A GNN-based Architecture for Group Detection from Spatio-temporal Trajectory Data

Maedeh Nasri[1 [0000−0002−8547−0946]], Zhizhou Fang[2 [0000−0002−7013−8182]], Mitra Baratchi[2 [0000−0002−1279−9310]], Gwenn Englebienne[3 [0000−0002−3130−2082]], Shenghui Wang[3 [0000−0001−8149−1590]], Alexander Koutamanis[4 [0000−0002−0355−1276]], and Carolien Rieffe[1,3,5 [0000−0002−7584−6698]]

[1] Department of Developmental Psychology, Leiden University, The Netherlands
[2] Leiden Institute of Advanced Computing, Leiden University, The Netherlands
[3] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands
[4] Faculty of Architecture & the Built Environment, Delft University of Technology, The Netherlands
[5] Department of Psychology and Human Development, University College London, UK

Abstract. Detecting and analyzing group behavior from spatio-temporal trajectories is an interesting topic in various domains, such as autonomous driving, urban computing, and social sciences. This paper revisits the group detection problem from spatio-temporal trajectories and proposes "WavenetNRI", a graph neural network (GNN) based method. The proposed WavenetNRI extends the previously proposed neural relational inference (NRI) method (an unsupervised learning approach for inferring interactions from observational data) in two directions: (1) symmetric edge features and edge updating processes are applied to generate symmetric edge representations corresponding to the symmetric binary group relationships; (2) a gated dilated residual causal convolutional (GD-RCC) block is adopted to capture both short and long dependency of the edge feature sequences. We evaluated the performance of the proposed model on three simulation datasets and three real-world pedestrian datasets, using the Group Mitre metric to measure the quality of the predicted groups. We compared WavenetNRI with four baseline methods, including two clustering-based and two classification-based methods. In these experiments, NRI and WavenetNRI outperformed all other baselines on the group-interaction simulation datasets, while NRI performed slightly better than WavenetNRI. On the pedestrian datasets, the WavenetNRI outperformed other classification-based baselines. However, it did not compete against the clustering-based methods. Our ablation study showed that while both proposed changes cannot be effective at the same time, either of them can improve the performance of the original NRI on one dataset type.

Keywords: Group Detection · spatio-temporal data · Deep learning.

1 Introduction

Detecting group behaviors based on users' spatio-temporal trajectories has numerous social and urban applications [1,10,4]. For example, detecting groups of pupils playing in a schoolyard facilitates psychologists in understanding pupils' social behavior [10]. Most previous studies in group detection tasks relied on heavy feature engineering [13,16]. These approaches extract selected features from raw trajectory data based on domain knowledge specific to an application area. This restricts generalization to other similar problems. This approach may also ignore informative underlying spatio-temporal patterns that are present in the raw data.

Recently, graph neural networks (GNNs) showed strong potential for relational reasoning [2]. GNNs could be used in group detection by modeling agents (or members of a community) as nodes and their relationships as edges. For example, Thompson et al. [14] proposed a graph convolutional network (GCN) to detect conversational groups among static agents involved in the same conversation.

In contrast with static groups, moving groups might dynamically change their distance from other groups in the same environment. This adds extra challenges to the group detection task. Kipf et al. [7] proposed a GNN-based method, Neural Relational Inference (NRI), which applied a GNN to infer the interactions between moving particles given their spatio-temporal sequences in a physical system. In this work, the interactions in a physical system are assumed constant among certain pairs of particles over the given time window. In a realistic social group setting, however, individuals often change their interaction partners. This renders the group detection problem a more challenging task compared to the interaction detection tasks considered by Kipf et al. [7]. For example, while the atoms in a molecule constantly interact with particular atoms over time, children playing in a playground might switch their playmates.

The strong performance of the NRI model in recovering the ground-truth interaction graphs makes it a suitable candidate to be further investigated in group detection tasks. The current study extends the original NRI method in two directions to extend its use from interaction detection to the more complex and realistic social group detection task: (1) We propose a GNN architecture for capturing both short and long dependence in the group detection task where the interactions between agents may change over time. For this purpose, the 1D convolutional layer in NRI is replaced with a gated dilated residual causal convolutional (GD-RCC) block, as proposed by Wavenet [11]. (2) The original NRI builds and updates edge features by simply concatenating the node features, which does not satisfy the symmetric property of group relationships. We propose using symmetric temporal edge features and symmetric edge updating to tackle this problem.

Overall, this paper makes the following contributions:

- We propose a framework for group detection building upon the NRI interaction detection method. Our framework can capture short and long dependencies in the spatio-temporal data and can satisfy the symmetric property of group behavior.

- We extend NRI by applying the Louvain community detection algorithm to transform the predicted interactions into predicted groups.
- We evaluate our group detection framework using three group-interaction simulation datasets and three pedestrian datasets and further compare our method against four state-of-the-art methods.
- We investigate the effectiveness of our two proposed changes, namely, the GD-RCC block and symmetric temporal edge feature with symmetric edge updating processes, on the original NRI in an ablation study.

The rest of the paper is organized as follows. In Section 2, we formulate the group detection problem. Section 3 discusses the related works. We present our proposed methodology in Section 4. In Section 5, the experiments are discussed. Finally, Section 6 presents conclusions and future research directions.

2 Problem Formulation

Assume given the spatio-temporal trajectories of N agents in a time window with a duration of T time steps, where the spatio-temporal measurements (e.g., position, speed, acceleration, etc.) of each agent $i \in 1, ..., N$ at a time step $t \in 1, ..., T$ is denoted by X_i^t and the spatio-temporal sequences of all agents are denoted by $X_{1:N}^t$. The goal is to detect groups $C = \{c_j | j = 1, ..., K\}$ of agents, where $K \leq N$ is the number of groups, assuming that the group relationships are constant in a time window, while agents could interact with other agents from a different group. We aim to learn the probability of pairwise interactions $\hat{I}$ between agents within the time window given $X_{1:N}^{1:T}$, i.e., $P(\hat{I}|X)$, such that the predicted pairwise interactions reflect the group memberships of agents in community detection algorithms.

Our proposed method to solve this problem employs a GNN encoder to predict pairwise interactions $\hat{I}$. The Louvain community detection algorithm [3] transforms the predicted pairwise interactions $\hat{I}$ into predicted groups $\hat{C}$. We train the GNN encoder in a supervised way using the ground-truth pairwise group relationships G where $G_{(i,j)} = 1$ denotes that agent i and agent j are in the same group and otherwise $G_{(i,j)} = 0$. In the training phase, the goal is to minimize the difference between G and $\hat{I}$ by minimizing the weighted cross-entropy loss function.

3 Related work

This section discusses the related work in group detection algorithms and further explores studies that proposed GNN models for spatio-temporal data.

Group Detection: Many previous studies in group detection tasks are based on classic machine learning methods with hand-crafted features [16,13]. Yamaguchi et al. [16] proposed an SVM-based framework applying normalized histograms of distances, velocity, and direction features to classify the binary group detection. Using supervised clustering, Solera et al. [13] proposed a structural SVM [15] framework to find groups of pedestrians based on hand-crafted features, e.g., distance, motion causality, trajectory shape, and paths convergence. Despite acceptable results, generating hand-crafted features needs domain knowledge. Besides, these features usually depend on particular data types and applications, e.g., the features created for detecting pedestrians walking on streets may not apply to other complex social settings (e.g., children playing).

To address this problem, many recent studies proposed deep learning-based methods. In GD-GAN [5], an LSTM-based generator predicts future trajectories. In this work, groups are detected by clustering the hidden states of this LSTM-based generator. Contrary to GD-GAN, which predicts future trajectories, our work predicts the pairwise interactions using a GNN encoder. This is beneficial because it can be directly trained with the ground-truth group relationships without special optimization algorithms, such as the Block-coordinate Frank-Wolfe (BCFW) algorithm, in a computationally efficient way. [9].

GNN for spatio-temporal Data: Most GNN-based works for spatio-temporal data, such as TrafficGraphNet [8], focus on improving the performance of forecasting tasks. This approach learns the node representations by aggregating the nodes' neighborhoods and does not directly model the pairwise interactions or group relationships needed for group detection tasks. Methods such as NRI [7] that focus on predicting the edges between nodes can denote the interaction or relation types between nodes. For instance, the encoder part of NRI [7] applies a GNN-encoder to predict the interaction types between particles in a physical system. In our study, we extended the encoder part of NRI, which predicts the interactions between agents for a group detection task.

4 Methodology

In this section, we first present the interaction model implemented using a GNN encoder. Next, the two main proposed features of this model, (i) symmetric edge features and (ii) GD-RCC, are each discussed separately. We employ a GNN encoder, based on NRI [7], and a GD-RDCC block, based on Wavenet [11], to create our proposed model "WavenetNRI".

4.1 GNN Encoder: Interactions Modelling

The core part of the proposed method is a GNN encoder proposed in NRI [7], which predicts the distribution of the interaction and non-interaction edges. In NRI, the initial edge features and edge updating are implemented by concatenating the features of the end nodes as follows:

$$e^t_{(i,j)} = [X^t_i, X^t_j](t \in 1, ..., T), \quad h^1_j = f_v(\sum_{i \neq j} h^1_{(i,j)}), \quad h^2_{(i,j)} = f_e([h^1_i, h^1_j]) \quad (1)$$

Where the spatio-temporal sequence of agent i at period of time $t \in 1, ..., T$ is denoted by ${X^t}_i$. The initial edge feature of the agents i and j at time step t is denoted by $e^t_{(i,j)}$. $[\cdot, \cdot]$ denotes concatenation. h^1_j and f_e denotes node representation of the agent j and edge updating function, respectively. The edge and node updating functions f_e and f_v are multilayer perceptrons (MLPs). NRI further applies a 1D convolutional layer with attentive pooling to transform the edge sequence $e^t_{(i,j)}$ into the vector representations of edges $h^1_{(i,j)} = \sum_t a^t s^t$, where a and s are attention score and edge representation, respectively (details are shown in Figure 1).

There are several limitations in the GNN encoder of the original NRI method: (1) Building and updating edge features, and representations by simply concatenating the node features (shown in Equation 1) cannot explicitly model the spatial differences of agents. Furthermore, the results of this concatenation are not symmetric, which may not satisfy the symmetric nature of group relationships. (2) Using only one convolutional layer may not capture the long-term interactions of the sequences of edge features. To tackle these limitations, we made the following changes to the original NRI:

- We included the spatial differences between agents and temporal increments in the initial temporal edge features $e^t_{(i,j)}$ and updated the edge features by element-wise product of the end nodes' representations. Consequently, the final edge vector representations $h^2_{(i,j)}$ are symmetric and capture both spatial differences between the agents and their movements (explained in Section 4.2.
- We replaced the single 1D convolutional layer in NRI with a GD-RCC block based on Wavenet model [11] to learn the temporary edge features and capture both short and long-term interactions of the edge feature sequences (explained in Section 4.3).

4.2 Symmetric edge features and updating

In our proposed method, the edge features are constructed by concatenating the spatial differences of the node measurements and the temporal increments, which is formulated as follows:

$$e^t_{(i,j)} = [\|X^t_i - X^t_j\|, \Delta X^t_i \odot \Delta X^t_j], \quad t \in 1, ..., T-1, \quad \Delta X^t_i = X^{t+1}_i - X^t_i \quad (2)$$

Where the Euclidean distance between agent i and agent j is denoted by $\|X^t_i - X^t_j\|$ and is used to model the spatial difference between agents and their movements (temporal increments). The element-wise production of the increments of the two agents is denoted by $\Delta X^t_i \odot \Delta X^t_j$. We achieve two benefits with this formulation: (i) the temporal edge $e^t_{(i,j)}$ captures both the spatial difference between agent i and agent j as well as the temporal increments of the agents; (ii) the edge features are symmetric, i.e., $e^t_{(i,j)} = e^t_{(j,i)}$, corresponding to the symmetric properties of the pairwise group relationships.

The edge sequences $e^t_{(i,j)}$ are passed to a GD-RCC block to get the vector representations of edges, denoted by $h^1_{(i,j)}$. For a node j, the vector representation

$h^1_{(i,j)}$ of incoming edges are aggregated and fed to a node updating function f_v to get a higher level node representation h^1_j of the node j, which is the same as the node updating process in NRI as shown in Equation 1.

These node representations are combined by element-wise production and fed to another neural network f_e to get final edge representations $h^2_{(i,j)}$, which represents the logits of categorical distributions of edges, shown in Equation 3. Through this process, the final edge representation $h^2_{(i,j)}$ captures not only the interaction between node i and node j but also the interactions of node i and node j with other nodes [7].

$$h^1_j = f_v(\sum_{i \neq j} h^1_{(i,j)}), \quad h^2_{(i,j)} = f_e([h^1_{(i,j)}, h^1_i \odot h^1_j]) \tag{3}$$

After supervised training, a community detection algorithm is applied to the interaction graphs to find clusters denoting groups.

4.3 GD-RCC block

A GD-RCC block [11] is used to transform the edge sequences $e^t_{(i,j)}$ into the vector representation $h^1_{(i,j)}$. The causal convolution preserves the order of the edge sequences by using features from past time steps. With dilated convolutional kernels, the receptive fields are expanded exponentially by staking convolutional layers [11]. The skip connection, a 1D CNN, solves the gradient vanishing problem when increasing the number of layers [6]. The gating activation function, as formulated in Equation 4, regulates the information flow and performs significantly better than rectified linear activation (ReLU) [11]:

$$e_{l+1} = tanh(W^1_l * e_l) \odot \sigma(W^2_l * e_l) \tag{4}$$

Where l is the layer index. W^1_l and W^2_l are two different learnable 1D-convolution parameters of the layer l; e_l denotes the hidden states of edge features of the layer l. $*$ denotes the convolutional operation. σ and $\odot$ denote the Sigmoid function and element-wise multiplication, respectively.

A 1D convolutional layer with attentive pooling over all time steps is applied afterward to get the vector representations of the edges $h^1_{(i,j)}$. This process is visualized in Figure 1.

During the supervised training phase, the ground-truth pairwise group relationships $G_{(i,j)}$ are used as labels; i.e., $G_{(i,j)} = 1$ denotes agent i and agent j are in the same group while $G_{(i,j)} = 0$ denotes otherwise. Due to the imbalanced distribution of the labels, the weighted cross-entropy $H(\hat{I}, G)$, as described in Equation 5, is used as a loss function in which the rare labels are assigned higher weights:

$$H(\hat{I}, G) = -\sum_{(i,j)} [w_G G_{(i,j)} log(I_{(i,j),2}) + w_{NG}(1 - G_{(i,j)}) log(I_{(i,j),1})], \tag{5}$$

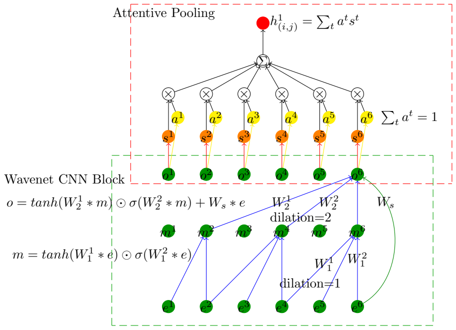

Fig. 1. 1D GD-RCC CNN block (green dashed line block) with Attentive Pooling (red dashed line block) calculated over the sequence of edges e^t. The edge feature sequences $e^{1:T}$ will be fed into a 1D GD-RCC CNN block with skip connections to get hidden states $o^{1:T}$. Here each m^i denotes a node in the first hidden layer. W_l^1 and W_l^2 denote two different learnable convolutional parameters of the layer l (the blue arrows). W_s denotes the 1D CNN skip connection (the green arrow). The hidden states $o^{1:T}$ will be fed into two 1D CNNs f_{pred} (predicts the edge representation s^t (the red arrows)) and f_{score} (predicts the attention score a^t (the yellow arrows)). The vector representations of edges is $h^1_{(i,j)}$.

Where $w_G = \frac{n_G + n_{NG}}{2n_G}$ and $w_{NG} = \frac{n_G + n_{NG}}{2n_{NG}}$ denote the weight of the group label and the weight of the non-group label, respectively. While n_G and n_{NG} are the number of group labels and non-group labels in the training dataset, respectively. By minimizing the weighted cross-entropy, the encoder is optimized to identify the "interaction" versus "no interaction" relation between agents.

5 Experiments

We studied the performance of our method on two types of datasets, i.e., real-world and simulated datasets. Before presenting the results, we first discuss these dataset types, the evaluation metrics, baseline measures, and the experimental setup.

5.1 Dataset

We trained and validated our model on three simulation datasets and three real-world pedestrian datasets. In pedestrian datasets, people walk in different group settings without interacting with other group members. In contrast, in the simulation datasets, cross-group interactions between particles are possible. The real-world datasets have been widely used by other researchers. Due to the lack of interaction between different groups in the pedestrian datasets, the developed methods can be tailored only to improve performance on these datasets and often are not applicable in real-world scenarios. Therefore, we chose to use simulation data, in addition to the pedestrian datasets, to increase the diversity of the datasets by considering the probability of cross-group interactions. This probability is mostly close to zero in pedestrian datasets. This enabled us to simulate more accurately real-world communities, such as schoolyards, where pupils from a particular group might have temporary interactions with peers from different groups over time.

Pedestrian datasets: We selected three public pedestrian datasets, namely *zara01*, *BIWI ETH* and *BIWI Hotel* [12]. We used the sequences of annotated locations of the pedestrians, i.e., the trajectories, as input features to detect pedestrians walking in groups. The duration of measurement, the number of pedestrians, and the number of groups are listed in Table 1 per dataset.

Table 1. The specification of pedestrian datasets

Dataset Name	Duration(s)	Number of Pedestrians	Number of Groups
zara01	360.4	148	45
BIWI ETH	713.4	360	65
BIWI Hotel	722.4	389	41

Group-interaction simulation datasets: To simulate group interactions, we extended the spring simulator introduced by Kipf et al. [7], which simulates the movement of particles randomly connected by a spring in a 2D box. We extended this simulation by defining groups of particles such that particles within a group have a higher probability of having interaction. In our proposed group-interaction simulation, the probability that particle v_i and particle v_j interact with each other given their group relation $G_{(i,j)}$ is formulated as follows:

$$P(I_{(i,j)} = 1 | G_{(i,j)}) = 1 - exp(-a(G_{(i,j)} + b)), (a > 0, b > 0) \tag{6}$$

Where interaction and group relationship between particles v_i and v_j is denoted by $I_{(i,j)}$ and $G_{(i,j)}$, respectively. $G_{(i,j)} = 1$ if v_i and v_j are in the same group otherwise $G_{(i,j)} = 0$. The values of a and b control group interaction and non-group interaction probabilities. Specifically, the value of a controls the overall magnitude of the probabilities, and the value of b impacts the non-group interaction probability. The specification of the three simulation datasets is described in Table 2. Each dataset has 2500 simulations, which include the locations and velocities of the particles over time. The duration of each simulation is 20 seconds, corresponding to 50 time steps.

5.2 Evaluation metrics

We applied Group Mitre $\Delta_{GM}(C, \hat{C})$ [13] to measure the quality of the predicted groups, where C and $\hat{C}$ are disjoint sets denoting the true groups and predicted groups, respectively. The exact procedure for calculating the Group Mitre (precision and recall) is presented in the work of Solera et al. [13], and we omitted the details due to the limit in space.

Table 2. The specification of the group-interaction simulation datasets.

Dataset	Number of Particles	a	b	Probability of Group Interaction	Probability of Non-Group Interaction
Simulation I	5	3	0.02	95.3%	5.8%
Simulation II	10	3	0.02	95.3%	5.8%
Simulation III	10	3	0.05	95.7%	13.9%

5.3 Baselines

We compared the results of our method with the following four baselines:

- **ATTR**[16] is a classification-based method that adopts a linear SVM to classify the binary group relationships based on hand-crafted histograms of distance, direction, and velocity. The regularisation parameter is set to 10.
- **S-SVM**[13] is a clustering-based method that uses a structured SVM to predict the pairwise similarities of the agents and further applies a correlation clustering component to predict the clusters. S-SVM is trained with the BCFW [9] algorithm. The regularisation parameter is set to 10.
- **GD-GAN**[5] is a clustering-based method that adopts an LSTM-based GAN to predict the future trajectory of agents. The DBSCAN algorithm is applied to the hidden states of the LSTM to find the groups. The dimensions of hidden states are set to 256.
- **NRI**[7] is a classification-based method extended by applying the Louvain community detection algorithm to transform the predicted pairwise interactions to the clusters denoting groups. The kernel size of the 1D convolutional layer is set to 5. The node updating and edge updating processes are MLPs with a hidden dimension of 256.

5.4 Experiment Settings

In our experiments, we set the kernel size of the GD-RCC block to five in Equation 4. The hidden dimension size of the node and edge functions in Equation 3 were set to 256. The stochastic gradient descent with a momentum equal to 0.9 was applied for optimization. The code to generate the group-interaction simulation datasets and to implement WavenetNRI is available in the Github repository[6].

5.5 Results

In this section, the results of our experiments are discussed. In each dataset, 60% of the samples were randomly chosen for training; 20% were randomly chosen as validation, and the remaining 20% were testing set. The results of both group-interaction simulation datasets and pedestrian datasets are listed in Table 3.

[6] https://github.com/fatcatZF/WavenetNRI

According to Table 3, NRI and WavenetNRI outperformed all other baselines, and NRI performed slightly better than WavenetNRI on simulation datasets in both recall and precision of group mitre Δ_{GW}. While on pedestrian datasets, GD-GAN [5] outperformed all other methods in both measures. The proposed WavenetNRI could outperform the original NRI [7] and ATTR [16] as the two classification-based baselines.

Concerning the impact of the population size (comparing *Simulation I* and *Simulation II*), we observed that by increasing the number of particles in simulation datasets, both precision and recall were decreased for all methods, except for NRI [7]. The same behavior was observed regarding the probability of non-group interactions (comparing *Simulation I* and *Simulation III*).

Furthermore, we calculated the average pairwise Euclidean distance between the group and non-group members of the two datasets. Our investigation of the differences between these two types of datasets showed that in the pedestrian datasets, the pairwise average Euclidean distances between group members (0.950 meters) were much lower than those from different groups (4.698 meters), i.e., the pedestrians were closer to their group members than other groups. While in the group-interaction simulation datasets, the differences between the Euclidean distances of the same groups (1.039 meters) and that of different groups (1.725 meters) were not significant.

Thus, distinguishing between group members and non-group members is more challenging in the simulation datasets compared with pedestrian datasets. Moreover, the fact that baselines do not generalize to simulation datasets suggests that available research might not be applicable to real-world scenarios where there is a chance for cross-group interactions.

Table 3. Experimental results of recall (R) and precision (P) based on Group Mitre Δ_{GW}. The best average values of recall and precision are highlighted with bold text.

	Simulation I		*Simulation II*		*Simulation III*		*zara01*		*ETH*		*Hotel*	
	R	P	R	P	R	P	R	P	R	P	R	P
ATTR[16]	0.579 ±0.017	0.481 ±0.020	0.512 ±0.009	0.388 ±0.015	0.511 ±0.006	0.386 ±0.005	0.889 ±0.076	0.879 ±0.077	0.745 ±0.067	0.746 ±0.087	0.833 ±0.072	0.841 ±0.068
S-SVM[13]	0.664 ±0.075	0.600 ±0.067	0.529 ±0.039	0.413 ±0.017	0.459 ±0.037	0.382 ±0.030	0.893 ±0.026	0.906 ±0.033	0.887 ±0.027	0.911 ±0.021	0.925 ±0.024	0.927 ±0.030
GD-GAN[5]	0.531 ±0.003	0.430 ±0.004	0.514 ±0.003	0.383 ±0.004	0.512 ±0.003	0.383 ±0.004	**0.949** ±0.046	**0.934** ±0.051	**0.931** ±0.037	**0.950** ±0.028	**0.925** ±0.084	**0.944** ±0.058
NRI[7]	**0.995** ±0.002	**0.994** ±0.003	**0.997** ±0.002	**0.994** ±0.002	**0.998** ±0.001	**0.996** ±0.001	0.801 ±0.096	0.737 ±0.108	0.663 ±0.083	0.669 ±0.080	0.577 ±0.122	0.565 ±0.122
Wavenet-NRI	0.990 ±0.010	0.988 ±0.013	0.985 ±0.005	0.970 ±0.010	0.986 ±0.004	0.972 ±0.007	0.893 ±0.090	0.900 ±0.107	0.793 ±0.078	0.815 ±0.079	0.748 ±0.106	0.790 ±0.086

5.6 Ablation Study

Our proposed approach applied two changes to the original NRI (i.e., adding symmetric edge features and symmetric edge updating process and the GD-RCC block). In this section, we explored the effects of these changes by performing an ablation study. To test the impact of the symmetric edge features and symmetric edge updating process, the same 1D convolutional as the original NRI with the symmetric edge features and the symmetric edge updating process was applied. This model is called "NRI-Symmetric". To test the effects of the GD-RCC block, "Wavenet-GD-RCC" was designed, which used the GD-RCC block with the same edge features and edge updating process as the original NRI. We compared the performance of these two methods with the proposed WavenetNRI and the original NRI on the simulation and pedestrian datasets. The results of both experiments are listed in Table 4. According to the results listed in Table 4, the Wavenet-GD-RCC performed slightly better than NRI, while the performance of NRI-Symmetric was lower than NRI. Therefore, the GD-RCC block could slightly improve the performance of NRI on the group-interaction datasets, and the symmetric edges and symmetric edge updating process negatively affected the original NRI. Additionally, the NRI-Symmetric performed better than the NRI, and Wavenet-GD-RCC performed similarly to NRI on the pedestrian data sets. Therefore, the symmetric edge features with the symmetric edge updating process could improve the performance of NRI on the pedestrian data sets, and the GD-RCC block did not significantly affect NRI's performance. Thus, the results were consistent per dataset type but not overall. We also noticed that either change could add value to one dataset category. As discussed earlier, the complexity of the simulation datasets in the behavior and interactions of the group members and non-group members might explain the inconsistent performance in these two types of datasets.

Table 4. Ablation study results of recall (R) and precision (P) based on Group Mitre Δ_{GW}. The best average values of recall and precision are highlighted with bold text.

	Simulation I		*Simulation II*		*Simulation III*		*zara01*		*ETH*		*Hotel*	
	R	P	R	P	R	P	R	P	R	P	R	P
NRI[7]	0.995 ±0.002	0.994 ±0.003	0.997 ±0.002	0.994 ±0.002	**0.998** ±0.001	0.996 ±0.001	0.801 ±0.096	0.737 ±0.108	0.663 ±0.083	0.669 ±0.080	0.577 ±0.122	0.565 ±0.122
NRI-Symmetric	0.990 ±0.004	0.987 ±0.006	0.981 ±0.007	0.964 ±0.013	0.981 ±0.007	0.961 ±0.009	0.851 ±0.093	0.813 ±0.091	0.679 ±0.094	0.686 ±0.096	0.708 ±0.121	0.739 ±0.115
Wavenet-GD-RCC	**0.998** ±0.002	**0.997** ±0.001	**0.999** ±0.001	**0.997** ±0.002	**0.998** ±0.001	**0.997** ±0.001	0.719 ±0.138	0.625 ±0.165	0.542 ±0.146	0.530 ±0.147	0.566 ±0.169	0.554 ±0.163
Wavenet NRI	0.990 ±0.010	0.988 ±0.013	0.985 ±0.005	0.970 ±0.010	0.986 ±0.004	0.972 ±0.007	**0.893** ±0.090	**0.900** ±0.107	**0.793** ±0.078	**0.815** ±0.079	**0.748** ±0.106	**0.790** ±0.086

6 Discussion and Conclusions

The present study explored the application of GNN by extending the NRI model [7] for group detection in two directions: (1) by applying symmetric edge features with symmetric edge updating processes and (2) by replacing the 1D convolution layer with a GD-RCC block, as proposed by Wavenet [11]. We compared the performance of WavenetNRI with other baselines on the three group-interaction simulation datasets and three pedestrian datasets. NRI and WavenetNRI outperformed all other baselines on the group-interaction simulation datasets. Although the pedestrian datasets were captured in real-world setups, the simulation datasets were better reflecting complex group interactions with larger groups, which stresses the importance of the obtained results. On the pedestrian datasets, although our proposed method did not compete against the clustering-based baselines, i.e., GD-GAN [5] and S-SVM [13], it outperformed classification-based methods, i.e., ATTR [16] and the original NRI [7]. Yet, baseline methods did not generalize very well to the simulation datasets. We further evaluated the effects of our changes to the original NRI in the ablation study. We found that on the group-interaction data sets, the GD-RCC block slightly improved the performance of NRI. Simultaneously, the symmetric edge features with symmetric edge updating processes negatively affected the performance of NRI. On the pedestrian data sets, the symmetric edge features with symmetric edge updating processes improved the performance of NRI, while the GD-RCC block had no significant effect on NRI.

Our analysis demonstrates that WavenetNRI is highly effective at predicting pairwise interactions, which ultimately reflect the group memberships of agents in an interacting environment. One drawback of the proposed method is its dependency on ground truth data. Unsupervised methods such as GD-GAN are preferable if ground truth is not available for a particular study. Many real-world communities, such as sports clubs and schoolyards, can be understood as a dynamic interacting system, where applying a trained WavenetNRI model can be helpful in predicting group memberships within the system.

The current study can be improved by investigating how to adapt the proposed neural network design more efficiently to different datasets using meta-learning. Additionally, it is worth studying how to extend the proposed classification-based method to a supervised clustering task. And finally, designing a fully supervised model by adding a final layer to classify nodes into the group they belong to could be investigated in the future.

References

1. Baratchi, M., Meratnia, N., Havinga, P.J.: On the use of mobility data for discovery and description of social ties. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 1229–1236 (2013)
2. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al.: Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261 (2018)
3. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10), P10008 (2008)
4. Chon, Y., Kim, S., Lee, S., Kim, D., Kim, Y., Cha, H.: Sensing wifi packets in the air: Practicality and implications in urban mobility monitoring. In: Proceedings of

the ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 189–200 (2014)

5. Fernando, T., Denman, S., Sridharan, S., Fookes, C.: Gd-gan: Generative adversarial networks for trajectory prediction and group detection in crowds. In: Proceedings of the Asian Conference on Computer Vision. pp. 314–330. Springer (2018)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
7. Kipf, T., Fetaya, E., Wang, K.C., Welling, M., Zemel, R.: Neural relational inference for interacting systems. In: Proceedings of the International Conference on Machine Learning. pp. 2688–2697. PMLR (2018)
8. Kumar, S., Gu, Y., Hoang, J., Haynes, G.C., Marchetti-Bowick, M.: Interaction-based trajectory prediction over a hybrid traffic graph. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5530–5535. IEEE (2021)
9. Lacoste-Julien, S., Jaggi, M., Schmidt, M., Pletscher, P.: Block-coordinate frank-wolfe optimization for structural svms. In: Proceedings of the International Conference on Machine Learning. pp. 53–61. PMLR (2013)
10. Nasri, M., Tsou, Y.T., Koutamanis, A., Baratchi, M., Giest, S., Reidsma, D., Rieffe, C.: A novel data-driven approach to examine children's movements and social behaviour in schoolyard environments. Children **9**(8), 1177 (2022)
11. Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)
12. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 261–268. IEEE (2009)
13. Solera, F., Calderara, S., Cucchiara, R.: Socially constrained structural learning for groups detection in crowd. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(5), 995–1008 (2015)
14. Thompson, S., Gupta, A., Gupta, A.W., Chen, A., Vázquez, M.: Conversational group detection with graph neural networks. In: Proceedings of the International Conference on Multimodal Interaction. pp. 248–252 (2021)
15. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proceedings of the International Conference on Machine Learning. p. 104 (2004)
16. Yamaguchi, K., Berg, A.C., Ortiz, L.E., Berg, T.L.: Who are you with and where are you going? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1345–1352. IEEE (2011)