Generic Representation Learning for Dynamic Social Interaction

Yanbang Wang
ywangdr@cs.stanford.edu
Stanford University

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

Chongyang Bai
cy@cs.dartmouth.edu
Dartmouth College

V.S. Subrahmanian
vs@dartmouth.edu
Dartmouth College

Jure Leskovec
jure@cs.stanford.edu
Stanford University

ABSTRACT
Social interactions, such as eye contact, speaking and listening, are ubiquitous in our life and carry important clues of human’s social status and psychological state. With evolving dynamics fundamentally different from social relationships, the complex interactions among a group of people are another informative resource to analyze patterns of social behaviors and characteristics. Despite the great importance, previous approaches on extracting patterns from such dynamic social interactions are still underdeveloped and overly task-specific. We fill this gap by proposing a temporal network formulation of the problem, together with a representation learning framework, temporal network-diffusion convolution networks (TNDCN). The framework accommodates the many downstream tasks with a unified structure: we creatively propagate people’s fast-changing descriptive traits among their evolving gazing networks with specially designed (1) network diffusion scheme and (2) hierarchical pooling to learn high-quality embeddings for downstream tasks using a consistent structure and minimal feature engineering. Analysis show that (1) can not only capture patterns from existed interactions but also people’s avoidance of interactions that turn out just as critical. (2) allows us to flexibly collect different fine-grained critical interaction features scattered over an extremely long time span, which is also key to success while it empirically fails almost all the previous temporal GNNs based on recurrent structures. We evaluate our model over three different prediction tasks, detecting deception, dominance and nervousness. Our model not only consistently outperforms previous baselines but also provides good interpretation by implying two important pieces of social insight derived from the learned coefficients.

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

1 INTRODUCTION
Social interactions, referring to numerous and complicated actions among two or more people, have woven themselves into every piece of our daily life [33]. They are one of the most ubiquitous forms that people connect with each other, and can happen whenever several people meet physically or online with video meeting tools. Such interactions, featured by synchronously occurred high-frequent eye contacts, fast-changing facial expressions, voice features and even physical proximity, have evolving dynamics fundamentally different from acquaintance-based social networks.

With such informative signals that characterize complex human group behaviors, psychological state and social-economic status, social interactions become critical data resources for social scientists to study patterns of human behaviors and make further inferences [25]. For example, where, when and how people interact with others may provide informative cues to solve various social tasks including deception detection [4, 13], dominance identification [5, 8], personality traits characterization [30] and friendship inference [16].

However, we recognize that existing literature [4, 5, 8, 14, 15, 41] on each of these tasks primarily rely on handcrafted features that are hardly transferable to each other. This makes the modeling process overly task-specific and high-demanding for domain expertise. In that regard, we propose a more generic framework that formulates social interactions with such vigorous dynamics by a temporal network for more unified representation learning. Central to this prototype is the usage of people’s evolving eye-gazing or speak-and-listen-to relationships for temporal network construction. With eye contacts as the key to signal information flows [36] in a social interaction, both the explicit messages and people’s underlying influence on each other can now be naturally modeled into a graph diffusion process, which essentially instantiates a variant of the powerful (temporal) graph neural network [23]. We term this temporal network model dynamic social interaction network.

There are several temporal GNN frameworks proposed recently for representation of generic temporal networks. However, they are not well-suited to our downstream tasks. One important reason is that they are primarily designed to fit the occurrence/attributes of temporal edges and thus almost always place an imbalanced weight on events towards the sequence’s end [11, 29, 39, 47]. However, our tasks (deception detection for example) essentially need to collect the different, sporadic and potentially overlapping traits of communication throughout the interaction. Such mismatch becomes especially noticeable with long, fine-grained interaction sequences.
We further denote the universal node set works \([21, 28, 35, 37]\). A static network can be represented as a graph \(A\). 2 PROBLEM FORMULATION

Our contribution in this paper is summarized as the following:

- We propose the formulation of social interactions into a temporal network prototype to enable unified representation learning for the many downstream tasks with minimal level of knowledge-based feature engineering.
- We propose an end-to-end neural-network-based model, temporal network-diffusion convolution networks (TNDCN) that both intuitively model the information flow with enhanced graph convolution and flexibly collect scattered fine-grained patterns over long time with special hierarchical pooling.
- We evaluate TNDCN over three different social tasks including deception, dominance and nervousness detection. Our model consistently outperforms a variety of our baselines. In-depth analysis shows that the learnt coefficients further yield interesting insights on interaction patterns.

2 PROBLEM FORMULATION

In this section, we first introduce notations of static graphs in general. Then, we introduce our problem into the context by further formulating it into the prototype of temporal graphs. We use capital letters \(N, M, M'\) to denote some positive integers.

2.1 General Graph Notations

A static network can be represented as a graph \(G = (V, E)\) where \(V\) denotes the set of nodes and \(E\) the set of edges, \(N = |V|\). Networks that we discuss include both directed and undirected. As an undirected graph can be viewed as a special case of directed one, we assume \(G\) is directed hereafter unless specified.

Graph \(G\) is associated with adjacency matrix \(A \in \mathbb{R}^{N \times N}\) where \(A_{uv} = 1\) if and only if \((u, v) \in E\). Also, we define the diagonal out-degree matrix \(D_{\text{out}} \in \mathbb{R}^{N \times N}\) where its \(u\)-th diagonal component is \(d_{\text{out}, u} = \sum_{v \in V} A_{uv}\). The random walk matrix over \(G\) is defined as

\[
W = D_{\text{out}}^{-1}A.
\]

2.2 Dynamic Social Interaction Networks

Central to the prototype’s formulation are two things: nodes, which represent people, and timestamped edges, which represent interactions between two people and are usually mapped from people’s evolving eye-gazing or speak-listen relationships. Dynamic personal traits are further associated with nodes and communication-based properties like gazing probabilities are associated to edges.

To record social interactions, the most common practice is to leverage a variety of sensors to record snapshots of interaction scenes of high time resolution. Therefore, we define our data structures using temporal graph snapshots: \(\{G_t\}_{1 \leq t \leq T}\) where \(G_t = (V_t, E_t)\). We further denote the universal node set \(V = \bigcup V_t, \forall t \in [1, T]\), so that \(V\) is fixed across different snapshots.

As mentioned, dynamic social interaction networks can be associated with both dynamic node and/or edge attributes. and dynamic edge attributes. For node attributes, we have \(\{X_t\}_{1 \leq t \leq T}\), where the row of \(X_t\) corresponding to node \(u\)-s initial attributes. For edges, we also allow the network edge associated with a quantified attribute, denoted by a weighted adjacency matrix \(A\) in generalized form. For example, an edge can carry the likelihood of one person looking at the other, or the wireless signal strength that one’s smart device receives from the other’s. Generalization to multi-type edges yields multi-view temporal network, which beyond our discussion of this work and left for future work. Note that our definition allows both edge and node attributes to evolve dynamically though.

Our goal is to learn representations for nodes in these networks that capture important patterns from their social interaction behaviors. Once the representations are learned, prediction/inference for certain tasks can be accomplished by feeding these representations into inference blocks for downstream tasks. We claim our approach can be used for general node-level prediction tasks that require patterns to be extracted from dynamic social interaction networks. We demonstrate this capability by considering the following three tasks: detection lying people, dominant people, and nervous people in a social interaction event. The specific inference blocks and training objectives will be specified in Section 5.

3 PROPOSED MODEL

Our model temporal network-diffusion convolutional network (TNDCN) consists of two main components, Network Diffusion of node attributes, and Set-Temporal Convolution-based hierarchical pooling over time, as shown in Figure 1. The obtained node embeddings will be further fed into a simple task-driven output block to either compute loss (during training) or make inference. The loss is specified by tasks so we introduce it in Section 5.

3.1 Network Diffusion Component

We propose using network diffusion process to most intuitively model the information flows carried by interactive behaviors in our dynamic network: given people’s personal traits in a certain snapshot quantified by node attributes \(X^{(0)} \in \mathbb{R}^{N \times M}\), the \(k\)-hop network diffusion can be written as

\[
X^{(k)} = (W^T)^k X^{(0)}
\]

where \(W^T\) is the transpose of \(W\). This process is further enhanced by two sets of parameters:

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 [32], and some deceivers may tend to be abnormally quiet in front of others [41] due to their low-level self-confidence. However, different phenomena could happen between a follower and his leader [42]. 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 \(\bar{A} = I - A\) where \(I\) is the

\[\footnote{We assume the components of \(A\) are normalized within interval \([0, 1]\).} \]
identity matrix. Then, we introduce another parameter $\beta \in [0, 1]$ to merge these two networks to obtain a new adjacency matrix via

$$A' = \beta A + (1 - \beta)(1 - A).$$

(3)

Apparently, this parameter $\beta$ have physical implication. A greater $\beta$ suggests making interaction is more informative to the prediction task, while a smaller $\beta$ emphasizes that avoiding interaction may be the key clue. From now on, we will diffuse node attributes over the random walk matrix derived from this parameterized network $A'$ based on (1), i.e., replace $W$ in Eqn. 2 by $W' = D^{-1}A'$. Admittedly, using graph would lead to a densely connected network which can lead to large computation cost in large graphs. However, this issue would not appear here since dynamic social interactions in our context can hardly involve people of more than hundreds in a single interaction event.

Parameter $\Gamma_k$ for different-hop interactions. The model is now to perform different-step network diffusion. 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 $X^{(0)} \in \mathbb{R}^{N \times M}$ based on network diffusion as

$$H = \sum_{k \geq 0} H^{(k)} \Gamma_k = \sum_{k \geq 0} (W')^k H^{(0)} \Gamma_k, \quad H^{(0)} = f(X^{(0)})$$

(4)

where $f(\cdot) : \mathbb{R}^{N \times M} \to \mathbb{R}^{N \times M'}$ 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 $y_{k,q}$, is the weight for $q$’s output channel. In practice, typically only the first several hops could be informative so we may set a upper bound to the number of hops: $5 \sim 10$ steps provide good enough results in practice.

Note that the formulation (4) has many implications. Consider the sequence $\{y_{k,q}\}_{k \geq 0}$ for any $q$ and suppose $f$ is identity mapping. From the perspective of graph spectral convolution, $\{y_{k,q}\}_{k \geq 0}$ corresponds to weights on different levels of the smoothness of $q$-th node attributes. Moreover, different fixed formulations of $y_{k,q}$ also provide different types of ranks of nodes: $y_{k,q} \propto \Delta^k$ corresponds to PageRank [31]; $y_{k,q} \propto \frac{h^k}{k}$ corresponds to heat-kernel PageRank [10]. Extensive feature engineering shows that different formulations of ranks could be important signals to detect deceivers or leaders among group of people [4, 5]. Our formulation based on learnable parameters allows for more representation power to cover multiple prediction tasks. Moreover, for model interpretation, as $W'^T$ is column stochastic, it will keep the $\ell_1$-norm of every column of $H^{(k)}$ unchanged (with non-negative features) and thus naturally hold normalizing property. Therefore, the value $|y_{k,q}|$ and the sign of $y_{k,q}$ are naturally interpreted as the effect of $k$-hop diffusion of $q$-th node attribute to the final representation. Even when we choose $f$ as 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 for model interpretation.

Note that there could be variants of (4) 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 of graph convolution networks [23]. However, adding non-linearity per step increases the difficulty for training, which limits the steps of propagation to 2-3, and could simultaneously hurt the interpretation of models. 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 [24, 43]. However, to the best of our knowledge, we are the first to show the success of this manner to process dynamic networks.

3.2 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. As mentioned, there are two challenges in designing the block: being able to handle a extremely long sequence of snapshots because of the high time resolution, and being able to capture local dynamics typically subtle and scattered randomly in the whole time span. Correspondingly, our S-TCN block consists of two components: multi-layer temporal convolution and set pooling.

Multi-layer Temporal Convolution. The input of this block is a sequence of node features $\{H_t\}_{t \leq T}$ where $H_t \in \mathbb{R}^{N \times M'}$ denotes the node features for each snapshot $t$ obtained via (4). The $l$-th
Set Pooling. As opposed to online social networks that often show seasonal patterns, there are seldom periodical patterns in offline social interaction networks. Consider eye contact in conversations/meetings among a group of people. Informative patterns of interactive behaviors of people are usually randomly scattered in the long time span. Therefore, based on 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 the message scattered within this long sequence. We observe that the following form is generally effective across different applications: First, we impose max pooling (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.

\[
Z_{max} = \max\text{-pooling}_{1 \leq t \leq T^{(L)}}(Z_t^{(L)}), \quad Z'_t = Z_t^{(L)}
\]

\[
Z_{out} = \text{mean-pooling}_{1 \leq t \leq T^{(L)}}(\text{ReLu}(\Theta_1 Z_t^{(L)} + \Theta_2 Z_{max}))
\]

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 [46] for our setting.

4 RELATED WORK
The research related to our problem spans two broad areas:

Methods to Analyze Social Interactions. Lots of research has been conducted to identify human behaviors and relationship during social interactions such as leadership [8], dominance [5, 22], friendship [9], and deception [4, 14]. These works focus on designing task-specific features in short periods and aggregate them to long-term feature vectors via statistical methods. The obtained features are then fed into standard classifiers (e.g. SVM, Random Forest). These engineered features, although shown to be powerful in their corresponding tasks, are less general and often requires specific domain knowledge in social science and psychology theories (e.g. visual dominance ratio [15], emotions and deception [41]). Moreover, the hand-crafted features become more noisy when building upon raw features. To effectively aggregate features over the long temporal domain, extensive statistical methods are employed such as summation, median and variance [8, 14], histograms and bag-of-words [4, 5]. These indifferenciable aggregations make potential models untrainable. In contrast, due to our neural-network-based module, we obtain a differentiable model that connects raw networks directly to the social tasks and allows for an end-to-end training. By taking advantage of this training procedure, the model naturally learns the effect of interacting networks for various social tasks without using extensive statistical analysis.

Representation Learning for Dynamic Networks. The success of representation learning for dynamic social interaction networks strongly depends on processing the interweaving high-dynamic node attributes and interactions. So we first partition previous approaches to learn representation of dynamic networks into two categories regarding whether dynamic node attributes can be processed. We do not provide a detailed review of works unable to take dynamic node attributes including [1, 11, 20, 29, 38, 39, 47, 48]. Works that were claimed to digest dynamic node attributes all work on networks snapshots [21, 28, 35, 37]. They generally follow the pipeline by first propagating node attributes of each network snapshot and then aggregating them over time. Works [21, 28, 37] use graph convolution networks [23] for the first step while Sankar et al. [35] leverages graph attention networks [40]. For the second step, works [21, 28, 37] use RNN and its variants to aggregate node representations, while Sankar et al. [35] uses self-attention mechanism. However, all these approaches share the issue of limited memory capacity when #snapshots > 100. Moreover, despite proposed to process dynamic node attributes, they have not been evaluated in the setting with highly dynamic node attributes as those in dynamic social interaction networks.
5 EMPIRICAL STUDY

5.1 Experiment Settings

Overview. Our proposed model is evaluated on three different node-level prediction social tasks over 4 datasets:

(1) Dominance Detection: RESISTANCE-1 [6], ELEA [34];
(2) Deception Detection: RESISTANCE-2 [6];
(3) Nervousness Detection: RESISTANCE-3 [6];

Task (1) and (3) aims to identify the most dominant/nervous person in a social interaction event happening between 3-8 people. Task (2) aims to detect all the hidden liars in a social interaction event happening between 5-8 people. Therefore, we consider Task (1), (3) as a one-vs-rest classification problem, and consider Task (2) as a binary classification problem.

In its original format, each dataset is a collection of videos. Each video records a conversation whose duration ranges from around 5 to 30 minutes, with different conversation contexts, which we will introduce slightly afterwards. We preprocess each video using vision-based and audio-based techniques of various sources, which for each conversation generates around 800-4500 dynamic network snapshots (on 3 FPS, as described in Section 2.2). In the rest of this section, we will explain the dataset background, ground-truth label collection, feature preprocessing, as well as other implementation details including baselines and model tuning.

Dataset: RESISTANCE-1, 2, 3. All three dataset are a collection of videos and surveys recording 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" conducted by discussion and election, which involves frequent deception behavior (presumably only from Team B) and argument, query and persuasion (from all parties). In order to persuade, very often people tend to be dominant and avoid appearing nervous. Please see supplementary material for more background. The three dataset share about 50% videos in common while for the rest they each differ slightly due to several practical reasons such as label missing or mismatch.

Labels for RESISTANCE-1, 3 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 nervous level of the other people. We treat the median score of each person (rated by others) as its ground truth score, ties broken by further comparing the mean. Labels for RESISTANCE-2, which are Team A & B’s identity of each game, are given by the dataset. Considering the gaming rules, it is presumable that Team B (spies) are essentially lying throughout the game.

Dataset: ELEA The dataset [34] is a widely used benchmark for modeling and detecting personal traits such as dominance [5]. The dataset can be accessed here. In each video, 3-4 participants collaboratively performed a ‘winter survival task’ by having peaceful discussions. We perform dominance detection task on the dataset as other labels are unavailable.

Labels are generated in a slightly different way following the protocols of [2, 5]: we are detecting more dominant people instead of the most. This also provides a new angle of evaluation. There are two categories of dominance scores: (1) perceived dominance (DPom), which is scores rated by the game organizers who hosted and monitored the game; (2) ranked dominance (RDom), which is scores rated by game participants to each other. We assign binary dominance labels to people by thresholding their dominance scores by the median values of people in each video.

Feature Extraction. We extract the following social interaction features from videos on frequency of 3 FPS using a combination of toolkit:

(1) Emotion: intensity of eight emotions + two facial traits (smile, open eyes) by Amazon Recognition;
(2) FAU: intensity of 17 facial action units using OpenFace [7];
(3) MFCC: voice features widely used in audio analysis [12];
(4) Speak Prob.: Probability a person is speaking estimated from lip movement [17];
(5) Gazing Prob.: probability that each person looks at each other players estimated from eye movement [3]. For each person his Gazing Prob. towards other people sums up to 1.

Our dynamic social interaction network is constructed from the last feature.

Output Layer & Loss. As mentioned, we consider Task (2) a binary classification problem, so the output block for processing each person’s representation is a simple logistic regression instantiated by a densely connected neural network layer plus the Sigmoid nonlinearity.

For task (1) and (3) to select one out of a set 3-8 representations we further use the following transformation: Let \( Z_{out} \in \mathbb{R}^{N \times e} \) taken from Eqn. 6 be the learned representation of all the people in one interaction event, where \( N \in [3, 8] \) is the number of people ad \( e \) is the representation length. The \( i \)-th people’s output logit is given by:

\[
Z_i = Z_iW_1 + \text{Mean-pooling}([Z_i]_{1 \leq i \leq N})W_2, \quad \text{for } 1 \leq i \leq N \tag{7}
\]

\[
\hat{y} = \text{softmax}(ZW_3) \tag{8}
\]

where \( W_1, W_2 \in \mathbb{R}^{e \times e} \) and \( W_3 \in \mathbb{R}^{e \times 1} \) are projection matrices whose parameters are to be learned. We use cross entropy loss for all tasks and back-propagate the errors to all aforementioned learnable parameters for optimization.

Baselines. Our framework is compared with two sets of baselines. The first set is task-specific baselines proposed uniquely for each task:

- Dominance Detection: MKL [8], which is based on handcrafted features like voice pitch and speaking rate; two versions of the GDP [5] which primarily relies on their handcrafted feature called DomRank: GDP with random forest classifier (GDP-RF), and GDP with multi-layer perceptron classifier (GDP-MLP); DELF [5] also uses DomRank.
- Deception Detection: ADD [44] based on handcrafted micro facial expression and NLP features; TGCN-L [27] based on gazing probabilities, and LiarRank [4] based on all the features we used but aggregating them in a special manner;

Supplementary material: https://bit.ly/36ipjoO
ELEA dataset: https://www.idiap.ch/dataset/elea

\textsuperscript{3}ElEA dataset: https://www.idiap.ch/dataset/elea
we explain that effectively capturing the temporal cues is the key (TGCN, Logistic., Random F.), Fisher Vector encoding (FacialCues), Table 2 compares performance of our model and other baselines on 10 times using different random seed to initialize parameters.

4i.e. without any network-level operation

to success. While almost all the baselines come with proper graph convolution or careful feature engineering work, their processing of temporal information falls insufficient, either mean pooling (TGCN, Logistic., Random F.), Fisher Vector encoding (FacialCues), histogram encoding (DELF), or many-to-one LSTM (GCN-LSTM). Also notice the GCN-LSTM’s failure on most tasks, which shows the insufficiency of temporal sequence modeling techniques based on recurrent structures.

5.3 Model Interpretation

The linearity of parameters in network diffusion provides model interpretation. Next, we investigate these parameters and explain the induced social insights.

Interpretation I: Balancing Weight $\beta$. Recall from Section 3.1 that $\beta$ is the learnable parameter that directly controls the relative importance of proactive interaction versus avoidance of interaction. Figure 3 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. In particular, the deception detection task $\beta$ significantly drops to around 0.2, which indicates that avoidance of interaction may be much more important than contacts in detecting deception. Interestingly, this phenomenon coincides with findings from a psychological study [32] on eye movement of people in various contexts. It is also quite understandable that dominant people are more easily identified with their aggressive way of reaching out to people (rather than escape to do so). Nervous people, on the other hand, seem to be identified with a mixture of the two extremes. The analysis on $\beta$ shows its prediction power and high value to understanding human’s avoidance of social interactions.

Interpretation II: Diffusion Weights $\Gamma_k$. Recall from Equation 4 that $\{\Gamma_k\}_{0 \leq k \leq K}$ 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. Therefore, when we finish training we would obtain $K+1$ diffusion weights for each of the $M'$ features. Analyzing these diffusion weights provides us with important insights of how the interaction network actually helps shape the original features during the diffusion. Fig. 4 displays such weights for four of the features’ (here diffusion steps $K = 10$) after being trained on the nervousness.

Table 1: General statistics of the dynamic networks used for representation learning.

<table>
<thead>
<tr>
<th>Task No. and Task</th>
<th>Dataset</th>
<th>Dynamic Network Sequences</th>
<th>Time Steps (Avg.)</th>
<th>Nodes</th>
<th>Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Dominance Identification</td>
<td>RESISTANCE-1</td>
<td>956</td>
<td>2,514</td>
<td>4780</td>
<td>$4.007 \times 10^6$</td>
</tr>
<tr>
<td>(1) Dominance Identification</td>
<td>ELEA</td>
<td>21</td>
<td>1,350</td>
<td>84</td>
<td>$6.474 \times 10^3$</td>
</tr>
<tr>
<td>(2) Deception Detection</td>
<td>RESISTANCE-2</td>
<td>2,157</td>
<td>1,800</td>
<td>10,785</td>
<td>$2.439 \times 10^7$</td>
</tr>
<tr>
<td>(3) Nervousness Detection</td>
<td>RESISTANCE-3</td>
<td>1,097</td>
<td>1,800</td>
<td>5,485</td>
<td>$4.910 \times 10^7$</td>
</tr>
</tbody>
</table>

† We count all the interactions with gazing probability $\geq 0.5$.

5.2 Performance Comparison

Table 2 compares performance of our model and other baselines on the three tasks. It shows that our model significantly outperforms the strongest baselines in Task 2,3 and also achieves comparable result with the strongest baseline on Task 1. Notice that our best model achieves this with less than one-fifth parameters of DELF and GDP. We also notice that our model has more significant gain in the challenging tasks 2 and 3. The two tasks are challenging essentially because they are both probing for something that the interaction participants are purposely concealing. For such cases we explain that effectively capturing the temporal cues is the key to success. While almost all the baselines come with proper graph convolution or careful feature engineering work, their processing of temporal information falls insufficient, either mean pooling (TGCN, Logistic., Random F.), Fisher Vector encoding (FacialCues), histogram encoding (DELF), or many-to-one LSTM (GCN-LSTM). Also notice the GCN-LSTM’s failure on most tasks, which shows the insufficiency of temporal sequence modeling techniques based on recurrent structures.

Since very little previous literature exists on this task, we further implement two baselines that use a simple classifier to process all our input features independently, i.e. without any network-level operation.
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 bottom two features is observed. While both of them can propagate quite long via the interactions, the way diffusion modifies the original features are quite different. We attribute such distinction to the two features’ different real-world implications: node in-degree feature in a gazing network snapshot can be interpreted as the attention one received from other people at the corresponding moment, and thus indicates the interaction engagement level of that person. In contrast, the node self-loop feature can be interpreted as the probability that the person looked at his/her own screen at the corresponding moment, making the person look introverted and preservative. The different practical meanings entailed by the feature determines they are propagated via the interaction network in distinctive manners. Finally, the "smile" and "happy" emotion seem to be able to diffuse two steps while not beyond.

5.4 Ablation study

Ablation study on Task 1 is further conducted on RESISTANCE-1 to evaluate the usefulness of TNDCN building blocks independently.

As shown by table 3, Eval. 2-3 further verify RNN’s insufficiency on handling both extremely long time sequence and weak local dynamics. Interestingly, the very simple mean pooling can significantly outperform RNN and achieve close result even when compared with our set pooling technique. Eval. 4-8 focus on the graph-level techniques. Eval. 4 shows the importance of using network for prediction. Eval. 5-8 indicates the usability of GCN despite its serious decay because of over-smoothing and too many nonlinearities when going deep. In contrast, our network diffusion can propagate as long as 10 hops without significant decay in performance.

5.5 Further Scope Study

While we claim that TNDCN is especially helpful to deal with interaction sequences that are extremely long and high-frequent, one interesting question to ask is how it will perform if the sequence is relatively short and the dynamics are less frequent? We investigate this problem by further running TNGCN on CIAW [18], a dataset recording more than 92 people’s timestamped proximities (of up to 1.2 meters) in a workplace over 20 days. The goal is to infer each person’s department based solely on their interaction data. Notice that there is only one dynamic network, which we partition
into only 20 snapshots. Since no previous works were done on we focus on comparing with generic temporal GNN models. Please see supplementary material for more details of the settings and results.

Our conclusion is that our model is still able to perform quite well even in this special scenario. We attribute this to the high flexibility of our S-TCN module to deal with sequences of various lengths.

6 CONCLUSION

In this work, we introduced a new neural-network-based representation learning model, TNDCN, which is particularly designed for dynamic social interaction networks. Dynamic social interaction networks contain highly dynamic node attributes with interactions with duration, which makes previous dynamic network embedding approaches not applicable. Our TNDCN model contains a network diffusion block that is capable of extracting patterns from complex interweaving of highly dynamic node attributes and interaction. It also leverages combination of TCN and set pooling that may learn the subtle and local patterns of social interactions randomly scattered in a long time span. TNDCN has been evaluated on three node-level prediction social tasks outperformed all previous baselines. The learned coefficients of TNDCN also give interesting social insights. Overall, TNDCN provides social scientists a powerful tool to automatically analyze social interactions to solve social tasks and extract knowledge for social science.

REFERENCES

[1] [n.d.]


