Universiteit Leiden

Master Computer Science

Group Detection from Spatiotemporal data using Social Context

Name:	Thomas Maliappis
Student ID:	s3249484
Date:	28/11/2023
Specialisation:	Data Science
1st supervisor:	Dr. Mitra Baratchi
2nd supervisor:	Prof. dr. Carolien J. Rieffe
Daily supervisor:	Maedeh Nasri

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

In this study, we address the group detection problem using spatiotemporal data from human trajectories. We leverage the trajectories of surrounding agents, referred to as 'context', when determining if two agents are part of the same group during multiple consecutive timeframes known as 'scene'. Our approach is built upon the Deep Affinity Network for Clustering Conversational Interactants (DANTE). The main advancement in our method lies in the incorporation of Recurrent Neural Networks (RNN) layers, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), within the neural network architecture. This addition aims to capture the temporal dynamics inherent in the trajectories of agents in the datasets. Our method, the so-called T-DANTE, combines temporal features with the base model. Our ablation studies demonstrate that the utilization of context, combined with the processing of temporal dynamics, yields promising results for the group detection task, across real-world pedestrian datasets and spring simulation datasets. This is evident and validated across these datasets. Moreover, we compared the performance of T-DANTE with NRI, WavenetNRI, GDGAN and the original DANTE baselines. Our method outperformed baselines in terms of Group Correctness metric by at least 17.97% for pedestrian datasets Although some baselines perform better for simulation datasets, the difference is not significant.

Contents

Chapter 1

Introduction

Group detection from spatiotemporal trajectory datasets has wide-ranging applications. Group detection algorithms are crucial in studying human mobility and their activities within communities [4, 7, 18, 19, 20, 21, 32], that could add insights into human behavior in social sciences and psychology [15]. Social patterns can also be detected at schools to increase inclusiveness in social activities and avoid marginalization of children [16, 14]. Moreover, group detection algorithms aid in understanding migration patterns and animal group behavior [12, 24], forecasting natural phenomena, such as predicting landfalls by grouping with old occurrences to find similar behavior [12], and developing carpool sharing platforms for efficient transportation systems [24].

The conventional landscape of group detection research has predominantly focused on traditional machine learning methodologies involving feature engineering [19, 31]. These approaches typically require manual extraction and selection of features to train models for identifying groups among individuals, a process that can be time-consuming and potentially introduce bias. Recent advancements in the field are transitioning the group detection task in spatiotemporal data into detecting communities in a graph representation of movement trajectories. This is achieved by constructing a social graph representing trajectories and applying community detection methods to capture agent groups. Sen et al. [18] employ an SVM classifier to create a social graph based on custom agent similarity features, subsequently applying a clustering algorithm to get the underlying groups in the graph. Although the results were promising, the performance is limited to the selected features.

Recently, deep neural networks gained traction for modeling interactions within spatiotemporal data [11, 15], as they are capable of detecting complex nonlinear relationships between variables, finding possible interactions between predictor variables, and being trained using different algorithmic methods. GD-GAN [7], NRI [11] and WavenetNRI [15] are all deep neural network-based approaches aiming to decode the spatiotemporal patterns and to identify group behavior among agents. The preliminary limitation of this line of research is the utilization of the entire spatial data per time frame, which means that all

agents impact each other even if the distance between them is too large.

Another attempt to detect group behaviors in the spatiotemporal data has been made by Swofford et al. [21]. They introduced DANTE which tackles conversational group detection by incorporating context features, that represent spatial surroundings, into the input of their proposed model to learn a graph representation for a single-frame scene by a neural network. This method leverages the importance of considering someone's surroundings when estimating conversational group membership. The limitation of this study is the reliance of the neural network architecture only on multilayer perceptron (MLP) design and the consideration of only a single frame to detect the groups. Specifically, MLPs have several disadvantages such as being computationally intensive and not being well-suited for sequential data such as time series, and spatiotemporal data.

Although several research studies have been carried out on detecting groups in spatiotemporal data, they often employ manually constructed features or neural networks that overlook temporal aspects of spatiotemporal data. To address this gap, the present study introduces an approach, building upon DANTE [21], that represents agents and their spatiotemporal data as a social graph using a deep neural network. Our proposed model incorporates layers encountered in Recurrent Neural Networks (RNN) to account for the temporal aspect of agent movements, which was not included in the original model and has shown positive effects in predicting vehicle trajectory in previous studies [6]. Moreover, our approach builds on the concept of context information that showed promising results in identifying group behavior [21], and is further refined in our work by including scenes with multiple timeframes. One challenge of using this approach is the need to preprocess the datasets and format them accordingly to become suitable for being the input of our network. Another challenge of the approach is the need to use an input layer that can process timeseries data. In most previous studies, the context information includes the data of all available agents, making it more complex to estimate the affinities of agents. However, our proposed model uses a specific number of agents for the context information based on the characteristics of the dataset, such as the average number of agents per timeframe. Subsequently, a community detection algorithm is applied to identify groups among agents. Moreover, our experimental investigations explore the impact of different hypeprparameters such as including different numbers of agents in context information (i.e., context size) and different types of layers in our model across multiple real-world and simulation datasets. The main contributions of this work are:

- Introducing a novel framework that extends DANTE by including the ability to process input data of more than a single timeframe and employing RNN layers such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) to capture temporal dependencies.

- Conducting extensive ablation studies to investigate the impacts of context size and RNN layers on the performance of our work.

- Introducing a novel simulation dataset to model group behavior by including the concept of attraction points in the original spring simulation datasets [11, 15]. The inclusion of attraction points is a step towards making simulation datasets closer to reality. These points could represent a spot in a playground that children play around it or a pond in the desert when analysing animal movements. The use of them could help better manipulate the trajectories created by moving towards these specified or random points.

- Evaluating our proposed model using five pedestrian datasets and six simulation datasets against baselines. The pedestrian datasets differ in terms of number of agents, duration and number of groups included. The simulation datasets have the number of particles and number of groups parameters that leads to creating diverse datasets. The baselines include NRI [11], WavenetNRI [15], GDGAN [7], and the original DANTE [21] using Group Correctness and Group Mitre as the evaluation metrics.

The present study is organized as follows. In the introductory Section 2, the problem formulation is introduced, delineating the parameters and goal associated with the group detection problem. The subsequent Section 3 delves into a thorough review of related work in the field, offering a comprehensive background and contextualization for the proposed approach. The methodology Section 4 elucidates the specifics of the proposed approach. It provides insights into the neural network architecture and the subsequent application of a community detection algorithm to identify groups within the constructed social graphs. Moving forward, the experiments Section 5 details the experimental setup, encompassing information about the datasets, the evaluation metrics, and the selected baselines for comparison. The results Section 6 presents the findings obtained from the experiments, offering a detailed analysis of the outcomes and the performance of the proposed approach. The concluding Section 7 summarizes the entire study, encapsulating key findings and insights. It also discusses potential avenues for future research and the broader implications of the proposed approach.

Chapter 2

Problem Formulation

In order to define the group detection problem in spatiotemporal data, it is necessary to first define the notations representing the data. Given N agents and a scene of T consecutive time steps, X_i^t is the location and velocity of agent $i \in 1, ..., N$ in time step $t \in 1, ..., T$. The trajectory of agent i can be represented by $X_i^{1:T}$, which includes data about the location and velocity of agent i during the scene T. We are interested in detecting groups $C = \{c_j | j \in [1, K]\}$ in which each agent belongs, where $1 \leq K \leq N$ is the number of groups. Agents being in the same group means that they are sharing similar spatial behavior over a scene of T time steps. The assumption is that the group relationships do not change during a scene. Duration T of the scene is fixed. The problem of detecting groups in the spatiotemporal data can now be formulated as obtaining a representation of graph $G = (V, E)$ in each scene of T time steps, by learning the pairwise affinities between the agents. $V = \{v_1, v_2, ..., v_N\}$ being the set of nodes corresponding to the agents and E the set of edges, where $e_{ij} = (v_i, v_j) \in E$ if the pairwise affinity a_{ij} of agents i and j over the scene T is higher than a parameter thr. The graph representation G is the $n \times n$ adjacency matrix $A = (a_{ij})_{n \times n}$ where $a_{ij} = 1$ if an edge connects nodes i and j.

Afterward, the adjacency matrix A of the graph representation will be used as the input of a graph community detection algorithm to detect the communities C in the graph. These discovered communities C represent the groups of agents that have similar spatiotemporal behavior during the scene T.

Our proposed method for solving the problem is based on a deep neural network. Specifically, our proposed model approximates the pairwise affinities between agents in each scene and produces the corresponding adjacency matrix. The social graph represented by each adjacency matrix then will be given to the Dominant Sets (DS) [9] community detection algorithm to discover communities in the graph.

Chapter 3

Related work

To date, various group detection methods have been proposed. For convenience, we categorized methods into (i) classical and (ii) deep neural network approaches. For the latter category, the concepts of graph neural networks, recurrent neural networks, and context information are being introduced, which includes frameworks that combine one or more of them.

3.1 Classical Approaches

The first attempts at solving the group detection task have mainly focused on traditional machine learning methodologies. Yamaguchi et al. [31] approached the task as a binary classification problem over pairwise trajectory features and used an SVM classifier to estimate if the two agents with their corresponding trajectories are in the same group. Solera et al. [19] built a Structural SVM-based learning framework that uses proxemics, which is the study of how space is used in human interactions, and causality-related features to solve the group detection task. Such approaches, however, take a significant amount of time to select and extract the required features. Moreover, these models potentially introduce bias by ignoring some crucial aspects of the spatiotemporal data.

3.2 Deep Neural Network Approaches

Group detection task could not have escaped by the rise of deep neural networks. Most of the recent strides in the field have integrated a deep neural network into their frameworks as they seem to be more capable of capturing the complex dependencies between the data than models using manual feature extraction [1, 3, 22].

3.2.1 Graph Neural Networks

In the landscape of deep neural networks, Graph Neural Networks (GNNs) emerge as a distinct and powerful paradigm, particularly tailored for data with underlying graph structures. These networks feature specialized layers designed to effectively capture complex relationships and dependencies within graph-structured data, making them well-suited for tasks involving interconnected entities [29, 30, 34].

The GNNs can be used in different applications such as group detection tasks using spatiotemporal data. Conversational group detection is a special version of group detection. In the realm of conversational group detection, the emphasis lies in identifying F-formations, which represent the spatial arrangements of individuals during group conversations [9, 27]. To tackle this challenge, Thompson et al. [25] proposed a novel framework based on a message-passing GNN, offering a unique perspective on understanding conversational dynamics. In this study, the temporal dependencies in the spatiotemporal data are entirely overlooked, whereas the inclusion of specific layers, for example, Recurrent Neural Networks can integrate the temporal dynamics.

3.2.2 Recurrent Neural Networks

RNNs represent a specialized class of deep neural networks uniquely designed to model sequential data effectively. These networks feature specific layers that excel in capturing intricate dependencies within time series data, making them particularly well-suited for tasks involving sequential information [5, 8].

Neural Relational Inference (NRI) [11] is one of the RNN-based approaches that have introduced a paradigm shift by incorporating deep neural networks to model intricate interactions between individuals. This work takes advantage of both GNNs and RNNs to build an auto-encoder model to learn the latent vectors that represent the interaction graph. Building upon this foundation, Nasri et al. [15] introduced WavenetNRI, a model that integrates a gated Residual Dilated Causal Convolutional Block [26] in order to capture both short and long-term interactions in the sequences of edge features. This approach utilizes learned interactions to effectively extract and discern groups formed by interacting individuals, showcasing the evolving complexity in group detection methodologies. The main disadvantage of these approaches is the complete reliance on the model to understand which agents affect the trajectories of others. In contrast, our work only maintains the surrounding agents as part of the same group, thus not all agents are in our affinity learning process. In this way, the model focuses on the interactions between agents that are close to each other, while excluding insignificant agents located at a distance from agents of interest. By excluding agents, we also aim to reduce the computational cost of our method.

3.2.3 Context information

The integration of contextual information has emerged as a pivotal aspect of group detection. Deep Affinity Network (DANTE) of Swofford et al. [21] is a notable example that utilizes a specified number of surrounding agents as context for clustering social interactants. This approach is limited due to the use of only one timeframe; thus, it does not exploit the temporal aspects of the problem. Similarly, Tan et al. [23] employ an approach where all agents in a given scene contribute to contextual information, feeding a neural network that predicts affinities between agents. These methodologies underscore the importance of considering broader contextual cues for accurate group detection in dynamic environments.

The significance of context information is further exemplified in vehicle trajectory prediction. Neural networks have been employed to forecast the behavior of vehicles based on the movements of surrounding vehicles. For instance, LSTM encoder-decoder model of Deo et al. [6], incorporating convolutional social pooling, showcases an innovative approach to predict the motion of surrounding vehicles for autonomous vehicles. Similarly, our proposed model integrates context information and temporal dynamics, but in a different application to solve the group detection problem.

In a nutshell, the present study is inspired by multiple ideas from the aforementioned approaches to solve the group detection problem using spatiotemporal data. The temporal dynamics of the data are captured by incorporating RNN layers in our model. The novelty of our approach is the combination of the temporal aspects of the data with context information when processing each pair of agents in a scene. More details about the methodology will be presented in the following section.

Chapter 4

Methodology

This section introduces the architecture of our method. Firstly, we explain our framework to learn the affinities between agents in a scene. Since our network is based on DANTE [21] combined with temporal features of spatiotemporal data (i.e., including RNN layers), we name our model T-DANTE. Secondly, we explain the Dominant Sets community detection algorithm which is used to obtain the groups from the affinity graph. Figures 4.1 and 4.4 provide a visual representation of our framework, respectively.

4.1 Affinity learning network

This section explains the deep neural network proposed to estimate an affinity graph representing the trajectories of agents during a scene. This neural network is called T-DANTE and is estimating the affinities between the agents which represent the edges in the affinity graph (visualisation in Figure 4.1). The architecture of T-DANTE tries to take advantage of types of information: (1) the information related only to the pair of agents that we are interested to check their affinity in the graph and (2) the information acquired from the surrounding agents, which are the context, of the agents, who form the pair of interest. Our T-DANTE advances this idea by using RNN layers (i.e., LSTM and GRU) to include temporal data, in addition to the spatial features, and decide the affinity score between two agents accordingly.

During the training of T-DANTE, the pairwise group relationships will be used as ground truth and the difference between A and $\hat{A}$ will be minimized using the log loss function. Equation 4.1 shows how the log loss $L(y, \hat{y})$ is calculated.

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{k=1}^{N}\left(y_k \cdot \log(\hat{y}_k) + (1 - y_k) \cdot \log(1 - \hat{y}_k)\right) \tag{4.1}$$

where N is the number of samples. y_k is the true label for the k-th sample (either 0 or 1). $\hat{k}_i$ is the predicted probability that the k-th sample belongs to

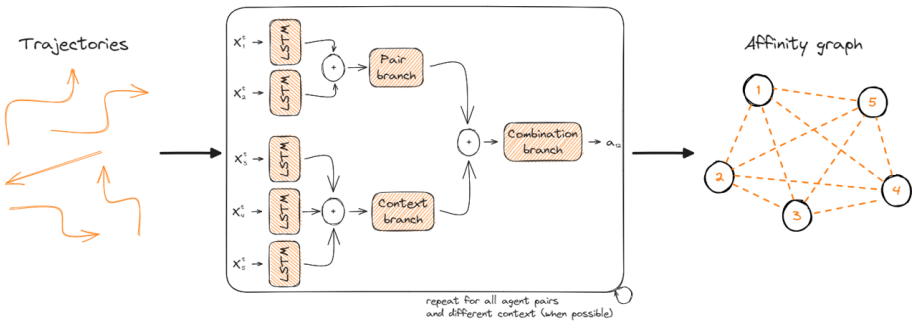


Figure 4.1: The locations and velocities of the agents are the input of our deep neural network. The pairwise affinities are learned in order to create the affinity matrix A which is the representation of graph G used in the community detection algorithm. In this figure, the value of a_{12} is approximated by processing the features of Agents 1 and 2 by the *Pair* Branch and the features of the other agents by the *Context* Branch of the network. The *Combination* Branch is the last part of the architecture responsible for the final output.

the positive class (between 0 and 1). Each y_k sample refers to the affinity a_{ij} of two Agents i and j. If $a_{ij} = 1$ means Agent i and Agent j are in the same group while $a_{ij} = 0$ means Agent i and Agent j are in different groups. For the rest of this work, we make the assumption that T-DANTE computes the affinity a_{ij} for the Agents i and j for graph G.

4.1.1 Pair Branch

The first part of T-DANTE is called Pair Branch and uses the data of Agents i and j to compute the local features that represent the interactions between these agents. An example visualisation can be found in Figure 4.2, where Agent $i = 1$ and Agent $j = 2$. A two row matrix, one for each Agent is used as input of this branch. The data of each Agent is separately passed to a RNN layer (i.e., LSTM or GRU layer) depending on the variation of T-DANTE, After the extracted features for the two Agent are combined to be processed together.

LSTM and GRU layers are types of RNN architectures, designed to capture and utilize temporal information in sequential data [8, 5]. The LSTM features memory cells and intricate gating mechanisms, including input, forget, and output gates, which allow them to selectively store and retrieve information over extended sequences. This capability is particularly beneficial for tasks where modeling long-term dependencies is crucial. Equation 4.2 formulates the LSTM

layer as follows:

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
\tag{4.2}
$$

Where i_t is the input gate activation vector. f_t is the forget gate activation vector. g_t is the candidate cell state (proposed change). o_t is the output gate activation vector. c_t is the updated cell state. h_t is the output hidden state. σ is the sigmoid activation function. tanh is the hyperbolic tangent activation function. $\odot$ is the element-wise multiplication. W_{ij} is the weight matrix for the input (i), forget (f), candidate (g), and output (o) gate. x_t is the input at time t. h_{t-1} is the hidden state at time $t-1$ and b_{ij} is the bias term for the input (i), forget (f), candidate (g), and output (o) gate.

On the other hand, GRU layers employ simpler update and reset gates, offering computational efficiency with fewer parameters. GRUs excel in tasks where capturing shorter-term dependencies and efficiently processing sequential information are essential. The GRU layer is formulated in Equation 4.2 as follows:

$$
\begin{aligned}
z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\
\tilde{h}_t &= \tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t
\end{aligned}
\tag{4.3}
$$

where z_t is the update gate activation vector. r_t is the Reset gate activation vector. $\tilde{h}_t$ is the candidate hidden state. h_t is the updated hidden state. σ is the sigmoid activation function. tanh is the hyperbolic tangent activation function. $\odot$ is the element-wise multiplication. W_z, W_r, and W_h are the weight matrices for the input (x_t) in the update, reset, and candidate hidden state equations, respectively. U_z, U_r, U_h are the weight matrices for the hidden state (h_{t-1}) in the update, reset, and candidate hidden state equations, respectively. b_z, b_r, and b_h are the bias terms for the update, reset, and candidate hidden state equations, respectively. x_t is the input at time t. h_{t-1} is the hidden state at time $t-1$.

Both LSTM and GRU layers have proven effective in a variety of applications, providing practitioners with versatile tools to address temporal dependencies in diverse datasets [8, 5]. The concatenation of the RNN layer (either LSTM or GRU) outputs is then managed by multiple x blocks of a series of a Convolutional or Dense layer for T-DANTE and T-DANTE GD, respectively, followed by a Dropout layer and a Batch Normalisation layer. The Dropout layer is responsible for reducing overfitting to the training dataset and improving the generalization of the final model. The Batch Normalisation layer is

used to avoid the covariate shift that occurs when the distribution of input features is changing during training. The Convolutional layers use ReLU activation functions as it is known to prevent the exponential growth in the computation required to operate the neural network.

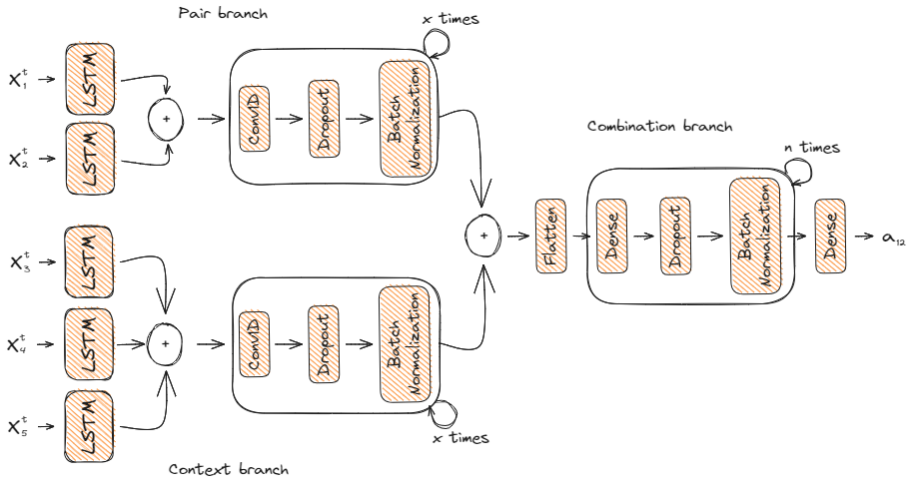


Figure 4.2: Visualisation of the architecture of T-DANTE using LSTM and Conv1D layers. This architecture is mentioned as T-DANTE throughout the thesis.

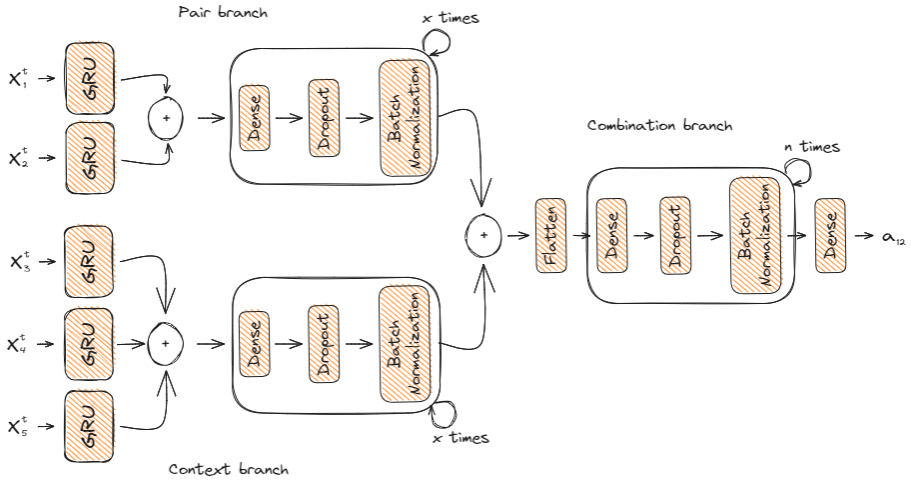


Figure 4.3: Visualisation of the architecture of T-DANTE using GRU and Dense layers. This architecture is mentioned as T-DANTE GD throughout the thesis.

4.1.2 Context Branch

The Context Branch of T-Dante computes the global feature representations of the social context of the pair of interests as depicted in Figure 4.2. The number of agents considered as social context is a hyperparameter of the model, as an example the context size is 3 agents in Figure 4.2. Similar to the Pair Branch, the Context Branch first applies RNN layers to the features of each agent in the context. The same sequence of the Convolutional/Dense layer, Dropout layer, and Batch Normalisation layer is repeated x times with different numbers of filters based on the prescribed configuration.

4.1.3 Combination branch

The Pair Branch and Context Branch are followed by a Concatenate layer in order to combine their acquired information. This combined branch is called the Combination Branch as depicted in Figure 4.2. The tensors are flattened and used by a sequence of a Dense layer, a Dropout layer, and a Batch Normalisation layer n times with various filters. The number of layers and the filter size depend on the complexity of the data. Attributes representing the complexity of the data could be the number of frames per sample, the batch size, and the amount of data. This sequence also uses ReLU activation functions. The last layer of the Combination Branch and of the whole network is a Dense layer using a Sigmoid activation function to constrain the output to the $[0, 1]$ range. This is the affinity score for the pair of agents of interest given the specified context.

4.2 Graph community detection

Once all the affinity values between pairs of individuals are computed within the social affinity graph, the subsequent step involves unraveling the inherent group structures embedded in the data. To achieve this, we turn to the DS algorithm, a powerful tool introduced by Hung et al. [9] specifically designed for the analysis of edge-weighted graphs. In the context of our study, the social affinity graph G serves as the canvas upon which we seek to reveal cohesive groups, and the DS algorithm plays a pivotal role in this endeavor.

The DS algorithm [9], as an extension of maximal cliques to edge-weighted graphs, facilitates this identification by sticking to specific criteria of mutual affinity. The DS algorithm not only identifies clusters based on high relative mutual affinity but also continues to search iteratively for new clusters that satisfy this criterion. This cluster identification process is not boundless; it concludes under two conditions. Either a newly considered cluster fails to meet the requirement of high relative mutual group affinity, or the mutual affinity within a group drops below a certain threshold. The DS algorithm is proficient at producing compact clusters, effectively representing F-Formations of varying sizes. While it can accommodate social affinity graphs with asymmetric affinities, empirical evidence suggests that symmetric affinities tend to yield more robust outcomes [9, 27]. Therefore, in our study, we adopt the assumption of

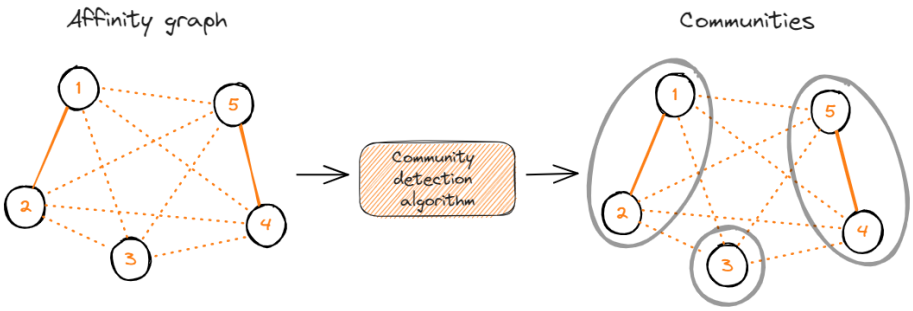


Figure 4.4: The obtained graph representation is used as the input of the community detection algorithm that discovers the communities between the agents.

symmetric affinities, achieved by setting edge weights to the average of predicted values a_{ij} and a_{ji} for $1 \leq i, j \leq N$ and $i \neq j$. This choice further ensures a cohesive and insightful community detection process within our spatiotemporal group detection framework.

Chapter 5

Experiments

In this section, we conducted several experiments in order to evaluate the performance of our model. The pedestrian and spring simulation datasets used in the experiments are explored, and the steps of preprocessing datasets to become suitable for our model are explained. Furthermore, the evaluation metrics and baselines will be outlined before presenting the results. The experiments will help us answer the following questions:

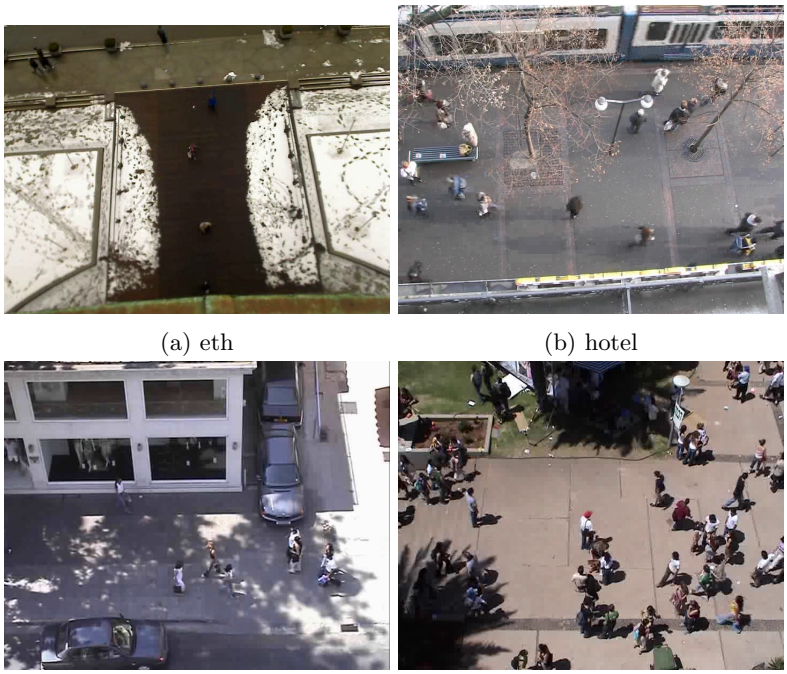
- Can the addition of the temporal aspect of data to the input of the neural network lead to better results on the group detection problem?
- Does the use of context information for each sample processed by the neural network help to gain better performance?
- Does the number of agents included in the context information of each sample affect the performance of the model?
- How does our method perform compared to other methods?

5.1 Datasets

5.1.1 Pedestrian datasets

Two types of pedestrian datasets are used in our experiments. The first is obtained from a study conducted by Pellegrini et al. [17] which introduced two datasets, namely *eth* and *hotel*. The second study conducted by Lerner et al. [13], which includes three datasets, namely *zara01*, *zara02* and *students03*. These datasets can be found in OpenTraj repository [1] [2] and are commonly used as benchmarks for group detection tasks on spatiotemporal data. The data of the aforementioned experiments consist of the location and the velocity of each agent for multiple timeframes. The ground truth of the agent groups is also included in these datasets.

[1] https://github.com/crowdbotp/OpenTraj



(a) eth (b) hotel

(c) zara01 and zara02 (d) students03

Figure 5.1: Reference photos of the pedestrian datasets.

Table 5.1 shows the differences between the pedestrian datasets concerning the duration of measurements in seconds, the number of agents, and the number of groups to which they belong. According to this Table, the *students*03 dataset contains the highest number of agents and groups with the shortest duration of measurement. Another equally important feature is that *eth* and *hotel* datasets have very similar values for all variables, probably due to being products of the same study [17]. Moreover, *zara*01 and *zara*02 datasets are both captured in the same location with a medium-range duration of measurement compared to the other datasets. In Figure 5.1, the reference photos for all the datasets are depicted, with *zara*01 and *zara*02 sharing the same photo as they were captured from the same position.

The *students*03 dataset has a higher number of groups than any other dataset with a group size of 2-4 members, as depicted in Table 5.1. Most of the datasets include small group sizes with 2 or 3 members. The only dataset that has large group sizes is the *eth* dataset with groups of 4 and 6 members.

Another aspect that was interesting to explore was the number of agents in scenes per dataset. As depicted in Figure 5.3, the *students*03 dataset has a higher number of agents in scenes, which makes it suitable for experiments with high context size. On the other hand, *eth*, *hotel*, *zara*01, and *zara*02 datasets have a lower number of agents in most scenes. Therefore, the characteristics of

Dataset	Duration (s)	Agents #	Groups #
eth	773.4	360	58
hotel	722.4	390	41
zara01	360.4	148	45
zara02	420.4	204	58
students03	215.6	428	101

Table 5.1: Information about the name, duration, number of agents, and number of groups in each pedestrian dataset.

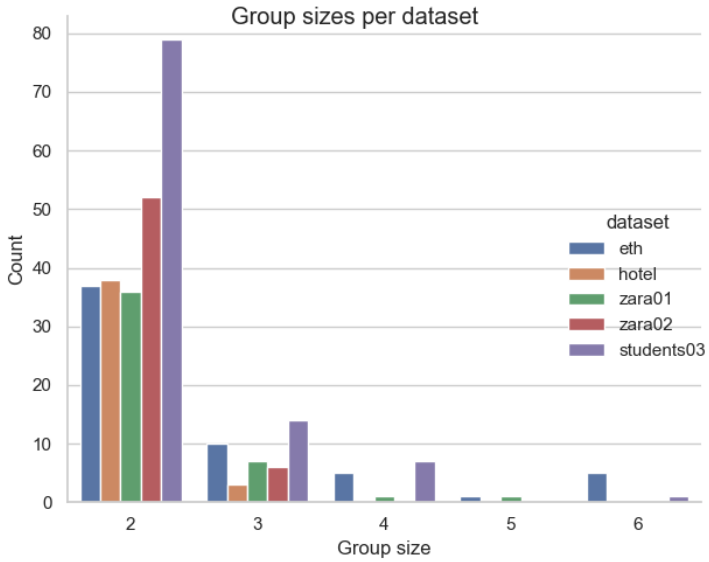
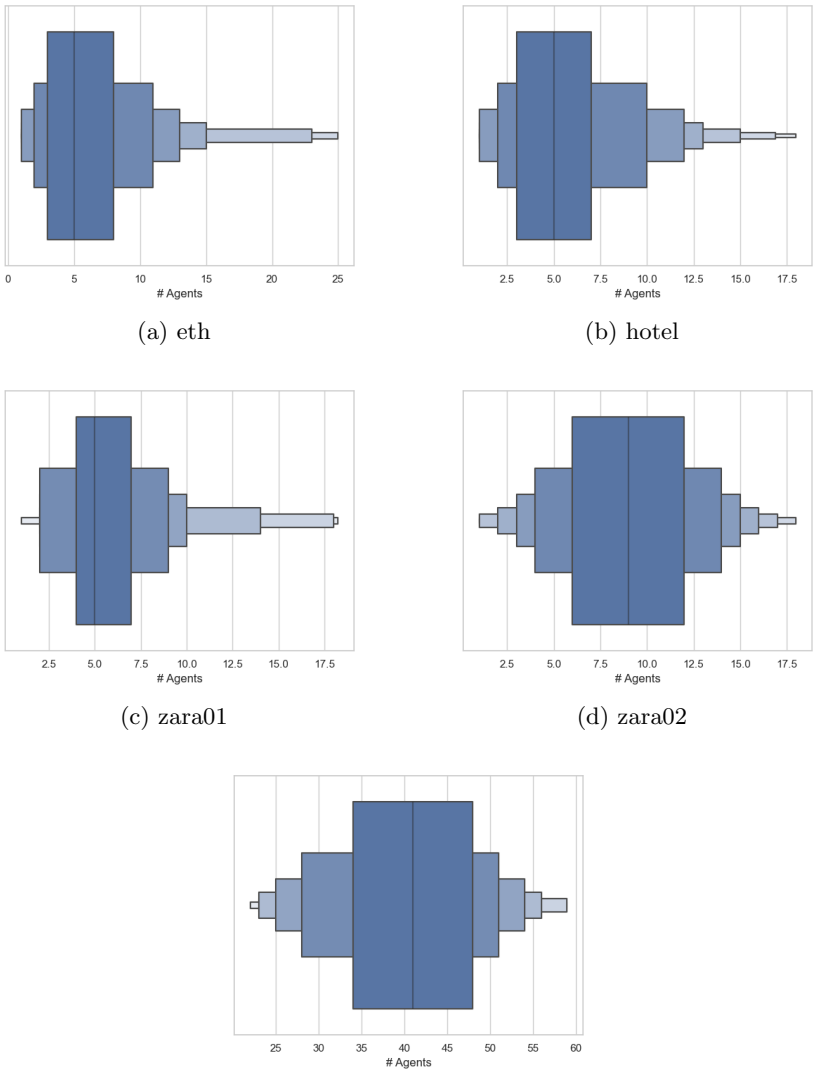


Figure 5.2: Bar plot of group sizes per pedestrian dataset.



(a) eth

(b) hotel

(c) zara01

(d) zara02

(e) students03

Figure 5.3: Distribution of number of agents in different scenes for each pedestrian dataset. The boxplot summarizes key statistical measures, such as quartiles and median for number of agents variable of scenes for each pedestrian dataset.

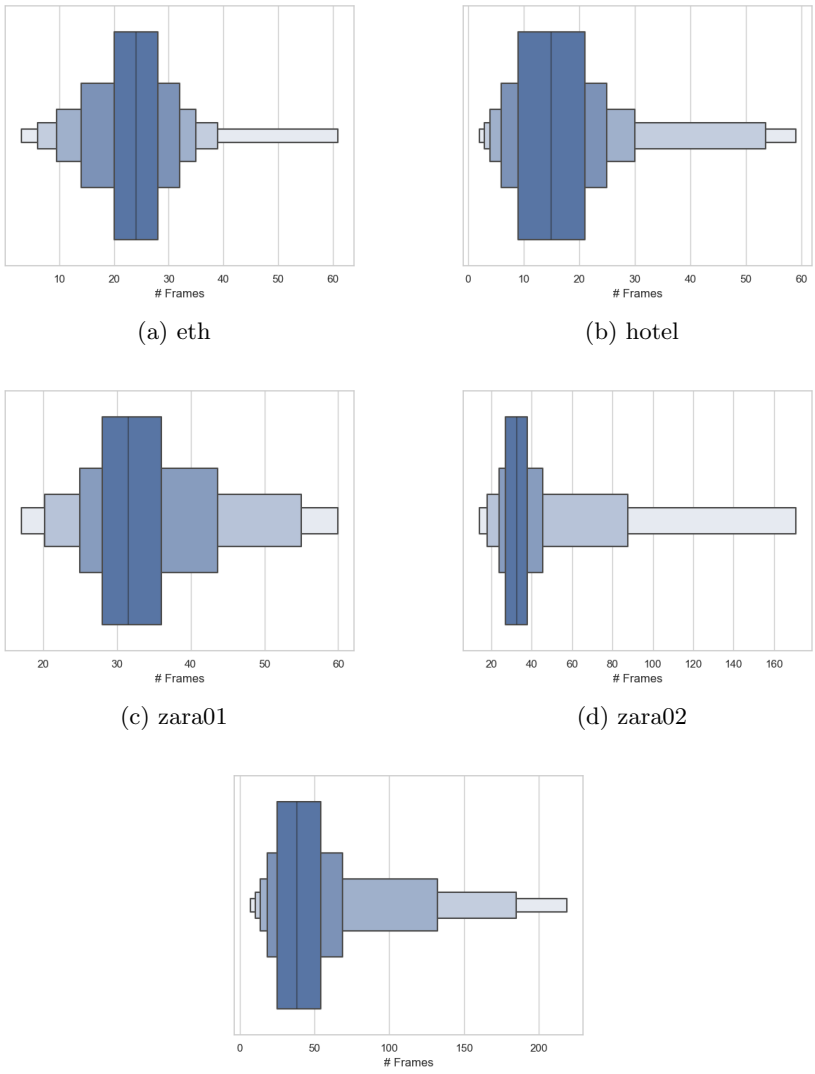
the pedestrian datasets lead us to explore context sizes of less than 10 agents, later in our experiments. This decision was made in order to restrict the number of samples with fake data, as *eth*, *hotel*, *zara*01, and *zara*02 datasets have a high number of scenes with only 5 agents present.

Lastly, we explored the distribution of the number of timeframes in which an agent appears in a scene. As shown in Figure 5.4, the *students*03 dataset has the highest number of timeframes in which agents appear with a mean value of slightly less than 50 timeframes. Both *eth* and *hotel* datasets, collected during the same study [17], have a very similar distribution with a mean value of approximately 25 and 15 timeframes, respectively. The *zara*01 and *zara*02 datasets have completely different distributions. The *zara*01 dataset presents more similarities with *eth* and *hotel* datasets. In contrary, the *zara*02 dataset, despite the significantly lower mean of around 30 timeframes, shares more similarities with *student*03 datasets. Since many datasets contain agents appearing only in few timeframes, we chose the scene size of 15 timeframes to avoid excluding essential information from our datasets.

5.1.2 Simulation dataset

In addition to pedestrian datasets, spring simulation data was used in our experiments. The advantage of simulation data is the availability of ground truth and the possibility of generating an infinite amount of data in order to train our model. The original spring simulation dataset was proposed by Kipf et al. [11] and further enriched by Nasri et al. [15] with particle group information. The basic idea is that a number of particles move in a 2-D space, simulating the concept of particles moving along with each other and affecting the trajectory of each other. The locations and velocities of the particles are part of the generated data as well as the group membership information. The particles were distributed randomly in different groups (the maximum number of groups is the number of particles), while the particles in the same group attract each other and repel particles from other groups.

In order to control the size of groups, the number of groups as well as the number of particles, was added as a parameter of the simulations. The method of initializing groups was modified accordingly to include these parameters. Another feature that is added to the spring simulation experiment is the concept of attraction points, which are locations in the simulation that groups could be attracted to.The attraction points could represent a spot in a playground that children play around or a pond in the desert which attracts animal movements. The attraction points are implemented by defining a force that points each particle toward an attraction point. All the forces has the same strength value, but the direction of them is based on the location of the particle compared to the location attraction point. The number of attraction points ap is another parameter of the simulation. At the start of each simulation, a list of attraction points AP_g, $0 < |AP_g| < ap$ is assigned to each group g. This list includes the location of attraction points for the particles in a group that are going to be attracted along their trajectories. Figure 5.5 shows some examples of simulations



(a) eth

(b) hotel

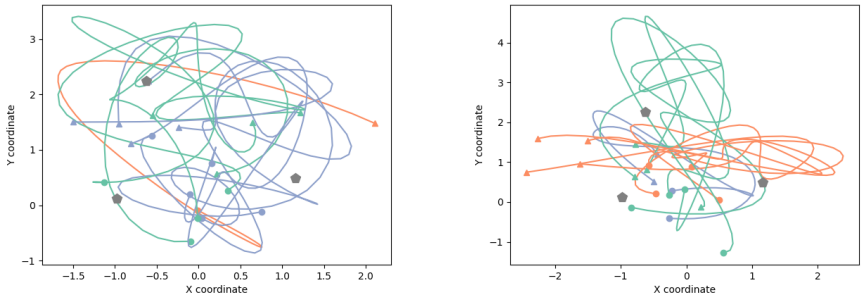
(c) zara01

(d) zara02

(e) students03

Figure 5.4: Distribution of the number of timeframes in which an agent appears in a scene for each pedestrian dataset. The boxplot summarizes key statistical measures, such as quartiles and median for number of timeframes variable of each agent for each pedestrian dataset.

generated with the addition of attraction points.



(a) (b)

Figure 5.5: Visualisations of spring simulations with attraction points. Circle markers are the starting points of the trajectory of a particle and triangle markers are the final points of the trajectory of a particle. Each group of particles share the same color and the pentagon markers represent the attraction points. In Figure 5.5a the blue group is first attracted by the right attraction point and at the end attracted by the top attraction point. Another example is Figure 5.5b where the orange group is first attracted by the right attraction point and at the end attracted by the top attraction point. These are not completely clear in the images due to the attraction and repulsion forces that also occur.

More spring simulation visualization examples can be found in the Appendix 9.1. Table 5.2 presents the parameters used to generate the six spring simulation datasets in our experiments. All the simulation datasets include 3 attraction points. The major difference between the simulation datasets compared to the pedestrian datasets is the arbitrary choice of sample size. In our experiment, each simulation dataset consists of 1000 samples with a specified number of particles and number of groups. For example sim_1 of Table 5.2, includes 1000 samples, that each one includes 8 particles split into 2 groups. Each sample has duration of 50 timeframes, which is also the length of the particles trajectories as we have data for all particles for the whole duration of the sample. This length was selected to experiment with richer data than the pedestrian datasets, where agents are present in more consecutive timeframes. Another feature of spring simulation datasets is the existence of groups with various sizes, whereas the pedestrian datasets mostly include smaller group sizes.

A more detailed representation of the group sizes of each simulation dataset can be seen in Figure 5.6. As shown in this figure, compared to the distribution shown in Figure 5.2, there is a higher number of groups with size over 3. Especially sim_2 and sim_4 simulation datasets have a small number of groups with sizes 1 and 2, as most of their groups contain sizes of 5, 6, and 7 particles. On the other side sim_1, sim_3, sim_5, and sim_6 mostly consist of groups with smaller sizes, but over 3 particles.

dataset	particles #	groups #	average group size
sim_1	8	2	4.01
sim_2	9	2	4.51
sim_3	9	3	3.10
sim_4	10	2	5.00
sim_5	10	3	3.40
sim_6	10	4	2.65

Table 5.2: The six simulation datasets generated and used in our experiments. From left to right, the columns denote the name of the dataset, the number of particles included in the dataset, the number of groups that the particles were split in, and the average group size in the dataset, respectively.

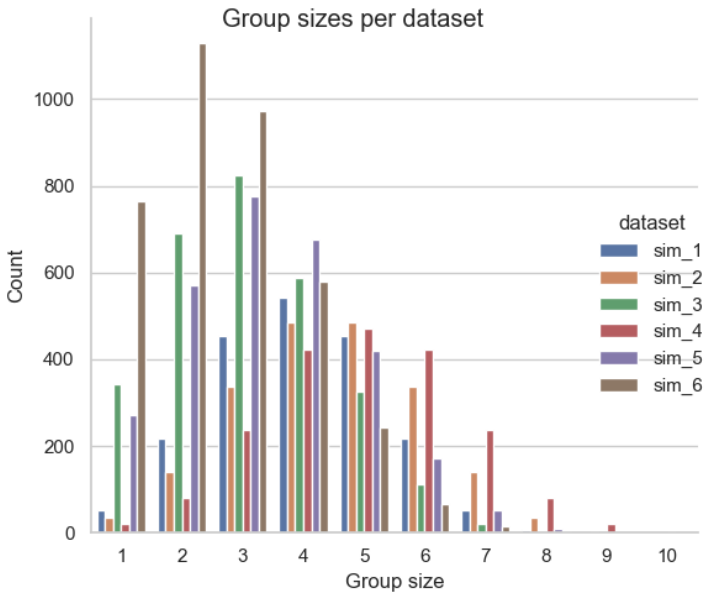


Figure 5.6: Bar plot of group sizes per simulation dataset.

5.1.3 Datasets preprocessing

Both pedestrian and simulation datasets need to be processed in order to be used by the input structure of T-DANTE. The baselines also use the same data, but possibly in different formats to fit the input of each model. The acceptable input structure for T-DANTE samples is an array of agents data $\{X_i^{1:T} | i \in [0, k]\}$, where $k \geq 2$ is the number of agents and T is the length of their trajectories. Each sample has at least the pair of agents that the affinity is estimated plus the 'context' agents, which could also be zero.

At the beginning of the process, we need to find all the possible scenes of size T (T is the number of timeframes that a scene consists of). In order for a number of timeframes to form a scene, they have to be consecutive. The constraint for a scene to be part of the dataset is to include at least 2 agents. During the process of finding the scenes, using these two constraints, we keep track of the groups and the agents that are included in the scene, and later use this information during the evaluation process.

Next, we create samples extracted from the pulled scenes, by using different agents as part of the context around the pair of interest. Thus, by having enough agents for each possible pair of agents in each scene, multiple samples are created.

Moreover, the information about the groups, the label of the affinity between the pair of agents of interest, and the timeframe IDs that constitute the scene are stored to be used in later stages, i.e., when splitting the dataset into folds and then evaluating the model. Additionally, different sampling rates are used for pairs of agents in the same group and pairs of agents in different groups. This aims to achieve a balanced dataset by minimizing disparities in the number of samples for pairs of agents within the same group and those in opposing groups.

Another point to consider is the frame in which the spatial features are reported. The datasets include location data that use a world reference W. For each of the agents that are part of the context of a sample, their data are transformed to represent their relative location to a local frame of reference L_{ij}, which is unique for each pair of agents i and j that are the pair of interest for the given sample. This local frame of reference is defined by the middle point of the line connecting agents i and j and is visualized Figure 5.7. This transformation of the context features enhances the learning and generalization capability of our approach.

For dataset splitting strategy, we employed a 5-fold cross-validation approach. In Figure 5.8, we visualize the division of samples into five parts, where three parts constitute the training set, one forms the validation set, and the remaining part serves as the test set in each iteration of the cross-validation process. This ensures a balanced distribution of scenes across the five sets. To maintain the integrity of temporal relationships within scenes, we took special care to allocate all samples from the same scene to the same set. This scene-level splitting strategy is crucial for constructing affinity graphs corresponding to scenes during both the training and testing phases. Consequently, each fold contains a diverse representation of scenes, allowing our model to learn and gen-

eralize effectively across different temporal contexts. The sizes of our training, validation, and test sets were held consistently across folds. This uniformity in set sizes facilitates a fair evaluation of our model's performance across diverse temporal scenarios. The model's performance is thoroughly evaluated using the metrics discussed in the evaluation metrics section 5.2, with the average performance metrics computed across folds. This provides a comprehensive assessment of models' generalization ability. The dataset splitting strategy, combined with 5-fold cross-validation with scene-level splitting, not only optimizes computational efficiency but also ensures robust model evaluation by preserving the temporal dependencies within each scene. This approach contributes to the model's ability to generalize well to unseen data, a critical factor in the context of our group detection task.

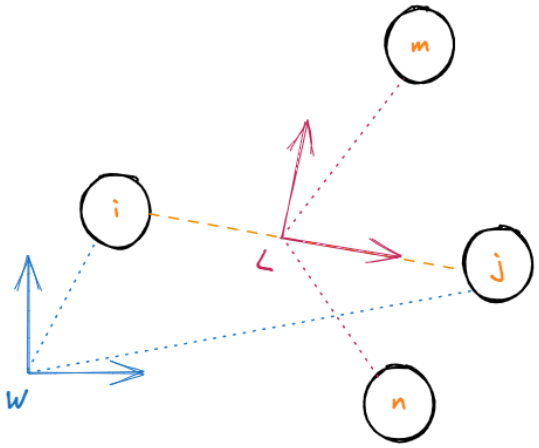


Figure 5.7: Visualisation of how the agents of the pair of interest (i and j) are using a global frame of reference W and the context agents (m and n) are using a local frame of reference L defined by the pair of interest.

5.2 Evaluation Metrics

The following section describes the two evaluation metrics that are used to assess the performance and effectiveness of the models in our experiments.

5.2.1 Group Mitre

Group Mitre was introduced by Solera et al. [20] built upon the work of Vilain et al. [28] in describing a scoring scheme for coreference tasks. The scoring process involves determining the minimal adjustments needed to transform the predicted groups into those of the ground truth groups. Specifically, the recall (as well as precision) error terms are computed by identifying the smallest number of links

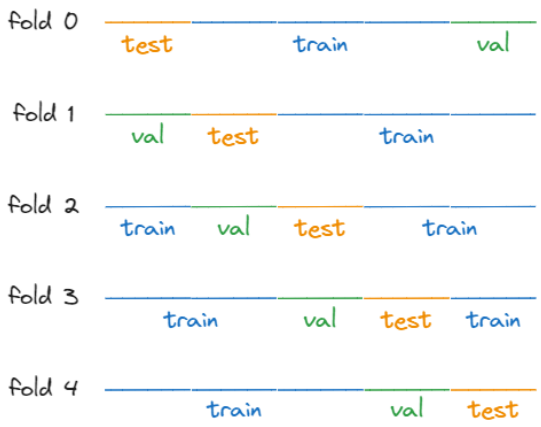


Figure 5.8: Visualisation of how the datasets has been split into folds.

required to align the predicted groups and the ground truth groups. Despite the apparent combinatorial complexity, the method leverages the concept of minimal spanning subsets, enabling the use of a simple counting approach to achieve efficient alignment. For any list of groups represented by a spanning forest, a spanning tree is an equivalence class within a group. Thus, the score can be calculated by accounting for the number of links that are needed to be added or removed in order to recover the spanning forest of the correct solution. This approach originally had the flaw of not including groups of a single individual. Thus, Solera et al. [20] proposed the addition of fake individuals to be connected to each isolated individual. In our experiments, we use the F1 Group Mitre score as an evaluation metric that combines information of both precision and recall, as formulated in Equation 5.1.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{5.1}$$

5.2.2 Group Correctness

In order to assess the effectiveness of the pipeline we created to face the group detection task, we employed an evaluation metric that has been widely recognized in previous studies [9, 21, 27, 33]. These works did not give a specific name to this metric, so we decided to refer to it as group correctness. The idea of this metric is that a specific percentage of the members in the ground truth group need to be part of the predicted group in order to consider the predicted group as a True Positive (TP). The formula used to check the minimum number of agents for a predicted group to be a TP is $P * |c_d|$, where P is a threshold parameter and $|c_d|$: indicates the cardinality or the size of the ground truth group d. If $P = 1$ then all the members of the ground truth group are needed, so the evaluation is stricter than when $P < 1$. For our experiments we checked

two values for P, $P = 2/3$ and $P = 1$. The F1 Group Correctness score was calculated to evaluate the performance of our group detection model.

5.3 Baselines

This section introduces the baselines utilized in the experiments to compare the performance of our model.

5.3.1 DANTE

Swofford et al. [21] presented a data-driven approach to detect conversational groups. Their approach introduced a novel Deep Affinity Network (DANTE) to predict the likelihood that two agents in the same scene can be part of the same conversational group, considering their social context. In more detail, DANTE is a neural network that takes the location and head orientation data of a single frame scene and tries to learn the pairwise affinities between the agents by identifying their spatial arrangements. The predicted results for all agent pairs in the scene are then used by the DS clustering algorithm to identify groups of various sizes. This pipeline was also used to test interaction scenarios between a robot and humans. In contrast to relying on head orientation, this baseline also utilizes the velocity data. The lack of temporal considerations in the design is expected to make it unable to capture temporal dependencies.

5.3.2 GDGAN

Fernando et al. [7] implemented a novel deep-learning framework for predicting human trajectories and detecting social group memberships in crowds. The framework includes a generative adversarial network that uses the spatiotemporal structure of the neighborhood around an agent in order to identify attributes that describe the social identity of the agents. The authors are approaching the problem from an unsupervised learning point of view, allowing them to apply the pipeline to various settings without the need for labeling.

5.3.3 NRI

Kipf et al. [11] introduced the NRI model. NRI is an unsupervised model that learns to estimate interactions while at the same time learning the dynamics using observational data. In more detail, this model is a variational auto-encoder that learns the interactions between agents and uses graph neural networks in order to reconstruct the data.

5.3.4 WavenetNRI

Nasri et al.[15] used an NRI [11] adaptation to perform group detection in spatiotemporal data. The model consists of a GNN encoder transformed by applying a Residual Dilated Causal Convolutional Block inspired by Wavenet

architecture [26]. This work includes both supervised and unsupervised training. Louvain community detection algorithm is used to find the clusters of the interaction graphs formed by the predictions of the model. For our experiments, we have used the supervised trained version as one of the baselines. This model uses whole scenes as samples, which is different from our approach where only a specific amount of surrounding agents is used to predict the affinities.

5.4 Implementation details

In this section, the values of the parameters of each model used in the experiments are specified.

5.4.1 DANTE

DANTE model uses two branches as our proposed approach. The branch that processes data of the main pair of agents of a sample uses three convolutional layers of different filters of values 16, 64, 128. The context branch also uses three convolutional layers, although the values are 64, 128, and 512. After the concatenation of the pair branch and context branch, two dense layers with filter values of 256 and 64 follow before the last dense layer which uses the sigmoid activation function. The dropout and regularization values used are 0.35 and 0.0000001 respectively. Adam optimizer is used during training with a learning rate value of 0.0001, beta_1 0.9, beta_2 0.999, decay 1e-5, amsgrad False, and clipvalue 0.5. beta_1 and beta_2 are parameters to control the exponential decay rate for the first and second moment estimations of the updating rule of the optimizer [10], respectively. Furthermore, binary crossentropy is selected as the loss function. The code was found on DANTE Github repository[2].

5.4.2 GDGAN

GDGAN baseline is based on the code found on WavenetNRI GitHub repository[3].

5.4.3 NRI

NRI baseline has the same parameters as WavenetNRI and our experiments were done based on the implementation found on WavenetNRI github repository[3].

5.4.4 WavenetNRI

The parameters used for this baseline were retrieved from the given code uploaded on Github[3]. The only values that were changed were the weight of group

[2]https://github.com/mswoff/DANTE
[3]https://github.com/fatcatZF/WavenetNRI

dataset	w_G	$w_{\bar{G}}$
eth	2.42	0.63
hotel	5.79	0.55
zara01	2.22	0.64
zara02	5.12	0.55
students03	11.82	0.52

Table 5.3: Group and non-group weight values used for each pedestrian dataset.

labels and the weight of the non-group labels. These values are specific for each dataset and calculated using the following equations.

$$w_G = \frac{n_G + n_{\bar{G}}}{2n_G} \tag{5.2}$$

$$w_{\bar{G}} = \frac{n_G + n_{\bar{G}}}{2n_{\bar{G}}} \tag{5.3}$$

Equation 5.2 is used to estimate w_G, which is the weight of group labels and equation 5.3 is used to estimate $w_{\bar{G}}$, which is the weight of non-group labels. Accordingly, Table 5.3 presents the values of w_G and $w_{\bar{G}}$ calculated for the pedestrian datasets experiments. On the other hand, for all simulation datasets the values of w_G and $w_{\bar{G}}$ used during the training process was 0.5.

5.4.5 T-DANTE

The architecture of T-DANTE is similar to DANTE, so most of the parameters used are kept the same. The difference is the number of filters used in the Pair branch layers and Context branch layers. The three Pair branch layers have 32, 128, and 256 filters and the three Context branch layers have 64, 128, and 256 filters. Another parameter that has been used for the T-DANTE GD variation of T-DANTE is the one that swaps the LSTM layers with GRU layers and the Conv1D layers with Dense layers. The number of filters for each branch is kept the same for the T-DANTE GD neural network structure.

In order to compare our model with WavenetNRI in pedestrian datasets, we chose scenes of 15 timeframes as the original work [15]. The NRI and GDGAN baselines are also built with scene sizes of 15 timeframes. On the other hand, DANTE uses just a single frame to evaluate pairwise affinities of agents due to its temporal limitation, which leads to higher number of samples. For the simulation datasets, all of the baselines and T-DANTE use 50 timeframes as scene size, which is the length of each simulation dataset sample.

Chapter 6

Results

This section presents the results of our experiment. Each pedestrian dataset has been split into 5 folds and for each fold, each method has been evaluated 5 times, in total 25 runs per method. On the other hand, the spring simulation datasets have not been split into folds, as they were generated under controlled conditions that do not exhibit the distribution shifts typically addressed by cross-validation. Thus, each method has been evaluated 25 times for each simulation dataset. The Wilcoxon signed rank test has been applied to compare the results of the experiments and check if there is any significant difference between the top two performing models. This statistical test was selected as it is a non-parametric version of the paired T-test and it provides a statistic that is easy to interpret. In the tables used for this section, the significance of the superiority of a model for a dataset is shown by the use of $*$ next to the value. Our first phase of experiments is related to the ablation study and the second phase of the results is the comparison of our model T-DANTE against four other baselines.

6.1 Ablation study

The ablation study is conducted per dataset type. For both the pedestrian and simulation datasets, we perform multiple experiments with different design layers and context sizes in our proposed T-DANTE model.

6.1.1 Pedestrian datasets

Table 6.1 and Table 6.2 show the results of T-DANTE when using various context sizes including 0, 4, and 8 context size and scene size of 15 timeframes. The table includes T-DANTE with LSTM and Conv1D layers mentioned as T-DANTE, and T-DANTE GD which includes GRU and Dense layers. According to this Table, the main conclusion is that including context information enhances the performance of T-DANTE, except in the *eth* dataset when using the Group Mitre as the evaluation metric. T-DANTE and T-DANTE GD show lower values

	eth	hotel	zara01	zara02	students03
T-DANTE	0.5859	0.5232	0.8103	0.8491	0.5421
no context	±0.0206	±0.0283	±0.0185	±0.0121	±0.0959
T-DANTE	0.5736	**0.5338***	**0.8224**	0.862	0.644
context 4	±0.0193	±0.0293	±0.0147	±0.0101	±0.0824
T-DANTE	**0.5901**	0.508	0.8215	**0.8699***	**0.6957**
context 8	±0.0301	±0.0434	±0.0149	±0.0108	±0.0563
T-DANTE	0.5593	0.5199	0.8027	0.8416	0.6332
GD no context	±0.027	±0.027	±0.0182	±0.0223	±0.114
T-DANTE	0.5453	0.5274	0.8149	0.8457	0.6661
GD context 4	±0.026	±0.0261	±0.0186	±0.0121	±0.0844
T-DANTE	0.5663	0.5241	0.8077	0.8536	0.6777
GD context 8	±0.0357	±0.0335	±0.022	±0.0191	±0.0855

Table 6.1: Group Correctness metric with $P = 1$ for T-DANTE variations in all pedestrian datasets. Context sizes of 0, 4 and 8 agents and scene size of 15 consecutive timeframes. ***** shows that this result is significantly different than all the other values in the same dataset.

for datasets *eth* and *hotel*, which can be explained by the high number of scenes with less than 5 agents (Figure 5.3), that leads the models to include zeros as fake agent data in order to fill the Context Branch which has fixed dimensions. A larger context size seems to work efficiently mostly for *zara*02 and *students*03 datasets where more agents appear in their scenes (Figure 5.3). This trend can be seen for both T-DANTE and T-DANTE GD models. However, the use of a larger context size can also deteriorate the performance of the model as we can see in Table 6.2 when analyzing *hotel* dataset, where using 8 agents as context significantly diminishes the performance of T-DANTE compared with using 0 or 4 agents. The distribution of agents in the scenes of *hotel* dataset, as high number of the scenes have less than 10 agents (2 pair agents + 8 context agents), is responsible for the low results. Similar features can be found when comparing Group Correctness results using $P = \frac{2}{3}$ (See Appendix 9.2). Regarding our experiments related to context size for pedestrian datasets, we conclude that each dataset may benefit from using a different context size depending on the features of the dataset, such as the average number of agents per scene. Another point is that T-DANTE performs better than T-DANTE GD in almost all of the pedestrian datasets. This leads us to the interpretation that the combination of LSTM and Conv1D layers are better equipped than the combination of GRU and Dense layers to process the information passed through our neural network.

	eth	hotel	zara01	zara02	students03
T-DANTE	**0.6743**	0.6024	0.8248	0.848	0.7135
no context	±0.015	±0.0203	±0.0166	±0.0137	±0.0386
T-DANTE	0.6693	0.6038	**0.838**	0.8628	0.7537
context 4	±0.012	±0.0213	±0.0143	±0.0106	±0.0367
T-DANTE	0.6651	0.5425	0.838	**0.8726***	**0.7805**
context 8	±0.0168	±0.0228	±0.0152	±0.0106	±0.0281
T-DANTE	0.6645	0.5998	0.8185	0.8415	0.7521
GD no context	±0.0176	±0.0186	±0.0171	±0.0147	±0.0549
T-DANTE	0.6614	**0.6116**	0.8326	0.8452	0.7668
GD context 4	±0.017	±0.0141	±0.0175	±0.0106	±0.0385
T-DANTE	0.6543	0.5574	0.8288	0.8532	0.7702
GD context 8	±0.0234	±0.0321	±0.0201	±0.0155	±0.0369

Table 6.2: Group Mitre metric for T-DANTE variations in all pedestrian datasets. Context sizes of 0, 4 and 8 agents and scene size of 15 consecutive timeframes. * shows that this result is significantly different than all the other values in the same dataset.

6.1.2 Simulation datasets

The result of Group Correctness and Group Mitre evaluation metrics of T-DANTE for different variations using the simulation datasets are presented in Tables 6.3 and Table 6.4, respectively. The most notable feature of these tables is that T-DANTE with a context of 8 agents does not perform any better than T-DANTE with no context and T-DANTE with a context of 4 agents. This is rational for the simulation datasets, which have a constant number of 8 and 9 agents in their scenes, in which the input of the Context Branch of the neural network is filled with fake data (zeros) to reach the given context size, as there are not enough agents to fill the context information. For the rest of the results in both tables, the T-DANTE model with context size of 4 agents ranks first in sim_2 and sim_3 datasets, and T-DANTE GD with context size of 4 performs better for sim_1, sim_4, sim_5 and sim_6 datasets. An important feature is that the performance of the model based on the Group Mitre metric in Table 6.4 with no context and context of 4 agents are very similar, but the corresponding values with the Group Correctness in Table 6.3 present larger differences in their performance. Another point is that the performance of all the simulation experiments is significantly better with respect to absolute values compared to the performance of the model using pedestrian datasets. This can be explained by the existence of more complete data and the lack of randomness between the samples in the simulation datasets.

	sim_1	sim_2	sim_3	sim_4	sim_5	sim_6
T-DANTE	0.9653	0.9541	0.9473	0.9477	0.9609	0.9323
no context	±0.0082	±0.009	±0.0137	±0.0088	±0.0055	±0.0087
T-DANTE	0.9693	**0.9798***	**0.9817***	0.9707	0.9593	0.9449
context 4	±0.0023	±0.0018	±0.0062	±0.0065	±0.014	±0.0113
T-DANTE	0.942	0.9639	0.9639	0.9333	0.906	0.8918
context 8	±0.0117	±0.0087	±0.012	±0.0124	±0.0298	±0.0162
T-DANTE	0.978	0.9665	0.9599	0.9666	0.9698	0.9455
GD no context	±0.0074	±0.0074	±0.0083	±0.006	±0.0068	±0.0084
T-DANTE	**0.9807**	0.9726	0.9703	**0.9749***	**0.9737***	**0.9604***
GD context 4	±0.0052	±0.0107	±0.0061	±0.0041	±0.0064	±0.0079
T-DANTE	0.9673	0.958	0.9564	0.9441	0.9385	0.9145
GD context 8	±0.0099	±0.0133	±0.0119	±0.0153	±0.0175	±0.0138

Table 6.3: Group Correctness metric with $P = 1$ for T-DANTE variations in all spring simulation datasets. Context sizes of 0, 4 and 8 agents and scene size of 50 consecutive timeframes. * shows that this result is significantly different than all the other values in the same dataset.

The interpretation of the ablation study results leads to the conclusion that in almost all cases the use of context is beneficiary for the performance of the model using either of the evaluation metrics. However, the best size of this context is not the same for all the datasets, as each dataset holds different characteristics like the number of agents that are present in every timeframe.

	sim_1	sim_2	sim_3	sim_4	sim_5	sim_6
T-DANTE	0.979	0.9699	0.9749	0.9792	0.9781	0.9742
no context	±0.0041	±0.0055	±0.0052	±0.0031	±0.0021	±0.0031
T-DANTE	0.9834	**0.9888***	**0.9876***	0.9866	0.9778	0.9763
context 4	±0.0016	±0.0012	±0.0028	±0.0023	±0.0057	±0.0032
T-DANTE	0.9724	0.9809	0.9783	0.9733	0.9528	0.9599
context 8	±0.0037	±0.0036	±0.0049	±0.0054	±0.0134	±0.0044
T-DANTE	0.9853	0.9776	0.9836	0.986	0.9813	0.9788
GD no context	±0.0042	±0.0035	±0.0037	±0.0021	±0.0041	±0.0026
T-DANTE	**0.9858**	0.9789	0.9837	**0.9876***	**0.9831**	**0.9797**
GD context 4	±0.0028	±0.0059	±0.0022	±0.0012	±0.0035	±0.0031
T-DANTE	0.9779	0.9692	0.9756	0.9759	0.9619	0.9659
GD context 8	±0.0053	±0.0065	±0.0041	±0.0046	±0.0088	±0.0043

Table 6.4: Group Mitre metric for T-DANTE variations in all spring simulation datasets. Context sizes of 0, 4 and 8 agents and scene size of 50 consecutive timeframes. ***** shows that this result is significantly different than all the other values in the same dataset.

6.2 T-DANTE vs Baselines

In this section, the results of T-DANTE compared with the baselines will be discussed. Table 6.5 and Table 6.6 show the results of the experiments for the pedestrian datasets and Table 6.7 and Table 6.8 present the results of the simulation dataset.

6.2.1 Pedestrian datasets

	eth	hotel	zara01	zara02	students03
DANTE	0.3195 ±0.0474	0.4306 ±0.0435	0.7307 ±0.051	0.6334 ±0.0376	0.0236 ±0.0107
NRI	0.201 ±0.062	0.1691 ±0.0544	0.285 ±0.0671	0.1058 ±0.0346	0.0065 ±0.0096
GDGAN	0.3243 ±0.0459	0.3224 ±0.0458	0.5373 ±0.0288	0.1887 ±0.0309	0.0811 ±0.019
WavenetNRI	0.2419 ±0.059	0.2021 ±0.0485	0.3611 ±0.0912	0.1839 ±0.0655	0.0011 ±0.0041
T-DANTE	**0.5901*** ±0.0301	**0.508*** ±0.0434	**0.8215*** ±0.0149	**0.8699*** ±0.0108	**0.6957*** ±0.0563

Table 6.5: Group Correctness metric with $P = 1$ for T-DANTE vs Baselines in all pedestrian datasets. * shows that this result is significantly different than all the other values in the same column.

T-DANTE with a context size of 8 agents was chosen for this part of the comparison because, on average, it performed best in the ablation study using the pedestrian datasets. According to 6.5, T-DANTE produces superior results than all baselines, i.e., DANTE, NRI, GDGAN, and WavenetNRI, for all pedestrian datasets using Group Correctness metric. Results in table 6.6 with Group Mitre metric do not agree completely with Group Correctness as GDGAN is producing the best results for *eth* and *hotel* datasets, and T-DANTE takes the second place in these datasets. This can happen due to the tendency of Group Correctness metric at penalising more the model, when a false negative group has been predicted. GDGAN creates many false negative groups for *eth* and *hotel* datasets. While in all other datasets, T-DANTE performed best. These results can lead us to the assumption that due to the superiority of T-DANTE versus DANTE, the addition of a temporal aspect using the LSTM layers ameliorates the performance of the model. Another notable aspect of the tables is the high standard deviation in NRI and WavenetNRI results compared to the rest of the results. This means that these models do not consistently learn

	eth	hotel	zara01	zara02	students03
DANTE	0.5479 ±0.0195	0.5859 ±0.0355	0.7932 ±0.0285	0.7054 ±0.0259	0.5019 ±0.0131
NRI	0.5707 ±0.0738	0.5399 ±0.0972	0.5969 ±0.0527	0.4172 ±0.019	0.28 ±0.026
GDGAN	**0.7058*** ±0.0257	**0.6117*** ±0.0514	0.7762 ±0.0218	0.5273 ±0.0237	0.286 ±0.0215
WavenetNRI	0.5526 ±0.0568	0.4549 ±0.0799	0.627 ±0.0659	0.4622 ±0.0401	0.2799 ±0.0237
T-DANTE	0.6651 ±0.0168	0.5425 ±0.0228	**0.838*** ±0.0152	**0.8726*** ±0.0106	**0.7805*** ±0.0281

Table 6.6: Group Mitre metric for T-DANTE vs Baselines in all pedestrian datasets. * shows that this result is significantly different than all the other values in the same dataset.

how to distinguish the different classes in every experiment run, which leads to the difference between their results that is represented by standard deviation. The low values produced by the baselines for the *students*03 dataset could be explained by the high number of agents of this dataset 5.1, as most of them take into account all the agents in a scene.

6.2.2 Simulation datasets

In the simulation dataset, the T-DANTE with a context size of 4 was selected as it showed better results in the ablation study. According to Table 6.7 and Table 6.8, WavenetNRI and NRI performed better than T-DANTE in all simulation datasets using both evaluation metrics. On the other hand, GDGAN was unable to capture the patterns in the simulation datasets, as its weak performance is shown in both tables. The existence of large groups in the simulation datasets could be the reason for these low values occurring. DANTE baseline is performing better than GDGAN, but it does not reach the performance of the other three models. T-DANTE performs worse than WavenetNRI and NRI, and better than DANTE and GDGAN. The WavenetNRI and NRI are able to process and analyze the entire scene, which might explain their superiority.

In general, the results demonstrate the effect of including the temporal dynamics of the datasets in the structure of the neural network. The results of T-DANTE vs DANTE clearly represent that the use of data from multiple timeframes in a single sample helps the model to perform better. Another point is that T-DANTE surpasses the baselines in the pedestrian datasets, but in the simulation datasets, NRI and WavenetNRI baselines share the first- and

	sim_1	sim_2	sim_3	sim_4	sim_5	sim_6
DANTE	0.2155 ±0.0075	0.1978 ±0.0079	0.0955 ±0.0108	0.1992 ±0.0108	0.0804 ±0.0107	0.0408 ±0.0075
NRI	0.9837 ±0.0041	0.9828 ±0.0067	**0.9884*** ±0.0043	0.9961 ±0.0026	**0.9897*** ±0.0049	**0.9885*** ±0.0071
GDGAN	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
WavenetNRI	**0.996** ±0.0061	**0.9953*** ±0.0042	0.9772 ±0.0082	**0.9983*** ±0.0035	0.9721 ±0.0117	0.953 ±0.0115
T-DANTE	0.9693 ±0.0023	0.9798 ±0.0018	0.9817 ±0.0062	0.9707 ±0.0065	0.9593 ±0.014	0.9449 ±0.0113

Table 6.7: Group Correctness metric with $P = 1$ for T-DANTE vs Baselines in all spring simulation datasets. * shows that this result is significantly different than all the other values in the same dataset.

	sim_1	sim_2	sim_3	sim_4	sim_5	sim_6
DANTE	0.717 ±0.0041	0.7012 ±0.0035	0.518 ±0.011	0.7117 ±0.0046	0.5272 ±0.007	0.4246 ±0.0089
NRI	0.9915 ±0.0024	0.9934 ±0.0025	**0.9946*** ±0.0023	0.9986 ±0.0009	**0.994** ±0.0027	**0.9948** ±0.0026
GDGAN	0.0067 ±0.0	0.0031 ±0.0003	0.0411 ±0.0	0.0024 ±0.0003	0.0278 ±0.0	0.092 ±0.0
WavenetNRI	**0.9984** ±0.0022	**0.9984*** ±0.0015	0.9882 ±0.0044	**0.9995** ±0.001	0.9878 ±0.0055	0.9685 ±0.009
T-DANTE	0.9834 ±0.0016	0.9888 ±0.0012	0.9876 ±0.0028	0.9866 ±0.0023	0.9778 ±0.0057	0.9763 ±0.0032

Table 6.8: Group Mitre metric for T-DANTE vs Baselines in all spring simulation datasets. * shows that this result is significantly different than all the other values in the same dataset.

second-best places. This behavior might be explained by different characteristics of the pedestrian and simulation datasets. Simulation datasets include more data samples and these samples include a higher number of scenes with groups of over 3 members than pedestrian datasets. The baselines were unable to capture the interactions of smaller groups that appear more frequently in

the pedestrian datasets. However, these baselines were able to effectively find patterns in larger groups that are majorly included in the simulation datasets. Group Correction metric seems to give harsher punishment, when false negative group are detected, than Group Mitre metric. This can be seen in all the tables of this section. The results are also affected by the differences between pedestrian datasets related to number of agents included in each one 5.1. The best results occur for the *zara*01 and *zara*02 pedestrian datasets that involve the least number of agents. On the opposite side, *eth*, *hotel* and *students*03 datasets, which include higher number of agents.

After our experimentation, we answer the questions we set at the beginning of the experiments section 5.

- *Can the addition of the temporal aspect of data lead to better results on the group detection problem?*
  The integration of temporal dynamics in T-DANTE enhanced the performance of the model. This is demonstrated in T-DANTE vs baselines section, as T-DANTE is performing better than DANTE, while DANTE only uses a single frame per sample.

- *Does the use of context information help to gain better performance?*
  Context information seems to enhance the performance of T-DANTE mostly when using the context of 4 agents compared to using no context information.

- *Does the size of the context information affect the performance of the model?*
  The best context size depends on the characteristics of the dataset. For some datasets context size of 4 agents was more suitable than 8 agents. For example, the simulation datasets did not perform well with the context size of 8 agents. However, T-DANTE with a context size of 8 agents have been the best for modeling *zara*02 and *students*03 datasets.

- *How does our method perform compared to other methods?*
  Our proposed method outperformed other baselines almost in all cases in pedestrian datasets. To be more precise, in 3 out of 5 cases using Group Mitre metric and 5 out of 5 cases using Group Correctness metric with $P = 1$, our model produced higher outcomes than the baselines. However, in simulation datasets, NRI and WavenetNRI baselines show superior performance than T-DANTE.

Chapter 7

Conclusion

In conclusion, this thesis has delved into the critical task of group detection within spatiotemporal data, showcasing its broad applications spanning intelligent surveillance, human mobility modeling, animal trajectory analysis, and natural phenomena forecasting. Our proposed methodology, drawing inspiration from the neural network architecture introduced by Swofford et al. [21] and enriched with RNN layers, demonstrates its effectiveness in capturing the temporal dynamics inherent in agent movements. The subsequent application of the DS community detection algorithm unveils latent groups within the constructed social graph. The insights gathered from our experiments show the value of context size, temporal information, and the model's ability to discern meaningful groups in both pedestrian and simulation datasets. Our experiments demonstrate that T-DANTE is the superior model for the group detection task using pedestrian datasets. However, T-DANTE does not surpass other baselines in simulation datasets but seems to be competitive. The conclusion of the experiments versus the baselines executed for both pedestrian and simulation datasets demonstrated that T-DANTE was the only approach with acceptable performance in all kinds of datasets.

The major limitation of our work is the need to feed the Context Branch of T-DANTE, even when there are not enough agents in a scene. The incorporation of dynamic context size per scene based on the presented number of agents would be a solution that could be explored in order to feed the model with actual data.

Nevertheless, the complexity of spatiotemporal data and the dynamic nature of group interactions present challenges that need further exploration. Future research endeavors could deepen our understanding by refining the proposed methodology, exploring alternative neural network architectures, and investigating the impact of better hyperparameter tuning on model performance. The generalizability of our approach across different datasets and its scalability to real-time applications could be another future approach. Last but not least, an intriguing direction would be experimenting with a spectrum of community detection algorithms. Diverse algorithms, beyond the DS algorithm applied in our current work, might offer better results.

Chapter 8

Ethical Considerations

The publicly available datasets that we used were captured with the use of cameras. The camera footage was then transformed to location and velocity data for each of the agents involved. The use of such camera footage should be used only with the consent of the people included and should be anonymised properly in order to avoid being used for unethical purposes. In our case we use our data and our results only for research purposes and in no way intend to create any commercial benefit by manipulating this data.

Bibliography

[1] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):53, Mar 2021.

[2] Javad Amirian, Bingqing Zhang, Francisco Valente Castro, Juan Jose Baldelomar, Jean-Bernard Hayet, and Julien Pettre. Opentraj: Assessing prediction complexity in human trajectories datasets. In *Asian Conference on Computer Vision (ACCV)*. Springer, 2020.

[3] Yoshua Bengio. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

[4] Hao Chen, Seung Hyun Cha, and Tae Wan Kim. A framework for group activity detection and recognition using smartphone sensors and beacons. *Building and Environment*, 158:205–216, 2019.

[5] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[6] Nachiket Deo and Mohan M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[7] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Gd-gan: Generative adversarial networks for trajectory prediction and group detection in crowds. In C. V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision – ACCV 2018*, pages 314–330, Cham, 2019. Springer International Publishing.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[9] Hayley Hung and Ben Kröse. Detecting f-formations as dominant sets. In *Proceedings of the 13th International Conference on Multimodal Inter-*

faces, ICMI '11, page 231–238, New York, NY, USA, 2011. Association for Computing Machinery.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[11] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2688–2697. PMLR, 10–15 Jul 2018.

[12] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, page 593–604, New York, NY, USA, 2007. Association for Computing Machinery.

[13] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[14] Maedeh Nasri, Mitra Baratchi, Yung-Ting Tsou, Sarah Giest, Alexander Koutamanis, and Carolien Rieffe. A novel metric to measure spatio-temporal proximity: a case study analyzing children's social network in schoolyards. *Applied Network Science*, 8(1):50, 2023.

[15] Maedeh Nasri, Zhizhou Fang, Mitra Baratchi, Gwenn Englebienne, Shenghui Wang, Alexander Koutamanis, and Carolien Rieffe. A gnn-based architecture for group detection from spatio-temporal trajectory data. In Bruno Crémilleux, Sibylle Hess, and Siegfried Nijssen, editors, *Advances in Intelligent Data Analysis XXI*, pages 327–339, Cham, 2023. Springer Nature Switzerland.

[16] Maedeh Nasri, Yung-Ting Tsou, Alexander Koutamanis, Mitra Baratchi, Sarah Giest, Dennis Reidsma, and Carolien Rieffe. A novel data-driven approach to examine children's movements and social behaviour in schoolyard environments. *Children*, 9(8):1177, 2022.

[17] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.

[18] Rijurekha Sen, Youngki Lee, Kasthuri Jayarajah, Archan Misra, and Rajesh Krishna Balan. Grumon: Fast and accurate group monitoring for heterogeneous urban spaces. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, SenSys '14, page 46–60, New York, NY, USA, 2014. Association for Computing Machinery.

[19] Francesco Solera, Simone Calderara, and Rita Cucchiara. Structured learning for detection of social groups in crowd. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 7–12, 2013.

[20] Francesco Solera, Simone Calderara, and Rita Cucchiara. Socially constrained structural learning for groups detection in crowd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):995–1008, 2016.

[21] Mason Swofford, John Peruzzi, Nathan Tsoi, Sydney Thompson, Roberto Martín-Martín, Silvio Savarese, and Marynel Vázquez. Improving social awareness through dante: Deep affinity network for clustering conversational interactants. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW1), may 2020.

[22] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

[23] Stephanie Tan, David M.J. Tax, and Hayley Hung. Conversation group detection with spatio-temporal context. In *Proceedings of the 2022 International Conference on Multimodal Interaction*, ICMI '22, page 170–180, New York, NY, USA, 2022. Association for Computing Machinery.

[24] Lu-An Tang, Yu Zheng, Jing Yuan, Jiawei Han, Alice Leung, Wen-Chih Peng, and Thomas La Porta. A framework of traveling companion discovery on trajectory data streams. *ACM Trans. Intell. Syst. Technol.*, 5(1), jan 2014.

[25] Sydney Thompson, Abhijit Gupta, Anjali W. Gupta, Austin Chen, and Marynel Vázquez. Conversational group detection with graph neural networks. In *Proceedings of the 2021 International Conference on Multimodal Interaction*, ICMI '21, page 248–252, New York, NY, USA, 2021. Association for Computing Machinery.

[26] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016.

[27] Sebastiano Vascon, Eyasu Z. Mequanint, Marco Cristani, Hayley Hung, Marcello Pelillo, and Vittorio Murino. Detecting conversational groups in images and sequences: A robust game-theoretic approach. *Computer Vision and Image Understanding*, 143:11–24, 2016. Inference and Learning of Graphical Models Theory and Applications in Computer Vision and Image Analysis.

[28] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, page 45–52, USA, 1995. Association for Computational Linguistics.

[29] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[30] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *CoRR*, abs/1810.00826, 2018.

[31] Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. Who are you with and where are you going? In *CVPR 2011*, pages 1345–1352, 2011.

[32] Chao Zhang, Keyang Zhang, Quan Yuan, Luming Zhang, Tim Hanratty, and Jiawei Han. Gmove: Group-level mobility modeling using geo-tagged social media. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1305–1314, New York, NY, USA, 2016. Association for Computing Machinery.

[33] Lu Zhang and Hayley Hung. Beyond f-formations: Determining social involvement in free standing conversing groups from static images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1086–1095, 2016.

[34] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Chapter 9

Appendix

This appendix contains data that would have taken up too much space to include in the main thesis.

9.1 Simulation dataset visualisations

More visual representations of the simulation samples can be found below.

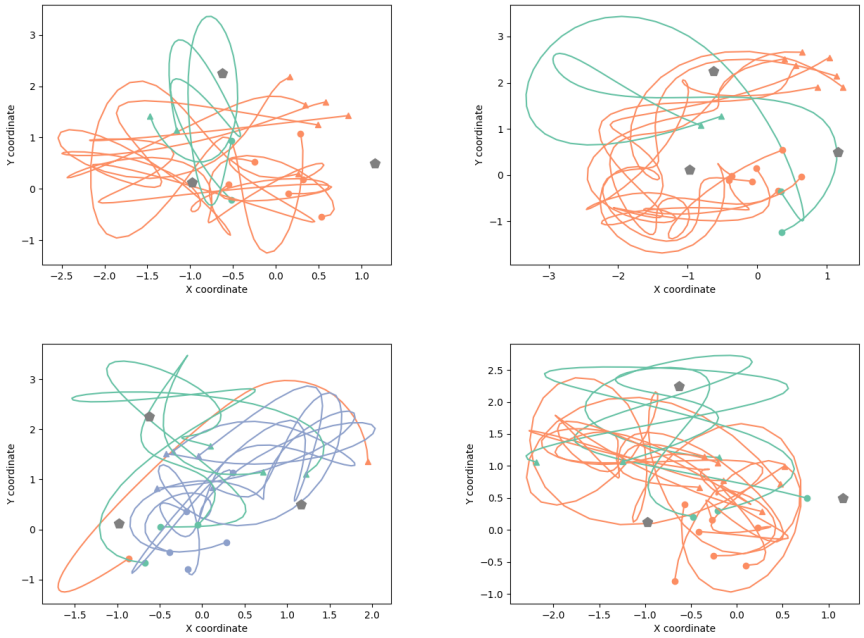


Figure 9.1: Visualisations of spring simulations with attraction points. Circle markers are the starting points of the trajectory of a particle and triangle markers are the final points of the trajectory of a particle. Each group of particles share the same color and the pentagon markers represent the attraction points.

9.2 Result tables

This part of the appendix shows result tables for Group Correctness evaluation using threshold value 2/3.

9.2.1 Ablation study

	eth	hotel	zara01	zara02	students03
T-DANTE	**0.6094**	0.5306	0.8188	0.8552	0.7084
no context	±0.0185	±0.0272	±0.0127	±0.0079	±0.075
T-DANTE	0.6063	0.5367	0.8253	0.8641	0.7768
context 4	±0.0156	±0.0284	±0.0129	±0.0093	±0.045
T-DANTE	0.609	0.533	**0.8256**	**0.8715***	**0.807**
context 8	±0.0221	±0.0253	±0.0136	±0.01	±0.0374
T-DANTE	0.6057	0.5258	0.8134	0.8508	0.7752
GD no context	±0.0174	±0.026	±0.015	±0.0111	±0.069
T-DANTE	0.5983	0.5306	0.8236	0.8522	0.7919
GD context 4	±0.019	±0.0261	±0.0158	±0.0097	±0.057
T-DANTE	0.5999	**0.5382**	0.8202	0.8592	0.7979
GD context 8	±0.0255	±0.0283	±0.0163	±0.0122	±0.0493

Table 9.1: Group Correctness metric with $P = \frac{2}{3}$ for T-DANTE variations in all pedestrian datasets. Context sizes of 0, 4 and 8 agents and scene size of 15 consecutive timeframes. ***** shows that this result is significantly different than all the other values in the same column.

	sim_1	sim_2	sim_3	sim_4	sim_5	sim_6
T-DANTE	0.978	0.9737	0.97	0.9668	0.9745	0.9448
no context	±0.0051	±0.0059	±0.0064	±0.0059	±0.004	±0.007
T-DANTE	0.9734	**0.9854***	**0.9869***	0.9803	0.9745	0.9578
context 4	±0.0021	±0.0019	±0.0039	±0.0029	±0.008	±0.0093
T-DANTE	0.9555	0.9771	0.9755	0.9616	0.9408	0.9169
context 8	±0.0114	±0.0052	±0.0079	±0.0074	±0.019	±0.01
T-DANTE	0.9848	0.9799	0.9784	0.9764	0.9808	0.9557
GD no context	±0.0038	±0.0048	±0.0058	±0.0031	±0.0049	±0.0064
T-DANTE	**0.9857**	0.983	0.9825	**0.9854***	**0.9832***	**0.9687***
GD context 4	±0.0028	±0.0046	±0.003	±0.0031	±0.0036	±0.0053
T-DANTE	0.98	0.9749	0.974	0.9693	0.9603	0.9346
GD context 8	±0.0048	±0.0067	±0.0064	±0.0093	±0.0121	±0.0145

Table 9.2: Group Correctness metric with $P = \frac{2}{3}$ for T-DANTE variations in all spring simulation datasets. Context sizes of 0, 4 and 8 agents and scene size of 50 consecutive timeframes. * shows that this result is significantly different than all the other values in the same dataset.

9.2.2 T-DANTE vs Baselines

	eth	hotel	zara01	zara02	students03
DANTE	0.5145 ±0.026	0.4757 ±0.0349	**0.8278** ±0.0292	0.7841 ±0.0408	0.1111 ±0.0231
NRI	0.3554 ±0.074	0.2587 ±0.0695	0.5231 ±0.0581	0.2497 ±0.0432	0.0143 ±0.0177
GDGAN	0.4438 ±0.0364	0.3291 ±0.0421	0.6246 ±0.0334	0.2754 ±0.0356	0.1865 ±0.0285
WavenetNRI	0.3942 ±0.0545	0.2831 ±0.0496	0.5551 ±0.0769	0.3298 ±0.0577	0.0044 ±0.0083
T-DANTE	**0.609*** ±0.0221	**0.533*** ±0.0253	0.8256 ±0.0136	**0.8715*** ±0.01	**0.807*** ±0.0374

Table 9.3: Group Correctness metric with $P = \frac{2}{3}$ for T-DANTE vs Baselines in all pedestrian datasets. * shows that this result is significantly different than all the other values in the same dataset.

	sim_1	sim_2	sim_3	sim_4	sim_5	sim_6
DANTE	0.5189 ±0.0089	0.5601 ±0.0092	0.284 ±0.0197	0.5299 ±0.0161	0.2621 ±0.0221	0.1677 ±0.0288
NRI	0.9886 ±0.0035	0.987 ±0.0044	**0.9906*** ±0.0036	0.9985 ±0.0017	**0.9922*** ±0.0036	**0.9932*** ±0.0032
GDGAN	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.001 ±0.001
WavenetNRI	**0.9993** ±0.0015	**0.9983*** ±0.0023	0.9849 ±0.0049	**0.9996*** ±0.0013	0.9793 ±0.0096	0.9641 ±0.0086
T-DANTE	0.9734 ±0.0021	0.9854 ±0.0019	0.9869 ±0.0039	0.9803 ±0.0029	0.9745 ±0.008	0.9578 ±0.0093

Table 9.4: Group Correctness metric with $P = \frac{2}{3}$ for T-DANTE vs Baselines in all spring simulation datasets. * shows that this result is significantly different than all the other values in the same dataset.