Social Relation Reasoning Based on Triangular Constraints

Yunfei Guo[1, 2], **Fei Yin**[1, 2], **Wei Feng**[1, 2], **Xudong Yan**[3], **Tao Xue**[3], **Shuqi Mei**[3], **Cheng-Lin Liu**[1, 2, 4]
[1]National Laboratory of Pattern Recognition (NLPR),
Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[3]T Lab, Tencent Map, Tencent Technology (Beijing) Co., Ltd., Beijing 100193, China
[4]CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing 100190, China
guoyunfei2019@ia.ac.cn, {fyin,wei.feng}@nlpr.ia.ac.cn, {owenyan,emmaxue,shawnmei}@tencent.com, liucl@nlpr.ia.ac.cn

Abstract

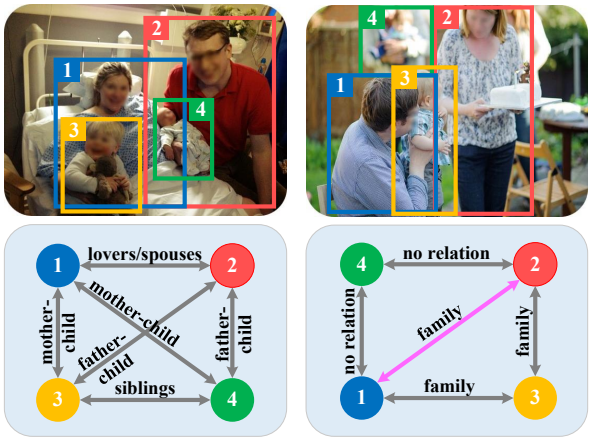
Social networks are essentially in a graph structure where persons act as nodes and the edges connecting nodes denote social relations. The prediction of social relations, therefore, relies on the context in graphs to model the higher-order constraints among relations, which has not been exploited sufficiently by previous works, however. In this paper, we formulate the paradigm of the higher-order constraints in social relations into triangular relational closed-loop structures, i.e., triangular constraints, and further introduce the triangular reasoning graph attention network (TRGAT). Our TRGAT employs the attention mechanism to aggregate features with triangular constraints in the graph, thereby exploiting the higher-order context to reason social relations iteratively. Besides, to acquire better feature representations of persons, we introduce node contrastive learning into relation reasoning. Experimental results show that our method outperforms existing approaches significantly, with higher accuracy and better consistency in generating social relation graphs.

Introduction

Social relations play a core role in social networks that everyone lives in. The prediction of social relations from images has emerged as an important topic in computer vision (Kukleva, Tapaswi, and Laptev 2020; Yan et al. 2021; Goel, Ma, and Tan 2019; Zhang et al. 2019), which has many vital applications such as social event understanding, intelligent robots, and intelligent personal assistants.

Social relation recognition is to classify the relations between persons given the input image and bounding boxes of persons as shown in Figure 1. Usually, multiple persons exist in an image, and due to the transitivity of social relations, potential connections, i.e., high-order relation constraints, are common in their relations (Wang et al. 2022; Li et al. 2020). Most previous methods (Li et al. 2017; Wang et al. 2018; Goel, Ma, and Tan 2019) adopted the pairwise prediction, where the features of two persons are extracted, fused, and classified to predict between-person relations. These methods handled multiple pairwise relations in the same image independently, ignoring the higher-order constraints. Recently, Li et al. (Li et al. 2020) tried to address this problem with graph neural networks, which constructed a social relation graph for each image and conducted joint reasoning through graph convolutional networks (GCN) (Kipf and Welling 2016) and gated recurrent units (GRU) (Cho et al. 2014). This method exploits better the context information than the independent prediction of pairwise relations, but due to its message propagation mechanism from traditional graph neural networks, it does not utilize higher-order constraints in graphs sufficiently. Although some other computer vision tasks, such as scene graph generation (Xu et al. 2017; Yang et al. 2018; Zellers et al. 2018) and group activity recognition (Ibrahim and Mori 2018; Wu et al. 2019; Tang et al. 2019), achieved success in relation reasoning, the relations they studied have weak logical constraints (Li et al. 2020). Thus most of their methods still employ simple message propagation mechanisms without delving into higher-order constraints.

To exploit the higher-order context in graphs, we take a look back at how human beings reason about social relations. As shown in Figure 1(b), supposing that we do not know the relation (the pink arrow) between node 1 (the man) and node 2 (the woman), we will look for an intermediary — node 3 (the child) — between them. Thereafter we predict the relation between the man and woman via their rela-



(a) (b)

Figure 1: Images and corresponding annotations from (a) the PIPA dataset and (b) the PISC dataset. For better visualization, we omit some persons and their relations.

tions with the child, naturally and interpretably. We call this ternary constraint the triangular constraint. Obviously, triangular constraints can assist relation reasoning. In case of Figure 1(a), every two-node relation can be predicted within a triangular relation by indirect reasoning. However, not all triangular relations contain strong triangular constraints. For example, node 4 in Figure 1(b) is not a good intermediary for the pink relation because "no relation" is of little help to us. Therefore, we conclude that higher-order constraints can be utilized through triangular constraints and valid triangular constraints are beneficial to social relation prediction.

Based on the above analysis, we propose a novel graph neural network for social relation recognition, which we call Triangular Reasoning Graph Attention neTwork (TRGAT). To incorporate the triangular constraints into message propagation for an edge, we first seek all triangular relations containing the edge; then we generate attention coefficients by evaluating the importance or validity of each triangular relation to the edge; and finally, we perform weighted aggregation of features in these triangular relations to obtain the final edge features for classification. Such triangular relation reasoning combines the direct reasoning from paired persons and the indirect reasoning from triangular constraints, encouraging more consistent and robust relation recognition than pairwise prediction. Besides, to acquire better feature representations of persons to enhance the reasoning ability of our TRGAT, we introduce contrastive learning (He et al. 2020; Chen et al. 2020; Grill et al. 2020) into relation reasoning. The proposed so-called node contrastive learning can learn better representations of persons by designing contrastive loss between features of the same person or different persons. Along with the supervised learning of social relations, the node contrastive learning enhances the learning ability and improves the performance of our model.

To summarize, our contributions are in three folds:

- We propose the triangular reasoning graph attention network based on the triangular constraints, which can integrate higher-order constraints efficiently and perform social relation reasoning better.
- We propose node contrastive learning to guide our model to acquire better feature representations of persons and improve the final performance.
- Our proposed method outperforms existing methods significantly, with higher accuracy and better consistency in generating social relation graphs.

Related Work

Social Relation Recognition

Previous methods for social relation recognition can be divided into two types based on their technical routes: pairwise prediction methods (Li et al. 2017; Wang et al. 2018; Goel, Ma, and Tan 2019; Zhang et al. 2019) and joint reasoning methods (Li et al. 2020). Li et al. (Li et al. 2017) proposed a dual-glance model based on pairwise prediction to explore the image context. Zhang et al. (Zhang et al. 2019) introduced additional pose cues of persons and adopted graph convolutional networks (GCN) (Kipf and Welling 2016) for pairwise prediction. Without considering the constraints among relations, these methods did not achieve satisfactory performance. To address this problem, Li et al. (Li et al. 2020) constructed a social relation graph for the entire input image and proposed $\mathrm{GR}^2\mathrm{N}$ based on GCN and GRU for joint relation reasoning. Despite great success made by this method, it did not sufficiently exploit higher-order relation constraints and integrate them into the message propagation mechanism of graph networks.

Higher-Order Networks

Higher-order interaction is a vital issue in graph theory. Since pairwise interaction networks are deemed insufficient for many complex systems (Battiston et al. 2020, 2021), many works (Iacopini et al. 2019; Wang et al. 2022) explored higher-order interaction by adopting higher-order networks. Inspired by the Weisfeiler-Lehman algorithm, Morris et al. (Morris et al. 2019) introduced a high-order graph neural network and applied to molecular learning, showing excellent performance. However, visual relation reasoning works have not adopted well higher-order networks. Gao et al. (Gao et al. 2021) adopted high-order graph networks but only for input feature fusion rather than context aggregation. For social relations with strong high-order constraints, an efficient high-order network is necessary.

Contrastive Learning

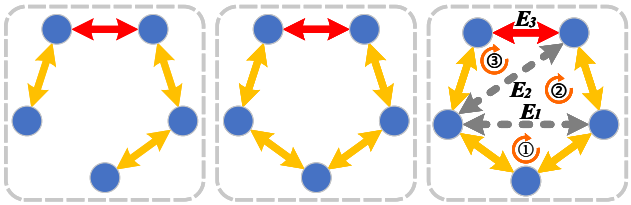
Contrastive learning is a self-supervised learning method to learn general representations from datasets without labels. Since remarkable advances (He et al. 2020; Oord, Li, and Vinyals 2018; Chi et al. 2021) have been made, it has been applied to a variety of downstream tasks. Recently, some methods (Dai and Lin 2017; Ma et al. 2021; Li et al. 2021) introduced contrastive learning to assist supervised training. Dai et al. (Dai and Lin 2017) applied contrastive learning to image captioning to encourage discriminativeness. Hu et al. (Hu, Cui, and Wang 2021) proposed region-aware contrastive learning for supervised semantic segmentation. Inspired by these methods, we deploy contrastive learning as an auxiliary training method to guide the model to obtain better feature representations for persons.

Triangular Relation Reasoning

In this section, we give a comprehensive explanation of how the triangular relation reasoning of our TRGAT captures higher-order constraints to perform indirect relation reasoning in theory.

Observation 1. *Social relations in each image can be represented in a graph whose nodes stand for the persons and edges stand for their relations. In the graph, higher-order constraints always exist in relational closed loops.*

Due to the transitivity of social relations, the chain-like relations acquire information redundancy only after forming a closed loop (Figure 2(b)), thus producing the high-order constraints. In contrast, non-closed-loop structures do not have information redundancy. As illustrated in Figure 2(a), the red edge does not have external constraints, and the relation it represents depends only on its two endpoints.



(a) (b) (c)

Figure 2: Several examples of graph structures. (a) a relational non-closed-loop graph structure; (b) a relational closed loop; (c) a case of decomposing the relational closed loop into triangular relational closed loops.

Such kind of high-order relations is helpful as it provides each of its edges with more robust contextual cues than visual features. In some cases, final relation result can even be determined by indirect reasoning from its higher-order constraint as Figure 1(a). Unfortunately, it is hard to make full use of all relational loops. In a fully connected graph, the number of relational closed loops grows exponentially with the number of nodes.

Observation 2. *The closed-loop structure can be decomposed into a series of triangular loops by adding edges, and its constraint information can also be transferred to them.*

We denote these triangular loops as the triangular relational closed loops. The indirect relation reasoning from any closed loops can be realized by a series of triangular relation reasoning on these triangular relational closed loops. As a case shown in Figure 2(c), to get E_3 during reasoning, we can first get E_1 via loop ①, then E_2 via loop ②, and finally E_3 via loop ③. Through such iterations, the higher-order constraint of the entire closed loop can be perceived by its every edge member. Moreover, the amount of computation is reduced to a polynomial level. Theoretically, the more iterations, the higher order constraints can be captured.

Method

The overview of our framework is diagrammed in Figure 3. Given the input image and bounding boxes of persons, first, a backbone network (CNN+FPN) is used to extract features for the entire image, then visual features are extracted by RoIAlign (He et al. 2017) and spatial maps are generated from bounding boxes. The extracted visual and spatial features are fused into the input graph to perform relation reasoning through our TRGAT.

Graph Construction

We model the persons and social relations into a relational graph. To acquire the feature representations of nodes and edges, we first extract visual features of the entire image using a convolutional neural network (CNN), followed by a feature pyramid network (FPN) (Lin et al. 2017a) to fuse high-level semantic and low-level textural information. Thus we get the backbone feature F_b.

Node Features. For persons, we employ RoIAlign (He et al. 2017) to extract features from F_b based on their bounding boxes and scale these features to a fixed size $C_v \times S_v \times S_v$, where C_v is the original channels of the features and S_v denotes width and height of the scaled visual features. We call these features visual features of persons F_{pv}. Considering that the diverse pose of persons incurs a lot of background inside many bounding boxes, we apply the spatial attention mechanism, which is composed of convolutional and activation layers, to focus on salient regions. At last, we use ReLU (Glorot, Bordes, and Bengio 2011) for attention coefficient truncation. This process can be formulated as:

$$\hat{F}_{pv} = F_{pv} \odot ReLU\left(RA\left(F_{pv}\right)\right), \tag{1}$$

where $\hat{F}_{pv}$ is the visual feature of a person after region attention; $\odot$ stands for element-wise product; $RA : R^{C_v \times S_v \times S_v} \rightarrow R^{1 \times S_v \times S_v}$ is our region attention network.

To fuse spatial information, we generate spatial map features from bounding boxes of persons. The spatial location of a particular person in the entire image is mapped into a feature map sized $S_s \times S_s$, with value 1 inside the mapped bounding box and 0 outside it, where S_s denotes the width and height of spatial maps. We call these features the spatial features of persons $F_{ps} \in R^{S_s \times S_s}$. Then the final node features F_N are obtained from the sum of transformed $\hat{V}_{pv}$ and V_{ps}, both of which are 1D vectors stretched from $\hat{F}_{pv}$ and F_{ps} respectively.

$$F_N = f_{pv}\left(\hat{V}_{pv}\right) + f_{ps}\left(V_{ps}\right), \tag{2}$$

where $f_{pv} : R^{C_v S_v^2} \rightarrow R^D$ and $f_{ps} : R^{S_s^2} \rightarrow R^D$ are fully connected layers.

Edge Features. For the relation between two persons, we materialize it as the region of the union box of the two persons. The features are also extracted from F_b by RoIAlign. We call them relational visual features $F_{rv} \in R^{C_v \times S_v \times S_v}$. The relational spatial features $F_{rs} \in R^{S_s \times S_s}$ are defined and generated in the same manner as the persons. For relational visual features, we do not employ the spatial attention mechanism. The edge features can be generated by the sum of the transformed V_{rv} and V_{rs}, both of which are 1D vectors stretched from F_{rv} and F_{rs} respectively:

$$F_E = f_{rv}\left(V_{rv}\right) + f_{rs}\left(V_{rs}\right), \tag{3}$$

where $f_{rv} : R^{C_v S_v^2} \rightarrow R^D$; $f_{rs} : R^{S_s^2} \rightarrow R^D$.

Graph Structure. We build a fully connected graph as the input graph because we require fully connected graphs to decompose all relational closed loops and transfer relation constraints to them. In our graph, each undirected edge is represented by two oppositely directed edges with the same initial features. When testing, we average the features of the two edges and then classify them.

Triangular Reasoning GAT

Based on the proposed triangular relation reasoning and principle of graph attention network (GAT) (Veličković et al. 2018), we design a novel graph neural network called Triangular Reasoning Graph Attention neTwork (TRGAT). The main differences from GAT lie in two update steps of each layer in TRGAT: node aggregation and edge aggregation.

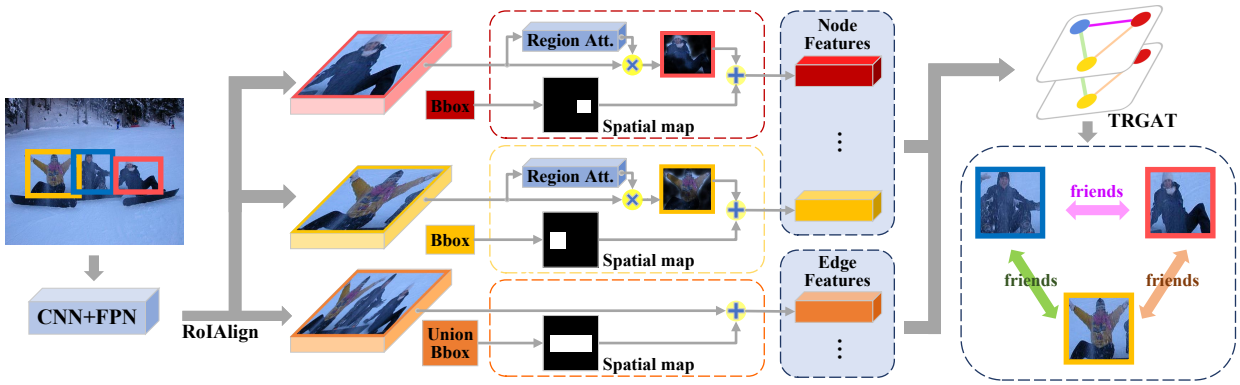


Figure 3: The overview of our framework.

Node Aggregation. We denote node features before node aggregation as $F_N \in R^D$ and edge features as $F_E \in R^D$, while denote the node features after aggregation as $F_N^{'} \in R^D$, where D is the dimension of node and edge feature vectors. During the aggregation, multi-head attention mechanism is deployed for a weighted sum of inflow edge features. For clarity, we adopt one-head attention to explain. The attention coefficient between i-th node N_i and inflow edge E_{ji} from j-th node N_j can be calculated by:

$$\alpha_{ji} = \frac{\exp\left(\sigma\left(f_A\left(F_{E_{ji}} \parallel F_{N_i}\right)\right)\right)}{\sum\limits_{k \in \mathbb{N}_i} \exp\left(\sigma\left(f_A\left(F_{E_{ki}} \parallel F_{N_i}\right)\right)\right)}, \tag{4}$$

where we use leaky ReLU (Maas et al. 2013) as the activation function $\sigma(*)$; $f_A : R^{2D} \rightarrow R$ is a fully connected layer for the attention mechanism between node and edge; $\parallel$ is the concatenation operation; $\mathbb{N}_i$ stands for the neighbor of N_i.

Then we perform weighted aggregation through the attention coefficients. This process can be explained as follows:

$$F_{N_i}^{'} = \sigma\left(f_N\left(\sum_{j \in \mathbb{N}_i} \alpha_{ji} F_{E_{ji}}\right) \parallel F_{N_i}\right), \tag{5}$$

where $f_N : R^{2D} \rightarrow R^D$ is a fully connected layer to transform node features.

Edge Aggregation. Edge aggregation first adopts the multi-head attention mechanism to fuse the two endpoint features of a specific edge. The fused features, represented by $F_E^{'}$, can be calculate by:

$$F_{E_{ji}}^{'} = \sigma\left(f_E\left(\left(\alpha_{ij} F_{N_j} + \alpha_{ji} F_{N_i}\right) \parallel F_{E_{ji}}\right)\right), \tag{6}$$

where $f_E : R^{2D} \rightarrow R^D$ is a fully connected layer to transform edge features; α_{ji} and α_{ij} are the same to the ones in node aggregation. The above aggregation is inspired by (Li, Yin, and Liu 2020; Ye et al. 2019), which combines the source node information and the target node information of the edge, but ours is more concise.

Then we fuse features of triangular relational closed loops into each edge. For a specific edge E_{ji}, its triangular relational closed loops can be obtained by enumerating all common neighboring nodes of its two endpoints N_j and N_i. Each of these nodes forms a triangular relational closed loop with the two endpoints. Coefficients of multi-head attention are calculated by evaluating the importance of a specific triangular relational closed loop $\triangle\, jki$ to the edge E_{ji}:

$$\beta_{\triangle jki} = \frac{\exp\left(\sigma\left(f_{TRA}\left(F_{E_{jk}}^{'} \parallel F_{E_{ki}}^{'} \parallel F_{E_{ji}}^{'}\right)\right)\right)}{\sum\limits_{\triangle jhi \in \Omega_{ji}} \exp\left(\sigma\left(f_{TRA}\left(F_{E_{jh}}^{'} \parallel F_{E_{hi}}^{'} \parallel F_{E_{ji}}^{'}\right)\right)\right)}, \tag{7}$$

where $\triangle\, jhi$ demonstrates the triangular relational closed loop composed by E_{jh}, E_{hi}, and E_{ji}; $f_{TRA} : R^{3D} \rightarrow R^D$ is a fully connected layer for the attention mechanism between the edge and its triangular relational closed loop; Ω_{ji} is a set of all triangular relational closed loops of E_{ji}.

Finally, the aggregation procedure is achieved by performing an attention-weighted summation of all relational bypass features in triangular relational closed loops, concatenating with the original edge feature, and transforming it via a fully connected layer.

$$F_{E_{ji}}^{''} = \sigma\left(f_{TR}\left(\sum_{\triangle jki \in \Omega_{ji}} \beta_{\triangle jki}\left(F_{E_{jk}}^{'} \parallel F_{E_{ki}}^{'}\right) \parallel F_{E_{ji}}^{'}\right)\right), \tag{8}$$

where $f_{TR} : R^{3D} \rightarrow R^D$ is a fully connected layer to transform the concatenated features; $F_{E_{ji}}^{''}$ is the final feature of E_{ji} after edge aggregation.

Reasoning in Graph. The above aggregations integrate triangular constraints in triangular closed loops into each edge of the graph. To obtain higher-order constraints, we stack multiple these above layers for iterative reasoning. As explained previously, the more layers we stack, the higher order constraints our network can capture.

To acquire final relation results, we adopt a C-way linear classifier CLS with a binary focal loss (Lin et al. 2017b) l_f for edge classification. C is the number of relation classes. This process can be formulated by:

$$L_{rel} = \frac{1}{C}\sum_{i=1}^{C} l_f\left(CLS^i\left(F_E\right), T_E^i\right), \tag{9}$$

where CLS^i is the i-th way in CLS; F_E is the output edge feature of our TRGAT; T_E^i is a boolean denoting whether the edge is in the class i.

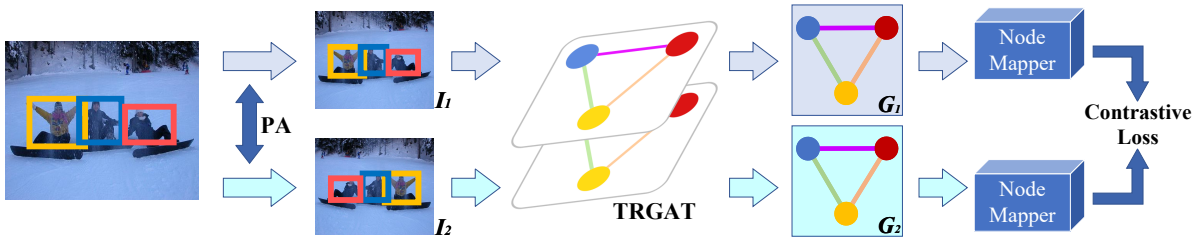


Figure 4: The procedure of node contrastive learning. There are two parallel processing lines (in different colors) on the same image, and we finally obtain two graphs G_1 and G_2 with the same structure. The two node mappers share the same parameters. "PA" means Pairwise Augmentation.

Node Contrastive Learning

Contrastive learning is a way of self-supervised learning, which does not require annotations for supervising. Contrastive learning has been widely used in computer vision (Hu, Cui, and Wang 2021; Wu et al. 2018; Dai and Lin 2017). But most methods use contrastive learning as a pre-training approach. In this paper, we introduce contrastive learning as an auxiliary training method to provide a cheap supervision for nodes.

In relation reasoning, we hope that the graph neural network can learn the characteristic of each person. Since bounding boxes of persons may have serious overlapping, and many bounding boxes contain two or even more persons, a good relation reasoning network is expected to distinguish the dominant person from others within the bounding box. Node contrastive learning provides us with a usable way to guide our graph neural network to obtain better and more discriminative feature representations of persons.

The procedure of node contrastive learning is illustrated in Figure 4. For an input image I, we first perform pairwise augmentation, involving random flipping, small-angle tilting, and random blurring, to get two forms of a single image I_1 and I_2. This pair of images are input into our framework and finally output a pair of graphs G_1 and G_2 with the same structure. In the end, we employ a MultiLayer Perception (MLP) as a node mapper to map the nodes of the two graphs into output space. Here we define the number of nodes in G_1 or G_2 as n, the union set of nodes in G_1 and G_2 as U, whose length is $2n$. Then we take the nodes representing the same person, e.g., N_i in G_1 and N_{i^*} in G_2, as positive samples for each other, while the nodes denoting different persons are negative samples, whether they are in the same graph. Contrastive loss (Wu et al. 2018) is exploited to achieve the effect that the features of the same person are more similar and features of different persons are more differentiated. This loss L_{contr} can be expressed as follows:

$$L_{contr} = -\frac{1}{n}\sum_{i=1}^{n}\log l_{contr}\left(N_i, N_{i^*}\right), \tag{10}$$

$$l_{contr}\left(N_i, N_j\right) = \frac{\exp\left(SIM\left(F_{N_i}, F_{N_j}\right)/\tau\right)}{\sum\limits_{N_k \in U, k \neq i}\exp\left(SIM\left(F_{N_i}, F_{N_k}\right)/\tau\right)}, \tag{11}$$

where F_N is the output node feature from MLP; τ is a hyper-parameter representing the temperature coefficient; SIM is the similarity measure function, where we adopt cosine similarity. Notably, we extend the above loss to a batch, where the nodes of graphs generated by different images are negative samples of each other.

The total loss is a weighted sum of relation loss L_{rel} and node contrastive loss L_{contr} with a hyperparameter λ.

$$L_{total} = L_{rel} + \lambda L_{contr}. \tag{12}$$

Experiments

To evaluate the performance of the proposed method, we conduct extensive experiments on the two popular benchmark datasets: PISC (Li et al. 2017) and PIPA (Sun, Schiele, and Fritz 2017).

Experiment Setting

Dataset. The People in Social Context (PISC) dataset (Li et al. 2017) is a large-scale social relation dataset with 22670 images. The average number of persons per image is 3.11. It has two types of social relation categories: 3 coarse relations and 6 fine relations. The coarse setting has a train/-val/test split of 13142, 4000 and 4000 images while the fine is 16828, 500 and 1250 images. The evaluation metrics are to calculate the per-class recall for each relation and the mean average precision (mAP) over all relations. The People in Photo Album (PIPA) (Sun, Schiele, and Fritz 2017) dataset contains 8570 images. The average number of persons per image is 2.56. The relations it defines have 16 categories covering 6 domains. For fair comparisons, we adopt the standard train/val/test split introduced by (Sun, Schiele, and Fritz 2017), which has a train/val/test split of 5857, 261, and 2452 images. The top-1 classification accuracy of all relation pairs is the final metric.

Implementation Details. Our implementation is based on PyTorch (Paszke et al. 2019) and MMDetection (Chen et al. 2019) framework. Following other methods' backbone setting (Li et al. 2020), our stem network adopts ResNet-101 (He et al. 2016) inheriting parameters pre-trained on ImageNet dataset (Krizhevsky, Sutskever, and Hinton 2012). To balance the performance and speed, we set the layer number of our TRGAT to 2. Features representing persons and relations are set to 2048-D vectors. In training, we scale the long

Method	PISC-Coarse				PISC-Fine							PIPA
	Int	Non	NoR	mAP	Fri	Fam	Cou	Pro	Com	NoR	mAP	Acc
Pair CNN (Li et al. 2017)	70.3	80.5	38.8	65.1	30.2	59.1	69.4	57.5	41.9	34.2	48.2	58.0
Dual-Glance (Li et al. 2017)	73.1	84.2	59.6	79.7	35.4	68.1	76.3	70.3	57.6	60.9	63.2	59.6
GRM (Wang et al. 2018)	81.7	73.4	65.5	82.8	59.6	64.4	58.6	76.6	39.5	67.7	68.7	62.3
MGR (Zhang et al. 2019)	-	-	-	-	64.6	67.8	60.5	76.8	34.7	70.4	70.1	64.4
SRG-GN (Goel, Ma, and Tan 2019)	-	-	-	-	25.2	80.0	100.0	78.4	83.3	62.5	71.6	53.6
GR^2N (Li et al. 2020)	81.6	74.3	70.8	83.1	60.8	65.9	84.8	73.0	51.7	70.4	72.7	64.3
TRGAT	82.2	75.2	73.0	**86.1**	50.1	69.4	78.9	72.4	79.1	71.2	**76.2**	**64.8**
GR^2N-NCL	80.4	76.1	74.6	85.5	55.4	70.2	78.9	71.5	70.9	70.7	75.5	64.9
TRGAT-NCL	81.4	76.6	75.3	**87.6**	58.2	73.1	78.9	76.7	70.6	73.6	**78.2**	**65.3**

Table 1: Comparison with other methods on the PISC and PIPA dataset. The results of PISC include per-class recall and mean Average Precision (mAP) and the results of PIAP are top-1 accuracy, all of which are in %. "NCL" stands for our node contrastive learning. (The abbreviations and corresponding full names of relation types are as follows. Int: Intimate, Non: Non-Intimate, NoR: No Relation, Fri: Friends, Fam: Family, Cou: Couple, Pro: Professional, Com: Commercial)

edge of input images to 600, 720, or 960 randomly while keeping the aspect ratio and 720 in testing. We train our model with the stochastic gradient descent method (SGD). All the experiments are conducted with a batch size of 16 on 2 GPUs. The implementation is on a workstation with a 2.40GHz 56-core CPU, 256G RAM, GTX Titan RTX, and 64-bit CentOS.

Comparison with the State-of-the-Art

To prove the superiority of our model, we compare it with several existing state-of-the-art methods. As shown in Table 1, the first five methods are based on pairwise input of persons and pairwise prediction. Our experimental results in Table 1 show that our method, with higher-order knowledge, surpasses these five methods significantly — nearly 7% of fine relations on PISC. Since most of the images in the PIPA test set have only two persons and additional annotations are utilized by most of these methods, the 1% improvement is not so obvious.

GR^2N tries to grasp the logical constraints between different types of social relations through joint reasoning in a social relation graph. But due to its simple way of message passing, it fails to exploit higher-order constraints among relations sufficiently, while our model successfully incorporates the triangular constraints into message propagation of edges, takes full use of higher-order context in graphs, and achieves better performance. As shown in Table 1, our method surpasses GR^2N by 4.5% on coarse relations, 5.5% on fine relations of PISC, and 1% on PIPA, which shows our model's superiority.

Ablation Studies

We conduct some ablation studies to verify the effectiveness of our framework, including comparison with vanilla GAT (Veličković et al. 2018) and the use of region attention, spatial features, reweighting, and node contrastive learning. The reweighting is to relieve the imbalance in the fine relations of PISC dataset. Concretely, we first freeze all the weight parameters except the linear classification layer, and then reweight the loss function with the coefficients calculated from the training set to refine the linear classification layer.

Method	RAtt	Spa	RW	C-mAP	F-mAP
GAT	✗	✗	✗	81.3	69.3
TRGAT	✗	✗	✗	85.7	74.6
TRGAT	✓	✗	✗	85.8	74.9
TRGAT	✓	✓	✗	86.1	75.5
TRGAT	✓	✓	✓	**86.1**	**76.2**

Table 2: Ablation studies on PISC dataset. (RAtt: Region Attention; Spa: Spatial features; RW: Reweighting method; C/F-mAP: mAP (in %) of coarse/fine relations.

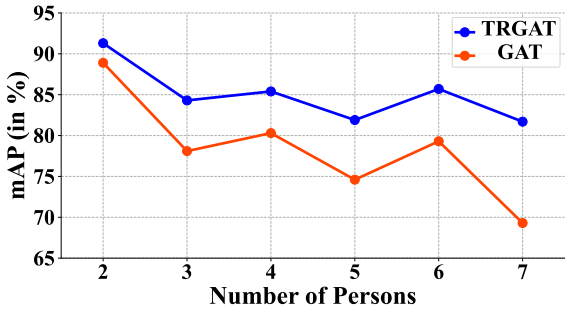


Figure 5: Variation of mAP (in %) with the number of persons in a image.

Comparison with GAT. We implement the original GAT on the PISC dataset. Following our backbone network, features of persons are extracted and embedded as nodes in the input graph. Then the input graph is sent to the GAT for refinement. Finally, we fuse the features of paired persons for joint reasoning of relations. Beyond that, all training and test settings are consistent with TRGAT.

The performance of GAT on the PISC dataset is shown in Table 2, lower than our method by 4.4% on coarse relations and 5.3% on fine relations. The reason is that the vanilla GAT can only aggregate the information from neighboring nodes in message propagation. Lacking the incorporation

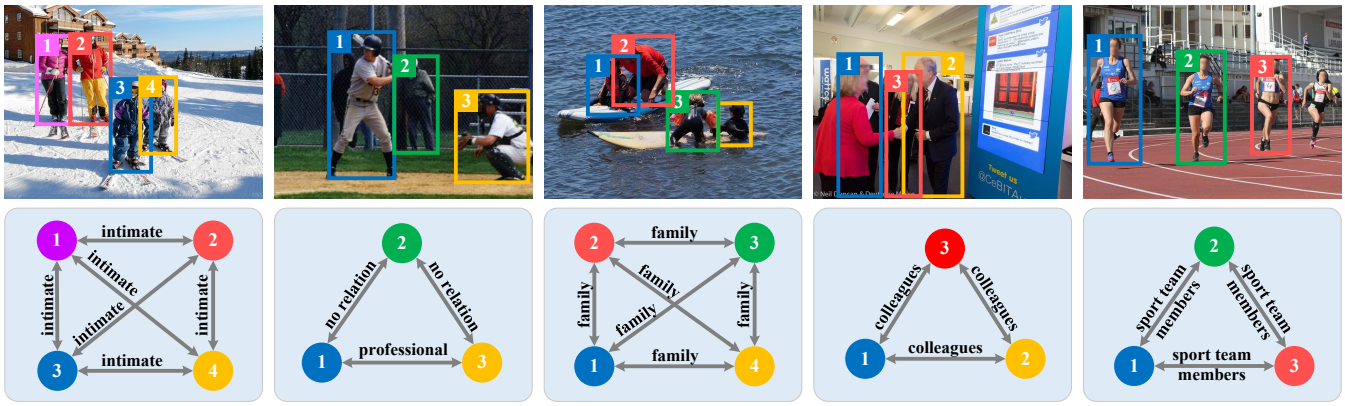


Figure 6: Visualization of qualitative results. The graph below each image is the social relation graph generated by our TRGAT.

Method	Number of Layers	F-mAP (in %)	FPS
TRGAT	1	77.4	19.2
TRGAT	2	78.2	15.5
TRGAT	3	78.1	12.7
GAT	3	69.3	19.1

Table 3: Ablation studies about setting of layers of our TRGAT on the PISC dataset.

of triangular relation features makes it hard to perceive the higher-order context. A more intuitive experiment is conducted and shown in Figure 5. We split the test set of coarse relations according to the number of persons existing in each image, and then test our TRGAT and GAT on images with different person numbers separately. Since the difficulty of relation reasoning is positively correlated with the number of persons, the reasoning ability of these models shows a downward tendency with the increase of persons. However, compared with GAT, our TRGAT drops more slowly. This illustrates the effectiveness of TRGAT for modeling relations between multiple persons with higher-order constraints.

RAtt & Spa & RW. The region attention mechanism weights the features after RoIAlign to make the model focus on some salient regions to acquire higher-quality feature representations of persons. Spatial features also exert their power in relation reasoning because two persons that are far apart in space are less likely to be intimate. Reweighting overcomes the imbalance in relations of the PISC dataset. These methods are simple but effective. Our model obtains a gain of 0.4% on coarse relations and 1.6% on fine relations.

Node Contrastive Learning. Our node contrastive learning aims at better feature representation of persons, thus conduce to relation reasoning. Here we compare the final performance of our model with and without node contrastive learning. As shown in Table 1, node contrastive learning brings 1.5% and 2.0% improvement to our model on coarse and fine relations respectively, which proves its effectiveness and significance. More importantly, our node contrast learning is general enough. We apply it to another graph-based social relation recognition network $\mathrm{GR}^2\mathrm{N}$, which can also bring 2.4% improvement in coarse relations, 2.8% in fine relations in the PISC dataset, and 0.6% on the PIPA dataset.

Setting of layers and Speed. We display the layer number setting, performance, and reasoning speed on the PISC dataset in Table 3. Since the relation annotations of PISC are dense enough, two layers are sufficient for capturing the higher-order constraints. Therefore, to balance performance and speed, we adopt a two-layer structure.

Qualitative Analyses

We visualize the results reasoned by our model in the two popular datasets. As is shown in Figure 6, our model is able to utilize the higher-order context by incorporating the triangular constraints, thus avoiding some unreasonable and contradictory relations and obtaining a consistent social relation graph. For example, in the first family image, all relations between family members should be intimate.

Conclusion

In this paper, we formulate the higher-order constraints in social relations into closed-loop structures, further incorporate the triangular relation reasoning into graph neural networks and propose TRGAT. Our TRGAT performs weighted aggregation of triangular relation features by introducing the attention mechanism, so as to perceive higher-order context in graphs and provide accurate reasoning results. To improve the learning ability of our TRGAT, we propose node contrastive learning that enables our TRGAT to grasp a better feature representation of each person. Experimental results show that our method outperforms existing methods significantly, with higher accuracy and better consistency in generating social relation graphs.

Acknowledgments

This work has been supported by the National Key Research and Development Program Grant 2018AAA0100400, and the National Natural Science Foundation of China (NSFC) Grants U20A20223 and 61721004.

References

Battiston, F.; Amico, E.; Barrat, A.; Bianconi, G.; Ferraz de Arruda, G.; Franceschiello, B.; Iacopini, I.; Kéfi, S.; Latora, V.; Moreno, Y.; et al. 2021. The physics of higher-order interactions in complex systems. *Nature Physics*, 1093–1098.

Battiston, F.; Cencetti, G.; Iacopini, I.; Latora, V.; Lucas, M.; Patania, A.; Young, J.-G.; and Petri, G. 2020. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 1–92.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. 2019. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 1597–1607.

Chi, Z.; Dong, L.; Wei, F.; Yang, N.; Singhal, S.; Wang, W.; Song, X.; Mao, X.-L.; Huang, H.-Y.; and Zhou, M. 2021. InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 3576–3588.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Dai, B.; and Lin, D. 2017. Contrastive learning for image captioning. *Advances in Neural Information Processing Systems (NIPS)*, 898–907.

Gao, J.; Qing, L.; Li, L.; Cheng, Y.; and Peng, Y. 2021. Multi-scale features based interpersonal relation recognition using higher-order graph neural network. *Neurocomputing*, 243–252.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep Sparse Rectifier Neural Networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (ICAIS)*, 315–323.

Goel, A.; Ma, K. T.; and Tan, C. 2019. An end-to-end network for generating social relationship graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11186–11195.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems (NIPS)*, 21271–21284.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9729–9738.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2961–2969.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hu, H.; Cui, J.; and Wang, L. 2021. Region-Aware Contrastive Learning for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16291–16301.

Iacopini, I.; Petri, G.; Barrat, A.; and Latora, V. 2019. Simplicial models of social contagion. *Nature communications*, 1–9.

Ibrahim, M. S.; and Mori, G. 2018. Hierarchical relational networks for group activity recognition and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 721–736.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS)*, 2361–2367.

Kukleva, A.; Tapaswi, M.; and Laptev, I. 2020. Learning interactions and relationships between movie characters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9849–9858.

Li, J.; Wong, Y.; Zhao, Q.; and Kankanhalli, M. S. 2017. Dual-glance model for deciphering social relationships. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2650–2659.

Li, W.; Duan, Y.; Lu, J.; Feng, J.; and Zhou, J. 2020. Graph-based social relation reasoning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 18–34.

Li, X.-H.; Yin, F.; and Liu, C.-L. 2020. Page segmentation using convolutional neural network and graphical model. In *International Workshop on Document Analysis Systems (DAS)*, 231–245.

Li, Z.; Wu, W.; Shou, M. Z.; Li, J.; Li, S.; Wang, Z.; and Zhou, H. 2021. Contrastive Learning of Semantic and Visual Representations for Text Tracking. *arXiv preprint arXiv:2112.14976*.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017a. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2117–2125.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017b. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2980–2988.

Ma, S.; Zeng, Z.; McDuff, D.; and Song, Y. 2021. Contrastive Learning of Global and Local Video Representations. *Advances in Neural Information Processing Systems (NIPS)*, 7025–7040.

Maas, A. L.; Hannun, A. Y.; Ng, A. Y.; et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, 3.

Morris, C.; Ritzert, M.; Fey, M.; Hamilton, W. L.; Lenssen, J. E.; Rattan, G.; and Grohe, M. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 4602–4609.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NIPS)*, 8026–8037.

Sun, Q.; Schiele, B.; and Fritz, M. 2017. A domain based approach to social relation recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3481–3490.

Tang, J.; Shu, X.; Yan, R.; and Zhang, L. 2019. Coherence constrained graph LSTM for group activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 636–647.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Wang, H.; Ma, C.; Chen, H.-S.; Lai, Y.-C.; and Zhang, H.-F. 2022. Full reconstruction of simplicial complexes from binary contagion and Ising data. *Nature Communications*, 1–10.

Wang, Z.; Chen, T.; Ren, J.; Yu, W.; Cheng, H.; and Lin, L. 2018. Deep reasoning with knowledge graph for social relationship understanding. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1021–1028.

Wu, J.; Wang, L.; Wang, L.; Guo, J.; and Wu, G. 2019. Learning actor relation graphs for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9964–9974.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3733–3742.

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L. 2017. Scene graph generation by iterative message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5410–5419.

Yan, C.; Liu, Z.; Li, F.; Cao, C.; Wang, Z.; and Wu, B. 2021. Social Relation Analysis from Videos via Multi-entity Reasoning. In *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*, 358–366.

Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D. 2018. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 670–685.

Ye, J.-Y.; Zhang, Y.-M.; Yang, Q.; and Liu, C.-L. 2019. Contextual stroke classification in online handwritten documents with graph attention networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 993–998.

Zellers, R.; Yatskar, M.; Thomson, S.; and Choi, Y. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5831–5840.

Zhang, M.; Liu, X.; Liu, W.; Zhou, A.; Ma, H.; and Mei, T. 2019. Multi-granularity reasoning for social relation recognition from images. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 1618–1623.