TEDIC: Neural Modeling of Behavioral Patterns in Dynamic Social Interaction Networks

Yanbang Wang  
ywangdr@cs.stanford.edu  
Stanford University

Chongyang Bai  
chy@cs.dartmouth.edu  
Dartmouth College

Pan Li  
panli@purdue.edu  
Stanford University, Purdue University

Jure Leskovec  
jure@cs.stanford.edu  
Stanford University

ABSTRACT
Dynamic social interaction networks are an important abstraction to model time-stamped social interactions such as eye contact, speaking and listening between people. These networks typically contain informative while subtle patterns that reflect people’s social characters and relationship, and therefore attract the attentions of a lot of social scientists and computer scientists. Previous approaches in extracting those patterns primarily rely on sophisticated expert knowledge of psychology and social science, and the obtained features are often overly task-specific. More generic models based on representation learning of dynamic networks may be applied, but the unique properties of social interactions cause severe model mismatch and degenerate the quality of the obtained representations. Here we fill this gap by proposing a novel framework, termed TEmporal network-Diffusion Convolutional networks (TEDIC), for generic representation learning on dynamic social interaction networks. We make TEDIC a good fit by designing two components: 1) Adopt diffusion of node attributes over a combination of the original network and its complement to capture long-hop interactive patterns embedded in the behaviors of people making or avoiding contact; 2) Leverage temporal convolution networks with hierarchical set-pooling operation to flexibly extract patterns from different-length interactions scattered over a long time span. The design also endows TEDIC with certain self-explaining power. We evaluate TEDIC over five real datasets for four different social character prediction tasks including deception detection, dominance identification, nervousness detection and community detection. TEDIC not only consistently outperforms previous SOTA’s, but also provides two important pieces of social insight. In addition, it exhibits favorable societal characteristics by remaining unbiased to people from different regions. The code can be downloaded from http://bit.ly/tedic_supplement

1 INTRODUCTION
Social interactions, referring to numerous and complicated actions among two or more people, have woven themselves into every piece of daily life [35]. These interactions, such as eye contact, speaking and listening, physical proximity between people, evolve over time and can be used to establish dynamic networks, which we term dynamic social interaction networks later. Dynamic social interaction networks, as a structured way to represent social interactions over time, have become critical data resources for social scientists to study the human behavioral patterns and make inferences about human social characters and relationship [25]. Specifically, where, when and how people interact with others provide informative cues for deception detection [2, 12], dominance identification [3, 6], personality traits characterization [31] and friendship inference [7, 15].

Despite their significance, mining indicative features from dynamic social interaction networks introduces great challenges. Such networks consist of two components that distinguish themselves from the relationship-based social networks arising typically from social media: 1) Highly dynamic attributes of individuals when they make contact, such as facial expressions, gestures and sounds; 2) Complicated and various ways of interactions, such as gazing, speaking and listening. Indicative features often come from the subtle interweaving of them and are concealed in a long-term complex interaction background. For instance, a lying person tends to quickly switch the eye contacts among different people due to low confidence [34], but such combination of behaviors may appear only a few times in a long conversation among a large group of people. As a more concrete example, we visualize in Fig. 1 dynamics of people’s behavior sampled from a person-to-person social-interaction game “RESISTANCE” (one of the datasets used in Sec. 5).

Previous works on these social tasks typically focus on designing hand-crafted features that are task-specific [2, 3, 6, 12, 31] and rely on domain knowledge in social science and psychology (e.g., visual dominance ratio [14], emotions and deception [42]). Consider the above example about inferring who has lied: Bai et al. [2] have demonstrated that the temporal distribution of the ranks of “gazing” probabilities among people is an informative feature. However, this feature has no obvious connection to one’s friend relationship.

Recent works on representation learning of dynamic networks seem to be a powerful alternative that allows for a generic extraction of features with little domain knowledge [26]. However, they have
been mostly evaluated on generic tasks such as link prediction and are not directly applicable. The mismatch comes from the unique patterns of dynamic social interaction networks, as shown in Fig. 1: 1) Social interactions such as "looking at" and "speaking to" have important duration information; 2) Multiple social interactions may be concurrent and overlapping in time domain. Previous methods to process streams of interactions typically focus on the starting point of each interaction but cannot handle concurrent interactions and their duration information [9, 24, 30, 40, 48].

A way to handle the concurrent and overlapping problems is to break dynamic networks into snapshots. However, network snapshots in our case should be partitioned in high time resolution to capture the important duration information of social interactions and the highly dynamic node attributes, which finally leads to a long sequence (\( \geq 1000 \)). Moreover, indicative patterns from subtle interweaving of highly dynamic node attributes and interactions are typically scattered in long-time. Both facts make previous methods on generic dynamic networks fail [19, 20, 29, 37, 38].

Present work. In this paper, we propose a neural network based model, temporal network-diffusion convolutional networks (TEDIC), to learn node representations of dynamic social interaction networks in a rather general manner, which fit into various node-level prediction tasks. The first part of TEDIC is network diffusion of node attributes that naturally captures the interweaving between highly dynamic node attributes and interactions. Note that graph diffusion procedure works in some sense similar to graph convolutional networks (GCN) [22] but without using non-linear activation neurons. This simplification allows tracking the effects of long-hop interactions and also improves the model’s explaining power. The second part of TEDIC is a temporal convolutional network (TCN) accompanied with set pooling to aggregate representations of nodes over a long time span. Due to the locality of temporal convolution kernels, TCN is able to extract patterns from interactions with various durations as these interactions may appear alternatively across multiple consecutive snapshots, and set-pooling is useful to collect subtle patterns scattered over a long-time span. Moreover, TEDIC is end-to-end trainable, and therefore provides an opportunity for social scientists to automatically process dynamic social interactions and obtain insights from the data simultaneously.

We evaluate TEDIC over four different node-level prediction tasks, including identification of people’s dominance, nervousness, lying behavior, as well as underlying community, on five different real social interaction networks. From the perspective of making inference, TEDIC significantly outperforms previous baselines that are either based on feature engineering designed for certain tasks or on neural networks for generic dynamic networks.

We further analyze TEDIC’s explaining power and broader societal implication by examining its learned coefficients and hidden embeddings: 1) We find that direct interactions (e.g., looking, speaking) among individuals may be more informative for dominance and nervousness detection, while signals of avoiding direct interactions are strongly informative for deception detection. This observation coincides with previous findings in psychology via extensive statistical analysis [14, 42]. 2) We also find that difference between the quantified attributes of one individual and those of his/her interacted neighbors is a stronger signal to indicate his/her dominance and nervousness, when compared with his/her own attributes. 3) We additionally show that TEDIC remains least biased to people from different regions despite its strong classification power on given tasks.

The paper is organized as follows: Section 2 reviews related research. Section 3 introduces notations and problem formulation. Section 4 introduces the TEDIC model. Section 5 evaluates TEDIC over extensive experiments and shows model interpretation.

2 RELATED WORK

The research related to our problem spans two broad areas.

Methods to Analyze Social Interactions. Many works have been conducted to analyze social interactions to identify human behaviors and relationship [2, 3, 6, 7, 12, 13, 15, 21, 31]. These works commonly adopt extensively statistical methods to analyze a combination of social interactions such as speaking and looking [6], physical proximity [7, 15] with individuals’ attributes including facial emotions and action units [2, 13], voice pitch and energy [6]. Task-specific features are extracted and are then fed into standard classifiers (e.g. SVM, Random Forest) to make inference. These engineered features, albeit powerful in their corresponding tasks, often require specific domain knowledge in social science and psychology theories and thus are less general.

Representation Learning for Dynamic Networks. The success of representation learning for dynamic social interaction networks strongly depends on extracting the interweaving of highly dynamic node attributes and interactions. A few works process a sequence of interactions between nodes, but they are unable to take dynamic node attributes [9, 19, 24, 30, 39, 40, 48, 49]. Works that were claimed to digest dynamic node attributes all work on network snapshots [20, 29, 37, 38]. They generally follow the framework by first propagating node attributes of each network snapshot and then aggregating them over time. The first step adopts either graph convolution networks [22] or graph attention networks [41]. The non-linearity in propagation therein prevents the model from learning long-hop interweaving between node attributes and edges.
TEDIC: Neural Modeling of Behavioral Patterns
in Dynamic Social Interaction Networks

Figure 2: Temporal Network-Diffusion Convolutional Network (TEDIC). The Graph Diffusion component captures the interaction patterns among people at any given time snapshot; its Set-Temporal Convolution further filters, transforms, and aggregates important signals over time in a hierarchical manner.

In this section, we introduce the notation and problem formulation.

**Notation.** A static network can be represented as a graph $G = (V, E)$ where $V$ denotes the set of nodes and $E \subseteq V \times V$ denotes the set of edges. Let $N = |V|$. An edge refers to a pair of vertices $(u, v) \in E$. Networks that we discuss may be directed or undirected. An undirected network can be viewed as a special case of directed ones given the condition $(u, v) \in E \iff (u, v) \in E$. In the later discussion, we implicitly assume $G$ is directed unless specified. Graph $G$ is associated with adjacency matrix $A \in \mathbb{R}^{N \times N}$, $G$ is assumed to be positive, normalized, and weighted: $A_{uv} \in (0, 1]$ if $(u, v) \in E$ and otherwise $A_{uv} = 0$. The diagonal degree matrix is defined as $D = \mathbb{R}^{N \times N}$ whose $u$-th diagonal component is $d_u = \sum_{v \in V} A_{uv}$.

We use $M$ and $M'$ to denote feature’s dimensions. Given a multi-variant multi-dimension time-series $\{X_t \}_{t \in \mathbb{Z}}$ where $X_t \in \mathbb{R}^{N \times M}$, we define temporal convolution as $Y_t = X_t \ast C_t \triangleq \sum_{t' \in \mathbb{Z}} X_{t' \ast t} C_{t'}$, where $\{C_t \}_{t \in \mathbb{Z}}$ are kernels and $C_t \in \mathbb{R}^{M \times M'}$. Note that for a finite length kernel $C_t$, the sum contains finite terms.

**Problem Definition.** Dynamic social interaction networks originally consist of streams of interactions with duration. In practice, researchers leverage sensors to sample snapshots of these networks in high temporal resolution. Therefore, we directly define our data structures as dynamic graph snapshots: $\{G_t \}_{1 \leq t \leq T}$ where $G_t = \{V_t, E_t\}$. Note that, in general, the node set $V_t$ could change over time. However, in our case, $V_t$ (denoting participants) is assumed to be fixed, i.e. $\forall t, V_t = V$, which comes from the property of data for social interaction networks of interest: In most cases network data is collected from sensors pre-allocated among participants of an experiment, e.g., tracking behaviors of multiple agents in a game/conference. Increasing the number of sensors during the experiments is not relevant to the target of the experiment. In contrast, the edge set $E_t$, denoting interactions between people could evolve significantly during the whole time period. In our problem, the network is associated with dynamic node attributes: $\{X_t \}_{1 \leq t \leq T}$, where the row of $X_t$ corresponding to node $u$, $X_{t,u}$, denotes initial attributes of node $u$.

Our work is to learn the node representations in these networks to capture important patterns from people’s social interaction behaviors. Once the representations are learnt, prediction/inference on certain tasks can be accomplished by feeding these representations into task-specific inference blocks. We claim that our approach can be used for general node-level prediction tasks that require patterns from dynamic social interaction networks, while specifically in this work, we consider the following four tasks: deception detection, dominance identification, nervousness detection and community detection. Note that the specific inference blocks and training objectives will be specified in Section 5.

4 PROPOSED MODEL: TEDIC

In this section, we introduce our model, TEmporal netwORk-DIffusion Convolutional Network (TEDIC). It consists of two main components: Network Diffusion of node attributes, and Set-temporal convolution-based aggregation over time (Fig. 2) plus a readout layer. Input to TEDIC is a long sequence of dynamic interaction features and network snapshots, $X_t$ and $G_t$. TEDIC outputs an embedding for each person encoding his/her behavioral traits through the interaction events, which can be directly piped to a simple classifier for prediction. Each component of the model is designed to capture properties of dynamic social interaction networks on a different aspect.
4.1 Network Diffusion Component

To learn interactive effects of people for various social tasks via the network diffusion process, we parameterize the diffusion process by two categories of parameters with proper physical meanings. The first category is to distinguish the implications of people making interactions and avoiding interactions. The second category of parameters is to characterize the effect of interactions with different hops over dynamic networks.

Parameters $\beta$ for making or avoiding interactions. One speciality of social interaction networks is that the behavior to avoid interactions could be very informative. For example, deceivers tend to avoid gazing at others [34], and some deceivers may tend to be abnormally quiet in front of others due to their low-level self-confidence. However, different phenomena could happen between a follower and his leader [43]. So we consider graphs corresponding to the original interaction networks and their complement graphs simultaneously. Concretely, for each type of interaction network with adjacency matrix $A$, we also consider the corresponding adjacency matrix of the complement network $\tilde{A} = 11^T - A$ where $11^T$ is an all-one matrix. Then, we introduce another parameter $\beta \in [0, 1]$ to merge these two networks to obtain a new adjacency matrix via

$$A' = \beta A + (1 - \beta)\tilde{A} = (2\beta - 1)A + (1 - \beta)11^T. \quad (1)$$

Apparently, this parameter $\beta$ can have implications: a greater $\beta$ suggests making interaction is more informative to a prediction task, while a smaller $\beta$ emphasizes that avoiding interaction may be the key clue. Next, we do graph diffusion of node attributes based on the random walk matrix $W = D^{-1}A'$, where $D'$ is the diagonal degree matrix of $A'$.

Parameter $\Gamma_k$ for different-hop interactions. The model is now to perform different-step graph diffusion of node attributes based on the induced random-walk matrix $W'$. By assigning a group of learnable parameters $(\Gamma_k)_{k \geq 0}$, where $\Gamma_k$ is a diagonal matrix for the hop $k$, we consider the transformation of initial node attributes $X_t \in \mathbb{R}^{N \times M}$ for network snapshot $t$ based on network diffusion as

$$H_t = \sum_{k \geq 0} H_t^{(k)} \Gamma_k = \sum_{k \geq 0} (W_t^{(k)} H_t^{(0)} \Gamma_k) \quad H_t^{(0)} = f(X_t) \quad (2)$$

where $f(\cdot) : \mathbb{R}^{N \times M} \rightarrow \mathbb{R}^{N \times M'}$ could be as simple as identity mapping $(M' = M)$ or as complex as multi-layer perceptrons (MLP) that properly transform and normalize initial node attributes. Here, $M'$ is the dimension of output channel. $\Gamma_k \in \mathbb{R}^{M \times M'}$ provides the weights for the $k$-hop diffusion. The corresponding $q$-th diagonal component, denoted by $\gamma_{k,q}$, is the weight for $q$’s output channel. In practice, typically only the first several hops could be informative so we may set an upper bound to the number of hops: 5 ~ 10 steps provide good enough results in practice.

The Eq. (2) has many implications. Consider the sequence $(\gamma_{k,q})_{k \geq 0}$ for any $q$ and suppose $f$ is identity mapping. From the perspective of graph spectral convolution, $(\gamma_{k,q})_{k \geq 0}$ corresponds to weights on different levels of the smoothness of the $q$-th node attributes. Moreover, different fixed formulations of $\gamma_{k,q}$ provide different ranks of nodes: $\gamma_{k,q} \propto a^k$ corresponds to PageRank [32]; $\gamma_{k,q} \propto \gamma^k / k!$ corresponds to heat-kernel PageRank [8]. Extensive feature engineering shows that different formulations of ranks could be important signals to detect deceivers or leaders among groups of people [2, 3]. Our formulation based on learnable parameters, connecting to generalized PageRank [27], allows for bigger representation power to cover multiple prediction tasks. Moreover, for model self-explanation, as $W'\Gamma$ is column stochastic, it will keep the $I_k$-norm of every column of $H^{(k)}$ unchanged (with non-negative features) and thus naturally hold normalizing property. Therefore, the value $|\gamma_{k,q}|$ and the sign of $\gamma_{k,q}$ can be naturally interpreted as the effect of $k$-hop diffusion of $q$-th node attribute to the final representation. Even when $f$ is an MLP, decoupling parameters $\Gamma_k$ on diffusion and parameters on pure transformation of node attributes in $f(\cdot)$ keeps the effect of network diffusion distinguishable, which is useful in the model self-explanation.

Note that there could be variants of Eq. (2) to further increase model complexity and representation power. By adding nonlinear transformation of each step $H^{(k)}$ before letting it propagate, one may get the model GCN [22]. However, adding non-linearity per step increases the difficulty for training, which limits the steps of propagation to 2-3, and could simultaneously decrease the model’s self-explanatory power. As our experiments do not show any improvement based on non-linearity, a simpler model is preferred. Similar gain by removing non-linearity has also been observed in many recent literatures on graph neural networks [23, 44]. However, to our best knowledge, we are the first to show the success of this manner to process dynamic networks. The network diffusion formula is also relevant to the ChebNet [11] while the ChebNet was proposed for undirected unweighted graphs and used graph Laplacians instead of random-walk matrices for weighted graphs in our setting.

4.2 Set-Temporal Convolution Component

To aggregate node features over time, we propose a method called Set-TCN (S-TCN) to handle the complex and long-term temporal social interactions. The input of this block is a sequence of node features $\{H_t\}_{1 \leq t \leq T}$ where $H_t$ denotes the node features for each snapshot $t$ obtained via equation (2).

There are two challenges, as aforementioned, in building the block to aggregate temporal information. First, our model should be able to handle an extremely long sequence of snapshots. Second, indicative patterns, such as “switching his/her gaze”, are typically subtle and scattered randomly and in the whole time span. Their global orders may not matter so much (e.g. when exactly a person laughs), but recognition and collection of the local patterns may be highly important (e.g. who laughs with the person and how many times do they laugh in total). Our model should also be capable to deal with such subtleties and complexities of behavior signals.

The S-TCN block is built for this target with two components. The first component consists of multiple TCN layers to capture local dynamics. The second component is a set pooling to collect local patterns randomly scattered within the whole long time space.

Multi-layer Temporal Convolution. There are $L$ layers of temporal convolutions. Kernels of the $l$-th temporal convolution layer
can be represented by a sequence \( \{C_t^{(l)}\}_{1 \leq t \leq w} \) where \( w \) is its window length, which transforms the input \( H_t \) from Eq. (2) via
\[
Z_t^{(l)} = \text{ReLU}(Z_t^{(l-1)} \ast C_t^{(l)}), \quad \{Z_t^{(0)}\}_{1 \leq t \leq T} = \{H_t\}_{1 \leq t \leq T} \tag{3}
\]
\[
Z_t^{(l)} = \text{max-pool}((\tilde{Z}_t^{(l)}, \tilde{Z}_{t+1}^{(l)}), \quad \text{for } 1 \leq l \leq L \tag{4}
\]
where \( \ast \) is the convolution operator defined in Sec. 3. The number of layers \( L \) typically depends on the time scale of interactions we want to extract patterns from. It is related to the receptive field of convolution kernels. Convolution kernels are widely known for their capability to function as low-pass filters. By stacking different numbers of convolution layers, we can explicitly tune the capability to extract scattered local patterns.

### Set Pooling
The interaction happened at the two blue timestamps in layer \( l \) is captured by the blue timestamps in layers \( l + 1 \) and \( l + 2 \) through convolution operation.

![Figure 3: Receptive field of temporal convolution](image)

**Figure 3:** Receptive field of temporal convolution: The interaction happened at the two blue timestamps in layer \( l \) is captured by the blue timestamps in layers \( l + 1 \) and \( l + 2 \) through convolution operation.

Given a proper depth of TCN \((L \in [2, 4])\), to obtain a proper size of receptive field, the length of the final layer could still be long (\( \geq 50 \)) because of the original long time series \((T \gtrsim 1000)\). Thus, next we leverage set pooling to extract scattered local patterns.

**Set Pooling.** As opposed to online social networks that often show seasonal patterns, there are seldom periodical patterns in the offline social interaction networks we study. Consider eye contact in conversation/meeting among a group of people. Informative patterns of interactive behaviors of people are usually randomly scattered in the long time span. Therefore, with the local patterns captured by TCN, we use set pooling to obtain the set sequence \( \{Z_t^{(l)}\}_{1 \leq t \leq T} \) to extract messages scattered within this long sequence. We observe that the following is generally effective across different applications:
\[
Z_{\text{out}} = \text{mean-pool}_{1 \leq t \leq T}((\text{ReLU}(Z_t^{(l)} \Theta_1 + Z_{\text{max}} \Theta_2))) \tag{6}
\]

First, we impose the max pooling Eq.(5) on \( \{Z_t^{(l)}\}_{1 \leq t \leq T} \) to emphasize the critical local patterns; Then, we linearly merge the output of max pooling into each \( Z_t^{(l)} \) to let each \( Z_t^{(l)} \) capture global information; Finally, after a simple ReLU activation, we obtain the output via mean pooling.

Note that the max pooling captures the essence of randomly scattered patterns while the second step based on linear combination and the mean pooling is found out to be useful to improve the robustness of feature aggregation. Note that this set-pooling technique properly tailors Deep Sets [47] for our setting.

### 4.3 Readout Layer
Up to Eq.(6) we derive for each person a representation that encodes his/her behavioral patterns throughout the interaction. The purpose of appending an additional readout layer is to further model the whole interaction context which involves all people’s representations. Therefore, we use the readout layer:
\[
Z_{\text{out}}^{(l)} = Z_{\text{out}}^{(l)} \Theta_3 + \text{mean-pool}_{1 \leq j \leq N}(Z_{\text{out}}^{(j)} \Theta_4), \quad \forall 1 \leq i \leq N, \tag{7}
\]
where \( Z_{\text{out}}^{(l)} \) is the \( i \)-th row of \( Z_{\text{out}} \) and \( \Theta_3, \Theta_4 \) are learnable weights. The second part of Eq. (7) models the whole interaction context.

### 5 EXPERIMENTS
Our proposed model is evaluated over five datasets on four node-level classification social tasks: detecting dominant, deceptive (lying), and nervous people, as well as people’s underlying community. Table 1 summarizes the task settings and dataset statistics. We will refer back to this table as we walk through the experiment settings in Sec. 5.1. Also, since Task 5 holds different properties compared to Tasks 1 ~ 4 while is a common task to evaluate representation learning of dynamic networks, we postpone its introduction and analysis to Sec. 5.5.

#### 5.1 Experimental Setup
**Raw data & Preprocessing.** The raw data of datasets 1-4 is a collection of videos. Each video records a group conversation that ranges from 5 to 40 minutes and contains frontal views of each individual in the group. The preprocessing of these videos involves two steps: feature extraction and time coarsening.

To extract numerical behavioral features of people in each video, we employ several vision-based and audio-based techniques following a similar pipeline of [3]. The extracted features cover a wide range of channels people use to convey messages. We briefly summarize here: I. Emotion: intensity of eight emotions e.g. happiness, anger, calm, etc. and two facial traits (smile, open eyes), provided by Amazon Rekognition; II. FAU: intensity of 17 facial action units extracted by OpenFace [5]; III. MFCC: voice features widely used in audio analysis [10]; IV. Speak Prob.: probability that a person is speaking estimated from lip movement [4]; V. Gazing Prob.: probability that person i looks at person j estimated from [4]. Note that gazing from person i to himself (j = i) means that the person looks at his own computer in front. The sum of each person’s Gazing Prob. towards all targets is 1. These features are extracted every 1 second from videos. We use Features I ~ IV as our dynamic node features, and use Feature V to construct a densely connected dynamic interaction network. In the network, the nodes are the participants and the dynamic edges are weighted by the gazing probabilities that participants look at each other over time.

With the features extracted, the time coarsening process deals with another critical aspect of our interaction sequences: the time
resolution. In this step, we smooth both the node attributes and the edge weights of the snapshot sequence by taking the mean value of each feature dimension every $\Delta$ seconds along the time axis. $\Delta$ is a hyperparameter having been extensively tuned for both TEDIC and all baselines as different models show different sensitivity to the time granularity. We will provide an in-depth empirical analysis on the values of different $\Delta$’s in Sec. 5.2.

**Dataset: RESISTANCE-D, S, -N.** These three datasets record people’s performance in a role-playing party game called the Resistance: Avalon [16]. Each game has 5 to 8 players secretly split into two rivaling teams (“spy” and “not spy”) before the game starts. In order to win, people need to collaborate with each other, argue persuasively, avoid appearing nervous, and even extensively lie if they are assigned a “spy” role. The three datasets share about 50% videos in common. The rest differs due to several practical constraints to collect labels.

Labels for RESISTANCE-D and RESISTANCE-N are generated by referencing surveys taken by all participants after each game. The surveys take the form of questionnaires, asking each participant to rate the dominance and nervousness levels for each other. Based on these scores, we rank all the people in each game and use the ranks as ground truth. Since the tasks on these two datasets (i.e. Tasks 1 & 4) is to identify the most outstanding person from the group (the most dominant person, the most nervous person), we consider them as a *multiclass* classification problem, where the number of classes is the number of players.

Labels for RESISTANCE-S, which are all people’s identity of each game (i.e. Spy or not), are pre-generated by the dataset. We know from the game’s setting that spies have to keep lying and thus regard those identities as the ground truth labels for deception detection (i.e. Task 3). Since there can be more than one spies in each game, we regard this task as a *binary* classification problem on each person.

**Dataset: ELEA.** The dataset [36] is a widely used public benchmark for modeling and detecting people’s behavior [3]. In each video, 3-4 participants performed a “winter survival task” by having collaborative discussions. External annotators watch video games and assign a dominance score for each player. Then, the generated dominance labels indicate a slightly different meaning: whether a person is more dominant in the group instead of the most dominant. This is done by thresholding dominance scores with the median dominance score and by assigning binary labels accordingly. The subtle difference is in place to follow the protocols of most previous works such as [1, 3]. This provides another angle of evaluation compared to most dominant person prediction in RESISTANCE-D.

We treat the task on ELEA (i.e. Task 2) as a person-wise *binary* classification task.

**Baselines.** Our framework is compared with two groups of baselines. The first group are task-specific baselines which were proposed uniquely for each task by integrating domain knowledge into handcrafted features. The second group are generic baselines originally proposed to model generic dynamic network structures. We briefly introduce them here.

For task-specific baselines, we select for each type of task a handful of previous methods to compare with:

**Dominance Detection.** MKL [6] is a method based on hand-crafted features like voice pitch and speaking rate. GDP [3] is a method relying on a special kind of handcrafted feature called DomRank, with two versions: one using random forest classifier (GDP-RF), and the other using multi-layer perceptron classifier (GDP-MLP). DELF [3] is a method also reported in the same work as GDP, and uses DomRank in a slightly differently way. FacialCues [18] is a method leveraging the facial action units from [5].

**Deception Detection.** DDV [45] is a method combining hand-crafted micro facial expression with NLP features. TGCN-L [28] is a method based on gazing probabilities. LiarRank [2] is based on all the features we used but aggregates them in a way so that several pieces of their domain knowledge get integrated.

**Nervousness Detection.** This is a new task which, to our best knowledge, few methods were proposed in a similar problem setting. Among all the previously introduced baselines, we think the LiarRank and FacialCues are two baselines that will most possibly work to help detect people’s nervousness. Therefore, they also become the baselines for this task.

For generic baselines, the SOTA methods that claim to handle dynamic networks well are primarily based on various architectures of temporal GNNs, among which we select three most representative ones to evaluate across all tasks: CD-GCN [29] is one of the latest methods on dynamic graph classification tasks. It combines a skip-connected GCN with a returning sequence LSTM. EvolveGCN [33] is a latest method on dynamic network sequence modeling tasks, especially for link prediction and link type classification. It uses a recurrent module to update the projection weight of a GCN module. GCN [38] is another method for modeling dynamic network sequence constructed from images and point clouds, using a convolutional module to update the internal weights of a LSTM module that deals with sequential node features.

To ensure fair comparison, all baselines share with our proposed model the same readout layer and loss function.

<table>
<thead>
<tr>
<th>No.</th>
<th>Task</th>
<th>Dataset</th>
<th>Classification</th>
<th>Networks</th>
<th>Avg. Time Steps</th>
<th>Group Size</th>
<th>Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dominance (R)</td>
<td>RESISTANCE-D</td>
<td>multiclass</td>
<td>956</td>
<td>2,514</td>
<td>5 ~ 8</td>
<td>$4.007 \times 10^5$</td>
</tr>
<tr>
<td>2</td>
<td>Dominance (E)</td>
<td>ELEA</td>
<td>binary</td>
<td>27</td>
<td>2,545</td>
<td>3 ~ 4</td>
<td>$6.474 \times 10^5$</td>
</tr>
<tr>
<td>3</td>
<td>Deception</td>
<td>RESISTANCE-S</td>
<td>binary</td>
<td>2,157</td>
<td>2,258</td>
<td>5 ~ 8</td>
<td>$2.439 \times 10^7$</td>
</tr>
<tr>
<td>4</td>
<td>Nervousness</td>
<td>RESISTANCE-N</td>
<td>multiclass</td>
<td>1,097</td>
<td>2,528</td>
<td>5 ~ 8</td>
<td>$4.910 \times 10^7$</td>
</tr>
<tr>
<td>5</td>
<td>Community</td>
<td>CIAW</td>
<td>multiclass</td>
<td>1</td>
<td>20</td>
<td>92</td>
<td>$2.149 \times 10^4$</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the dynamic network datasets. *: The time steps are before coarsening (with time granularity $\Delta = 0.33s$). †: We count all the interactions with gazing probability $\geq 0.5$. 
Training and Evaluation. We randomly partition our data into K folds, reserving 1/K for testing and the rest for training. Following [2, 3], we use K = 10 for all RESISTENCE datasets, and K = 27 for ELEA. To compute the logits, we add a single-layer NN plus a sigmoid or softmax nonlinearity on top of the readout layer (see Section 4.3). We use the cross entropy loss and use Adam to optimize all the models. To evaluate our method as well as baselines, Mean Accuracy over the K folds is reported. There are several hyperparameters related to the tuning process, including the time resolution $\Delta$ in data preprocessing, the number of layers for set-temporal pooling, etc. All hyperparameters, both for our model and all baselines (except DDV whose code is not available), are extensively tuned with grid search and the best performance is reported. Please refer to the supplementary material for the detailed search ranges of hyperparameters.

5.2 Experimental Results

Table 2 compares the performance all models on Tasks 1 ∼ 4. Here for brevity we only report the top 2 results of our task-specific baselines, and leave the complete evaluation table to the supplement. From the comparison, we observe that TEDIC consistently shows high performance across all the tasks: it significantly outperforms the strongest baselines in Tasks 2 & 4 and also achieves better result on Tasks 1. Interestingly, we also see that our model has most statistically significant gain in most challenging Tasks 3 & . These are the two tasks where the interacting participants purposely conceal the indicative signals of their labels because they do not want others to know that they are lying or nervous. We attribute such success to the fact that TEDIC effectively captures the temporal cues. While almost all the baselines come with proper graph convolution or careful feature engineering work, their ways to process temporal information are insufficient by simply using mean pooling (TGCN [28]), Fisher Vector (FacialCues [18]), histogram encoding (DELF [2]), or many-to-one LSTM (GCRN [38]). In particular, the generic baseline’s failure on Tasks 1 & 4 implies the lack of robustness with temporal sequence modeling techniques based on recurrent structures.

Ablation study. We further demonstrate the usefulness of each TEDIC building block by conducting ablation study on RESISTENCE-D. Results are shown in Table 3. In the table, Ab. 2 ∼3 further verify RNN’s insufficiency on handling both extremely long time sequence and weak local dynamics. Interestingly, the simple mean pooling can outperform RNNs. Ab. 4 ∼ 8 focus on graph-level techniques by replacing the network diffusion module. Ab. 4 shows the importance of using network for prediction. Ab. 5 ∼ 8 indicates the usability of GCN despite its serious decay because of over-smoothing when going deep. In contrast, our network diffusion can propagate as long as 10 hops without significant performance decay.

Effect of Time Resolution. The time resolution of input interaction sequence is a hyperparameter that controls the level of temporal smoothing in preprocessing stage. Data with high time resolution betters the opportunity to capture subtle short-time interaction patterns while introducing more noise. In that regard, we attribute the failure of RNN-based baselines [29, 33, 38] to their sensitivity to such noise. To validate this understanding, we conduct further experiments by adjusting the original time resolution $\Delta = 0.33s$ that is the highest time resolution to collect the data. Specifically, we averaged input feature sequence and edge weights for every few seconds ($\Delta = 1s, 3s, 15s, 60s$) to change the time resolution.

Fig.4 plots the performance of our model and the generic baselines on all datasets. The competing effect previously mentioned is clearly demonstrated by the trend of the lines: the accuracy of each method usually peaks at a certain time resolution and drops sideways. Also note that on three out of the four datasets our method’s performance peaks with high time resolution compared with baselines. This indicates that our model does well in extracting knowledge from more detailed and subtle behavioral patterns, while also staying robust to the adverse effect introduced by varying the sequence lengths and noise levels.

5.3 Model Interpretation

The linear dependence on the parameters $\beta$ and $\{\Gamma_k\}$ in network diffusion provides certain self-explaining power that further induces some social insights.

Interpretation I: Balancing Weight $\beta$. Recall that $\beta$ (Eq.) is the learnable parameter that directly controls the relative importance of proactive interaction versus avoidance of interaction. Fig. 5 displays how the $\beta$ converges during the training (initialized to 0.5, i.e. neutral). For each task we ran multiple times by introducing small perturbation to $\beta$’s initialization. The figure shows that the parameter exhibits very different convergence behavior across different tasks. For the deception detection task, $\beta$ significantly drops to around 0.2, which indicates that avoidance of interaction may be much more important than contacts to detect deception. Interestingly, this phenomenon coincides with findings from a psychological study [34] on eye movement of people in various contexts. Moreover, for the dominance identification task dominant, the convergence to a large $\beta$ implies that people are more easily identified with their aggressive way of reaching out to others. For the nervousness detection, there seems to be no dominating values of $\beta$. 

![Figure 4: Accuracy as a function of time resolution of the interaction sequence.](image-url)
Table 1: Accuracy of detecting dominance, deception and nervousness. Mean Accuracy ± 95% confidence interval is reported.

<table>
<thead>
<tr>
<th>Method</th>
<th>Task</th>
<th>Dominance (R)</th>
<th>Dominance (E)</th>
<th>Deception</th>
<th>Nervousness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-1 Method</td>
<td></td>
<td>0.918±0.013</td>
<td>0.769±0.019</td>
<td>0.668±0.021</td>
<td>0.733±0.022</td>
</tr>
<tr>
<td>Top-2 Method</td>
<td></td>
<td>0.887±0.015</td>
<td>0.673±0.015</td>
<td>0.638±0.016</td>
<td>0.729±0.015</td>
</tr>
<tr>
<td>CD-GCN[29]</td>
<td></td>
<td>0.687±0.042</td>
<td>0.794±0.022</td>
<td>0.673±0.018</td>
<td>0.534±0.084</td>
</tr>
<tr>
<td>GCRN[38]</td>
<td></td>
<td>0.587±0.096</td>
<td>0.795±0.032</td>
<td>0.643±0.045</td>
<td>0.356±0.104</td>
</tr>
<tr>
<td>EvolveGCN[53]</td>
<td></td>
<td>0.602±0.061</td>
<td>0.739±0.077</td>
<td>0.623±0.042</td>
<td>0.397±0.099</td>
</tr>
<tr>
<td>Proposed TEDIC</td>
<td></td>
<td>0.923±0.009</td>
<td>0.815±0.019</td>
<td>0.689±0.012</td>
<td>0.769±0.023</td>
</tr>
</tbody>
</table>

Table 3: Ablation study on Task 1.

<table>
<thead>
<tr>
<th>Ab. Replacement</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Original</td>
<td>0.923±0.009</td>
</tr>
<tr>
<td>2 S-TCN → LSTM</td>
<td>0.758±0.009</td>
</tr>
<tr>
<td>3 S-TCN → Mean Pool</td>
<td>0.842±0.023</td>
</tr>
<tr>
<td>4 Diff. → None</td>
<td>0.829±0.019</td>
</tr>
<tr>
<td>5 Diff. → GCN-1 Layer</td>
<td>0.844±0.020</td>
</tr>
<tr>
<td>6 Diff. → GCN-2 Layer</td>
<td>0.889±0.015</td>
</tr>
<tr>
<td>7 Diff. → GCN-4 Layer</td>
<td>0.784±0.026</td>
</tr>
<tr>
<td>9 Freeze β = 1</td>
<td>0.889±0.014</td>
</tr>
</tbody>
</table>

Analysis of β indeed provides some information related to people’s social status and their willingness of making social interactions.

Figure 5: Different convergence behaviors of β during training, 95% confidence intervals shaded. Trained on RESISTANCE-D,-S,-N respectively.

Interpretation II: Diffusion Weights \( \{Γ_k\} \). Recall from Eq. 2 that \( \{Γ_k\}_{0≤k≤K} \) is a sequence of diagonal matrices where \( Γ_k ∈ \mathbb{R}^{M'×M'} \) contains the weights corresponding to \( M' \) features’ \( k \)-hop diffusion. After training the model, we would obtain \( K + 1 \) diffusion weights for each of the \( M' \) features by taking the diagonal of each \( Γ_k \). Analyzing these weights provides insights of how the interaction network helps shape the original features during the diffusion. Fig. 6 shows the weights for four of the features (diffusion steps \( K = 10 \)) when the model is trained for the nervousness detection task. The diffusion weights have been normalized such that the 0-hop weight is 1. First, we observe that the 0-hop weight is significantly the largest, meaning that the original node features are very important to prediction. Therefore, the role of graph diffusion in this task can be roughly regarded as a fine-tuning process over the original features. Second, a clear contrast between the top two features is observed. Both of them can propagate far via the interactions, the way diffusion modifies the original features are different (\( Γ_k > 0 \) for node in-degree v.s. \( Γ_k < 0 \) for node self-loop degree). We attribute such distinction to different implications of those two features: The self-loop-degree of a node can be interpreted as the probability that the person looked at his/her own camera at the corresponding moment. Person with this behavior patterns tend to be more introverted and preservative, and may affect the conversation and make other people talkless as well. Therefore, if there is one person that looks frequently at his own computer, other people may appear a bit nervous to our algorithm as well. In contrast, the in-degree of a node can be interpreted as the attention that one received from other people at the moment. Those “other people” may therefore appear less nervousness. Finally, the “smile” and “happy” emotions seem to be able to diffuse two steps while not beyond, which indicates they are less influential to other people.

5.4 Fairness of the Model

Given our task’s nature, we paid special attention to TEDIC’s societal implications, hoping to ensure that it does not discriminate against people from certain communities. People involved in this study can be identified as four communities based on the regions where the videos are recorded: North America, Africa, Middle East Asia and Far East Asia. In particular, we examine two things: 1) Whether our model is able to distinguish people’s ethnic or cultural background; 2) How much bias our model may introduce to the decision process. For this case study we specially focus on Task 3 (deception detection) because it is more ethically sensitive and also the task that we have most complete regional identity labels of people.

We investigate the first question by visualizing the learned hidden embeddings of each person, which are extracted from three positions of our model: the input layer, the Graph Diffusion layer, and the S-TCN layer. Fig. 7 plots those embeddings projected into the low-dimensional space. Here, the color of one dot encodes one’s...
corresponding regional identity. To quantify the level of mixing identities, we use all those embeddings as input features and run a logistic classifier to check how accurate those embeddings can be used to identify one’s region. We see from both the plot and the numbers that the embeddings are well mixed together and the classification accuracy remains almost unchanged through our model. This implies that our framework collects very limited amount of regional information, which lowers the risk of it being biased.

To answer the second question, we summarize the proportions of people from different places that our model predicts to be lying. Table 4 shows the average percentages and their 95% confidence intervals. Statistically we can also conclude that no significant clue (i.e. \( p < 0.05 \)) is found with our model being discriminatory towards people with different regional identities.

### 5.5 General Applicability

In the end, we want to demonstrate that TEDIC is applicable to a wider range of social interaction networks. Previous experiments have shown its success to process networks with high-frequent interactions spanned in a long-time range. Here we further investigate how TEDIC works when the dynamics is less vibrant and the time range is relatively short, for which we conduct evaluation over another dataset CIAW [17]. CIAW is a dynamic network built upon 92 people’s timestamped proximities (of up to 1.2 meters) in a workplace over 20 days. Our goal is to infer one’s department identity based on the dynamic network. This evaluation setting differs from our main setting in several aspects: The network is significantly less dynamic with much less snapshots (Table 1); It also has a different instantiation of “social interaction” defined by physical proximity instead. More details of this dataset and experiment settings are left to our supplementary materials. We compare our model with the generic baselines and obtain the results in Table 5. We observe that our model can still perform well in such a very different scenario, though the accuracy gain to the strongest baseline is comparatively marginal. We attribute the consistent performance of our model to the high robustness of the S-TCN block to sequences with various length. It would also be interesting to further test our model in other cases where the nature of interaction sequences fundamentally differs, which we leave for future study.

## 6 CONCLUSION

In this work, we introduce TEDIC as a new neural-network-based model particularly designed to extract people’s behavioral patterns from dynamic social interactions via a light-feature-engineering manner. Benefiting from its well-coordinated building blocks, TEDIC not only achieves SOTA performance to identify various individuals’ social characteristics, but also enables certain self-explaining power to yield in-depth understanding of people’s behaviors used in their social interaction. Experimental results also demonstrate the fairness of TEDIC when it processes the patterns of people from different regions. Finally, we would like to claim that TEDIC’s broader societal implications can never be over-emphasized: Further investigation into its reliability, fairness and generality could inspire a series of meaningful studies in the future.

### REFERENCES


A DATASET DETAILS

**RESISTANCE-D, -S, -N.** These three dataset record people’s performance in a role-playing party game called the Resistance: Avalon. Each game has 5 to 8 players secretly split into two rivaling teams before the game starts: the resistance team ("good" people, Team A, accounting for about 70%) and the spy team ("bad" people, Team B, the rest 30%). Team B knows everyone’s real identity but Team A do not. Both teams’ goal is to beat each other in the “missions” in the form of discussion and voting. This process involves frequent deception behavior (presumably only from Team B) and argument, query and persuasion (from all parties). To persuade others, people often need to be dominant and avoid appearing nervous. The three dataset share about 50% videos in common. For the rest they each differ slightly because of several practical constraints to collect labels.

Labels for RESISTANCE-D, RESISTANCE-N are generated by referencing surveys taken by all participants after each game. The surveys take the form of questionnaires, asking each participant to rate the dominance and nervousness level of all the other people. We treat the median and the mean value of each person’s scores rated by others as the “ground truth scores” for that person. The person with highest “ground truth scores” in a group is regarded as the most dominant / nervous one. Specifically, we first compare the majority of each person’s scores rated by others as the “ground truth scores”, and those with the most votes are regarded as the most dominant / nervous one. Finally, we classify each person into the following three categories based on their scores:

- **Dominant** (scores at least 3): Team A, dominant player.
- **Passive** (scores less than 3): Team A, passive player.
- **Team B** (randomly assigned): Team B, passive player.

Since the tasks on these two dataset (i.e. Task 1 & 4) is to identify one most outstanding person from a group, we consider them as a multiclass classification problem.

Labels for RESISTANCE-S, which are all people’s identity of each game (i.e. spy or not), are pre-generated by the dataset. By the game’s setting, it is guaranteed that all spys are lying throughout the game and the rest are not. Since there can be more than one spies in each game, we treat the task on this dataset (i.e. Task 3) as a binary classification problem. We take K = 10 for K-fold validation to evaluate our model.

**ELEA ELEA** is widely used public benchmark for modeling and detecting personal traits such as dominance [3]. In each video, 3-4 participants collaboratively performed a “winter survival task” by having peaceful discussions. We perform only dominance detection task on the dataset as other types of labels are unavailable. The dataset can be downloaded from [https://www.idiap.ch/dataset/elea](https://www.idiap.ch/dataset/elea). Following the protocols of [1, 3], labels for ELEA are generated in a slightly different way than RESISTANCE. First, the “perceived dominance
score” (PDS) is used as the ground truth. These are the scores rated for each participant by the game organizers who hosted and monitored the game. Second, based on the PDS, binary labels are assigned by thresholding each group of interacting people with their group’s PDS median. In other words, people that receive relatively high PDS would be marked as being dominant with label 1, and the rest with label 0. Because of these, we regard task performed on ELEA (i.e. Task 2) as a binary classification task. This differs from Task 1 and provides a new angle of evaluation. Like [1, 3], we use leave-one-out validation to validate our model.

CIAW The Contacts in a Workplace (CIAW) dataset contains the temporal network of contacts between individuals measured in an office building in France during the 10 work days from June 24 to July 3, 2013. Participants wear a body sensor that detects other sensors within 1.5 meters. A time-stamped contact would be recorded if the proximity is maintained for more than 20 seconds. Each participant belongs to one of the five departments, which we regard as the “community” ground truth. The goal is to detect people’s community by examining their interactions during that period of time. It can be downloaded from http://www.sociopatterns.org/datasets/contacts-in-a-workplace/

The contact records were split into 20 time intervals, each covering about half a day. Multiple contacts within one interval lead to edges with higher weight. Since the contact are mutual, undirected graphs were constructed. In terms of node-level features, we stick with [2]’s features which are essentially node embeddings generated by running DeepWalk [3] on each graph snapshot.

Since no previous work was done on this dataset, we adapt our generic baselines for comparison: CD-GCN, EvolveGCN, and GCRN. Since there is only one single dynamic network constructed upon the 92 people’s interaction, in our evaluation process the train-test split is no longer conduct across the complete dynamic networks but across nodes (i.e. people) within the single dynamic network. We take $K = 10$ for K-fold validation to evaluate our model.

B HYPERPARAMETERS TUNING

In this section, we provide additional details of the hyperparameter values or search range to tune our model and baselines.

B.1 TEDIC (proposed)

time resolution $\Delta$: 0.33s, 1s, 3s, 15s, 60s; batch size: 32; learning rate: 1e-4; dropout: 0.1, 0.3, 0.5, 0.7; L2 regularization: 0, 1e-4; graph diffusion steps $k$: 2, 5, 10; graph diffusion features: 32, 64; S-TCN layers: 2, 3, 4; S-TCN kernel window size: 3; S-TCN features: 32, 64

B.2 Generic Baselines

CD-GCN: graph convolution features: 32, 64; LSTM hidden features: 32, 64; Dropout: 0, 0.1, 0.3;

GCRN: “Graph CNN” features: 32, 64; LSTM features: 32, 64; Dropout: 0, 0.1, 0.3;

B.3 Task-specific Baselines

Most of the task-specific baselines have relatively limited number of hyperparameters, the majority of which were chosen for some good reason(s) based on expert knowledge. Therefore, we stick exactly to the default settings and hyperparameter search range for most of the task-specific baselines, including GDP [3], DELF [3], MKL [6], TGCN-L [28], and LiarRank [2].

There are two exceptions where we made slight adaptation to our baselines: (1) For DDV [45], since our data does not contain transcripts from people, we remove this modality from the “late fusion” stage in [45]. We did not have the micro-expressions used in [45] either, so we replace these features with the Facial Action Units extracted by OpenFace [5]. (2) For FacialCues [18], we use a Random Forest with 50 trees on the features extracted from [18].

C COMPLETE PERFORMANCE TABLE