MGpi: A Computational Model of Multiagent Group Perception and Interaction

Navyata Sanghvi
Carnegie Mellon University
Pittsburgh, Pennsylvania
navyatasanghvi@cmu.edu

Ryo Yonetani∗
Carnegie Mellon University
Pittsburgh, Pennsylvania
ryo.yonetani@sinicx.com

Kris Kitani
Carnegie Mellon University
Pittsburgh, Pennsylvania
kkitani@cs.cmu.edu

ABSTRACT
Toward enabling next-generation robots capable of socially intelligent interaction with humans, we present a computational model of interactions in a social environment of multiple agents and multiple groups. The Multiagent Group Perception and Interaction (MGpi) network is a deep neural network that predicts the appropriate social action to execute in a group conversation (e.g., speak, listen, respond, leave), taking into account neighbors’ observable features (e.g., location of people, gaze orientation, distraction, etc.). A central component of MGpi is the Kinesic-Proxemic-Message (KPM) gate, that performs social signal gating to extract important information from a group conversation. In particular, KPM gate filters incoming social cues from nearby agents by observing their body gestures (kinesics) and spatial behavior (proxemics). The MGpi network and its KPM gate are learned via imitation learning, using demonstrations from our designed social interaction simulator. Further, we demonstrate the efficacy of the KPM gate as a social attention mechanism, achieving state-of-the-art performance on the task of group identification without using explicit group annotations, layout assumptions, or manually chosen parameters.

KEYWORDS
Social agent models; Socially interactive agents; Agent-based analysis of human interactions; Social group identification; Multiagent learning; Learning from demonstrations.

ACM Reference Format:

1 INTRODUCTION
In order to develop next-generation robots that can interact socially with humans, there is a first need for expressive computational models that can encode social intelligence [2, 19, 29, 59]. Broadly, an agent’s social intelligence is its ability to understand and respond appropriately to others, i.e., its: (1) social perception and (2) social interaction management skill. Social perception is the ability to analyze other agents’ social signals including non-verbal behavioral information such as facial or postural expressions [41, 47], physical distance (proxemics) [21] and body gestures (kinesics) [4]. Social interaction management is the ability to take appropriate action in response to information retrieved via social perception.

Toward the ultimate goal of robotic social intelligence, we propose a computational model of social interaction management based on social perception of behavioral signals in the context of multi-agent multigroup conversation. Consider a crowded social party like the ones depicted in Figure 1 to understand some important components of social intelligence.

(1) Group Formation and Social Signal Gating. When many people are gathered together, it is natural for people to form smaller groups and engage in conversation. In order to participate effectively in a group conversation, each person needs to be able to pay attention to the people in the group, while suppressing information (both verbal and non-verbal) from other nearby groups. This type of social attention mechanism, which we term social signal gating, is a critical aspect of social perception and is closely tied to group identification (discussed later).

(2) Dynamic Group Size and Influence. At a social party, people dynamically move from group to group to start new conversations and leave old ones. The ability to adapt to dynamic group size and a dynamic degree of influence from nearby people is an important aspect of social intelligence.
(3) **Short-Term Memory.** Communication within a group is successful when there is an appropriate balance of conversational actions. At every moment, each group member decides their future conversational actions based on the past course of interaction (e.g., speak, respond or leave). For example, if one person has been talking very long, group members might become disinterested and disengage from the group. In other words, a social agent’s short-term memory of past non-verbal and conversational interactions with neighbors is influential in determining its future social interactions.

![Figure 2: Overview of MGpi Network](image)

The example above highlights critical features of a multiagent multigroup interaction model. Keeping in mind these features, we propose **MGpi**, a Multiagent Group Perception and Interaction Network. As shown in Figure 2, MGpi is a deep neural network consisting of the following modules:

1. **Social Signal Gating Module** expresses appropriate social perception by processing incoming and past social signals from neighboring agents (both non-verbal and conversational). It decides their degree of influence using a social attention function, the Kinesic-Proxemic-Message (KPM) gate. To enable reasoning with a dynamic number of influencing neighbors, it uses a pooling operation to aggregate encoded social signals.

2. **Short-term Memory (STM) Encoders** encode the agent’s short-term memory of past interactions with each neighbor (Social-STM) and its own actions (Self-STM).

3. **Interaction Policy Module** processes outputs from aforementioned components to decide the agent’s next conversational action.

The MGpi network enables us to model social agents that make action choices in a decentralized manner. The parameters of MGpi are learned end-to-end via imitation learning, using a set of demonstrated group interaction sequences. In this work, demonstrations are generated by a social interaction simulator, but MGpi can also be learned from annotated real-world social interactions. Since modeling the full extent of social intelligence is very complex (e.g., power, dialogue, trust), in this work, we use abstracted expressions of social interaction (e.g., discrete conversational actions). We believe this simplification is a necessary step towards designing more complex models of real-world social interactions.

As mentioned earlier, social perception is closely tied to group identification. Thus, we further hypothesize that, if our proposed social signal gating module is modeled correctly, we will be able to map the learned social attention directly to group membership. We demonstrate the ability of the KPM gate to identify groups in a direct, unsupervised manner, achieving state-of-the-art performance against group detection methods that use hand-defined parameters, features and complex spatial assumptions on how people arrange themselves in social situations (F-formations [34]).

In summary, our contributions are as follows:

1. We present MGpi, a computational model of multiagent multigroup social interactions (Section 3).

2. In a dearth of prior work on multiagent multigroup social interaction simulators and in the absence of real-world datasets with multigroup behavioral annotations (e.g., listen, respond, strongly address), we design our own social interaction simulator, seeking inspiration from several works which study the multi-modal nature of small-group conversational dynamics [27, 28, 30, 33, 40]. We use this simulated data to train and evaluate our network (Section 5).

3. We demonstrate how the ability of group identification emerges as a result of learning a social interaction policy, achieving competitive performance against state-of-the-art methods with the explicit aim of detecting groups. Unlike prior work, we do so without the use of explicit group annotations, layout assumptions or hand-defined parameters (Sections 4, 6).

![Figure 3: Components of MGpi Network](image)
2 RELATED WORK

The analysis of intelligent agent grouping, behavior and communication is of broad interest to many disciplines including Human-Robot Interaction [6, 7, 10, 23, 46, 47, 61], Multiagent Systems and Machine Learning [5, 9, 24, 25, 36, 38, 43, 44, 48, 54, 55, 60], Neurobiology [8, 13, 18, 26, 31, 52], Computer Vision [3, 16, 17, 50, 51, 57, 62] and Psychology [2, 4, 20, 21, 41, 42]. Since this work is focused on developing computational models of multiagent communication and identifying social groups, we elaborate on selected related work.

Human-Robot Interaction: The effects of gaze and proxemics in human-robot interaction and social signal perception has been widely explored [23]. Recognizing these non-verbal cues’ social importance, we incorporate the perception of such cues as a central component in our model. Dialogue systems work [49, 53] concentrates on appropriately responding to conversational cues, while our work additionally deals with the non-verbal aspects of human communication. Detection and automatic analysis of a human’s engagement in social human-robot scenarios has been comprehensively studied [10, 46, 47, 61] using non-verbal cues such as body attitudes, facial video signals, and personality traits. Related to this, our work presents a method to automatically infer an interacting agent’s attention from non-verbal and conversational cues. Sociable robots have been defined and studied in the context of their degree of interaction with humans and their functionality [6, 19]. Toward enabling socially intelligent robots, we present a model of multiagent multigroup interaction. The importance of simulation theory and imitative interaction in the social understanding of neighbors by robots has been studied [7]. We use this insight in order to learn a computational model for socially intelligent robots through imitation of simulated interactions.

Multiagent Systems and Machine Learning: Emergent collective behaviors have been studied in multiagent organized swarms [5]. By contrast, we model individual behavior in a socially intelligent multigroup setting. Adversarial deceptive interactions have been studied between individuals [60] and groups of intelligent robots [48]. Instead, we assume a cooperative scenario, where interactions imitate multigroup human-to-human social reactions. Modeling multiagent communication is an active area of research in machine learning [9, 54]. Recent work makes use of deep reinforcement learning or imitation learning to discover policies for agent-to-agent communication, e.g., [24, 25, 36, 38, 43, 44, 55]. The motivation for these works is often collaboratively solving a particular task which might involve partial observability. Communication protocols best fit to accomplish that task are learned. By contrast, we learn to imitate social communication protocols of humans in a multigroup scenario. Some works assume the communication of internal states between agents, while others learn to communicate a particular ‘message’ in order to solve the task. Neighbor messages might be aggregated using a pre-defined pooling operation. By contrast, we explicitly encode particular social signals between agents and learn signal gating to appropriately weight incoming signals, showing superior performance to prior pooling strategies.

Neurobiology and Signal Gating: The concept of social signal gating introduced in our model is inspired by the neurobiological process of sensory gating [18, 26]. Sensory gating is the ability to filter out unnecessary or irrelevant external stimuli. The cocktail party effect [8] is one example of auditory sensory gating where a person is able to focus auditory attention to a specific target while ignoring other irrelevant audio input. Similar sensory gating is observed in other senses as well, to prevent overwhelming the primary cortical areas [15, 31, 52]. Recognizing the importance of modelling sensory gating in socially intelligent computational systems, we use our KPM gate to mimic this phenomenon when considering the importance of nearby agents’ social signals.

Social Group Interactions: For the purpose of evaluating our model, there are no large-scale public datasets of states and conversational actions of people interacting in multiple groups. Existing datasets [15, 45, 62] only provide sequences of physical positions, gaze directions and group assignments. More importantly, no conversational action annotations (e.g., listening, speaking, etc) are provided. Closely related to our ideas, a social simulator PsychSim [39] adopts a theory-of-mind approach to modelling interactions in scenarios with agent attributes such as power and hardship. By contrast, we consider scenarios of face-to-face interactions in social situations, and require agents to have conversational roles such as responding to being addressed. Further, we aim to automatically learn the levels of reasoning required to behave appropriately in a social situation. In the absence of prior work on simulators or real-world datasets with conversational action annotations, we design our own social interaction simulator (Section 5). Inspired by several works which study the multi-modal nature of small-group conversational dynamics [27, 28, 30, 33, 40], we design simulated interaction rules in order to generate data to train our network.

Social Group Identification: There is much work on identifying the members of a group in a social situation [52, 50, 51, 58], and more recently [56]. This prior work typically assumes agents’ arrangement in spatial layouts called F-formations [34], often attempting to find o-space centers [34] using heuristic strategies and hand-crafted parameters. By contrast, under no such layout assumptions, we learn parameters for a straightforward, automatic group identification mechanism (Section 6).

3 SOCIAL INTERACTION MANAGEMENT: THE MGPI NETWORK

Our goal is to design a computational model for multiagent multigroup interactions that incorporates signal gating, adaptation to dynamic group sizes and short-term memory encoding. Towards this goal, we propose the MGPI architecture (Figures 2, 3). In our design of MGPI, we consider several insights from social science literature: the importance of ‘situational awareness’ and ‘presence’ [2], and the impact of non-verbal social cues ‘kinesics’ [4] and ‘proxemics’ [21].

Notation. We denote the m-th agent by and its neighbors at time t by \( \{A^{n_i}\}_{i=1}^{J_i} \). At time t, \( A^{m} \) and its neighbor \( A^{n_i} \) have gaze directions \( g^{m} \), \( g^{n_i} \) \( \in \mathbb{R}^{2} \), and positions \( I^{m} \), \( I^{n_i} \) \( \in \mathbb{R}^{2} \). \( R(\phi) \) is the rotation matrix associated with angle \( \phi = \arctan(g^{m}) \). Relative gaze direction and relative rotated position of \( A^{n_i} \) w.r.t. \( A^{m} \) are respectively given by:

\[
\dot{g}_{i}^{(n_i \rightarrow m)} = R(\phi)g^{n_i}_{i} \quad \text{and} \quad \dot{I}_{i}^{(n_i \rightarrow m)} = R(\phi)(I^{n_i} - I^{m}).
\]

In our model, we use a past history of features. The history length (horizon) is denoted as H. The histories of relative gaze directions and relative rotated positions of \( A^{n_i} \) w.r.t. \( A^{m} \) are respectively given by:

\[
\mathcal{G}^{(n_i \rightarrow m)} = R(\phi)g^{n_i}_{i} \quad \text{and} \quad \mathcal{I}_{i}^{(n_i \rightarrow m)} = R(\phi)(I^{n_i} - I^{m}).
\]
### 3.1 Social Signal Gating Module

Conceptually, the Social Signal Gating (SSG) Module allows the agent to select only the most relevant social signals coming from other agents. Computationally, the SSG generates an encoding for the part of available information that depends on the neighboring agents (the part that depends on itself is described later). As shown in blue in Figure 3, SSG consists of three computational units: (1) Social Short-Term Memory Encoder, (2) Kinetic-Proxemic-Message Gate and (3) Signal Pooling.

**Social Short-Term Memory Encoder (Social-STM).** For agent \( A^m \), at any time \( t \), the role of Social-STM is to generate an encoding for each neighbor \( A^{ni} \)'s social signals. As shown in red in Figure 3, the Social-STM contains two encoders. (1) Non-Verbal Encoder \( N \) encodes a history of relative gaze directions and rotated positions \( \rho_{ni}^{t, hist} \). (2) Convolutional Encoder \( C \) encodes a history of the neighbor's conversational actions \( D_{ni}^{t, hist} \).

Non-Verbal Encoder \( N (\rho_{ni}^{t, hist}) \) summarizes \( A^{ni} \)'s observations of neighbor \( A^{ni} \)'s non-verbal communication history. Convolutional Encoder \( C (D_{ni}^{t, hist}) \) summarizes \( A^{ni} \)'s observations of \( A^{ni} \)'s past conversation. As shown in Figure 3, \( N \) and \( C \) are both linearly-activated gated recurrent units (GRU) [11]. Each encoder learns the associated temporal dynamics and outputs neighbor 'messages' \( N \in \mathbb{R}^d \) and \( C \in \mathbb{R}^d \). \( J_t \) pairs of messages are output, one pair for each neighbor. Next, these messages are weighted by the KPM gate's importance score, pooled and passed to the interaction policy.

**Kinesic-Proxemic-Message (KPM) gate.** The role of the KPM gate is to decide how much attention to give to each neighboring agent's concatenated social signal \( [N (\rho_{ni}^{t, hist}) \cdot C (D_{ni}^{t, hist})] \). The KPM gate \( K (g_{ni}^{t, hist}) \) is a function of the relative rotated position and orientation between \( A^m \) and neighbor \( A^{ni} \) at time \( t \). The KPM gate is a multi-layer perceptron, consisting of two feed-forward layers, respectively activated by an exponential linear unit (ELU) [12] and a hard-sigmoid function [14]. For each neighbor \( A^{ni} \), it outputs a scalar 'importance' weight \( K \in [0, 1] \), reflecting the neighbor’s degree of influence on \( A^m \)'s next conversational action.

**Signal Pooling.** The role of the signal pooling operator is to aggregate the weighted social signals for all \( J_t \) neighbors of \( A^m \) at time \( t \). As shown in Figure 3, each neighbor \( A^{ni} \)'s concatenated social signal is weighted by the KPM gate's respective importance score to obtain:

\[
x_{ni}^{t, hist} = K (g_{ni}^{t, hist}) \cdot N (\rho_{ni}^{t, hist}) \cdot C (D_{ni}^{t, hist})
\]

Next, similar to [55], we employ average pooling of neighbors' weighted messages. The pooled message \( x^m_t = (1/J_t) \sum_{i=1}^{J_t} x_{ni}^{t, hist} \) is passed to the interaction policy \( \pi \), along with the Self Short-Term Memory Encoder message (described next), to decide \( A^m \)'s next conversational action.

### 3.2 Self Short-Term Memory Encoder (Self-STM)

The role of the Self-STM is to encode the agents recollection of her own past actions at time \( t \). Similar to the Conversational Encoder, the Self-STM \( C' \) encodes a history of the agent \( A^m \)'s own conversational actions \( D_{t, hist}^m \). As shown in yellow in Figure 3, \( C' (D_{t, hist}^m) \) is a linearly-activated GRU, yielding a self 'message' \( C' \in \mathbb{R}^d \). This is concatenated with the pooled neighbor message \( x^m_t \) and passed to the interaction policy to decide \( A^m \)'s next conversational action.

### 3.3 Interaction Policy

The final module of MGpi is the interaction policy \( \pi \) which, at time \( t \), decides the agent's next action based on the agent’s observations of its neighbors and itself. Formally, the policy \( \pi \) takes as input the concatenated outputs \( [x^m_t, C' (D_{t, hist}^m)] \) from the SSG Module and the Self-STM (shown in green in Figures 2, 3). The interaction policy \( \pi \left( \left[ x^m_t, C' (D_{t, hist}^m) \right] \right) \) is modeled as two fully-connected layers, respectively activated by an ELU and soft-max function. It yields a \(|\mathcal{U}| \)-dimensional output, representing a probability distribution over the agent \( A^m \)'s next conversational action.

### 4 IMITATION LEARNING AND GROUP IDENTIFICATION

By modeling the MGpi Network’s modules \( N, C, G, C' \), and \( \pi \) as fully-differentiable functions, we ensure that it is trainable end-to-end via back-propagation, using demonstrations of multiagent multigroup state and action sequences.

---

1. Our code may be found here: https://github.com/navysanghvi/MGpi
The distance is simply computed from the weighted average of the influence on a neighbor is to be in the same group as the agent. Therefore, no public datasets of large-scale demonstrations of the physical movement of people (6-12 per layout) are organized into several groups (Fig. 1). Multiple Break [15], and (c) Cocktail Party [62] datasets to provide 100, 119, and 320 non-identical, independent layouts respectively. Multiple people (6-12 per layout) are organized into several groups (Fig. 1).

**Group Identification.** Once the MGpi network parameters are learned, we use the KPM gate activations to identify groups. In order to identify groups, at each time step, we define a distance \( D(n, j) \) between two agents \( n, j \) based on the output of learned gating function \( K: D(n, j) = 1 - \frac{1}{2}(K(g^{(j-n)}, l^{(j-n)}) + K(g^{(n-j)}, l^{(n-j)})) \). The distance is simply computed from the weighted average of the bi-directional weights computed by the KPM gate. We compose an affinity matrix of pairwise distances between all agents and run the DBSCAN clustering algorithm [22] to cluster people into conversational groups. As mentioned in our Introduction, this is a relatively simple and straightforward approach. It is compared in Section 6 with state-of-the-art approaches based on complex o-space estimation under assumptions of particular spatial layouts called F-formations [34].

**5 EXPERIMENTS: SOCIAL INTERACTION SIMULATOR**

**Conversational Action Data.** We aim to learn computational models that can encode the kinesics and proxemics in group formation and conversational actions (e.g., responding, distraction, etc) in multiagent multigroup scenarios. Unfortunately, there are no public datasets of large-scale demonstrations of the physical positions, gaze directions, group assignments, and communication actions of multiple socially interacting people. Existing datasets [15, 45, 62] only provide sequences of the physical positions, gaze directions and group assignments (but annotated only sparsely). More importantly, no conversational action annotations are provided. Similarly, there are no social interaction simulators for the kind of face-to-face behavioral actions of interest to us. In order to evaluate the capacity of our proposed model, we simulate multigroup human communication, drawing inspiration from several prior studies and observations of small-group conversational dynamics [27, 28, 30, 33, 40]. While we simulate sequences of non-verbal interactions, we point out that the physical layouts of agents are from real-world datasets.

**Initial Agent Layout Data:** We experiment with both artificially constructed and real-world data to form our initial multigroup layouts - specifically, we use (a) Synthetic [15], (b) Coffee Break [15], and (c) Cocktail Party [62] datasets to provide 100, 119, and 320 non-identical, independent layouts respectively. Multiple people (6-12 per layout) are organized into several groups (Fig. 1).

**Rules of Interaction:** To design multigroup communication protocols, we draw inspiration from several observations of small-group conversational dynamics in literature, namely: speaking and turn-taking, [27], addressing [33], interest in meetings [28, 30], and the relation of speaking time and dominance [40]. We design per-group rules for two scenarios: (1) **Static scenario**, where agents stay within a social group and do not transition to other groups. Six conversational actions are considered \( \mathcal{U}_{\text{stat}} = \{\text{Speaking, Listening, Distracted, Strongly Addressing, Weakly Addressing, Responding}\} \). (2) **Dynamic scenario**, where agents transition from group to group, leading to a temporally evolving group assignments. Seven conversational actions are considered \( \mathcal{U}_{\text{dyn}} = \mathcal{U}_{\text{stat}} \cup \{\text{Moving}\} \).

---

**Figure 4:** Examples of simulated communications (static two agent case)

**Figure 5:** Example physical layout [15] and subsequent conversational interaction. Arrows represent gaze directions, head colors indicate conversational action - red: speaking, yellow: distracted, green: listening.
While we recognize the gap between reality and our simulations, with a certain probability. After some time steps, we believe that, in the absence of existing data or other simulators, MGpi speaks for some time, it yields to the other. The other agent is then of the Distracted and 3

verbal actions. The same rules are applied to every group in stead, MGpi

in a dynamic scenario, a group member who is Might transition to weakly Addressing, followed by Listening (Figure 4(d)). The Speaking agent might transition to Weakly Addressing (Figure 4(e)). Additionally, in a dynamic scenario, a group member who is Distracted may start Moving toward another group. When a Moving agent joins a new group, it is welcomed as the next speaker.

Baselines. We now describe experiments and results of applying the MGpi network to predict communication actions of simulated social agents. We compare MGpi with the following baselines, including communication models used in prior work:

### Table 1: Results for Static Scenario: Action Prediction (mean average precision), and Test Loss (cross-entropy)

<table>
<thead>
<tr>
<th>Model (mAP’s)</th>
<th>J = 2</th>
<th>4</th>
<th>8</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbor States Only (NSO)</td>
<td>0.67</td>
<td>0.69</td>
<td>0.70</td>
<td>0.68</td>
</tr>
<tr>
<td>Self State Only (SSO)</td>
<td>0.76</td>
<td>0.76</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>Equal Pooling (EQPOOL)</td>
<td>0.87</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>Social Pooling (SOCPOOL)</td>
<td>0.85</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
</tr>
<tr>
<td><strong>MGpi Network</strong></td>
<td>0.88</td>
<td>0.88</td>
<td>0.89</td>
<td>0.88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model (cross-entropy losses)</th>
<th>J = 2</th>
<th>4</th>
<th>8</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbor States Only (NSO)</td>
<td>0.99</td>
<td>0.95</td>
<td>0.89</td>
<td>1.00</td>
</tr>
<tr>
<td>Self State Only (SSO)</td>
<td>0.29</td>
<td>0.29</td>
<td>0.29</td>
<td>0.29</td>
</tr>
<tr>
<td>Equal Pooling (EQPOOL)</td>
<td>0.21</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
</tr>
<tr>
<td>Social Pooling (SOCPOOL)</td>
<td>0.21</td>
<td>0.22</td>
<td>0.22</td>
<td>0.25</td>
</tr>
<tr>
<td><strong>MGpi Network</strong></td>
<td>0.20</td>
<td>0.21</td>
<td>0.20</td>
<td>0.21</td>
</tr>
</tbody>
</table>

Based on observations from literature described above, we design per-group probabilistic rules conditioned on a temporal history of past group actions, to decide the evolution of each agent’s conversational actions. The same rules are applied to every group in the scene. We simulate 600-step interaction episodes starting from each set of initial layouts ((a),(b) and (c) above) to obtain training data. Figure 5 shows an example of initial multigroup layout and subsequent interaction.³

We reiterate: group assignment information is used only for evaluation and not for supervisory training signals to MGpi. Instead, MGpi implicitly identifies groups using the KPM gate. We hypothesize that our automatic KPM gating mechanism will learn to discover relevant neighbors in both static and dynamic scenarios. While we recognize the gap between reality and our simulations, we believe that, in the absence of existing data or other simulators, this is a necessary first step towards understanding how to model multiagent multigroup interaction.

### Two-Agent Simulation Example. Using Figure 4, we illustrate our rules in a simple two-agent case. In the static scenario, as shown in Figure 4(a), one agent is Speaking while the other is Listening. The Listening agent may become Distracted (Figure 4(b)) with a certain probability. After some time steps, the Speaking agent may start Strongly Addressing the group, to draw back the attention of the Distracted member (Figure 4(c)). Once the Speaking agent has spoken for some time, it yields to the other. The other agent is then Responding followed by Speaking (Figure 4(d)). The Speaking agent might transition to Weakly Addressing (Figure 4(e)). Additionally, in a dynamic scenario, a group member who is Distracted may start Moving toward another group. When a Moving agent joins a new group, it is welcomed as the next speaker.

### Baselines. We now describe experiments and results of applying the MGpi network to predict communication actions of simulated social agents. We compare MGpi with the following baselines, including communication models used in prior work:

³Example simulator videos may be found here: https://github.com/navysanghvi/MGpi

<table>
<thead>
<tr>
<th>Model (mAP’s)</th>
<th>J = 2</th>
<th>4</th>
<th>8</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbor States Only (NSO)</td>
<td>0.57</td>
<td>0.62</td>
<td>0.65</td>
<td>0.64</td>
</tr>
<tr>
<td>Self State Only (SSO)</td>
<td>0.78</td>
<td>0.78</td>
<td>0.78</td>
<td>0.78</td>
</tr>
<tr>
<td>Equal Pooling (EQPOOL)</td>
<td>0.85</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>Social Pooling (SOCPOOL)</td>
<td>0.85</td>
<td>0.85</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td><strong>MGpi Network</strong></td>
<td>0.87</td>
<td>0.87</td>
<td>0.88</td>
<td>0.88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model (cross-entropy losses)</th>
<th>J = 2</th>
<th>4</th>
<th>8</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbor States Only (NSO)</td>
<td>1.24</td>
<td>1.12</td>
<td>1.05</td>
<td>1.07</td>
</tr>
<tr>
<td>Self State Only (SSO)</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>Equal Pooling (EQPOOL)</td>
<td>0.26</td>
<td>0.26</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>Social Pooling (SOCPOOL)</td>
<td>0.27</td>
<td>0.27</td>
<td>0.25</td>
<td>0.26</td>
</tr>
<tr>
<td><strong>MGpi Network</strong></td>
<td>0.25</td>
<td>0.23</td>
<td>0.25</td>
<td>0.22</td>
</tr>
</tbody>
</table>

**Neighbor State Only (NSO):** We omit Self-STM Encoder Cᵣ so as to only use cues from neighbors. This helps us test whether observations of the self are necessary for modeling social interaction.

**Self State Only (SSO):** We omit the Social Signal Gating Module so as to use cues of the agent’s self alone. This helps us test whether observations of nearby agents’ social signals are necessary for modeling social interaction.

**Equal Pooling (EQPOOL):** Prior work studies the role of message broadcasting in scenarios where one agent simply averages messages it receives from other agents [24, 43, 55]. We implement this message broadcasting baseline by omitting the KPM gate K so that all neighbor encoded ‘messages’ N and C are equally weighted, such that \( x̃_t = \left( p_{t, hist} \right) C \left( D_{t, hist} \right) \). This helps us test if equal social perception to neighbor agents is sufficient for modeling social interaction.

**Social Pooling (SOCPOOL):** We adopt social pooling [1, 37] as a second message broadcasting baseline. It uses a grid that divides the 2D world into non-overlapping regions and pools messages by region. In our experiments, we use a 4 × 4 grid around target locations, each of size 50 × 50 pixels (this configuration yielded best performance). For each region, we average \( x̃_t \) over neighbors \( A^\theta \) located in that region. For this baseline, the KPM gate and original Signal Pooling Mechanism of MGpi are replaced by an operator that performs grid-wise pooling and concatenates pooled messages. We expect this baseline to learn to implicitly gate messages by down-weighting messages in far away grids.

**Training/Evaluation Scheme.** Demonstrations of multigroup communication were collected as described previously. For each layout in each dataset (Synthetic, Coffee Break and Cocktail Party), agents changed modes 600 times, i.e., \( T = 600 \). We split demonstrations into two subsets with equal number of layouts from each dataset, and learned models on one subset to test the other. All metrics are averaged over the two test subsets for two-fold cross-validation. All models were trained for 30 epochs with mini-batches of size 4096, various numbers of agents \( J ∈ \{ 2, 4, 8, 12 \} \), and a history window \( H = 15 \), and we confirmed that neither further
training nor smaller mini-batches improved performance. Each network was trained to minimize categorical cross entropy (also called negative log likelihood) between predicted probabilities of actions \{\{U_{i-1}\}_{i=2}^T\} and actual demonstrated actions \{\{U_{i-1}\}_{i=2}^T\} via Adam [35]. We report the cross-entropy loss on testing subsets. As every model assigns maximum probability to one of six (static scenario) or seven (dynamic scenario) actions, we also evaluate the model on mean average precision (mAP) over actions.

**Results.** Tables 1 and 2 show all model performances in communication action prediction, with \(H = 15\), and various numbers of neighbors \(J\). We emphasize that \(J\) is only fixed during training; since a single Social-STM Encoder is learned, any number of neighbors may be considered at every time step during testing - in fact, different numbers of neighbors may be considered for each agent in the scene.

As hypothesized, MGpi effectively learns protocols in both static and dynamic scenarios, outperforming strong baselines in terms of both mean average precision (mAP) and test loss, validating our choices of various encoding modules and demonstrating each one’s necessity. Confusion matrices of action prediction for all networks (omitted here for space) show that NSO and SSO work complementarily: NSO predicts better if agents make strong or weak addressing decisions by observing others. On the other hand, SSO performs better when observation of the self is necessary, i.e., in order to predict if agents keep speaking, listening, are distracted or responding. MGpi outperforms baselines for most actions, both in the static and dynamic case, with lower ambiguity.

![Figure 6: Social Attention of KPM Gate: For various positions of a neighbor \(A^n_i\) (e.g., any blue figure) in a fine grid around \(A^m\) (magenta figure, black circles), warmer colors indicate higher outputs from the learned KPM gate \(K\). Gaze directions \(g^n_i = [-1, 0]\), \(g^m_i = [+1, 0]\). History windows \(H \in \{5, 10, 15\}\).](image)

**Learned Social Attention.** Figure 6 represents physical constraints learned by the MGpi Network’s KPM gate \(K\) to judge if neighbor agent \(A^n_i\) is in the same group as agent \(A^m\). These visuals are obtained after training in the static scenario. We visualize the output from \(K\) at various locations around \(A^m\) while fixing relative gaze \(g^{(n_i-m)}\). As expected, we see improved learning with larger history windows \(H\), and choose \(H = 15\) in our experiments. We confirm that increasing the history window does not improve performance, suggesting that, once trained on real-world scenarios, MGpi might be able to reflect the optimal length of short-term memory exhibited in humans. Also as expected, we see a higher output from our KPM gate when agents are looking toward each other, i.e., when the neighbor \(A^n_i\) is toward the left of the image, looking in +x direction, i.e., toward \(A^m\), who is looking in -x direction.

**6 EXPERIMENTS: GROUP IDENTIFICATION IN REAL DATA**

We have demonstrated the effectiveness of the MGpi network in learning group communication policy with simulated social agents. We now test the effectiveness of our learned automatic social signal gating mechanism, the KPM gate, in identifying communication groups in real-world data [15, 62].

**Training MGpi.** Similar to Section 5, we train MGpi using a dataset of simulated interactions from a mixture of initial layouts from the Synthetic [15], Coffee Break [15] and Cocktail Party [62] datasets. For the sake of equal representation in our training data, we use 100 initial layouts from each dataset, for a total of 300 demonstrations of \(T = 600\) steps each. Each dataset has several continuous frames with various numbers of people annotated with positions, orientations, and group assignments (Fig. 1). MGpi is trained for 20 epochs, in both static and dynamic scenarios, setting \(J = 12\), which is sufficiently larger than group membership (6 at most) in the datasets. This ensures that the network must learn to filter out irrelevant social cues from non-group agents, strengthening the contribution of the KPM gate to good network performance. Further, this enhances the chances that KPM gate’s learned social attention is effective for group identification.

**Unsupervised Group Identification.** As described in Section 4, once MGpi is trained, we define a distance \(D(n, j)\) between two agents \(n, j\) based on the output of learned gating function \(K\): \(D(n, j) = 1 - \frac{1}{2}(K(g^{(j-n)}_i, l^{(j-n)}) + K(g^{(n-j)}_i, l^{(n-j)}))\). For every initial layout in real-world datasets Synthetic, Coffee Break, and Cocktail Party [15, 62], we use this simply computed distance measure to form an affinity matrix of pairwise distances between agents in the scene. Using this affinity matrix, we run the DBSCAN clustering algorithm [22] to cluster people into conversational groups.

**Evaluation.** Group detection performance is measured by the F1 score (harmonic mean of Precision and Recall) under the \(|G|\) condition [15, 50]. Per frame, a group is judged to be detected correctly if all constituent people are grouped into a single cluster. Precision, Recall and F1 scores are averaged over frames for each dataset (Synthetic, Coffee Break, Cocktail Party) separately, as well as over all datasets. Due to the stochastic nature of training MGpi, we show average and standard deviation of KPM gate’s performance over 5 training runs.

**Comparisons.** We define a pose-only baseline, which runs DBSCAN using actual euclidean distances between agents in each scene, scaled by the maximum such distance. Unlike MGpi and other state-of-the-art methods, this does not use gaze or body orientation information. In addition to this simple baseline, several state-of-the-art group identification methods DS[32], HVFG-ms [50], GCFF [51], GPRG [58] are chosen for comparison. These methods are based on complex o-space estimation under assumptions of particular spatial layouts called F-formations [34], often using heuristic parameters. By contrast, KPM gate offers a direct, unsupervised approach to group identification.

**Results.** From Table 3 we see that our learning process of the KPM gate is very stable and has very low standard deviation across training runs. Our simple method takes a few milliseconds to run and could be used for real-time group identification. On the Synthetic dataset, our approach significantly outperforms all state-of-the-art
methods in terms of all metrics. On the Coffee Break dataset, our approach outperforms other methods in terms of Precision, and performs as well as the best (GCFF) in terms of F1 score. On the Cocktail Party dataset, our method performs slightly worse than the best (GCFF and GRUPO), but better than HVFF-ms. Overall, our method outperforms all methods in terms of Precision and F1 score, and performs comparably to the best (GCFF) in terms of Recall.

Discussion. Our work improves over prior work in ways that are even more significant than our superior overall performance: (1) *Imitation leads to group detection:* Most importantly, unlike prior work, our model is not trained explicitly for group detection. Instead, our focus is on behavioral imitation (learning a model that mimics human interaction). The surprising result is that, after training for behavior imitation, the output of KPM gate also performs successfully on the task of group detection.

(2) *Data-driven approach, features, parameters:* All methods we compare with rely on the assumption that people in groups position themselves in specific layouts (F-formations) developed in interactional socio-linguistics. By contrast, our models use no prior knowledge about layouts - we learn spatial layouts from data. Methods like GRUPO [58] use hand-defined parameters, features (e.g. stride, lambda, gaussian mixtures), and expensive iterative mode-finding sub-algorithms. By contrast, our method has a single learned feature (KPM gate) and performs low-cost $O(N^2)$ unsupervised clustering. GRUPO uses the additional feature of lower body orientation, which our method does not.

In summary, we have demonstrated state-of-the-art performance using a simple, direct method without the use of hand-crafted parameters, features or layout assumptions. The remarkable emergence of social perception in the form of group identification has been compared with various explicit methods for this task.

7 CONCLUSION

In this work, we presented MGpi, a deep neural network model of non-verbal and conversational interactions among multiple people and multiple groups. In the absence of real-world annotated data or simulators of multigroup face-to-face conversational behavior (e.g., speaking, listening, responding), we have designed our own social interaction simulator. While we consider discrete conversational actions among agents, we believe this is a necessary first step toward designing more complex models of real-world multigroup social intelligence.

We have demonstrated the necessity and efficacy of each interpretable component of MGpi, by evaluating several ablative baselines on their ability to predict the next appropriate action. We have demonstrated MGpi’s superior performance in scenarios with static and dynamically evolving group assignments. We reiterate that training MGpi requires no explicit group annotations. Instead, its Kinesic-Proxemic-Message gate (KPM gate) learns to express social attention based on non-verbal cues, as a result of training for a socially intelligent policy.

We have demonstrated the remarkable emergence of KPM gate’s ability to identify groups in real-world data, and achieved state-of-the-art results with our direct, unsupervised method, without using complex layout assumptions or hand-defined parameters.

In a dearth of real-world data on multiagent multigroup conversational interactions, valuable future work would involve large-scale collection of non-verbal and verbal exchanges between multiple groups of people. The availability of more diverse data of human-to-human exchange (e.g., body orientation, gestures, facial expressions, etc) would equip us to better design computational models like MGpi. Furthermore, such data would equip our models to better reason about social perception and propriety, by allowing the training of deep modules for a wider range of classes of behavioral actions, e.g., illustrators, emblems and attitudes [59].

Acknowledgment: This project was sponsored in part by JST CREST (JPMJCR14E1), NSF NRI (1637927) and IARPA (D17PC00340).
REFERENCES


