Conversational Group Detection with Graph Neural Networks

Sydney Thompson∗
Yale University
New Haven, CT, United States
sydney.thompson@yale.edu

Abhijit Gupta∗
Anjali W. Gupta
Yale University
New Haven, CT, United States

Austin Chen
Marynel Vázquez
Yale University
New Haven, CT, United States

Figure 1: Proposed approach for conversational group detection ((a) includes an example frame from MatchNMingle [5]).

ABSTRACT
We study conversational group detection in varied social scenes using a message-passing Graph Neural Network (GNN) in combination with the Dominant Sets clustering algorithm. Our approach first describes a scene as an interaction graph, where nodes encode individual features and edges encode pairwise relationship data. Then, it uses a GNN to predict pairwise affinity values that represent the likelihood of two people interacting together, and computes non-overlapping group assignments based on these affinities. We evaluate the proposed approach on the Cocktail Party and MatchNMingle datasets. Our results suggest that using GNNs to leverage both individual and relationship features when computing groups is beneficial, especially when more features are available for each individual.

CCS CONCEPTS
• Computing methodologies → Activity recognition and understanding.

KEYWORDS
F-formation; clustering; graph neural network

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:

∗Both authors contributed equally to this research.

1INTRODUCTION
Conversational group detection has a wide range of applications, including video surveillance [8, 15, 29], displays and exhibits [10, 16], co-located collaboration [21], and interactive playgrounds [17, 22]. Group detection can also enable better spoken language interaction with situated agents [4], non-verbal robot behavior generation [34], and socially aware robot navigation in human environments [26].

Similar to prior work, we approach the problem of conversational group detection by reasoning about human proxemics [12] and conversational formations. During free-standing conversations, people tend to form certain spatial patterns with each other, known as Face Formations or F-Formations in short [18]. F-Formations are varied, adapting to factors such as density and physical environmental constraints. They characterize conversational groups.

We describe a social scene as an interaction graph and explore using a Graph Neural Network (GNN) [2] for conversational group detection. Inspired by Swofford and colleagues [30], we use the GNN to predict pairwise affinities for the graph, which encode the likelihood that two people are part of an F-Formation. Then, we use the affinities to cluster people into conversational groups, as shown in Figure 1. While Swofford and colleagues [30] used a Deep Set [25, 40] architecture to aggregate context from graph nodes when predicting an affinity value, this work advocates in favor of a more general message-passing architecture for reasoning about information in both the nodes and edges. This allows us to reduce feature engineering and more explicitly leverage relational features.

In summary, our main contributions are threefold. First, we propose a novel approach for group detection which relies on a GNN. Second, we conduct experiments on two datasets with varied input features such as position, orientation, and top-down images of participants to demonstrate the efficacy of the proposed model. Third, we open-source our code to facilitate future reproducibility.1

2RELATED WORK
The problem of conversational group detection has traditionally been approached by hand-crafted heuristics and mathematical models [15, 29, 32]. However, advancements in machine learning have

1http://gitlab.com/interactive-machines/perception/group_gnn
enabled improved social awareness with greater generalization
[11, 14, 30]. In particular, the approach by Swofford et al. [30], called
DANTE, outperformed several traditional approaches. DANTE re-
ceives spatial features for people in a scene and constructs a fully-
connected interaction graph, using the input data as node features.
It then computes pairwise affinities by combining the dyad node
features with context aggregated using a Deep Set architecture
[25, 40]. These affinities are used to partition the graph with the
Dominant Sets algorithm [15, 23]. Because DANTE computes con-
text in tandem with a dyad, it relies heavily on hand-crafted feature
transformations to preserve rotation and translation invariance.

While DANTE [30] mainly reasons about information encoded
in the nodes of a graph, we propose to use a more general message-
passing GNN architecture [2] for affinity prediction. The GNN
consists of a collection of update and aggregate functions that
allow for node and edge information consolidation in a graph. This
GNN architecture is a superset of Deep Sets, as discussed in [2].

While GNNs have previously been used for node clustering
[3, 19, 31, 38], our problem differs in several key ways. Methods such
as [19, 31] require an input affinity matrix, while our GNN must
calculate the affinities itself. Also, several prior models for clustering
with deep learning require information about the number of clusters
[3, 38]; however, we do not know the number of conversational
groups in a scene in advance. Lastly, many models (e.g. [6, 13]) for
social interaction analysis are designed and evaluated on large
graphs (see [36] for five such datasets with an average number of
nodes ranging from 13 to 500). In our case, we make predictions
over smaller graphs (2-16 nodes) as there is a physical limit to how
many people can interact simultaneously in a given place [18].

Three reasons motivate us to predict an affinity matrix with a
message-passing GNN. First, the values in an affinity matrix can be
thought of as unidimensional edge features and, by design, GNNs
are well suited to predict this type of data. Second, a single GNN can
work on graphs with numbers of nodes, which is important when reasoning about varied environments. Third, strategic choices
about what features are encoded in the nodes and edges of a graph
can make GNNs invariant to spatial rotations and translations. This
reduces the amount of pre-processing transformations needed to
analyze a scene in comparison to DANTE.

Prior benchmarks in conversational group detection from still
images [28–30] commonly consider datasets with a limited number
of people, e.g., the Cocktail Party dataset [41] considers six people.
Given the relative simplicity of these datasets, we study group de-
tection performance using the recent MatchNMingle dataset [5]
made available by the Delft University of Technology. MatchNMin-
gle is a multi-sensor dataset of in-the-wild conversations for the
analysis of social interactions. It contains 4446 images of a scene
with up to 15 people per frame, as shown in Figure 1.

3 METHOD

This paper studies conversational group detection: partitioning a
set of people in a scene into non-overlapping clusters representing
interacting groups. Formally, assume that there are \( n \) people in the
scene and let \( P \) be a set of individual feature vectors, \( P = \{ f_k | 1 \leq k \leq n \} \). Then, the groups can be expressed via a clustering
schema \( C : P \rightarrow \{ 1, \ldots, n_c \} \), with \( n_c \) the number of clusters.

Given a dataset \( D \) of \( N \) examples, \( D = \{ (P_i, C_i), \ldots, (P_N, C_N) \} \),
we frame the group detection problem from a supervised learning
perspective as computing a function \( h(P_i) = \hat{C}_i \) that estimates
cluster assignments. Each predicted \( \hat{C}_i \) should be as close as possible
to the true \( C_i \) for all the examples \( i \) in \( D \). Note that in this problem
the number of clusters and people may differ across examples.

3.1 Clustering Conversational Interactants

We propose to construct the function \( h(P_i) = \hat{C}_i \) using a Graph
Neural Network (GNN), followed by the application of the Domin-
ant Sets (DS) algorithm [23]. To this end, we first create a fully-
connected interaction graph that describes the scene, \( G^0 = (N^0, E^0) \),
as illustrated in Figure 1. We assign each feature vector \( f_k \) to node
features \( n^k \) and edge features \( e^k_{jk} \) in the graph, such that:

\[
N^0 = \{ n^k | 1 \leq k \leq |P_i| \}, E^0 = \{ e^k_{jk} | 1 \leq j, k \leq |P_i|, j \neq k \}
\]

The proposed GNN is composed of two graph computation layers,
\( g(\cdot) = g^2(g^1(\cdot)) \). Each layer transforms an input graph \( G^{l-1} \)
to another graph \( G^l \), with \( l \) indicating the \( l \)-th layer without loss of
generality. At the last layer of the GNN, \( e^2_{jk} \in E^2_i \) represents the
pairwise affinity from node \( j \) to node \( k \) in the graph.

Based on the pairwise affinities output by the GNN, we construct
an affinity matrix \( A_l \) for the graph corresponding to the set \( P_l \).
Then, it is then passed through the DS algorithm [23], which iteratively groups
graph nodes into clusters by maximizing the quadratic program
\( \max_{x \in \mathbb{R}^{N}} x^T A_l x \), where \( S \) is the standard simplex in \( \mathbb{R}^{\left|P_l\right|} \). Here,
solutions to the quadratic program represent a group of people, the
dominant set in the input \( A_l \). Note that every iteration of DS reduces
the size of the affinity matrix by removing the data corresponding
to the last group that was predicted by the algorithm.

Oftentimes, there will be individuals in a scene that are not in a
group conversation. However, the peeling-off strategy employed
by DS tends to group together these individuals. To combat this
problem, we use the DS stopping criteria from [15] to consider the
global context of the complete graph when grouping people.

3.1.1 Graph Neural Network. Each computation layer of the pro-
gressed GNN is a graph network block comprised of two updates: one
for the edges and one for the nodes, following the message-passing
architecture described in [2]. If we define the node and edge fea-
tures for layer \( l \) as \( n^l_k \) and \( e^l_{jk} \), respectively, then the graph network
block operates as follows:

\[
e^{l+1}_{jk} = \text{edge_update} \left( e^l_{jk}, n^l_j, n^l_k \right)
\]

(1)

\[
n^{l+1}_k = \text{node_update} \left( n^l_k, \text{agg} \left( \left( e^{l+1}_{jk} | j \neq k \right) \right) \right)
\]

(2)

The edge_update(·) and node_update(·) functions are neural
networks that reason about edge or node features in relation to the
information in their neighborhood in the graph, as shown in Figure
2. The agg(·) function is a symmetric function that summarizes
information in the edge features connected to a given node.

Our motivation for designing our GNN with two graph network
blocks stems from the fact that we consider fully-connected in-
teraction graphs with no self loops in this work. Thus, two graph
network blocks suffice to make the output affinity values dependent
on the information encoded in all the nodes and edges in the graph.
We aggregate these annotations into 600 frames per camera per day. Because there was high variability in group sizes, spacing, proximity data is provided for 71/92 participants, all labeled at 20Hz. In 9 categories for each participant. Triaxial acceleration and binary from 3 cameras with annotated bounding boxes and “social actions” that indicates their occurrence per type over the last 0.5 seconds. Our main evaluation metric is the Group F1 metric.

### 4.2 Group Detection Methods

We consider three methods in our evaluation:

1. **Dist.** Hand-crafted baseline inspired by [15, 39]. The method computes an affinity matrix as $A_{ij} = \exp\left(-\frac{d_{ij}}{2\sigma^2}\right)$, where $d_{ij}$ is the distance between two participants and $\sigma = 2$ meters, following [15]. DS is then applied to obtain groupings, as in [15].

2. **DANTE.** We implement the DANTE neural network [30] in PyTorch and use DS for clustering, as in [15]. The dyad and context MLPs of DANTE had two layers with 32 and 64 units. The final MLP had 64 and 32 units. All but the last layer used a ReLU activation followed by batch normalization.

   For Cocktail Party, DANTE uses position and the orientations as node features, transforming each feature into a coordinate frame centered between each dyad, as in [30]. For MatchNMingle, it uses the position, transformed by the dyadic coordinate frame, and applies the rest of the features without additional transformations.

3. **GNN.** Our proposed combination of a GNN with DS for group detection. We implement the GNN using PyTorch Geometric to leverage sparse tensor computations. For the edge updates, we use two MLPs of dimensions 128, 64 and 32, 16 for each graph network block. The node updates use MLPs of dimensions 32, 16 and 16, 16. As in DANTE, we use ReLU activations and batch norm. These dimensions were chosen to produce a similar number of parameters to the DANTE models for all input feature combinations.

   For Cocktail Party, the GNN uses distance and both angles, transformed into point pair features [9], as edge features. For MatchNMingle, it uses distance for edge features and all other features as node features. When considering position-only features, however, we do not use any node features.

   Both DANTE and the GNN were trained using a learning rate of 1e-4 that decays to 1e-6 over 1000 epochs, a batch size of 512, and the Adam optimizer. Early stopping halted training if there was no decrease in the cross-entropy loss after 50 epochs.

### 4.3 Results

Our main evaluation metric is the Group F1 metric [32]. For a threshold $T$, the Group F1 metric considers a ground truth cluster with $n_g$ people to be correctly identified if at least $\lceil T \cdot n_g \rceil$ members are grouped together by the algorithm and no more than $\lfloor (1-T) \cdot n_g \rfloor$ recording were used for test, the next 10% of frames from beginning and end were used for validation, and the middle 60% were used for training.

We consider 4 types of features for individuals in MatchNMingle:

- **Position features** (pos) include $x$, $y$ coordinates for the corresponding person on a video recording.
- **Acceleration features** (accel) are the last 10 accelerometer readings for a person, covering a time window of 0.5 seconds.
- **Image features** (img) are visual embeddings for the person.
- **Semantic features** (label) encode person actions.

The features are computed by aggregating the actions into a 9-dimensional vector that indicates their occurrence per type over the last 0.5 seconds.
Table 1: Results on Cocktail Party, including average results and std. deviation ($\mu \pm \sigma$) over the test examples. Results in bold are significantly better than those with regular font.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Dist</th>
<th>DANTE</th>
<th>GNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1T1</td>
<td>$0.24 \pm 0.34$</td>
<td>$0.58 \pm 0.43$</td>
<td>$0.62 \pm 0.41$</td>
</tr>
<tr>
<td>F1T2/3</td>
<td>$0.53 \pm 0.32$</td>
<td>$0.71 \pm 0.35$</td>
<td>$0.70 \pm 0.37$</td>
</tr>
</tbody>
</table>

Table 2: Results on MatchNMingle, including average results and std. deviation ($\mu \pm \sigma$) over the test examples. Results in bold are significantly better than those with regular font.

<table>
<thead>
<tr>
<th>Features</th>
<th>Metric</th>
<th>Dist</th>
<th>DANTE</th>
<th>GNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>pos</td>
<td>F1T1</td>
<td>$0.28 \pm 0.26$</td>
<td>$0.32 \pm 0.28$</td>
<td>$0.30 \pm 0.26$</td>
</tr>
<tr>
<td>pos</td>
<td>F1T2/3</td>
<td>$0.38 \pm 0.29$</td>
<td>$0.43 \pm 0.30$</td>
<td>$0.40 \pm 0.28$</td>
</tr>
<tr>
<td>pos+accel</td>
<td>F1T1</td>
<td>-</td>
<td>$0.24 \pm 0.25$</td>
<td>$0.34 \pm 0.27$</td>
</tr>
<tr>
<td>pos+accel</td>
<td>F1T2/3</td>
<td>-</td>
<td>$0.30 \pm 0.28$</td>
<td>$0.43 \pm 0.28$</td>
</tr>
<tr>
<td>pos+img</td>
<td>F1T1</td>
<td>-</td>
<td>$0.28 \pm 0.29$</td>
<td>$0.31 \pm 0.28$</td>
</tr>
<tr>
<td>pos+img</td>
<td>F1T2/3</td>
<td>-</td>
<td>$0.34 \pm 0.29$</td>
<td>$0.40 \pm 0.29$</td>
</tr>
<tr>
<td>pos+accel+img</td>
<td>F1T1</td>
<td>-</td>
<td>$0.23 \pm 0.24$</td>
<td>$0.32 \pm 0.28$</td>
</tr>
<tr>
<td>pos+accel+img</td>
<td>F1T2/3</td>
<td>-</td>
<td>$0.28 \pm 0.26$</td>
<td>$0.42 \pm 0.29$</td>
</tr>
</tbody>
</table>

We presented an approach to predict conversational clusters in social scenes, where the number of clusters is unknown a priori. Our results indicate that GNNs can better take advantage of multi-modal data for group detection in comparison to baselines. In particular, the proposed GNN-based model outperformed the previous state-of-the-art approach [30] on the complex MatchNMingle dataset with all types of data except position-only, while requiring less data pre-processing. This suggests that leveraging relational inductive biases in data-driven methods for group detection is beneficial.

5 LIMITATIONS & FUTURE WORK

We demonstrated the successful application of GNNs to group detection. In principle, the inductive nature of the proposed approach allows our method to run in an online fashion, processing streams of data. However, more tests are needed to verify this in practice. Future work could also evaluate the GNN on other group detection datasets, like CoffeeBreak [7] or Salsa [1].

Unexpectedly, DANTE and the proposed GNN did not benefit from the added image features in the MatchNMingle dataset. Further, in the case of DANTE, performance tended to decrease with more features. There are several possible explanations for this phenomenon. First, the image features from the ResNet [20] model could have been too deep in the network. Low-level features from earlier in the network could be used to fix this issue. Second, the MatchNMingle cameras have visible radial distortion, which we did not correct for because the intrinsic camera parameters are not public. Lastly, the performance drop could be due to challenges combining feature modalities. In this respect, future work could explore using attention mechanisms to fuse data, e.g., as in [24, 33, 37].

We processed the data from different cameras in the MatchNMingle dataset as independent samples, although some of them contained information captured at the same time from different views. Likewise, we did not consider the temporal correlation of data across dataset samples, but this information could improve model prediction [32, 35]. Thus, future work could explore detecting groups across multiple camera views to understand more holistically the environment, and combining GNNs with recurrent neural networks to take advantage of temporal correlations, e.g., as in [27].

6 CONCLUSION

We presented an approach to predict conversational clusters in social scenes, where the number of clusters is unknown a priori. Our results indicate that GNNs can better take advantage of multi-modal data for group detection in comparison to baselines. In particular, the proposed GNN-based model outperformed the previous state-of-the-art approach [30] on the complex MatchNMingle dataset with all types of data except position-only, while requiring less data pre-processing. This suggests that leveraging relational inductive biases in data-driven methods for group detection is beneficial.

ACKNOWLEDGMENTS

Portions of the research in this paper used the MatchNMingle Dataset made available by the Delft University of Technology, Delft, The Netherlands. This work was supported by the National Science Foundation (NSF), Grant No. (IIS-1924802). The findings and conclusions in this paper are those of the authors and do not necessarily reflect the views of the NSF. The authors are also thankful to the Yale Hahn Scholars program for supporting A. W. Gupta.
REFERENCES


