN incorporates a dynamic denoising mechanism that alternates between optimizing the adjacency matrix and embedding representations. Nevertheless, RSGNN's heavy reliance on local neighborhood aggregation during the embedding process means there is room for further enhancement.

BESIDE expands on embedding-based approaches by jointly modeling triangular and bridge-edge relationships. It excels in datasets with sparse local features, such as amazon_software, where it achieves an accuracy of 71.1%. However, its performance drops when it comes to tasks that require modeling strong global contexts. SDGNN introduces a hierarchical aggregation mechanism that captures global structural features in signed networks while integrating structural balance theory to optimize embeddings, effectively balancing global context and local semantics. These innovations make SDGNN the best-performing baseline method, achieving an accuracy of 90.4% on the wiki-rfa dataset. Nevertheless, SDGNN's limitation lies solely in its dependence on graph structures. It cannot leverage textual information that could contain valuable evidence for the LSP task.

Our STLSP combines structural balance theory with thoroughly considering global structural information while generating node embeddings. Additionally, we have innovatively integrated textual information into our framework. Experimental results demonstrate that STLSP outperforms all baseline methods across various datasets, achieving the highest accuracy and Binary-F1 scores. Specifically, on the amazon_software dataset, STLSP attained an accuracy of 90.7% and a Binary-F1 score of 90.8%, representing a 12.4% improvement over SDGNN. Similarly, on datasets rich in textual information, such as amazon_crafts, STLSP reached a Binary-F1 score of 95.5%. It further emphasizes the significance of incorporating textual information into STLSP. These results confirm that STLSP outperforms all baseline methods, highlighting its clear advantage in the LSP task.

5.3 Comprehensive Evaluation of STLSP

Dependency Analysis on LLMs (RQ2). To evaluate the generalization capabilities, we tested the performance of STLSP with different LLMs such as GPT-4o, Gemma-7B, LLama3-7B, and Mistral-7B. GPT-4o, a variant of OpenAI's GPT-4 model, is estimated to have over 100 billion parameters designed for complex reasoning tasks. Gemma-7B is an open-sourced LLM with 7 billion parameters optimized for efficient language understanding in low-resource environments. LLama3-7B strikes a balance between performance and resource consumption. Mistral-7B, another open-sourced LLM, excels in multilingual tasks that provide high accuracy in low- and high-resource settings.

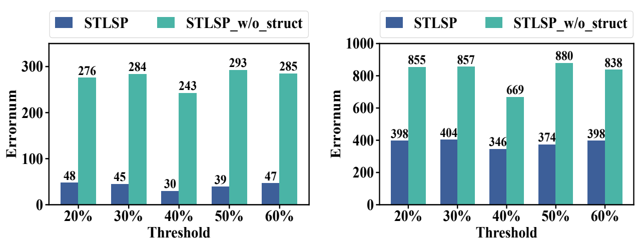
As summarized in Table 3, our STLSP framework demonstrated consistently strong results across all variants, showcasing its adaptability to LLMs with different scales and computational capacities. These findings underscore STLSP's remarkable adaptability to LLMs, rendering it an eminently

Method	wiki-rfa		amazon_software		amazon_office		amazon_video		amazon_crafts		Average	
	Acc.	Binary-F1	Acc.	Binary-F1	Acc.	Binary-F1	Acc.	Binary-F1	Acc.	Binary-F1	Acc.	Binary-F1
GPT-4o	**90.5%**	**90.2%**	<u>90.7%</u>	<u>90.8%</u>	91.7%	92.7%	89.9%	<u>90.8%</u>	94.9%	<u>95.5%</u>	<u>91.5%</u>	<u>92.0%</u>
Gemma-7B	83.4%	81.5%	82.8%	79.8%	**94.8%**	<u>93.6%</u>	<u>90.7%</u>	89.2%	**96.6%**	**95.8%**	89.7%	88.0%
LLama3-7B	<u>86.7%</u>	<u>87.2%</u>	**92.2%**	**92.5%**	93.5%	**94.2%**	**92.0%**	**93.1%**	95.2%	**95.8%**	**91.9%**	**92.6%**
Mistral-7B	82.9%	79.5%	78.8%	72.7%	<u>94.3%</u>	92.7%	89.2%	86.5%	<u>96.3%</u>	95.4%	88.3%	85.4%

Table 3: Evaluation results of STLSP with differenet LLMs on various datasets. Acc. is short for accuracy. The best scores in each metric are highlighted in **bold**, and the follow-ups are marked with <u>underline</u>.

Method	wiki-rfa		amazon_software	
	Accuracy	Binary-F1	Accuracy	Binary-F1
STLSP	90.5%	90.2%	90.7%	90.8%
STLSP_w/o_text	64.2%	65.3%	42.1%	44.3%
- difference	-26.3%	-24.9%	-48.1%	-46.5%

Table 4: Impact analysis about the textual information on STLSP. STLSP_w/o_text is a variation of STLSP that does not utilize textural details, i.e., the first context in the prompt of Figure 3 is omitted.



(a) wiki-rfa

(b) amazon_software

Figure 4: Impact analysis on LLM reliability of STLSP on the wiki-rfa and amazon_software networks. STLSP_w/o_struct is a variation in which graph structures are not utilized, i.e., the second context in the prompt of Figure 3. Answers with reliability below the threshold are considered errors. And Errornum indicates the number of errors.

suitable solution for the LSP task across a broad spectrum of LLM configurations. Notably, the combination with LLama3-7B achieved the highest scores on most datasets despite having fewer parameters than GPT-4o. It indicates that the proposed STLSP can be deployed on a local machine with a consumer-level GPU to avoid data exposure.

Impact Analysis of Integrating Text (RQ3). To validate the necessity of incorporating textual information, we compared STLSP with STLSP_w/o_text. As shown in Table 4, the performance of STLSP dropped significantly when textual information was removed. Specifically, on the amazon_software dataset, the score difference reached 48.1% in terms of accuracy. It denotes that STLSP effectively incorporated edge text, and the textual information contributed enormously based on LLMs for the LSP task.

Impact Analysis on LLM reliability (RQ4). Due to the probabilistic inherence, LLMs return answers with the highest probability, even if the reliability (i.e., confidence) of these answers is low. To evaluate the reliability of LLM inferences for the LSP task, we counted the number of low-reliability answers at varying reliability thresholds (20%, 30%, 40%, 50%, 60%), which were injected into the prompts. Figure 4 demonstrates that our method significantly reduced the number of

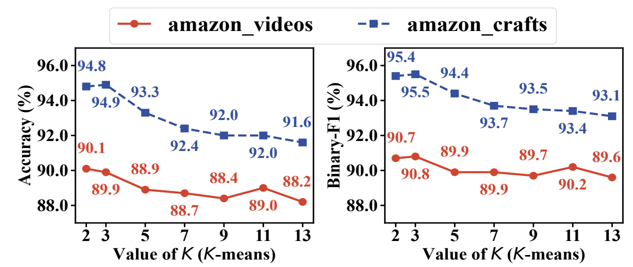


Figure 5: Sensitivity analysis results of the clustering parameter K of STLSP. The Accuracy and Binary-F1 scores for LSP are reported with varying K values.

low-reliability answers by fusing structure embeddings, improving the inference reliability of LLMs for the LSP task.

Note that the number of errors (Errornum) does not show a monotonically decreasing trend as the reliability threshold increases. It occurs because LLMs generated new tokens each time the threshold was adjusted, and the attention mechanism computed a different weight matrix. As a result, the threshold changes were independent, which explains the lack of a consistent downward trend in the error count.

Sensitivity Analysis of Clustering K. Sensitivity tests about the clustering parameter K were executed on two datasets, and the results are illustrated in Figure 5. The results show that STLSP can obtain good results under $K = 2$ and $K = 3$. In STLSP, the reasonable K value can be set as 2 (or 3), which is determined by the strong (weak) structural balance theory.

6 Conclusions

In this paper, we proposed STLSP for link sign prediction, integrating signed networks' structural and textual information using LLMs to improve accuracy and reliability. In STLSP, we design a structure-balanced method to encode and cluster graph structures and convert them into natural language. Then, through carefully designed prompts, we effectively combine structures and text as contextual input for LLMs, achieving state-of-the-art LSP results in terms of effectiveness, as shown in extensive experimental results. Besides, we evaluated the reliability of LLMs' outputs, demonstrating that the proposed STLSP significantly enhances the reliability of predictions. Furthermore, verifications on different LLMs illustrated that STLSP can work well with relatively small-scale LLMs, such as LLama3-7B. In future work, we plan to incorporate more prompt engineering techniques [Giray, 2023] to refine prompts further to enhance LLMs' ability to understand and infer signed links.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 62173236 and 62306287, the Shenzhen Natural Science Foundation under Grant JCYJ20240813141416022, the Tencent "Rhinoceros Bird" Scientific Research Foundation for Young Teachers of Shenzhen University, and the Scientific Foundation for Youth Scholars of Shenzhen University under Grant 868-000001032902.

References

[Beigi *et al.*, 2020] Ghazaleh Beigi, Jiliang Tang, and Huan Liu. Social science–guided feature engineering: A novel approach to signed link analysis. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1):1–27, 2020.

[Bi *et al.*, 2024] Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, and Xueqi Cheng. Lpnl: Scalable link prediction with large language models. In *Findings of the Association for Computational Linguistics (ACL)*, pages 3615–3625, Bangkok, Thailand, 2024. Association for Computational Linguistics.

[Chen *et al.*, 2018] Yiqi Chen, Tieyun Qian, Huan Liu, and Ke Sun. "Bridge": Enhanced Signed Directed Network Embedding. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pages 773–782, Torino, Italy, 2018. ACM.

[Cheng *et al.*, 2024] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10558–10578, 2024.

[Chiang *et al.*, 2011] Kai-Yang Chiang, Nagarajan Natarajan, Ambuj Tewari, and Inderjit S Dhillon. Exploiting longer cycles for link prediction in signed networks. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1157–1162, Glasgow, United Kingdom, 2011. ACM.

[Derr *et al.*, 2018] Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 929–934, Singapore, 2018. IEEE.

[Duan *et al.*, 2023] Keyu Duan, Qian Liu, Tat-Seng Chua, Shuicheng Yan, Wei Tsang Ooi, Qizhe Xie, and Junxian He. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565*, 2023.

[Dubey *et al.*, 2024] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[Fang *et al.*, 2024] Zhihong Fang, Shaolin Tan, and Yaonan Wang. A signed subgraph encoding approach via linear optimization for link sign prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):14659–14670, 2024.

[Giray, 2023] Louie Giray. Prompt engineering with chatgpt: a guide for academic writers. *Annals of Biomedical Engineering*, 51(12):2629–2633, 2023.

[Heider, 1946] Fritz Heider. Attitudes and cognitive organization. *The Journal of Psychology*, 21(1):107–112, 1946.

[Huang *et al.*, 2019] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. Signed graph attention networks. In *Proceedings of the Artificial Neural Networks and Machine Learning (ICANN)*, pages 566–577, Munich, Germany, 2019. Springer.

[Huang *et al.*, 2021] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. Sdgnn: Learning node representation for signed directed networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 196–203, Vancouver, Canada (Virtual Event), 2021. AAAI Press.

[Huang *et al.*, 2024] Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. Can llms effectively leverage graph structural information through prompts, and why? *Transactions on Machine Learning Research*, 2024.

[Jin *et al.*, 2024] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8622–8642, 2024.

[Kouvaros and Lomuscio, 2021] Panagiotis Kouvaros and Alessio Lomuscio. Towards scalable complete verification of relu neural networks via dependency-based branching. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2643–2650, Montreal, Canada (Virtual Event), 2021. ijcai.org.

[Leskovec *et al.*, 2010] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 641–650, Raleigh, North Carolina, USA, 2010. ACM.

[Lin and Li, 2024] Wanyu Lin and Baochun Li. Status-aware signed heterogeneous network embedding with graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):4580–4592, 2024.

[Liu *et al.*, 2023] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36:21558–21572, 2023.

[MacQueen, 1967] J MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*, pages 281–297, Oakland, CA, USA, 1967.

[Majeed and Hwang, 2024] Abdul Majeed and Seong Oun Hwang. Reliability issues of llms: Chatgpt a case study. *IEEE Reliability Magazine*, 1(4):36–46, 2024.

[Mekrache *et al.*, 2024] Abdelkader Mekrache, Adlen Ksentini, and Christos Verikoukis. Intent-based management of next-generation networks: an llm-centric approach. *IEEE Network*, 38(5):29–36, 2024.

[Ni *et al.*, 2019] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China, 2019. Association for Computational Linguistics.

[Team *et al.*, 2024] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[Wang *et al.*, 2017] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. Signed network embedding in social media. In *Proceedings of the SIAM International Conference on Data Mining (ICDM)*, pages 327–335, Houston, Texas, USA, 2017. SIAM.

[Wang *et al.*, 2023] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[West *et al.*, 2014] Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics*, 2:297–310, 2014.

[Xie *et al.*, 2022] Yu Xie, Yanfeng Liang, Maoguo Gong, A Kai Qin, Yew-Soon Ong, and Tiantian He. Semisupervised graph neural networks for graph classification. *IEEE Transactions on Cybernetics*, 53(10):6222–6235, 2022.

[Xu *et al.*, 2019] Pinghua Xu, Wenbin Hu, Jia Wu, and Bo Du. Link prediction with signed latent factors in signed social networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1046–1054, Anchorage, AK, USA, 2019. ACM.

[Zhang *et al.*, 2023] Zeyu Zhang, Jiamou Liu, Xianda Zheng, Yifei Wang, Pengqian Han, Yupan Wang, Kaiqi Zhao, and Zijian Zhang. Rsgnn: A model-agnostic approach for enhancing the robustness of signed graph neural networks. In *Proceedings of the ACM Web Conference (WWW)*, pages 60–70, Austin, TX, USA, 2023. ACM.

[Zhou *et al.*, 2023] Honglu Zhou, Advith Chegu, Samuel S Sohn, Zuohui Fu, Gerard De Melo, and Mubbasir Kapadia. Harnessing neighborhood modeling and asymmetry preservation for digraph representation learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 6519–6524, Macao,SAR, China, 2023. ijcai.org.