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

Yanbang Wang*
ywangdr@cs.stanford.edu
Stanford University

Pan Li*
panli@purdue.edu
Purdue University

Chongyang Bai
cy@cs.dartmouth.edu
Dartmouth College

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 on 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 **TE**mporal network-**DI**ffusion **C**onvolutional 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. Our project website is: http://snap.stanford.edu/tedic/.

ACM Reference Format:
Yanbang Wang*, Pan Li*, Chongyang Bai, and Jure Leskovec. 2021. TEDIC: Neural Modeling of Behavioral Patterns in Dynamic Social Interaction Networks. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3450096

1 INTRODUCTION

Social interactions, referring to numerous and complicated actions among two or more people, have woven themselves into every piece of daily life [39]. 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 [28]. Specifically, where, when and how people interact with others provide informative cues for deception detection [2, 12], dominance identification [3, 6], personality traits characterization [34] 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 [37], 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, 34] and rely on domain knowledge in social science and psychology (*e.g.,* visual dominance ratio [14], emotions and deception [47]). 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 [29]. However, they have

*Equal contribution.
WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450096

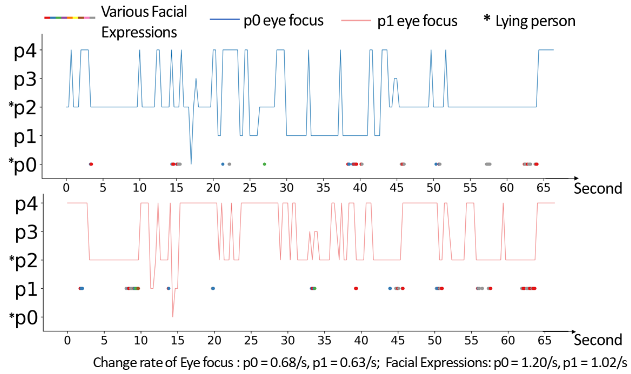

Figure 1: Visualization of two people's (p0 and p1) various behavioral traits in a social-interaction game "RESISTANCE" among 5 people p0-p5. Their eye focuses are used to built up the dynamic network and their facial expressions are used as node attributes. Both are interleaved and change over time in a highly dynamic fashion.

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, 27, 33, 45, 54].

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 ($\gtrsim$ 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 [20, 21, 32, 41, 42].

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) [24] 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, 47]. 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. 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, 12, 13], voice pitch and energy [6], or combination of multiple types of such features [3, 15, 22, 34]. 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, 17, 20, 27, 33, 38, 44, 45, 51, 54, 55]. Among them, it is worth mentioning that [17, 38, 51] explicitly look into the structural patterns on dynamic networks and provide many insights. However, they are not well-suited for our prediction tasks as the patterns does not incorporate dynamics of the attributes. Works that were claimed to digest dynamic node attributes all work on networks snapshots [21, 23, 32, 41–43]. [23, 43] study how to split the network into snapshots based on edge count or

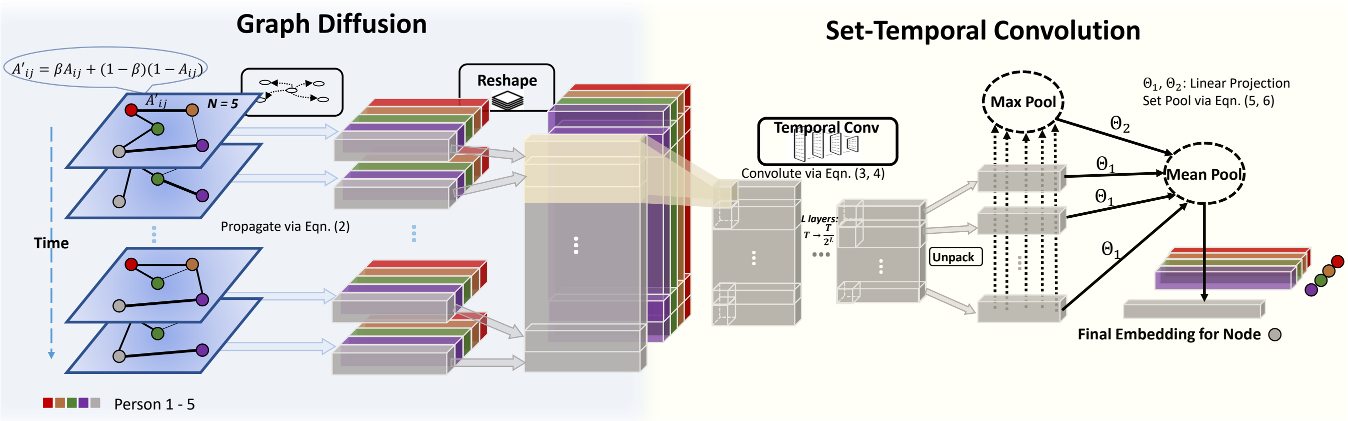

Figure 2: Temporal Network-Diffusion Convolutional Network (TEDIC). The Graph Diffusion component captures the interaction patterns among people at any given time snapshot; its Set-Temporal Convolution further filters, transforms, and aggregates important signals over time in a hierarchical manner.

structural maturity. After obtaining the snapshots, existing works 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 [24] or graph attention networks [46]. The non-linearity in propagation therein prevents the model from learning long-hop interweaving between node attributes and edges (interactions) and is not good for model interpretation [52]. The second step uses either variants of RNNs [21, 32, 42] or attention mechanism [41] to aggregate node representations, which limits the memory capacity and therefore cannot process a sequence of typically more than 100 snapshots. On the contrary, our model is robust to process more than 1000 snapshots. Moreover, although these works can process dynamic node attributes, they have not been evaluated in the settings with highly dynamic node attributes as those in dynamic social interaction networks.

3 PROBLEM DEFINITION

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 $(u, v) \in E \Leftrightarrow (v, u) \in E$. 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 $D \in \mathbb{R}^{N \times N}$ 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 $Y_t = X_t * C_t \triangleq \sum_{\tau \in \mathbb{Z}} X_{t-\tau} C_\tau$, 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*: $\{G_t\}_{1 \le t \le T}$ 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*: $\{X_t\}_{1 \le t \le T}$, where the row of X_t corresponding to node u, $X_{t,u}$, denotes intial 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, **TE**mporal network-**DI**fusion **C**onvolutional 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 β 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 [26], and some deceivers may tend to be abnormally quiet in front of others [47] due to their low-level self-confidence. However, different phenomena could happen between a follower and his leader [48]. 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 $\bar{A} = 11^T - A$ 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

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

Apparently, this parameter β can have implications: a greater β suggests making interaction is more informative to a prediction task, while a smaller β 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 Γ_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 Γ_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

$$H_t = \sum_{k\geq 0} H_t^{(k)}\Gamma_k = \sum_{k\geq 0} (W_t'^T)^k H_t^{(0)}\Gamma_k, \quad H_t^{(0)} = f(X_t) \quad (2)$$

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: $5 \sim 10$ 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: $\gamma_{k,q} \propto \alpha^k$ corresponds to PageRank [35]; $\gamma_{k,q} \propto h^k/k!$ 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 [30], allows for bigger representation power to cover multiple prediction tasks. Moreover, for model self-explanation, as W'^T is column stochastic, it will keep the ℓ_1-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 Γ_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 [24]. 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-explaining 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 [25, 49]. 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 **S**et-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

$$\bar{Z}_t^{(l)} = \text{ReLU}(Z_t^{(l-1)} * C_t^{(l)}), \quad \{Z_t^{(0)}\}_{1\leq t\leq T} = \{H_t\}_{1\leq t\leq T} \quad (3)$$

$$Z_t^{(l)} = \text{max-pool}(\{\bar{Z}_{2t-1}^{(l)}, \bar{Z}_{2t}^{(l)}\}), \quad \text{for } 1 \leq l \leq L \quad (4)$$

, where $*$ 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 networks (See Fig. 3)). The success of TCN in our setting comes from its clear and flexible receptive fields. If the size of max-pooling kernel is two as used in Eq. 4, then neurons in the last (L-th) convolution layer can perceive the signals with length 2^L. The size of receptive field is decided by two important points: **(1) Signal Denoising.** 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 of the network for signal smoothing; **(2) Temporal feature extraction from well-defined "locality".** By tuning the number of layers, one can actively search for the optimal receptive field length to gather meaningful features. Such length is also an important reference for us to understand individuals' interaction.

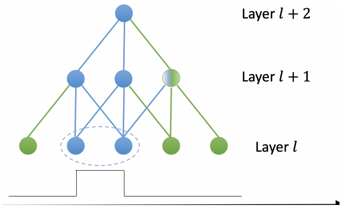

Figure 3: Receptive field of temporal convolution: The interaction happened at the two blue timestamps in layer l is captured by the blue timestamps in layers $l+1$ and $l+2$ through convolution operation.

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 over the obtained sequence $\{Z_t^{(L)}\}_{1\leq t\leq T^{(L)}}$ to extract messages scattered within this long sequence. We observe that the following is generally effective across different applications:

$$Z_{\max} = \text{max-pool}_{1\leq t\leq T^{(L)}}(Z'_t), \quad Z'_t = Z_t^{(L)} \quad (5)$$

$$Z_{\text{out}} = \text{mean-pool}_{1\leq t\leq T^{(L)}}(\text{ReLU}(Z'_t\Theta_1 + Z_{\max}\Theta_2)) \quad (6)$$

First, we impose the max pooling Eq.(5) on $\{Z_t^{(L)}\}_{1\leq t\leq T^{(L)}}$ 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 [53] 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 process where the final prediction for each person is made by *explicitly* considering all people's representations in the interaction event. In other words, the probability of each person being our target of interest should be conditioned on *both* the person's representation *and* the whole interaction context which involves all people's representations. Therefore, we use the readout layer:

$$Z_{out}^i = Z_{out}^i\Theta_3 + \text{mean-pool}_{1\leq j\leq N}(Z_{out}^j)\Theta_4, \ \forall\ 1 \leq i \leq N, \quad (7)$$

where Z_{out}^i is the i-th row of Z_{out} and Θ_3, Θ_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 many channels people use to convey messages. We briefly summarize them 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 camera in the front. The sum of each person i's Gazing Prob. towards all targets is 1. These features are extracted every 1/3 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

No.	Task	Dataset	Classification	Networks	Avg. Time Steps*	Group Size	Interactions†
1	Dominance (R)	RESISTANCE-D	multiclass	956	2,514	$5 \sim 8$	4.007×10^6
2	Dominance (E)	ELEA	binary	27	2,545	$3 \sim 4$	6.474×10^3
3	Deception	RESISTANCE-S	binary	2,157	2,258	$5 \sim 8$	2.439×10^7
4	Nervousness	RESISTANCE-N	multiclass	1,097	2,528	$5 \sim 8$	4.910×10^7
5	Community	CIAW	multiclass	1	20	92	2.149×10^4

Table 1: Statistics of the dynamic network datasets. $*$: The time steps are before coarsening (with time granularity $\Delta = 0.33$s). †: We count all the interactions with gazing probability ≥ 0.5.

participants and the dynamic edges are weighted by the gazing probabilities that participants look at each other over time.

With the extracted features, 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 Δ seconds along the time axis. Δ 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 Δ'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-given 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 spy in each game, we regard this task as a *binary* classification problem on each person.

Dataset: ELEA. The dataset [40] is a widely used public benchmark for modeling and detecting people's dominance [3]. In each video, 3-4 participants performed a "winter survival task" by having collaborative discussions. External annotators watch game videos 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 handcrafted 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 [19] is a method leveraging the facial action units from [5].

Deception Detection. DDV [50] is a method combining handcrafted micro facial expression with NLP features. TGCN-L [31] 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 [32] is one of the latest methods on dynamic graph classification tasks. It combines a skip-connected GCN with a returning sequence LSTM. EvolveGCN [36] 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. GCRN [42] 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.

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 Δ 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 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, 4 . Both are scenarios 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 [31]), Fisher Vector (FacialCues [19]), histogram encoding (DELF [3]), or many-to-one LSTM (GCRN [42]). 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 RESISTANCE-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 [32, 36, 42] 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.

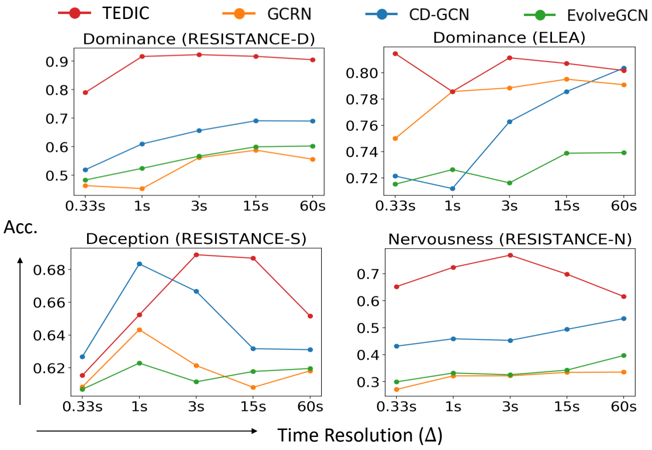

Figure 4: Accuracy as a function of time resolution of the interaction sequence.

5.3 Model Interpretation

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

Interpretation I: Balancing Weight β. Recall that β (Eq.) is the learnable parameter that directly controls the relative importance of proactive interaction versus avoidance of interaction. Fig. 5 displays how the β converges during the training (initialized to 0.5, i.e. neutral). For each task we ran multiple times by introducing small perturbation to β's initialization. The figure shows that the parameter exhibits very different convergence behavior across different tasks. For the deception detection task, β significantly drops to around 0.2, which indicates that avoidance of interaction may

Task Method		Dominance (R)	Dominance (E)	Deception	Nervousness
Task-specific	Top-1 Method	0.918±0.013	0.769±0.019	0.668±0.021	0.733±0.022
	Top-2 Method	0.887±0.015	0.677±N/A	0.638±0.016	0.729±0.015
Generic	CD-GCN[32]	0.687±0.042	0.794±0.022	0.673±0.018	0.534±0.084
	GCRN[42]	0.587±0.096	0.795±0.032	0.643±0.045	0.336±0.104
	EvolveGCN[36]	0.602±0.061	0.739±0.077	0.623±0.042	0.397±0.099
Proposed	TEDIC	**0.923**±0.009	**0.815**±0.019	**0.689**±0.012	**0.769**±0.023

Table 2: Accuracy of detecting dominance, deception and nervousness. Mean Accuracy ± 95% confidence interval is reported.

Ab.	Replacement	Accuracy
1	Original	0.923±0.009
2	S-TCN ⟶ LSTM	0.758±0.009
3	S-TCN ⟶ Mean Pool	0.842±0.023
4	Diff. ⟶ None	0.829±0.019
5	Diff. ⟶ GCN-1 Layer	0.844±0.020
6	Diff. ⟶ GCN-2 Layer	0.889±0.015
7	Diff. ⟶ GCN-4 Layer	0.784±0.026
9	Freeze $\beta = 1$	0.889±0.014

Table 3: Ablation study on Task 1.

be much more important than contacts to detect deception. Interestingly, this phenomenon coincides with findings from a psychological study [37] on eye movement of people in various contexts. Moreover, for the dominance identification task dominant, the convergence to a large β 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 β. Analysis of β indeed provides some information related to people's social status and their willingness of making social interactions.

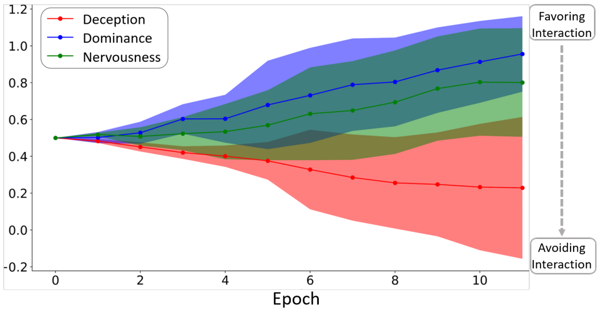

Figure 5: Different evolution behaviors of β during training, 95% confidence intervals shaded. Trained on RESISTANCE-D,-S,-N respectively.

Interpretation II: Diffusion Weights $\{\Gamma_k\}$. Recall from Eq. 2 that $\{\Gamma_k\}_{0\le k\le K}$ is a sequence of diagonal matrices where $\Gamma_k \in \mathbb{R}^{M'\times M'}$ 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 ($\Gamma_k < 0$ for node in-degree v.s. $\Gamma_k > 0$ for node self-loop degree). We attribute such distinction to different implications of those two features: The

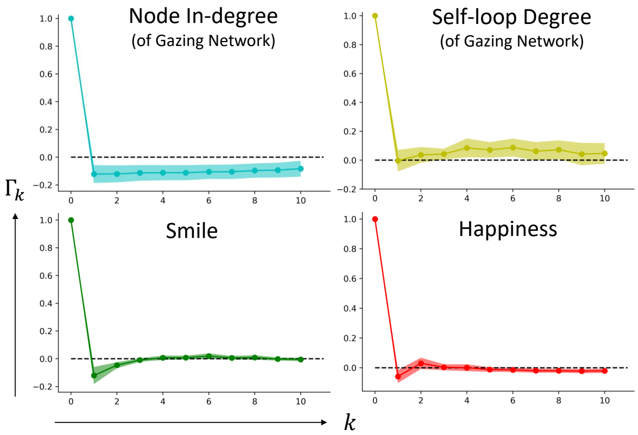

Figure 6: Diffusion weights of four features for the nervousness detection task (95% confidence interval shaded).

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: If a person smiles, the "smile" may tell something about the nervousness of the person itself; meanwhile, the likelihood of the nervousness of other people, who indirectly interact with the person through long "gaze chains", is barely affected.

5.4 Fairness of the Model

Given our task's nature, we paid special attention to TEDIC's societal implications. We hope to ensure that it does not discriminate against people from certain communities. People involved in this study can be roughly identified into 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

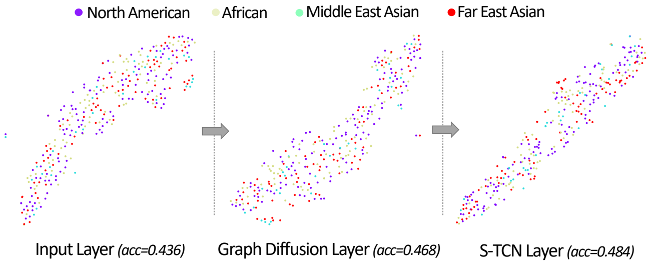

Figure 7: Projected embeddings output by different layers of our model with the corresponding regional identities.

Identity	N. American	African	Mid. East Asian	Far East Asian
%Lying Pred.	28.6±2.1	27.1±2.2	28.5±2.4	26.6±2.5

Table 4: Percentage of people predicted lying by our model.

or cultural background; **2)** How much bias our model may introduce to the decision process. For this case study we 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

Method	Accuracy
CD-GCN	0.819±0.021
GCRN	0.601±0.035
EvolveGCN	0.912±0.013
TEDIC	**0.929**±0.011

Table 5: Performance on CIAW.

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 [18]. 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.

Our method can also be applied to other real-world dynamic systems with similar properties as the social interaction networks. For example, the financial network among traders and companies in the stock market are usually high-frequent with strong local dynamic patterns and interweaving influence among nodes. Our Graph Diffusion and S-TCN modules can be easily adapted for node prediction tasks over such networks such as fraudulence detection. Traffic networks also have high-frequent traffic dynamics and complex relationships among traffic nodes. Therefore, TEDIC may be used to forecast the conditions of traffic nodes such as next-day flow volume and/or congestion level. In computing systems, communications between different softwares also share similar properties. Therefore, we can adapt TEDIC help with the malware detection task in software communication networks.

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.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of DARPA under Nos. N660011924033 (MCS); ARO under Nos. W911NF-16-1-0342 (MURI), W911NF-16-1-0171 (DURIP); NSF under Nos. OAC-1835598 (CINES), OAC-1934578 (HDR), CCF-1918940 (Expeditions), IIS-2030477 (RAPID); Stanford Data Science Initiative, Wu Tsai Neurosciences Institute, Chan Zuckerberg Biohub, Amazon, JPMorgan Chase, Docomo, Hitachi, JD.com, KDDI, NVIDIA, Dell, Toshiba, and UnitedHealth Group. J. L. is a Chan Zuckerberg Biohub investigator.

REFERENCES

[1] Oya Aran and Daniel Gatica-Perez. 2013. One of a kind: Inferring personality impressions in meetings. In *Proceedings of the 15th ACM on International conference on multimodal interaction.* 11–18.

[2] Chongyang Bai, Maksim Bolonkin, Judee Burgoon, Chao Chen, Norah Dunbar, Bharat Singh, VS Subrahmanian, and Zhe Wu. 2019. Automatic Long-Term Deception Detection in Group Interaction Videos. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1600–1605.

[3] Chongyang Bai, Maksim Bolonkin, Srijan Kumar, Jure Leskovec, Judee Burgoon, Norah Dunbar, and VS Subrahmanian. 2019. Predicting dominance in multi-person videos. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 4643–4650.

[4] Chongyang Bai, Srijan Kumar, Jure Leskovec, Miriam Metzger, Jay F. Nunamaker, and V. S. Subrahmanian. 2019. Predicting the Visual Focus of Attention in Multi-Person Discussion Videos. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4504–4510. https://doi.org/10.24963/ijcai.2019/626

[5] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 2016. Openface: an open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1–10.

[6] Cigdem Beyan, Francesca Capozzi, Cristina Becchio, and Vittorio Murino. 2017. Prediction of the leadership style of an emergent leader using audio and visual nonverbal features. *IEEE Transactions on Multimedia* 20, 2 (2017), 441–456.

[7] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1082–1090.

[8] Fan Chung. 2007. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences* 104, 50 (2007), 19735–19740.

[9] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Deep coevolutionary network: Embedding user and item features for recommendation. *arXiv preprint arXiv:1609.03675* (2016).

[10] Steven Davis and Paul Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing* 28, 4 (1980), 357–366.

[11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*. 3844–3852.

[12] Sergey Demyanov, James Bailey, Kotagiri Ramamohanarao, and Christopher Leckie. 2015. Detection of deception in the mafia party game. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. 335–342.

[13] Sergey Demyanov, James Bailey, Kotagiri Ramamohanarao, and Christopher Leckie. 2015. Detection of Deception in the Mafia Party Game. In *ACM ICMI*.

[14] John F Dovidio and Steve L Ellyson. 1982. Decoding visual dominance: Attributions of power based on relative percentages of looking while speaking and looking while listening. *Social Psychology Quarterly* (1982), 106–113.

[15] Nathan Eagle, Alex Sandy Pentland, and David Lazer. 2009. Inferring friendship network structure by using mobile phone data. *Proceedings of the national academy of sciences* 106, 36 (2009), 15274–15278.

[16] Don Eskridge. 2012. *The Resistance: Avalon*. Indie Boards & Cards.

[17] Wenjie Fu, Le Song, and Eric P Xing. 2009. Dynamic mixed membership blockmodel for evolving networks. In *Proceedings of the 26th annual international conference on machine learning*. 329–336.

[18] Mathieu Génois, Christian L Vestergaard, Julie Fournet, André Panisson, Isabelle Bonmarin, and Alain Barrat. 2015. Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. *Network Science* 3, 3 (2015), 326–347.

[19] G Giannakakis, Matthew Pediaditis, Dimitris Manousos, Eleni Kazantzaki, Franco Chiarugi, Panagiotis G Simos, Kostas Marias, and Manolis Tsiknakis. 2017. Stress and anxiety detection using facial cues from videos. *Biomedical Signal Processing and Control* 31 (2017), 89–101.

[20] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. 2020. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems* 187 (2020), 104816.

[21] Ehsan Hajiramezanali, Arman Hasanzadeh, Krishna Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. 2019. Variational graph recurrent neural networks. In *Advances in Neural Information Processing Systems*. 10700–10710.

[22] Dinesh Babu Jayagopi, Hayley Hung, Chuohao Yeo, and Daniel Gatica-Perez. 2009. Modeling dominance in group conversations using nonverbal activity cues. *IEEE Transactions on Audio, Speech, and Language Processing* 17, 3 (2009), 501–513.

[23] Di Jin, Sungchul Kim, Ryan A Rossi, and Danai Koutra. 2020. From Static to Dynamic Node Embeddings. *arXiv preprint arXiv:2009.10017* (2020).

[24] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[25] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).

[26] Srijan Kumar, Chongyang Bai, V.S. Subrahmanian, and Jure Leskovec. 2021. Deception Detection in Group Video Conversations using Dynamic Interaction Networks. In *Proceedings of the International AAAI Conference on Web and Social Media*.

[27] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1269–1278.

[28] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. 2009. Computational social science. *Science* 323, 5915 (2009), 721–723.

[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[30] Pan Li, I Chien, and Olgica Milenkovic. 2019. Optimizing Generalized PageRank Methods for Seed-Expansion Community Detection. In *Advances in Neural Information Processing Systems*. 11705–11716.

[31] Yozen Liu, Xiaolin Shi, Lucas Pierce, and Xiang Ren. 2019. Characterizing and Forecasting User Engagement with In-app Action Graph: A Case Study of Snapchat. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2023–2031.

[32] Franco Manessi, Alessandro Rozza, and Mario Manzo. 2020. Dynamic graph convolutional networks. *Pattern Recognition* 97 (2020), 107000.

[33] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. 2018. Continuous-time dynamic network embeddings. In *Companion Proceedings of the The Web Conference 2018*. 969–976.

[34] Shogo Okada, Laurent Son Nguyen, Oya Aran, and Daniel Gatica-Perez. 2019. Modeling dyadic and group impressions with intermodal and interperson features. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 1s (2019), 1–30.

[35] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The pagerank citation ranking: Bringing order to the web*. Technical Report. Stanford InfoLab.

[36] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E Leisersen. 2019. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *arXiv preprint arXiv:1902.10191* (2019).

[37] Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin* 124, 3 (1998), 372.

[38] Ryan A Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. 2013. Modeling dynamic behavior in large evolving graphs. In *Proceedings of the sixth ACM international conference on Web search and data mining*. 667–676.

[39] Rudolph J Rummel. 1976. Understanding conflict and war: vol. 2: the conflict helix. *Bev-erly Hills: Sage* (1976).

[40] Dairazalia Sanchez-Cortes, Oya Aran, Marianne Schmid Mast, and Daniel Gatica-Perez. 2011. A nonverbal behavior approach to identify emergent leaders in small groups. *IEEE Transactions on Multimedia* 14, 3 (2011), 816–832. Dataset: https://www.idiap.ch/dataset/elea.

[41] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2020. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 519–527.

[42] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*. Springer, 362–373.

[43] Sucheta Soundarajan, Acar Tamersoy, Elias B Khalil, Tina Eliassi-Rad, Duen Horng Chau, Brian Gallagher, and Kevin Roundy. 2016. Generating graph snapshots from streaming edge data. In *Proceedings of the 25th International Conference Companion on World Wide Web*. 109–110.

[44] Aynaz Taheri, Kevin Gimpel, and Tanya Berger-Wolf. 2019. Learning to Represent the Evolution of Dynamic Graphs with Recurrent Models. In *Companion Proceedings of The 2019 World Wide Web Conference*. 301–307.

[45] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2019. DyRep: Learning Representations over Dynamic Graphs. In *International Conference on Learning Representations*.

[46] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[47] Aldert Vrij. 2008. *Detecting lies and deceit: Pitfalls and opportunities*. John Wiley & Sons.

[48] Fred O Walumbwa and John Schaubroeck. 2009. Leader personality traits and employee voice behavior: mediating roles of ethical leadership and work group psychological safety. *Journal of applied psychology* 94, 5 (2009), 1275.

[49] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. 2019. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153* (2019).

[50] Zhe Wu, Bharat Singh, Larry S Davis, and VS Subrahmanian. 2018. Deception detection in videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[51] Eric P Xing, Wenjie Fu, Le Song, et al. 2010. A state-space mixed membership blockmodel for dynamic network tomography. *Annals of Applied Statistics* 4, 2

(2010), 535–566.

[52] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*. 9240–9251.

[53] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. Deep sets. In *Advances in neural information processing systems*. 3391–3401.

[54] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. 2018. Dynamic network embedding by modeling triadic closure process. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[55] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. 2018. Embedding temporal network via neighborhood formation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2857–2866.

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 know 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 median scores. Ties are broken by further comparing the mean score. 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-given by the dataset. By the game's setting, it is presumable that all spies 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 [40] is a 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.

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 timestamped 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 Δ: 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;
EvolveGCN: graph convolution features: 32, 64; GRU layers: 1, 2; 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-sepecific baselines, including GDP [3], DELF [3], MKL [6], TGCN-L [31], and LiarRank [2].

There are two exceptions where we made slight adaptation to our baselines: **(1)** For DDV [50], since our data does not contain transcripts from people, we remove this modality from the "late fusion" stage in [50]. We did not have the micro-expressions used in [50] either, so we replace these features with the Facial Action Units extracted by OpenFace [5]. **(2)** For FacialCues [19], we use a Random Forest with 50 trees on the features extracted from [19].

C FULL PERFORMANCE TABLE

See Tab. 6.

Task Group	Dominance (R)		Dominance (E)		Deception		Nervousness	
Task-specific Baselines	GDP-MLP [3]	0.918	GDP-MLP	0.800	DDV [50]	0.632	FacialCues	0.733
	GDP-RF [3]	0.848	GDP-RF	0.730	LiarRank [2]	0.668	LiarRank	0.729
	DELF [3]	0.887	DELF	0.769	TGCN-L [31]	0.550	-	-
	MKL [6]	0.879	MKL	0.677	-	-	-	-
	FacialCues [19]	0.746	FacialCues	0.702	-	-	-	-
Generic Baselines	CD-GCN[32]	0.687	CD-GCN	0.794	CD-GCN	0.673	CD-GCN	0.534
	GCRN [42]	0.587	GCRN	0.795	GCRN	0.643	GCRN	0.336
	EvolveGCN [36]	0.602	EvolveGCN	0.739	EvolveGCN	0.623	EvolveGCN	0.397
Proposed	TEDIC	**0.923**	TEDIC	**0.815**	TEDIC	**0.689**	TEDIC	**0.769**

Table 6: Full performance table.

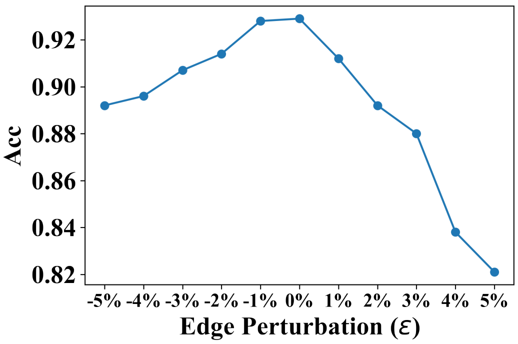

Figure 8: Performance under different level of noise on CIAW.

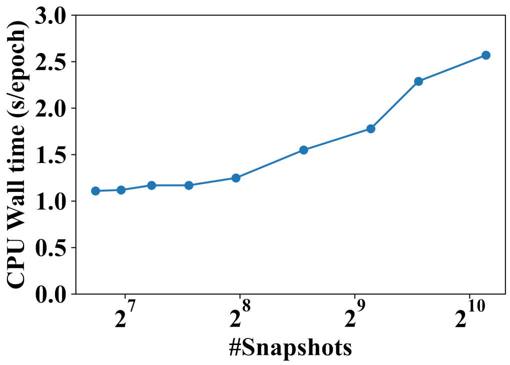

Figure 9: CPU Wall time to train one epoch when different number of snapshots are used. The experiment is run on RESISTENCE-S.

D MORE EXPERIMENT RESULTS

D.1 Robustness

Interaction networks built on data collected from sensors tend to contain noise. For example, in CIAW an interaction may be missed or mistakenly recorded by proximity sensors. We further investigate our TEDIC's robustness to such noise. The general procedure is that we randomly perturb the network structure up to a certain level and check how the performance gets affected. We use $|E|$ to denote the total number of interactions (edges) in the whole network. The noise rate is denoted as ϵ. A positive ϵ means that $\epsilon|E|$ edges are randomly added to the network. A negative ϵ means that $\epsilon|E|$ edges are randomly removed from the existing edges. In our experiments we consider up to 5% of the noise rate, *i.e.* $\epsilon \leq 5\%$.

Figure 8 shows the performance curve against different ϵ's. We can see the mistakenly recorded edges have relatively more impact on the performance than the missing edges. However, even at 5% noise level, our framework still outperforms the previous SOTA of CD-GCN on the original graph (Tab. 5). We attribute such robustness to the design of TEDIC, which naturally helps denoise the features: Our graph diffusion, temporal convolution, and set pooling functions are all smoothing operations reducing high-frequent signal components.

D.2 Scalability

We further report the empirical wall time used by TEDIC when it is trained on networks with different number snapshots on average. In this experiment, the number of snapshots on average is changed by setting the time resolution Δ to different values. We run the experiment on RESISTENCE-S. Fig. 9 shows the result. We see that the time per epoch is within very manageable range even when we deal with networks whose number of snapshots is in the magnitude of thousand. In principle, the time per epoch is linear against the number of snapshots. However, the usage of TCN in TEDIC allows our method to speed up by parallelizing the convolution in temporal dimension. This is an advantage over RNN-based methods such as EvolveGCN or CD-GCN, which have to be executed in sequential order.

The experiments were carried out on a Ubuntu 16.04 server with Xeon Gold 6148 2.4 GHz 40-core CPU, Nvidia 2080 Ti RTX 11GB GPU, and 768 GB memory.