Network Extraction, Analysis, and Prediction of Deception

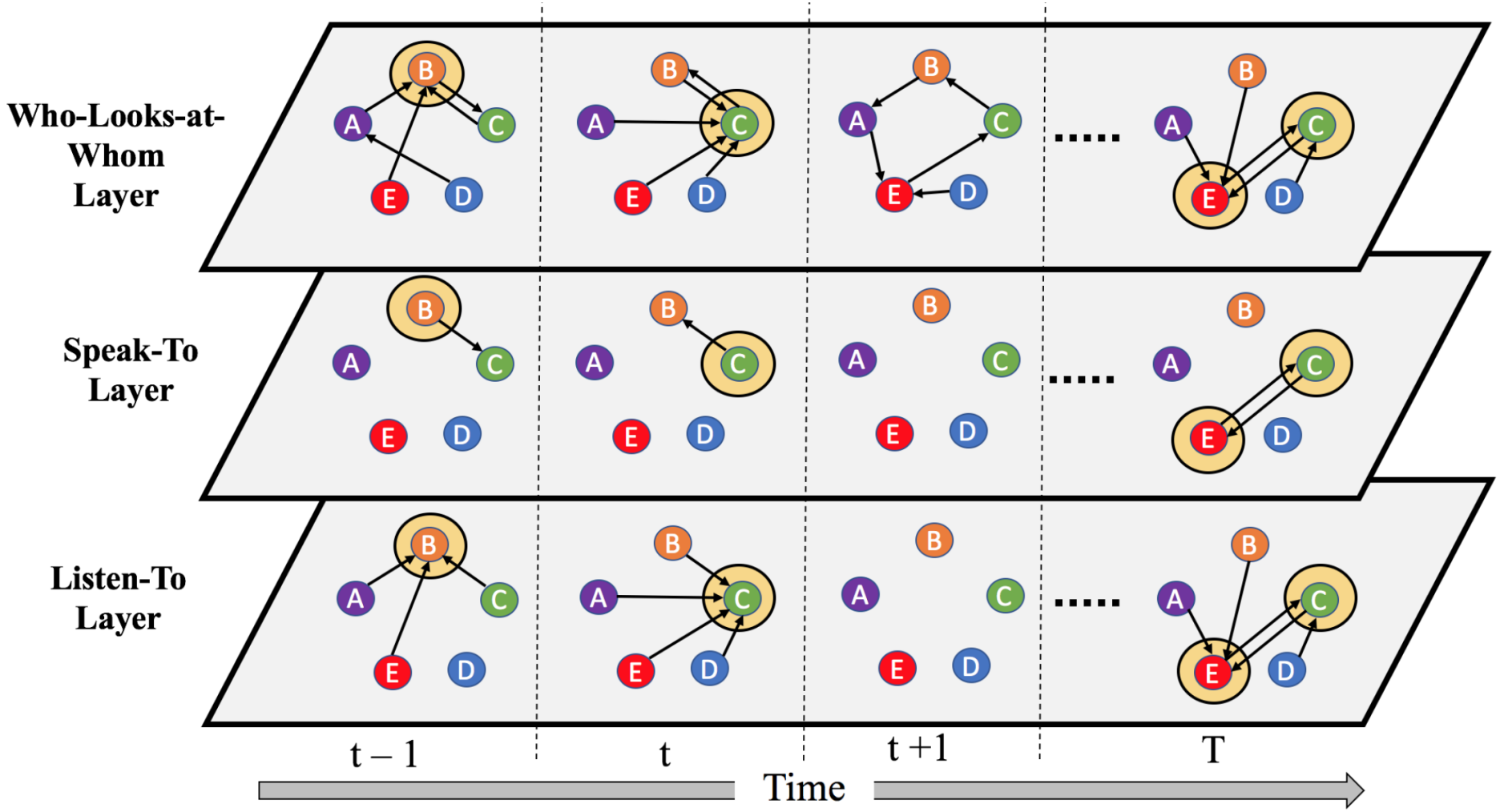
Jure Leskovec
Computer Science, Stanford University

Joint work with S. Kumar, C. Bai, VS. Subrahmanian.

Why Interaction Networks?

- **Individual analysis of each person ignores inter-personal interactions**
  - E.g., When person *A* looks at or talks to another person *B*, it creates an **interaction from *A* to *B***
- Networks give an efficient framework to **represent (verbal and non-verbal) interactions between people**
- Interaction networks can be used for **modeling** and **downstream prediction tasks**

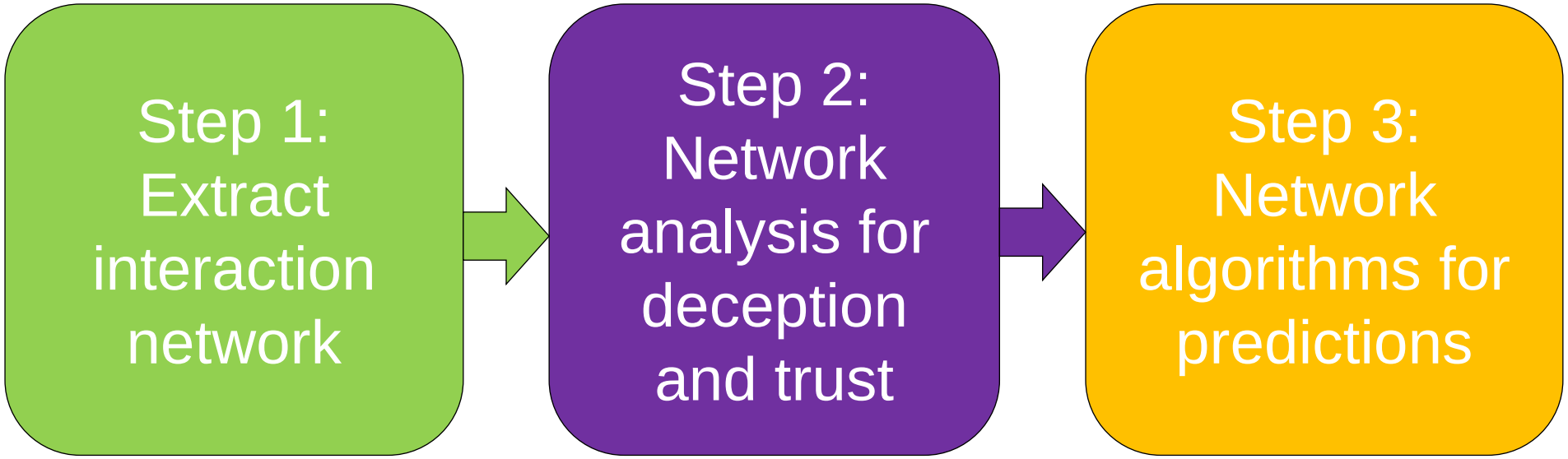
Our goal: Multi-layered Interaction Networks



Our context: Deception

- A game of 7 people who talk to each other
- People have assigned (but unknown) roles:
  - Deceivers
  - Truth-tellers
- In the end of the two groups wins

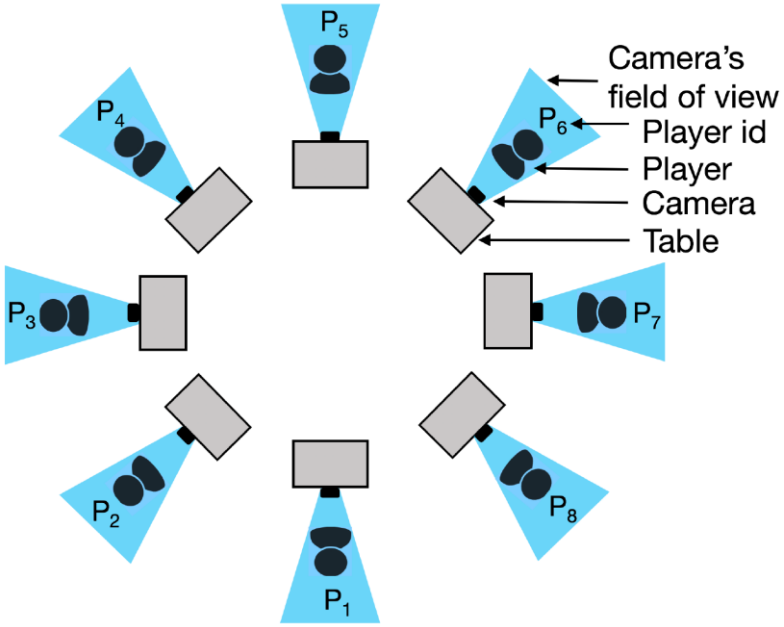
This talk:



Predicting the Visual Focus of Attention Prediction in Multi-person Discussion Videos.
Chongyang Bai, Srijan Kumar, Jure Leskovec, Miriam Metzger, Jay Nunamaker, V.S. Subrahmanian. IJCAI 2019

[IJCAI 2019]

Extracting Dynamic Networks from Video: Who Looks at Whom?



Joint work with Dartmouth, UCSB, Arizona

[IJCAI 2019]

Extracting Dynamic Networks from Video: Who Looks at Whom?

- **How do we extract dynamic interaction networks from video?**
- For **every 1/3rd second**, predict every person's **focus of attention**
  - Who is the player is looking at?
  - It takes 1/3rd second to focus attention (Rayner, 2009)

- **Candidates:**
  - Other players
  - Tablet

Joint work with Dartmouth, UCSB, Arizona

[IJCAI 2019]

Challenge #1

- Focus of attention **changes rapidly:**

Frame 25 Looking at **Player 6**

Frame 35 Looking at **Player 1**

Frame 45 Looking at **Player 1**

Frame 55 Looking at **Player 7**

1 second

[IJCAI 2019]

Challenge #2

- One's focus of attention is **affected by others' verbal and non-verbal behavior**

Player is speaking, so everyone looks at him

[IJCAI 2019]

Our solution: Multi-layered collective classification algorithm **(ICAF)**

- **ICAF:** a collective classification predictive model
- **Core idea:** Where a player looks influences where others look and is influenced by where others look.
  - Simultaneously make the prediction for all players.

First model that uses where others are looking at the moment to come up with joint prediction of who-looks-at-whom

Three major components of ICAF

At time *t*, three inputs are given to the predictor:

1. **Instantaneous input component: Where *u* looks at time *t*, depends on its head and eye features**
   - We extract raw features from **video** of the player's face at time *t* and use it as an input
     - Head pose (from OpenFace)
     - Eye gaze (from OpenFace)
     - Speaking probability (predicted)
2. **Collective classification component: Where *u* looks at time *t*, depends on where all others look at time *t***
   - We use output of other classifiers at time *t* as one input
3. **Temporal component: Where *u* looks at time *t*, depends on where *u* was looking at time *t-1***
   - We use output of *u's* classifier at time *t-1* as one input

ICAF details

- We train one classifier for each player
- Classifiers of all players are inter-dependent so that they influence one another
- All classifiers are trained jointly to make accurate predictions

Dataset creation

- **We manually annotated** 6,540 seconds (109 minutes) of videos from 35 8-person discussions to generate 7,604 labels ”who interacts with whom”.
- One label is generated for every 1/3rd second
- Tedious and time-consuming process: 40 hours needed to label total of ~2 hours of video

Baselines

We compare with 6 baselines

- GC: one classifier for all players in the game
- PC: one classifier per player
- These classifiers can use different features:
  - H: head pose feature
  - E: eye gaze feature
  - S: speaking probability

Result: Next Focus of Attention Prediction

- Learn a model from [0,t] seconds, predict in [t,t+1]
- Prediction accuracy of baselines and ICAF:

Model	RF	LR	SVM	NB
$GC(H)$	0.719	0.203	0.198	0.244
$GC(H,E)$	0.799	0.255	0.281	0.236
$GC(H,E,S)$	0.756	0.495	0.493	0.49
$PC(H)$	0.716	0.587	0.705	0.613
$PC(H,E)$	0.805	0.709	0.724	0.742
$PC(H,E,S)$	0.818	0.719	0.794	0.749
ICAF	**0.831**	**0.829**	**0.823**	**0.783**

ICAF consistently performs better than others, irrespective of the base classifier.

[IJCAI 2019]

Network Extracted by ICAF

Demo with more videos:
https://home.cs.dartmouth.edu/~cy/icaf/

[IJCAI 2019]

Lightly-supervised ICAF

- **Scalability** is a major challenge because labeling videos is resource intensive
- We extend ICAF with the **intuition**: **People are likely to look at the speaker**
- **Lightly-supervised ICAF:**
  - Find continuous speaking segment during the introduction round
  - Use this segment as the label of all other players

[IJCAI 2019]

Evaluating LightICAF

- **ICAF vs. LightICAF**
  - ICAF performs only slightly better than LightICAF
  - Average prediction accuracy of who is a person looking at:
    - **ICAF: 61%** (random guessing gets 1/8 12.5%)
    - **LightICAF: 56%**
- **We used LightICAF to generate 62 networks:**
  - Publicly released the networks to promote future research: http://snap.stanford.edu/data/comm-f2f-Resistance.html

Our Contributions

Step 1: Extract interaction network

Step 2: Network analysis for deception and trust

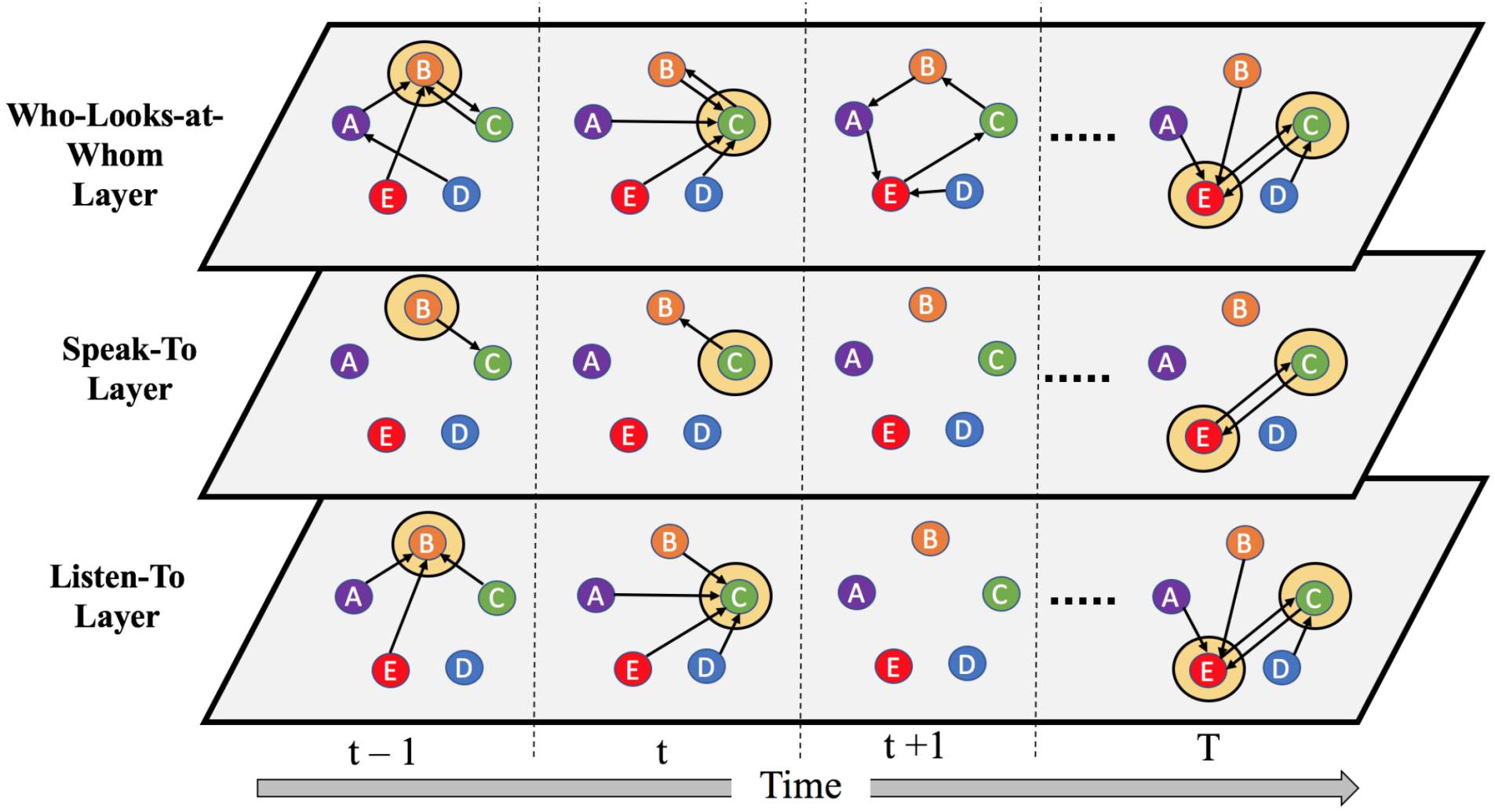
Step 3: Network algorithms for predictions

(1) Network extraction demo:
https://home.cs.dartmouth.edu/~cy/icaf/
(2) Mafia network dataset:
http://snap.stanford.edu/data/comm-f2f-Resistance.html

Our context: The “mafia” game

- A game of 7 people who talk to each other
- People have assigned (but unknown) roles:
  - Deceivers
  - Truth-tellers
- In the end of the two groups wins
- We also perform surveys during the game

Multi-layered Interaction Networks

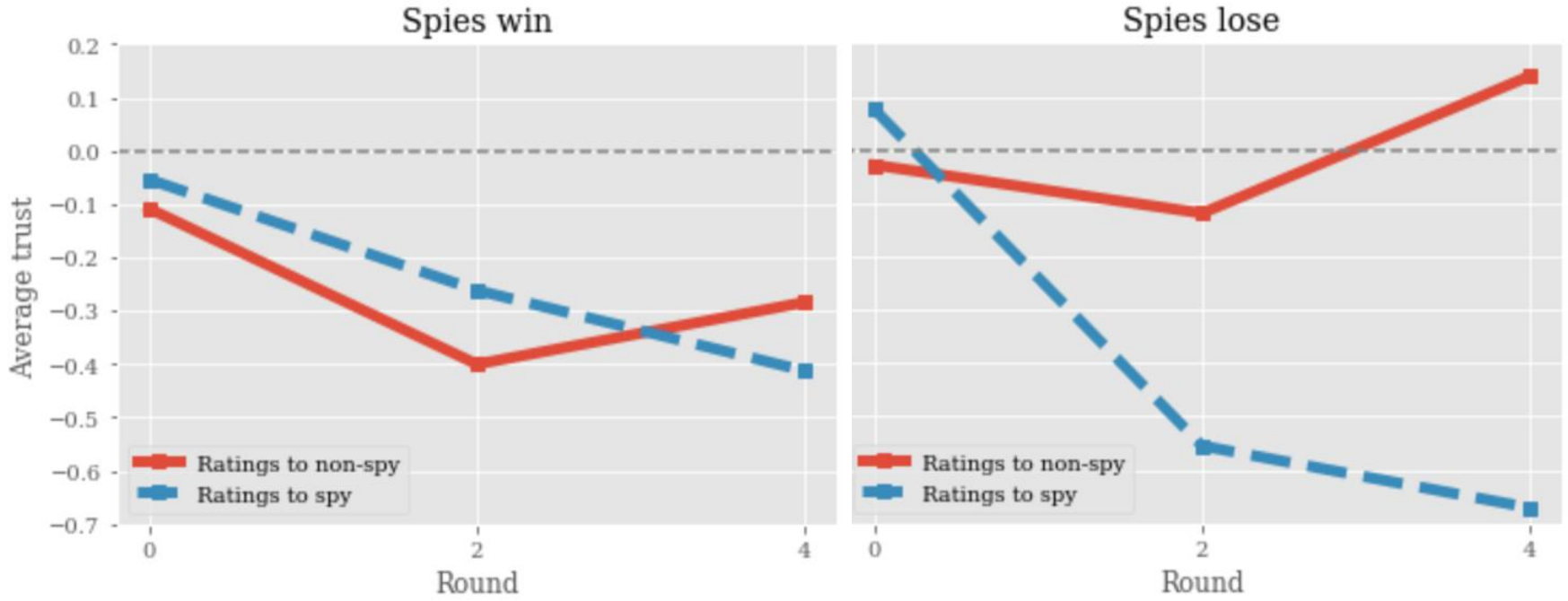


Key Questions

- **RQ1:** Does the behavior of Deceivers vary across games?
- **RQ2:** Do Deceivers and Truth-Tellers have distinct looking patterns?
- **RQ3:** How does the speaking pattern of Deceivers and Truth-Tellers differ?
- **RQ4:** Do Deceivers interact differently with other Deceivers compared to Truth-Tellers?

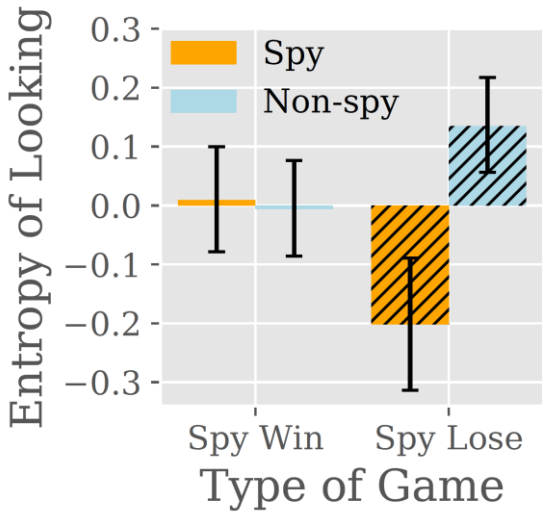
Analysis #1: Deceiver Win vs. Lose Games

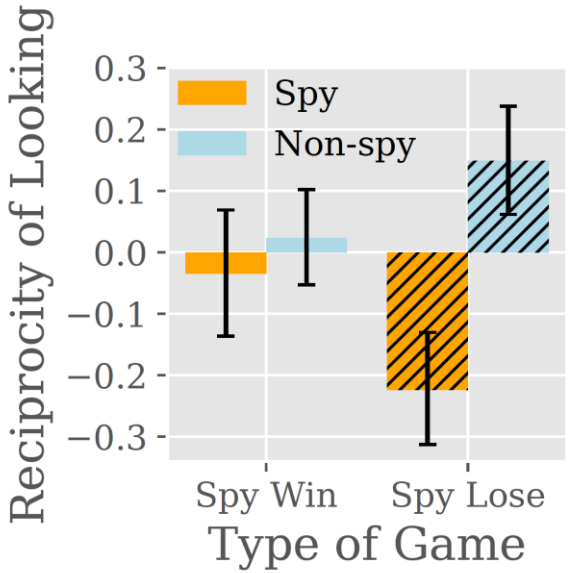
- **Finding 1:** Deceivers are more trusted than Truth-Tellers in the first 2 rounds in Deceiver-win games.
- **Finding 2:** Deceivers are identified in the first 2 rounds only in Deceiver-lose games.



Analysis #2: Look-at Network

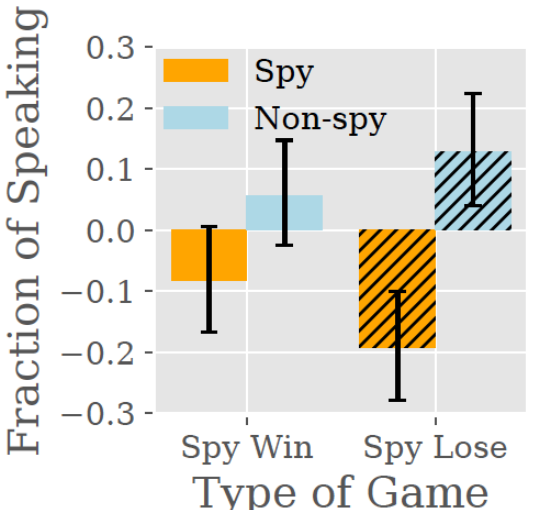
- **Finding 3:** Deceivers have a lower entropy and reciprocity of looking.
- **However, this is more nuanced: Depends on the game result**
  - **Finding 4:** Deceivers and Truth-Tellers have similar entropy and reciprocity of looking in Deceiver-win games
  - **Finding 5:** Deceivers have lower entropy and reciprocity of looking in Deceiver-lose games

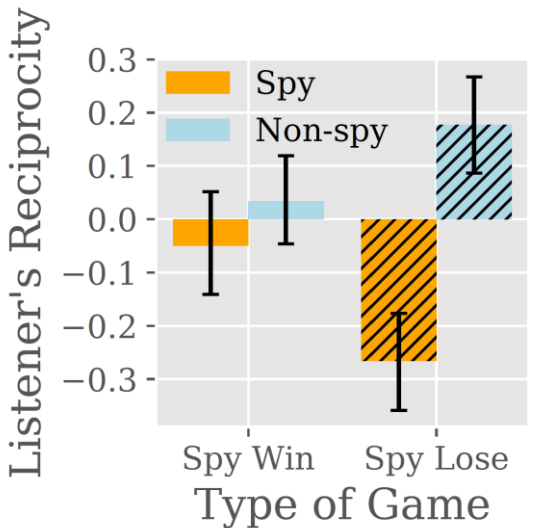


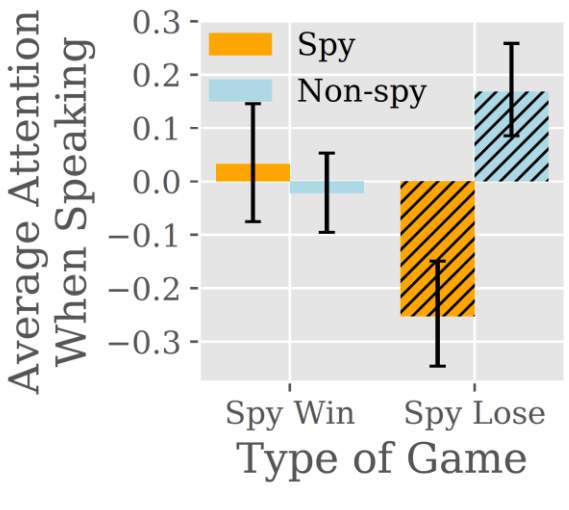


Analysis #3: Speaking Network

- **Finding 6:** Deceivers speak less and are listened to less. **Depends on the game outcome:**
- **Finding 7:** Deceivers in Deceiver-lose games speak less, not listened to, and get less attention (weighted indegree) than Truth-Tellers.
- **Finding 8:** Deceivers and Truth-Tellers in Deceiver-win games have similar speaking behavior.

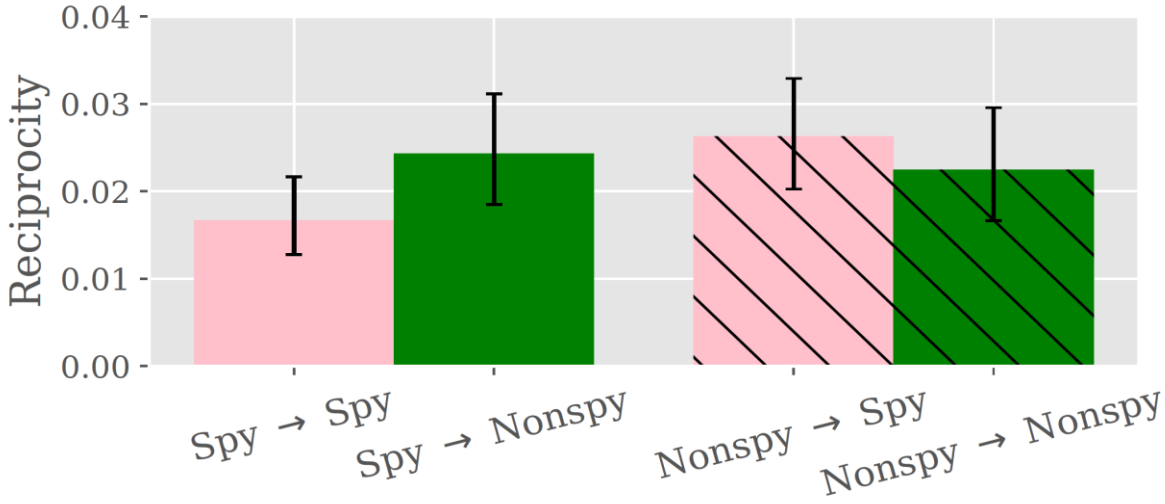


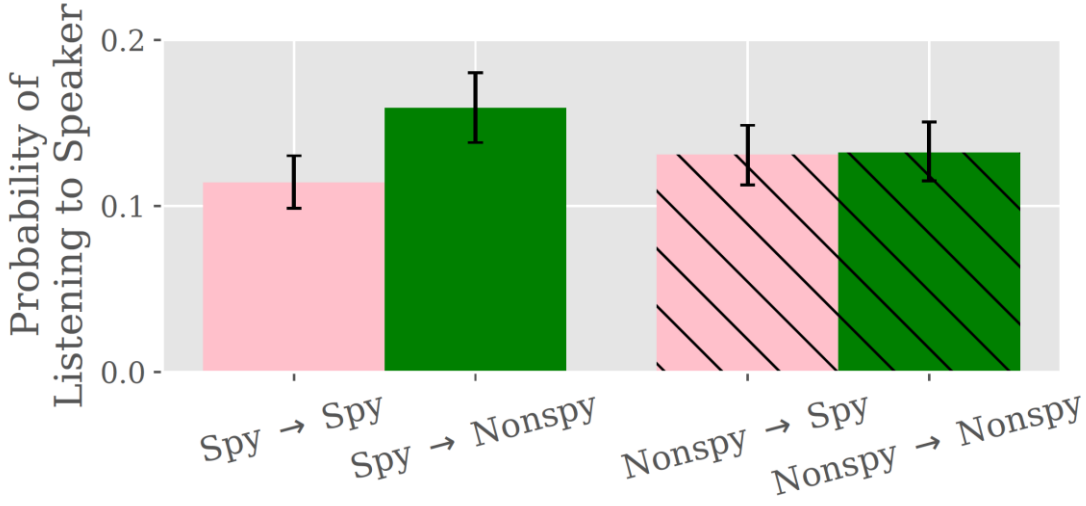




Analysis #4: Pairwise Interactions

- **Finding 9:** Truth-Tellers interact equally with everyone.
- **Finding 10:** Deceivers interact more with Truth-Tellers and ignore other Deceivers.





Answers to Key Questions

- **RQ1:** Does the behavior of Deceivers vary across games?
  - **Answer:** Deceiver behavior is different in Deceiver-win vs Deceiver-lose games
- **RQ2:** Do Deceivers and Truth-Tellers have distinct looking patterns?
  - **Answer:** Deceivers in Deceiver-lose games have lower entropy and reciprocity of looking.
- **RQ3:** How does the speaking pattern of Deceivers and Truth-Tellers differ?
  - **Answer:** Deceivers in Deceiver-lose games speak less and are listened to less.
- **RQ4:** Do Deceivers interact differently with other Deceivers compared to Truth-Tellers?
  - **Answer:** Deceivers ignore other Deceivers and interact more with Truth-Tellers.

Our Contributions

Step 1: Extract interaction network

Step 2: Network analysis for deception and trust

Step 3: Network algorithms for predictions

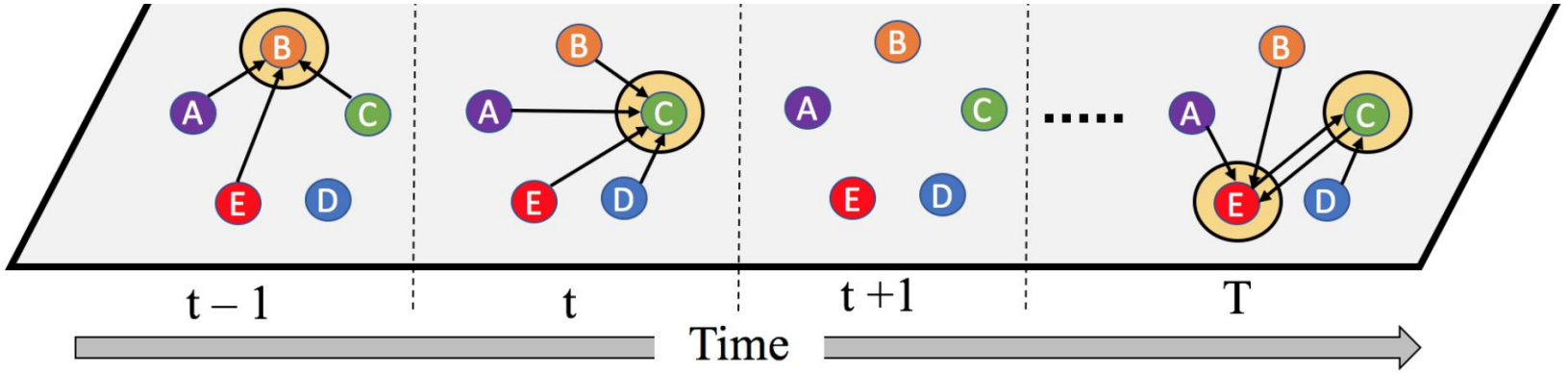
Key Prediction Questions

1. Can we accurately identify who is a Deceiver using the networks?
2. What is the length of observation (video/network) needed to make accurate prediction?

Model #1: Temporal Graph Convolution

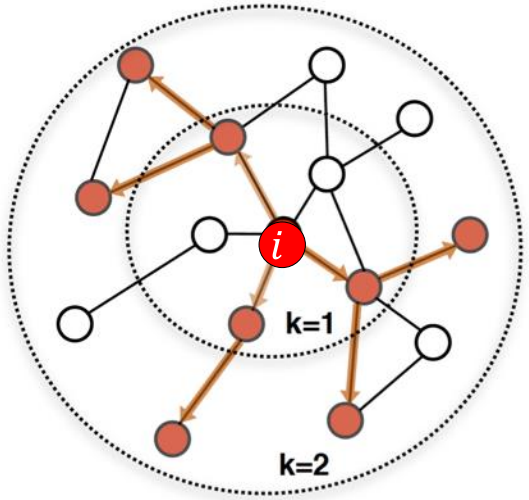
- **Input:** Sequence of graph snapshots
- **Output:** Node labels
- **Steps:**

1. Run graph neural network model on each network
2. Aggregate outputs from the sequence of graphs
   - Aggregations: average, input to LSTM, input to RNN, etc.

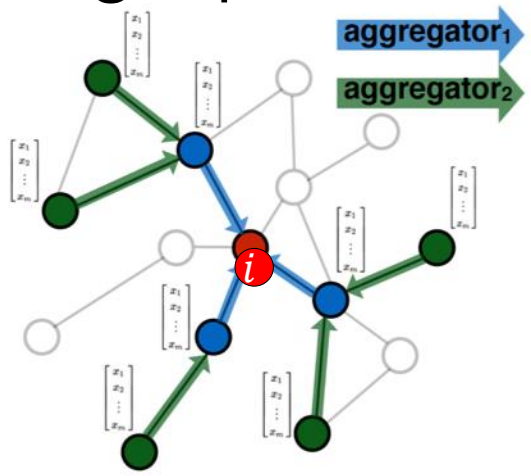


Graph Neural Networks

Idea: Node's neighborhood defines a computation graph



Determine node computation graph



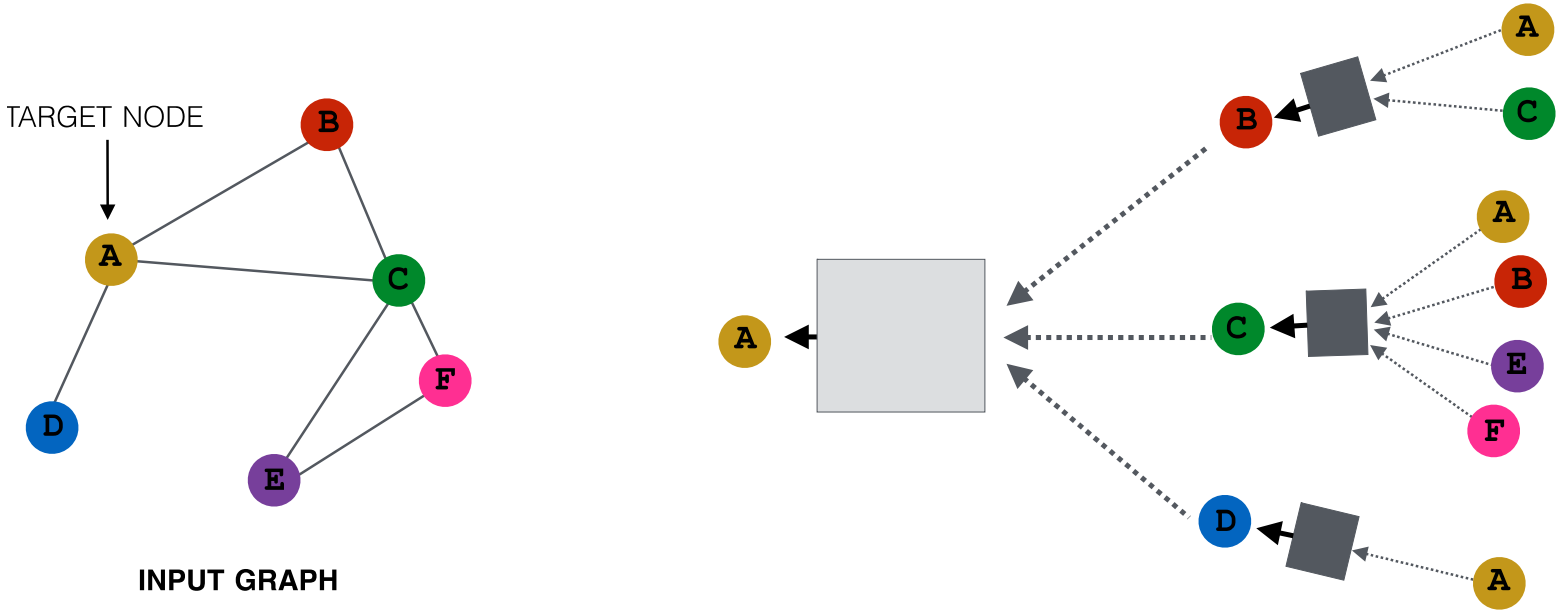
Propagate and transform information

Learn how to propagate information across the graph to compute node features

The Graph Neural Network Model. Scarselli et al. *IEEE Transactions on Neural Networks* 2005

Semi-Supervised Classification with Graph Convolutional Networks. T. N. Kipf, M. Welling, ICLR 2017

Graph Neural Networks



Each node defines a computation graph

- Each edge in this graph is a transformation/aggregation function

Inductive Representation Learning on Large Graphs. W. Hamilton, R. Ying, J. Leskovec. NIPS, 2017.

Our Model #2: Belief Propagation

1. **Create a negative network for each 1 second fragment:**
   - Replace each edge weight $w_{i,j}$ with $1 - w_{i,j}$
2. **Initialize nodes using a feature vector:**
   - **Node features:** Fraction of speaking, entropy of looking at, in-degree, in-degree while speaking
3. **Run till convergence on each network**

$$S_i \leftarrow \beta * (\Sigma_{(i,j) \in E} S_j * (1 - w_{j,i})) + (1 - \beta) S_i$$

4. **Average S scores over all networks/time steps**

Prediction Results #1

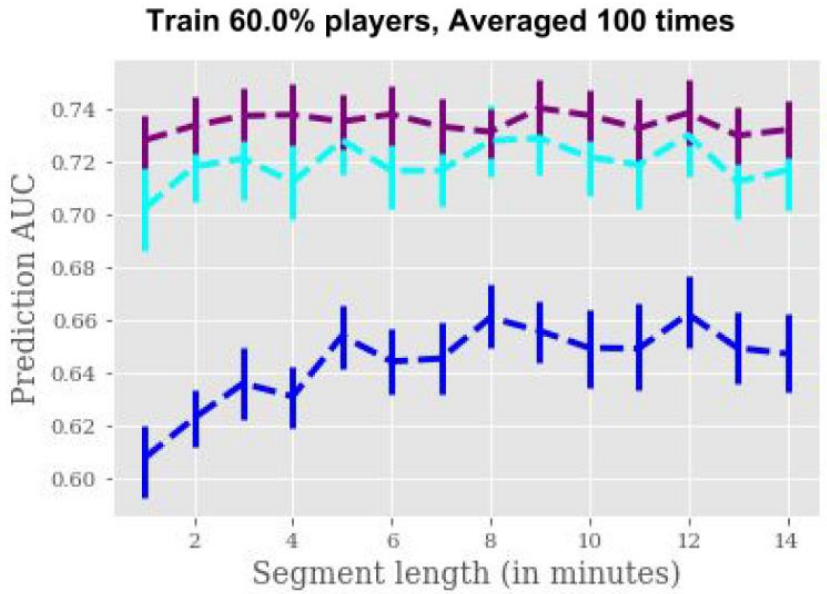
- **Task 1:** Identify who is a Deceiver using the networks
- **Setup:** Given 1 minute of interaction network, output the Deceivers
- **Setting:** 5-fold cross validation results:
  - Split by game to avoid label leakage: 80% games are used for training, 20% testing

Baselines

Method	Performance
Emotion	0.538 AUC
Head and eye movement	0.549 AUC
Facial action unit	0.569 AUC
Late fusion	0.587 AUC
Graph Neural Network	0.596 AUC
Belief propagation on negative network	**0.73 AUC**

Prediction Results #2

- **Task 2:** Measure impact of network duration
- **Result:** Performance of our model is consistent with the change in the length of the segment (still we do 1 network per second)



Belief Propagation + Features

Belief Propagation on negative network

Features

Next steps:

- **Multimodality:** Create multimodal framework to simultaneously model networks + language + videos
  - All modalities depend on one another and have cross-correlations
  - Treating them independently is sub-optimal
  - This requires creation of new models that leverage all three components together
- **Network modeling of trust and dominance:**
  - We have shown that networks are useful in detecting deception
  - Thus, we will use the networks to predict trust and dominance next
    - Existing network models may not be sufficient, so we may create new models

Next steps:

- **Cross-correlation between trust, deception, and dominance:**
  - Social affects are inter-dependent
  - We will create a multi-task learning framework based on networks to simultaneously predict deception, trust, and dominance
- **Cultural effects in networks:**
  - Current network models use all cultures to train and make predictions
  - We will use the networks to elicit culture-specific novelties
  - E.g., does our finding “Deceivers avoid looking at one another” hold for all cultures?

Deep Models for Temporal Networks

[KDD 2019]

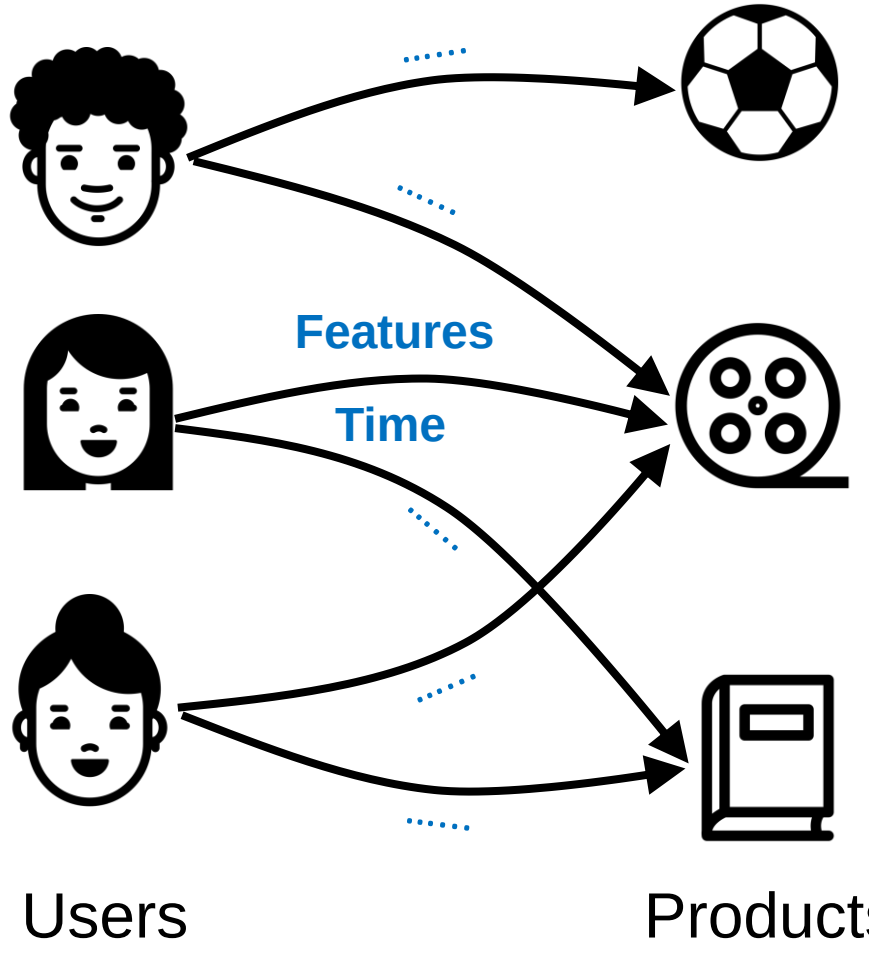
Deep Temporal Models

- **Goal:** Model the time-evolving player behavior
- **Our solution:** Deep learning model to learn dynamic embeddings
  - Mutually-coupled deep neural network model
  - Trained to predict future embedding
- **KDD 2019:** Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. S. Kumar, X. Zhang, J. Leskovec.
  - Highly accurate in identifying temporal anomalies
  - Code and data released to promote research: http://snap.stanford.edu/jodie

[KDD'19]

Temporal Interaction Networks

A flexible way to represent time-evolving relations



Representing as a sequence of interactions:

$$S_r = (u_r, i_r, t_r, f_r)$$

interaction user item time features

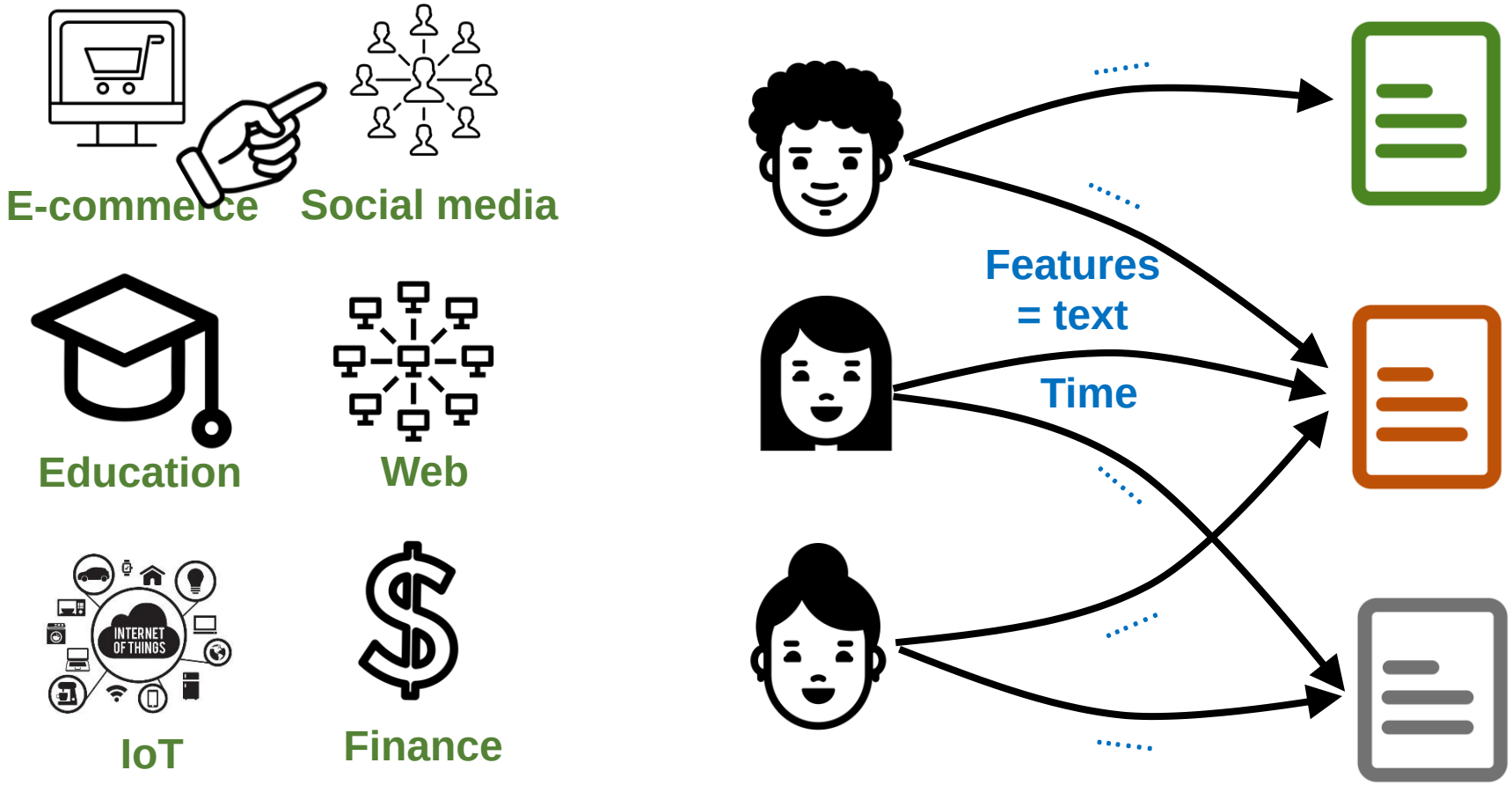
where

$$u_r \in \mathcal{U}, i_r \in \mathcal{I}, t_r \in \mathbb{R}^+$$

$$0 < t_1 \leq \ldots \leq T, f_r \in \mathbb{R}^d$$

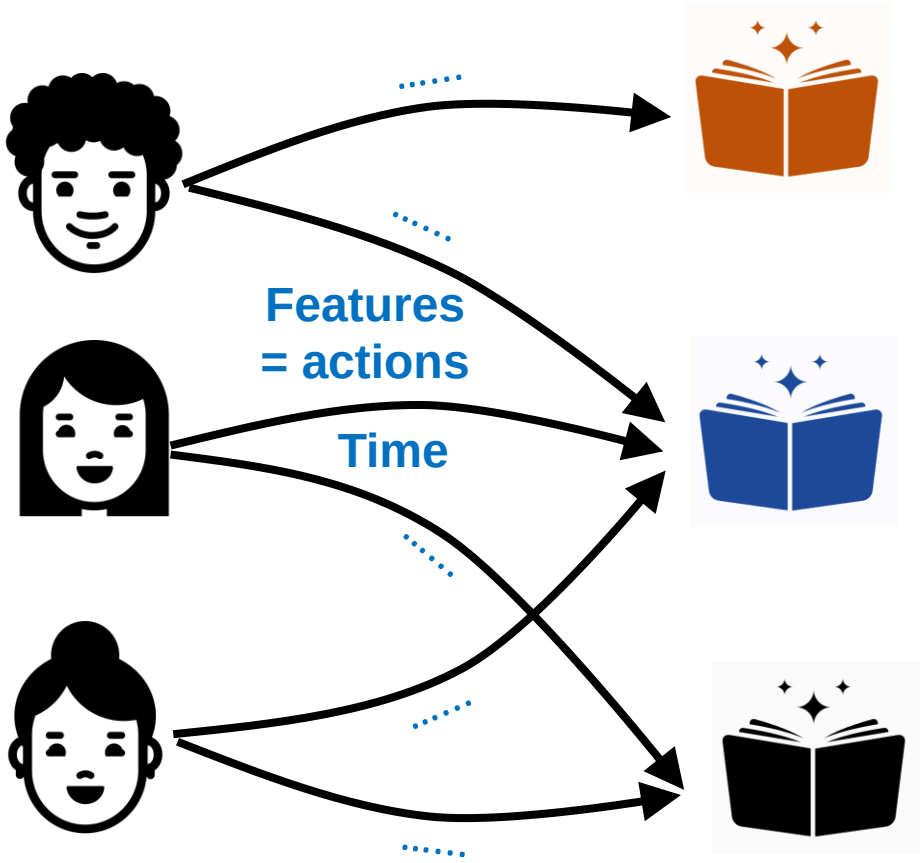
[KDD'19]

Temporal Interaction Networks



[KDD'19]

Temporal Interaction Networks



Application Domains

Students

Courses

Problem Setup

Given a temporal interaction network

$$S_r = (u_r, i_r, t_r, f_r)$$

interaction user item time features

where $u_r \in \mathcal{U}, i_r \in \mathcal{I}, t_r \in \mathbb{R}^+, 0 < t_1 \leq \ldots \leq T, f_r \in \mathbb{R}^d$

generate an embedding trajectory of every user

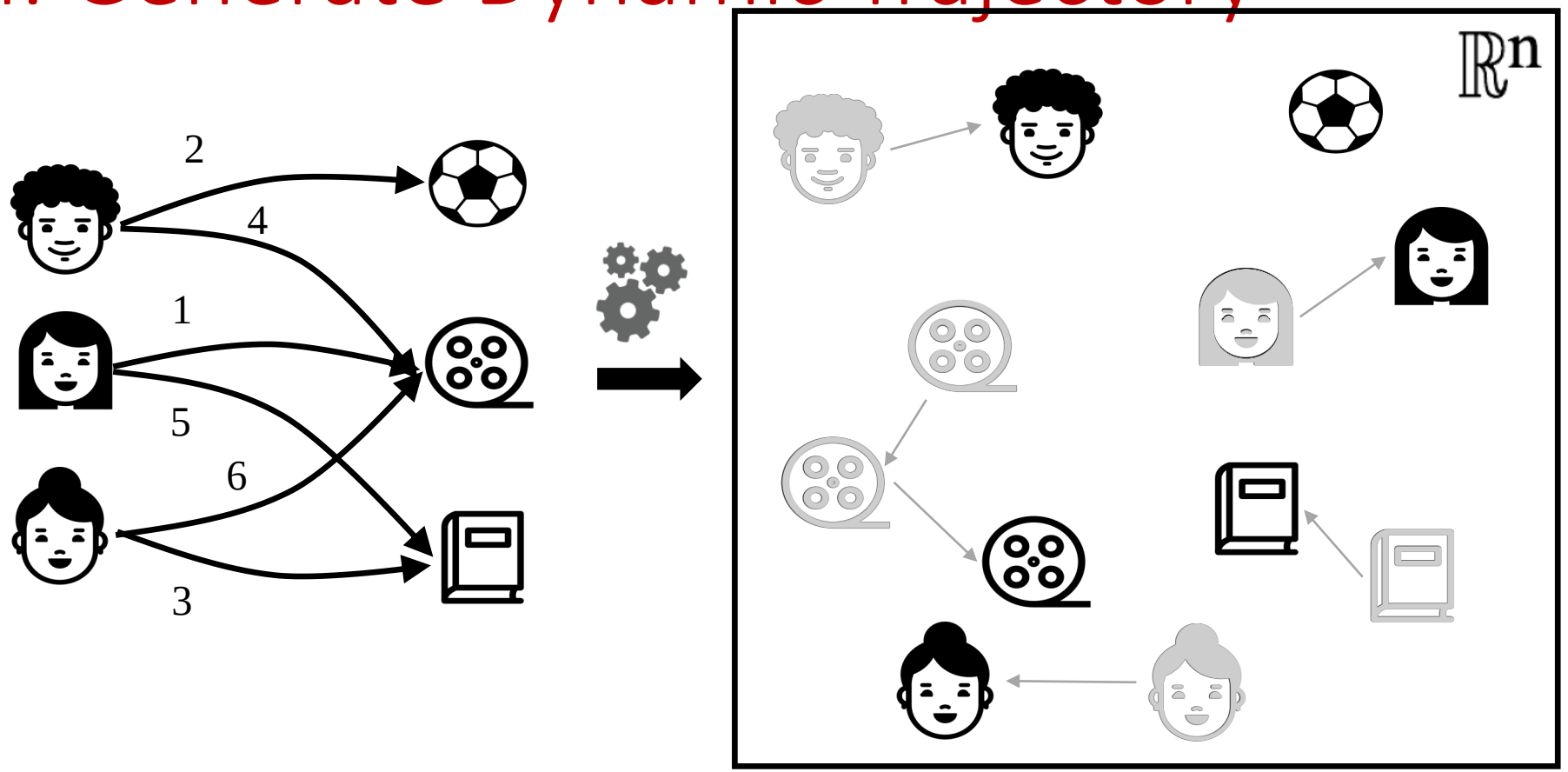
$$\mathbf{u(t)} \in \mathbb{R}^n \; \forall u \in \mathcal{U}, \forall t \in [0, T]$$

and an embedding trajectory of every item

$$\mathbf{i(t)} \in \mathbb{R}^n \; \forall i \in \mathcal{I}, \forall t \in [0, T]$$

[KDD'19]

Goal: Generate Dynamic Trajectory



Input: Temporal interaction network

Output: Dynamic trajectory in embedding space

[KDD'19]

Two Fundamental Problems

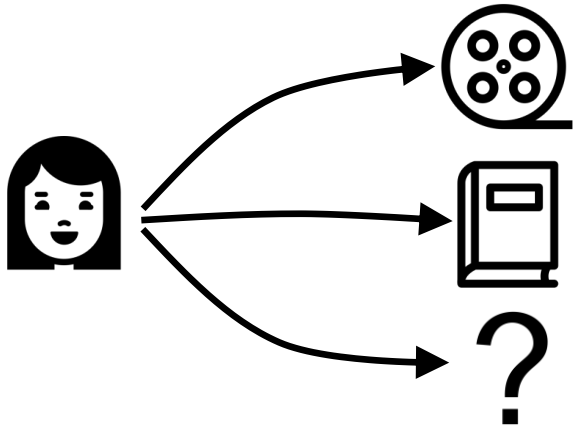
How to predict future user-item interactions?

Applications:

Recommendation

Network Evolution

Normal behavior modeling



How to detect anomalous change in user state?

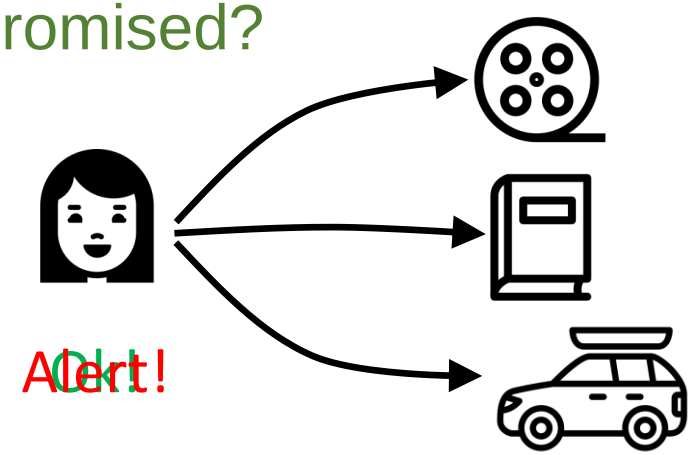
Example: Has an account been compromised?

Applications:

Anomaly detection

Device failure prediction

Churn prediction



Challenges in Dynamic Trajectories

Challenges in learning:

- **C1:** How to learn inter-dependent user and item embeddings?
- **C2:** Model should be able to make predictions at any time. How to generate embedding for every point in time?

Challenges in scalability:

- **C3:** During inference, how to make sub-linear time interaction predictions?
- **C4:** How to train temporal network in batches, not one-interaction-at-a-time?

[KDD'19]

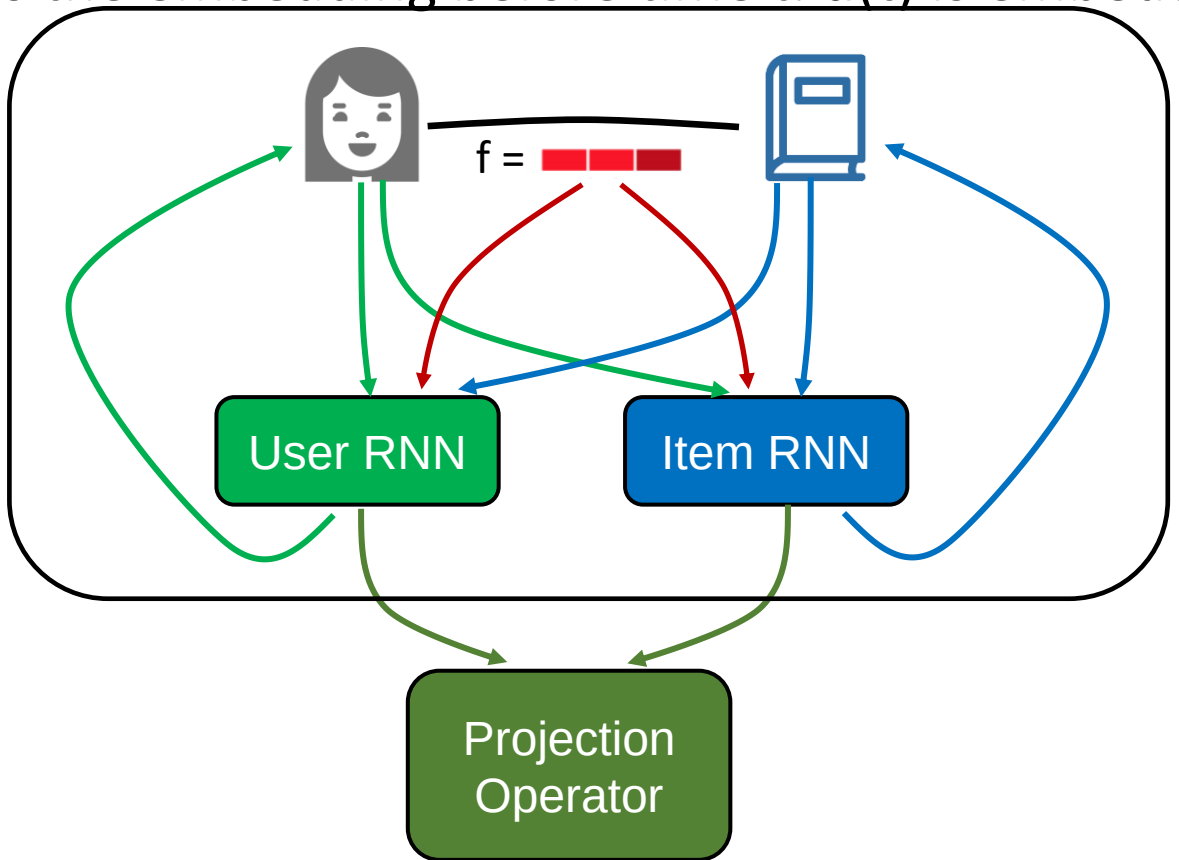
Existing Methods

	C1 Co-influence	C2 Embed any time	C3 Near constant prediction	C4 Train in batches
Deep recommender systems • Time-LSTM (IJCAI 2017) • Recurrent Recommender Networks (WSDM 2017) • Latent Cross (WSDM 2018)				✓
Dynamic co-evolution • Deep Coevolve (DLRS, 2016)	✓			
Temporal network embedding • CTDNE (BigNet, 2018)	✓			✓
Our model: JODIE	✓	✓	✓	✓

Our Model: JODIE

JODIE: Joint Dynamic Interaction Embedding

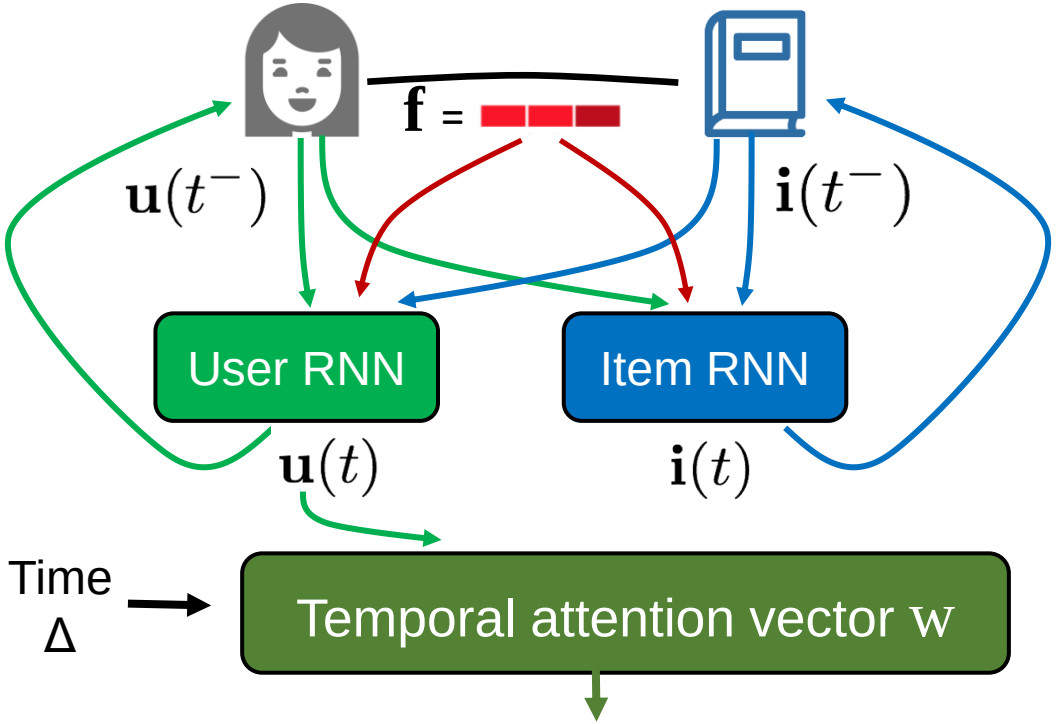
- Mutually-recursive recurrent neural network framework
- Notations: **u**(t^-) is the embedding before time t. **u**(t) is embedding at time t.

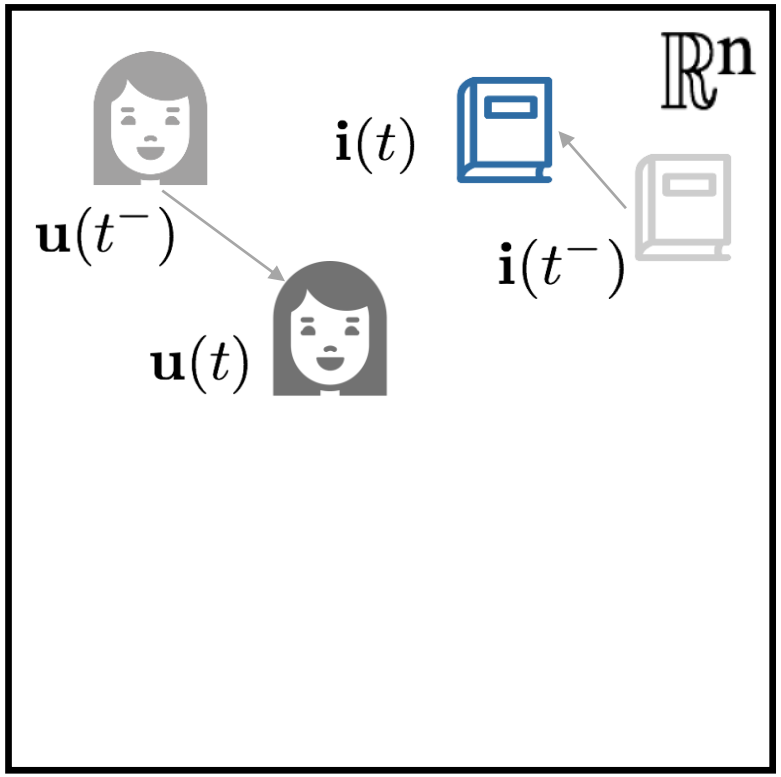


Update Component

Project Component

JODIE: Update Component





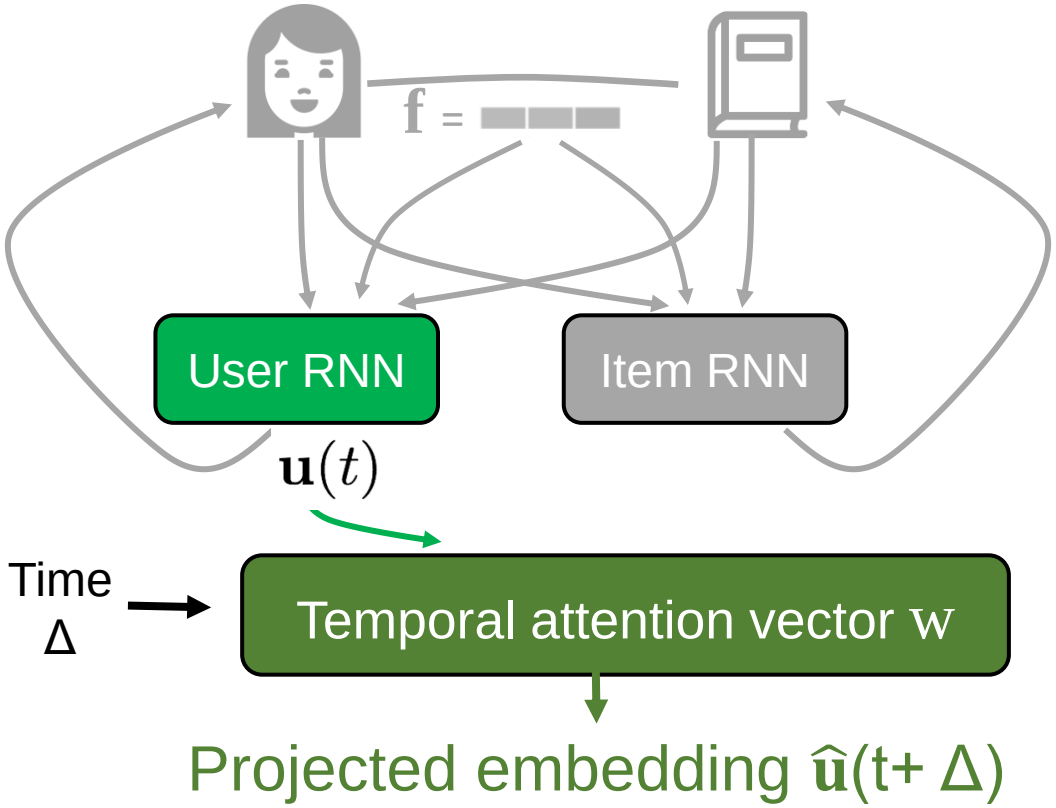
Projected embedding $\hat{\mathbf{u}}(t+\Delta)$

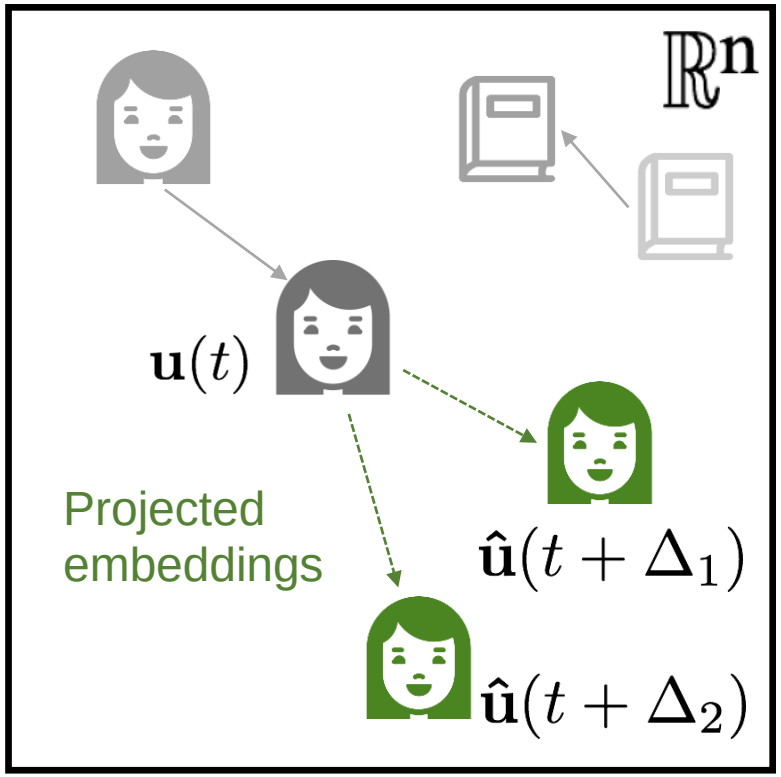
$$\mathbf{u}(t) = \sigma(W_1^u \mathbf{u}(t^-) + W_2^u \mathbf{i}(t^-) + W_3^u \mathbf{f})$$
$$\mathbf{i}(t) = \sigma(W_1^i \mathbf{i}(t^-) + W_2^i \mathbf{u}(t^-) + W_3^i \mathbf{f})$$

Weight matrices W are trainable

- Each user has one RNN. All users share the RNN parameters. Similar for items.

JODIE: Projection Component





$$\hat{\mathbf{u}}(t+\Delta) = (1+\mathrm{w}) * \mathbf{u}(t)$$

- Projected embeddings are linear scaling, based on time, of the observed embeddings

Challenges in Dynamic Trajectories

Challenges in learning:

- **C1:** How to learn inter-dependent user and item embeddings?
- **C2:** Model should be able to make predictions at any time. How to generate embedding for every point in time?

Challenges in scalability:

- **C3:** During inference, how to make sub-linear time interaction predictions?
- **C4:** How to train temporal network in batches, not one-interaction-at-a-time?

Interaction Predictions: Sub-linear

To predict interactions: which item $i \in I$ will a user u interact with at time t ?

- **Standard approach:**
  - Choose $j \leftarrow \arg\max_{i \in \mathcal{I}} p(u, i)$
  - $|I|$ **linear computations:** expensive during inference

- **How can we make predictions in near-constant time?**

Interaction Predictions: Sub-linear

JODIE predicts the item embedding directly

- At inference, output the item with the closest embedding: near-constant time predictions
- Prediction is done via a fully-connected linear layer

$$\tilde{\mathbf{j}} = W_5 \hat{\mathbf{u}}(t + \Delta) + W_6 \mathbf{i}(t + \Delta^-) + B$$

Predicted item embedding | **Projected embedding** | **Previous item embedding** | **Bias**

[KDD'19]

JODIE Formulation

Update:

$$\mathbf{u}(t) = \sigma(W_1^u \mathbf{u}(t^-) + W_2^u \mathbf{i}(t^-) + W_3^u \mathbf{f})$$
$$\mathbf{i}(t) = \sigma(W_1^i \mathbf{i}(t^-) + W_2^i \mathbf{u}(t^-) + W_3^i \mathbf{f})$$

Project:

$$\hat{\mathbf{u}}(t + \Delta) = (1 + w) * \mathbf{u}(t)$$

Predict:

$$\tilde{\mathbf{j}} = W_5 \hat{\mathbf{u}}(t + \Delta) + W_6 \mathbf{i}(t + \Delta^-) + B$$

Loss:

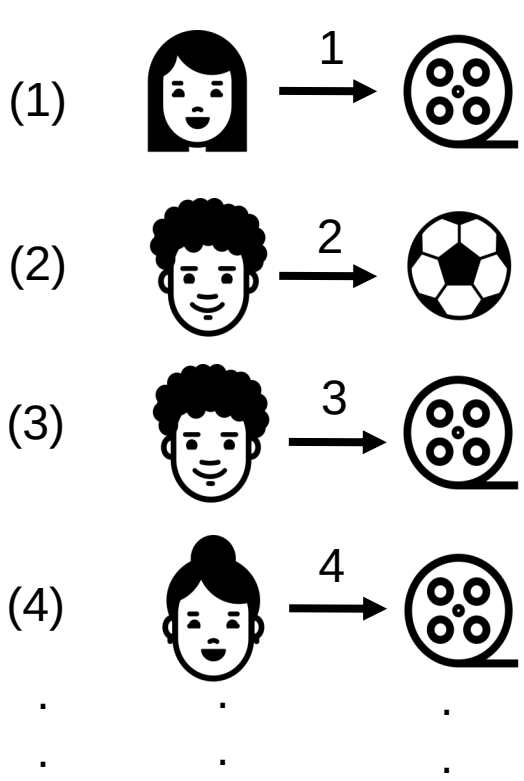
$$\sum_{S_r:(u,i,t,f)} ||\tilde{\mathbf{j}}(t) - \mathbf{i}(t^-)||_2 + \lambda_U ||\mathbf{u}(t) - \mathbf{u}(t^-)||_2 + \lambda_I ||\mathbf{i}(t) - \mathbf{i}(t^-)||_2$$

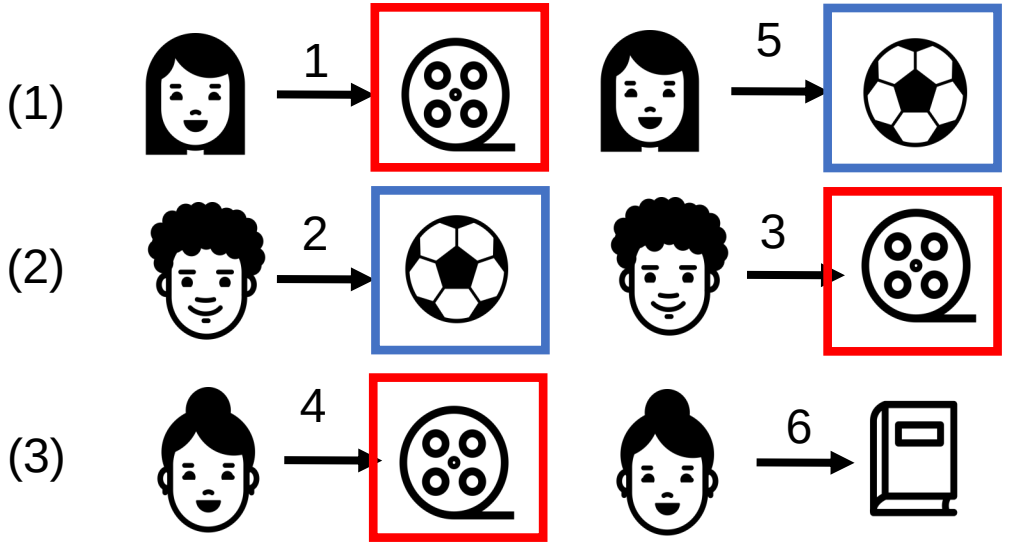
Predicted item embedding should be close to the real item embedding

Smoothness in evolving embeddings

Standard Training Processes: N/A

Training must maintain temporal order





Sequential processing:
not scalable

Split by user (or item):
not allowed

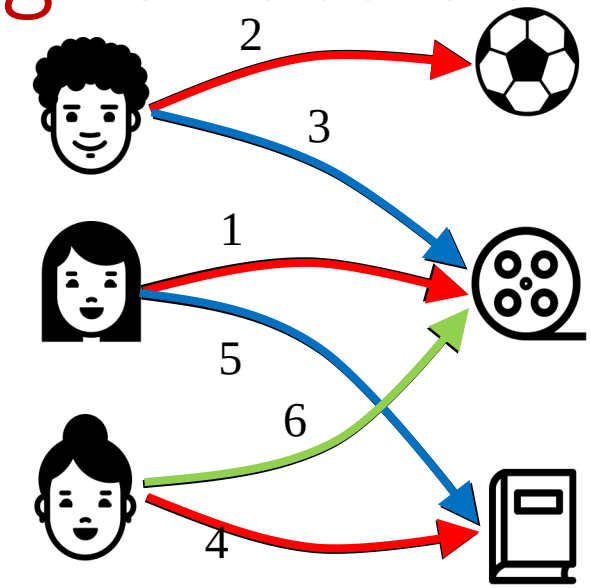
T-batch: Batching for Scalability

T-batch: Temporal data batching algorithm

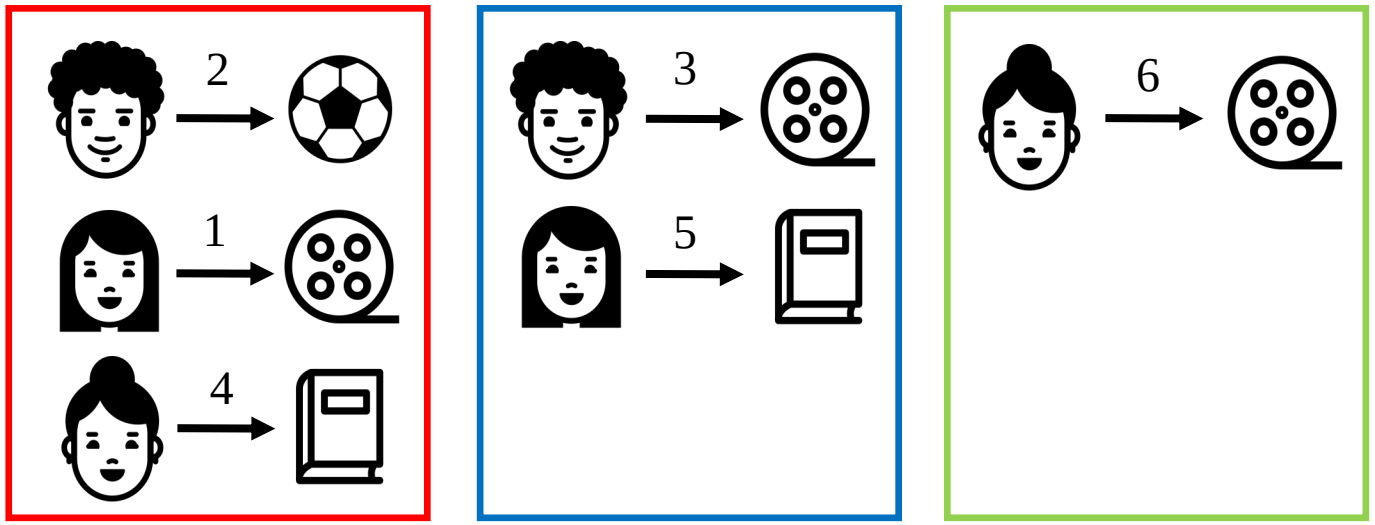
- Creates a sequence of batches
  - Interactions in each batch are processed in parallel
  - Processing the batches in sequence = processing each interaction in sequence

- **Main idea: create each batch as an independent edge set**

T-batch: Batching for Scalability



In each batch, edges are the **lowest time-stamped edges** incident on the user and the item



Experiments: Prediction Tasks

- **Interaction prediction:**
  - Given interactions till time t, which item $i \in I$ will user u interact with at time t?
- **Anomaly detection:**
  - Will an interaction lead to an anomalous change in the state of user u?
- **Settings:**
  - **Temporal Splits:** 60%, 20%, 20%
  - **Metrics:** Mean reciprocal rank, Recall@10, AUROC

Code and Data: https://snap.stanford.edu/jodie

Datasets

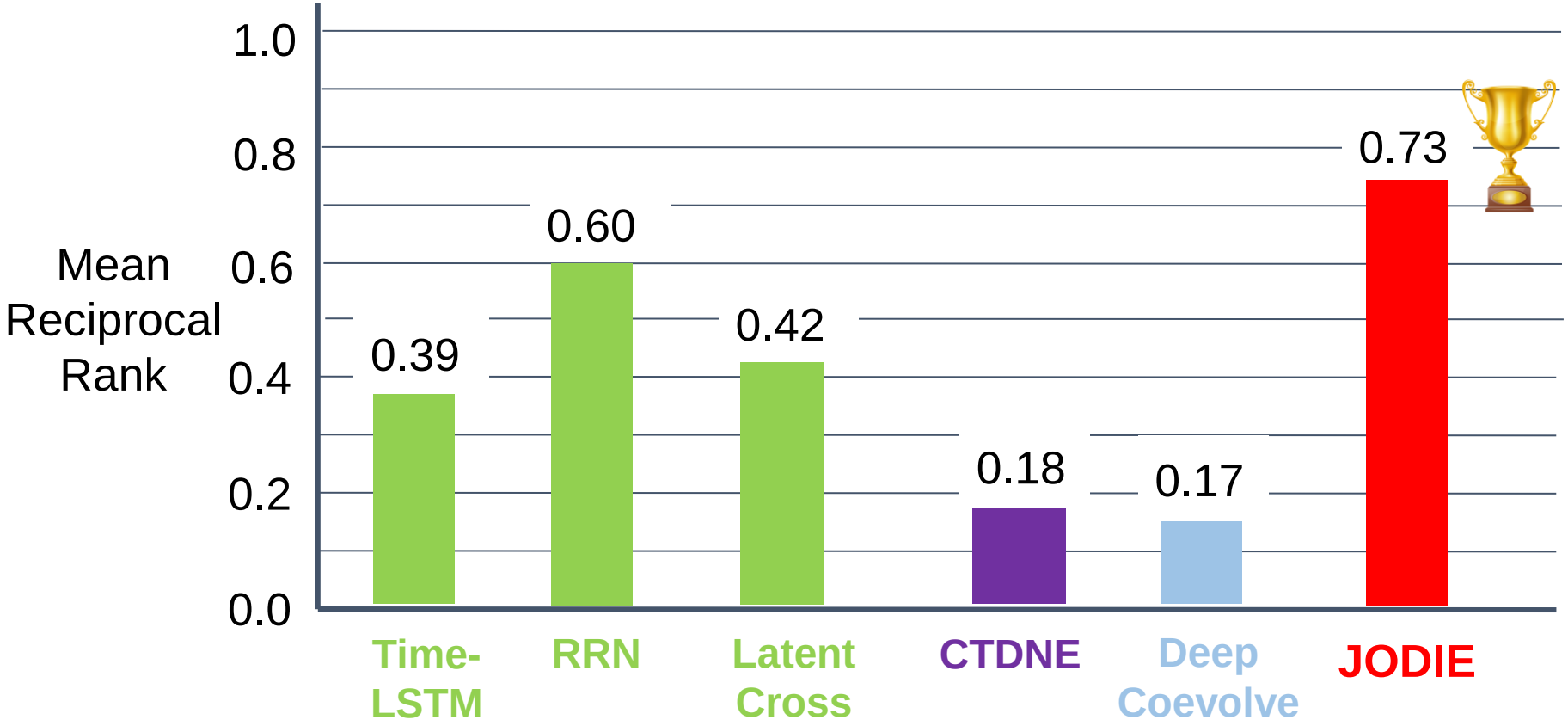
Data	Users	Items	Interactions	State Changes	% State Changes
Reddit	10,000	984	672,447	366	**0.05%**
Wikipedia	8,227	1,000	157,474	217	**0.14%**
LastFM	980	1,000	1,293,103	-	
MOOC	7,047	97	411,749	4,066	**0.99%**

Rare events!

Code and Data: https://snap.stanford.edu/jodie

[KDD'19]

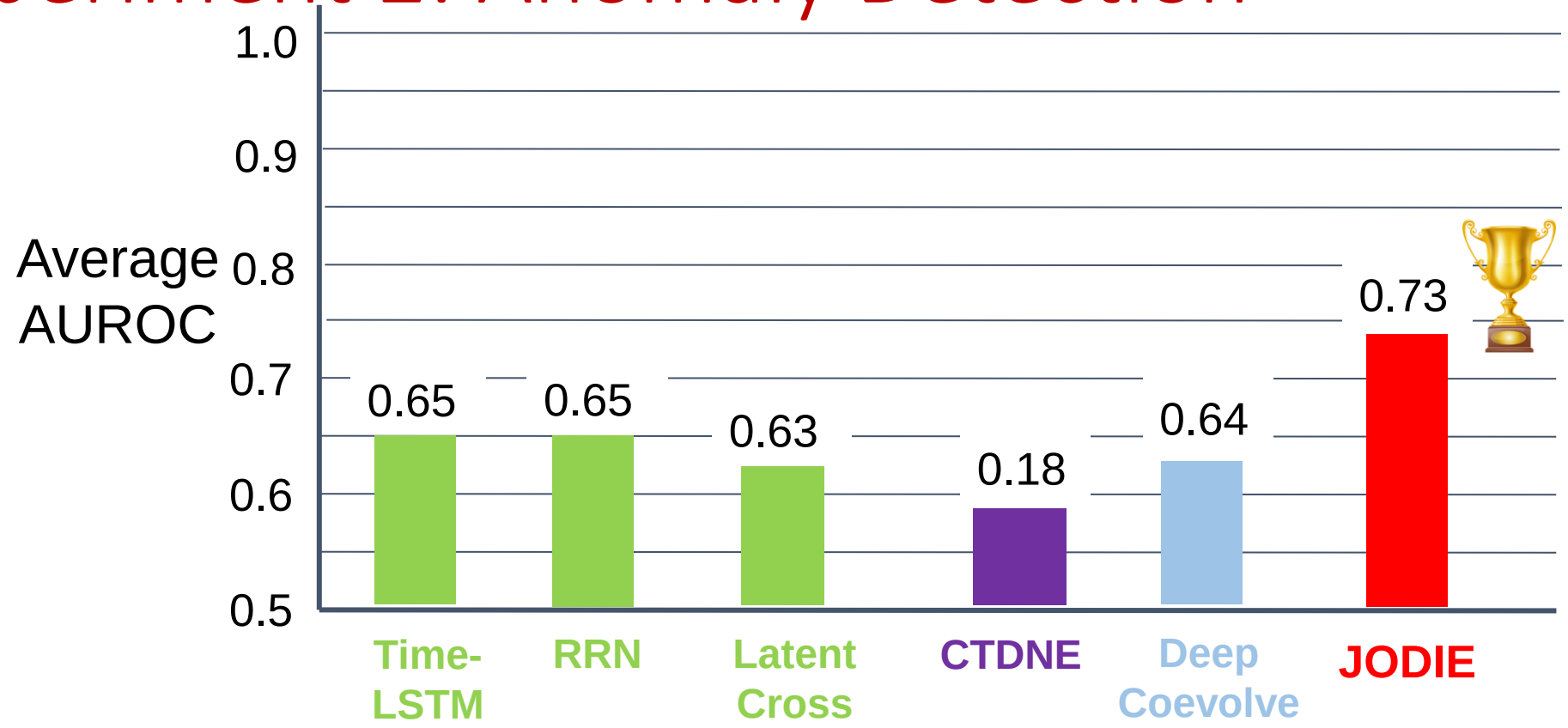
Experiment 1: Interaction Prediction



JODIE outperforms baselines by > 20%

[KDD'19]

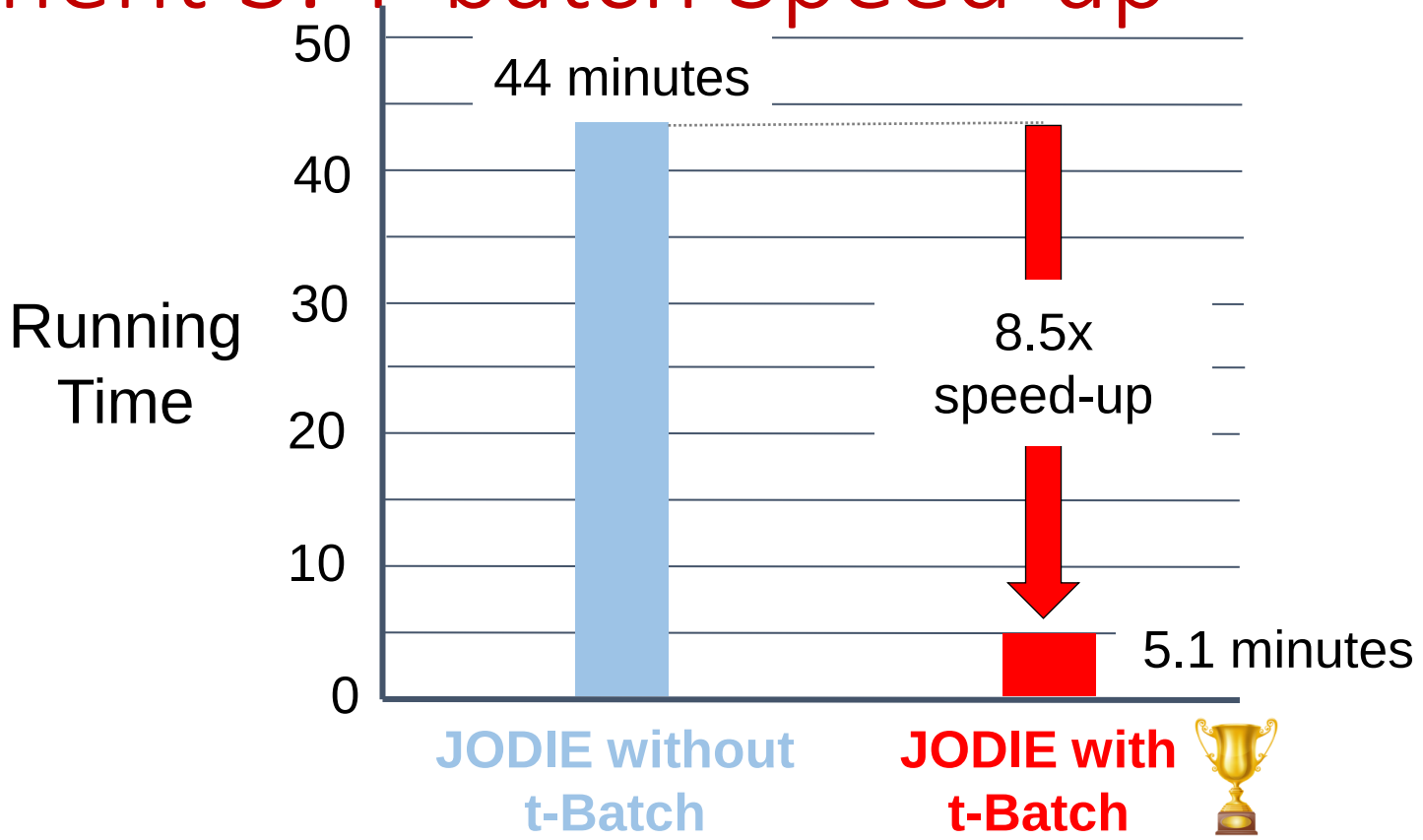
Experiment 2: Anomaly Detection



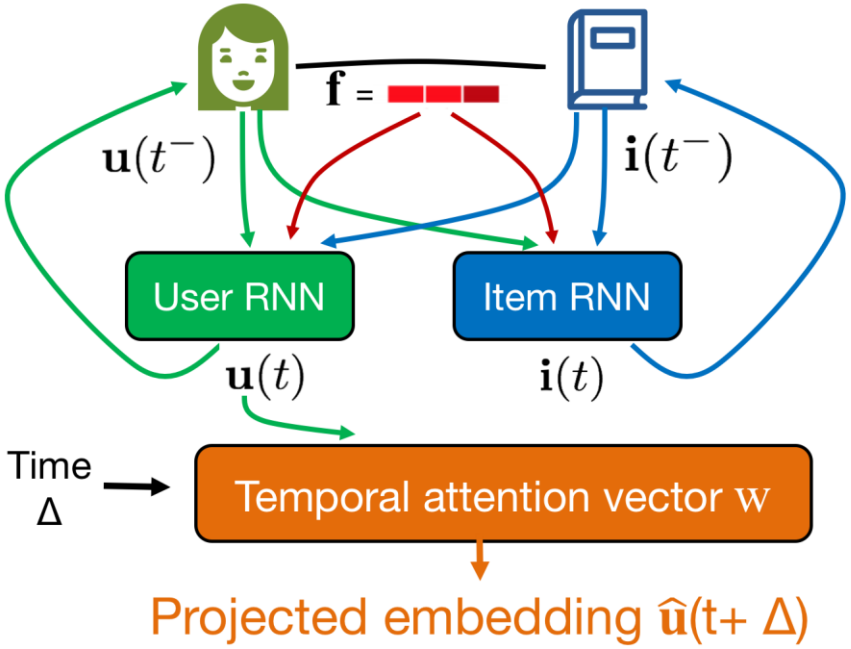
JODIE outperforms all baselines by >12%

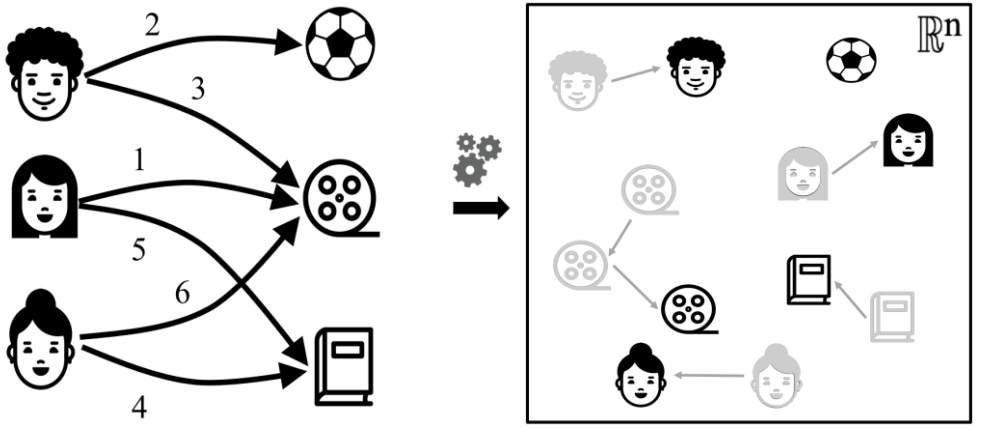
[KDD'19]

Experiment 3: T-batch Speed-up



T-batch leads to 8.5x speed-up in training





JODIE generates and projects **embedding trajectories**

Mutually-recursive RNN framework with temporal attention

- **T-batch:** 8.5x training speed-up
- **Efficient** in interaction prediction and anomaly detection
- **Extendible** to > 2 entity types

Project website: https://snap.stanford.edu/jodie