[WSDM'21 submission]

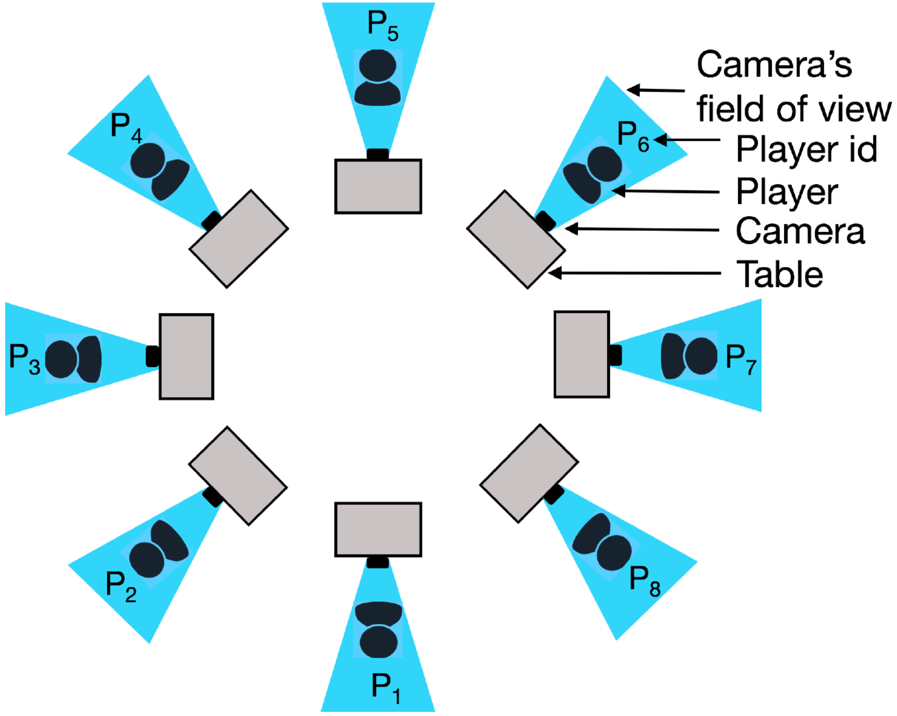
TEDIC: Neural Modeling of Behavioral Patterns in Dynamic Social Interaction Network

Jure Leskovec
Computer Science, Stanford University

Joint work with
Y. Wang, P. Li, C. Bai, VS. Subrahmanian.

Context: Social interaction events

- A game of 5-8 people who talk to each other in a party game
- People have assigned (but unknown) roles:
  - Deceivers
  - Truth-tellers
- After the game finished, participants are also surveyed about each other's characters exhibited in the game:
  - Level of dominance
  - Level of nervousness



Our task: Social prediction

- **Task:** Predicting the dominance, deception, nervousness among a group of people in an interaction event.
- **Input: "dynamic social interaction network"**
  - Dynamic Features: People's facial expressions, voice traits, etc.
  - Dynamic "gazing" data: Who looks at whom at what time.
- **Output:**
  - **Task 1:** The most/more dominant person in the group
  - **Task 2:** The most nervous person in the group
  - **Task 3:** Lying people ("spies") in the group

Joint work with Dartmouth, UCSB, Arizona

Challenge #1

Interactions and personal traits (facial expressions, …) can be **highly dynamic** and **complex**:

--- People shift their eye focus 60+ times on average within one minute.

--- Interleaving patterns are subtle and scattered in a long time span (30+min/game)

(Data sample visualization in the next slide)

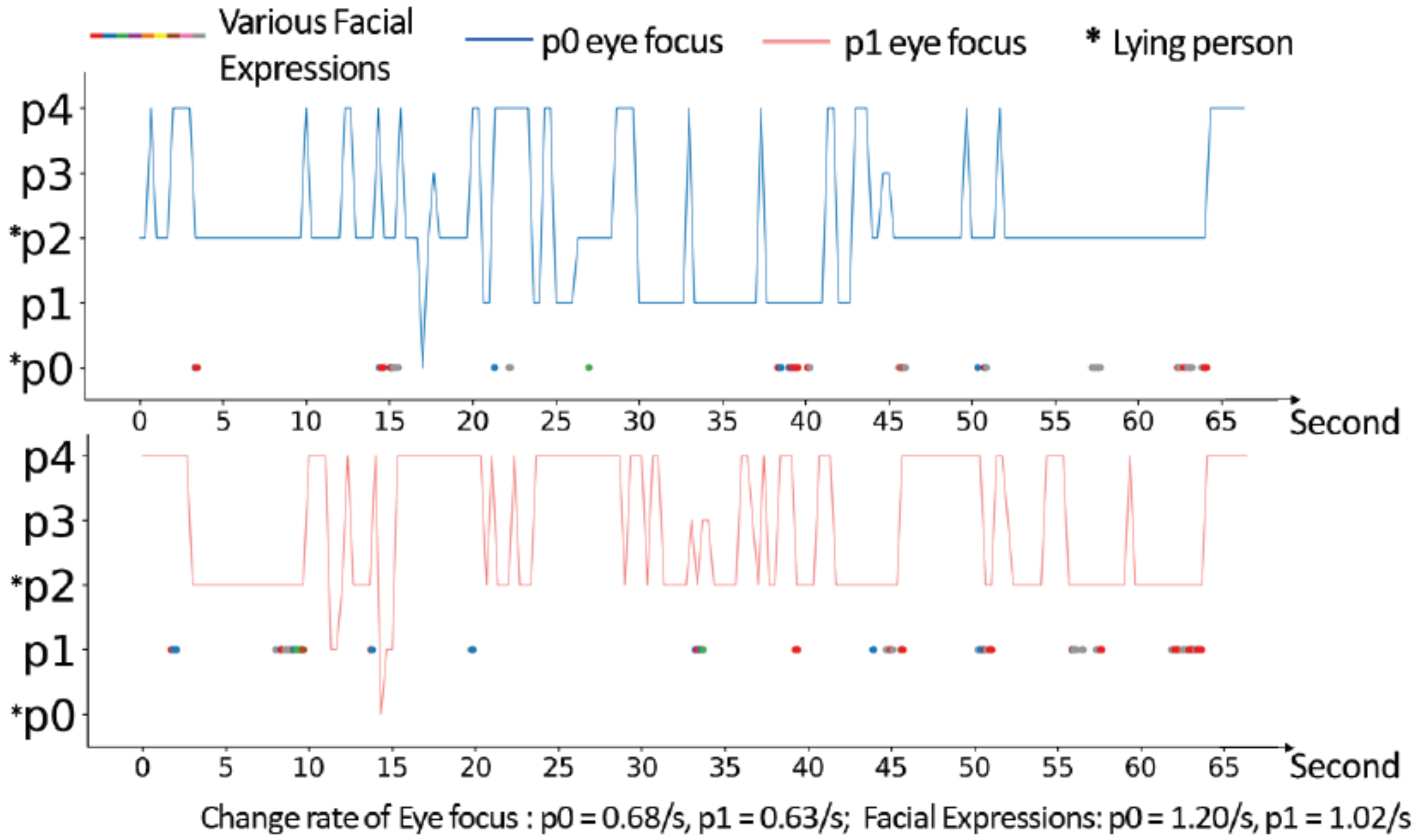


Figure 1: Visualization of two people's (p0 and p1) various behavioral traits in a social-interaction game "RESISTANCE" among 5 people p0-p5. Their eye focuses are used to built up the dynamic network and their facial expressions are used as node attributes. Both are interleaved and change over time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

Challenge #2

- Interactions can be extremely long and of very huge quantity
  - Well **beyond the memory capacity** of existing RNN-based deep models!
  - **Strong noise** accompanying **fine-grained behavioral patterns** over **long** time (so smoothing sequence along time also wipes out useful details!)

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 $t = 0.33$s). †: We count all the interactions with gazing probability ≥ 0.5.

Challenge #3

- Previous methods: Heavy process of **handcrafting features** using expert knowledge
  - time-consuming
  - experts not always readily available
  - new features required for different downstream tasks

Our solution: TEDIC

- A neural network model that is **performs well** across different prediction tasks:
  - *Detecting dominance, nervousness, deception, etc.*

- With other desirable features:
  - **Self-explaining power**: automatically provides social insights
  - **Fairness:** Judges people from different places equally
  - **General Applicability**: Can be applied to dynamic social networks of various kinds *(e.g. proximity-based one from body sensors)*

Why Neural Network ?

Address #Challenge 3:

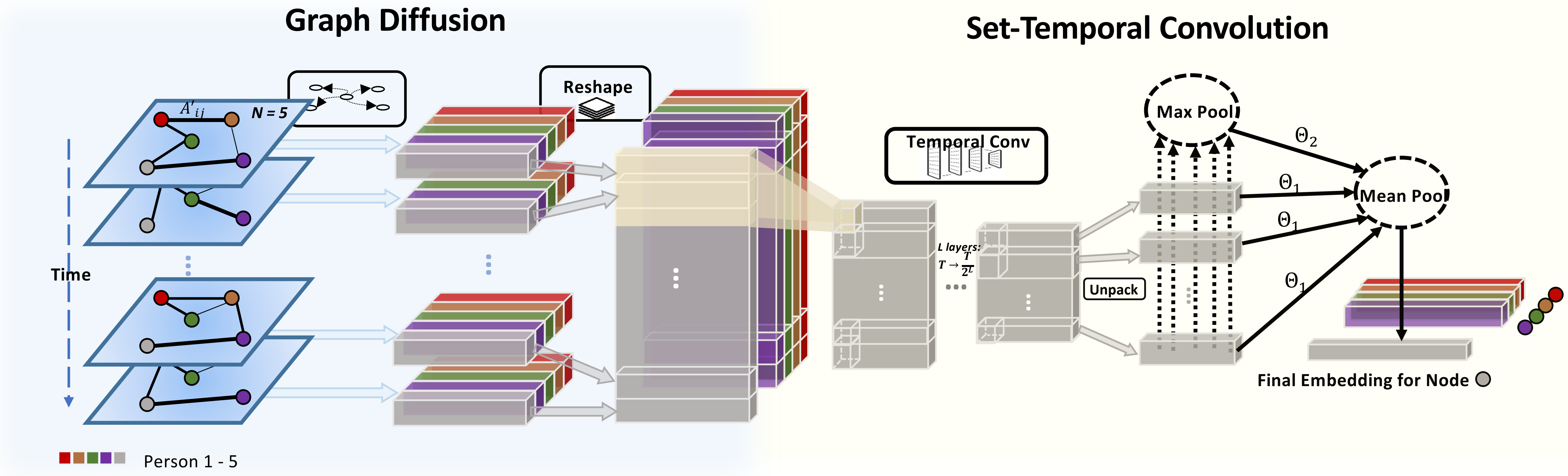
- **General**: depend much less on (task-specific) feature selection
- **Wxpressive:** capture complex patterns (may even beyond expert's manually designed ones!)

Summary of proposed framework: TEDIC

- **TE**mporal-Network **DI**ffusion **C**onvolutional Network **(TEDIC)**

- Two main modules:
  1. **Graph Diffusion module**: model how people **interact and influence each other** in real time via with their various communicative behaviors
  2. **Set-Temporal Convolutional Network (S-TCN)** module: a **hierarchical** structure to **filter, collect, and process** behavioral patterns scattered over time

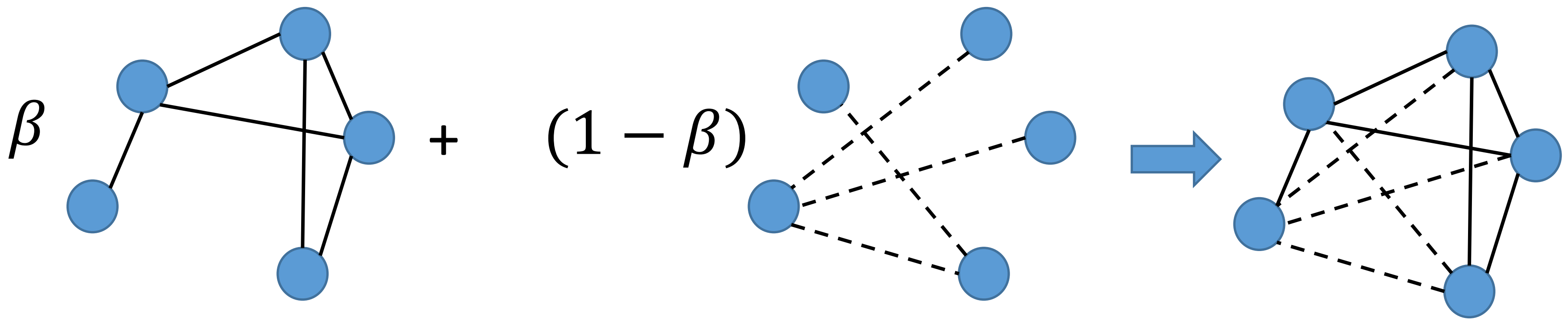
TEmporal-Network DIffusion Convolutional Network (TEDIC)



1. Graph diffusion refines node features in each network snapshot
2. Set-temporal convolution aggregates the refined node features over time

Graph Diffusion --- I

- The input graph combines positive and negative graphs via a trainable parameter.



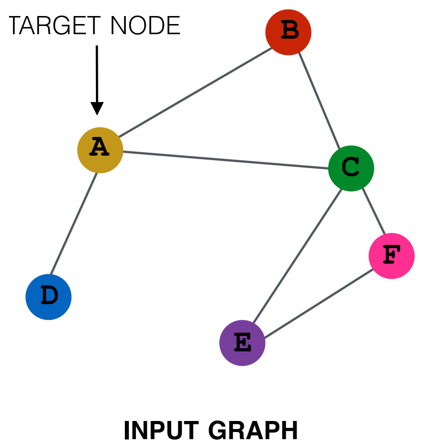
$$A' = \beta A + (1-\beta)\bar{A} = (2\beta - I)A + (1-\beta)I.$$

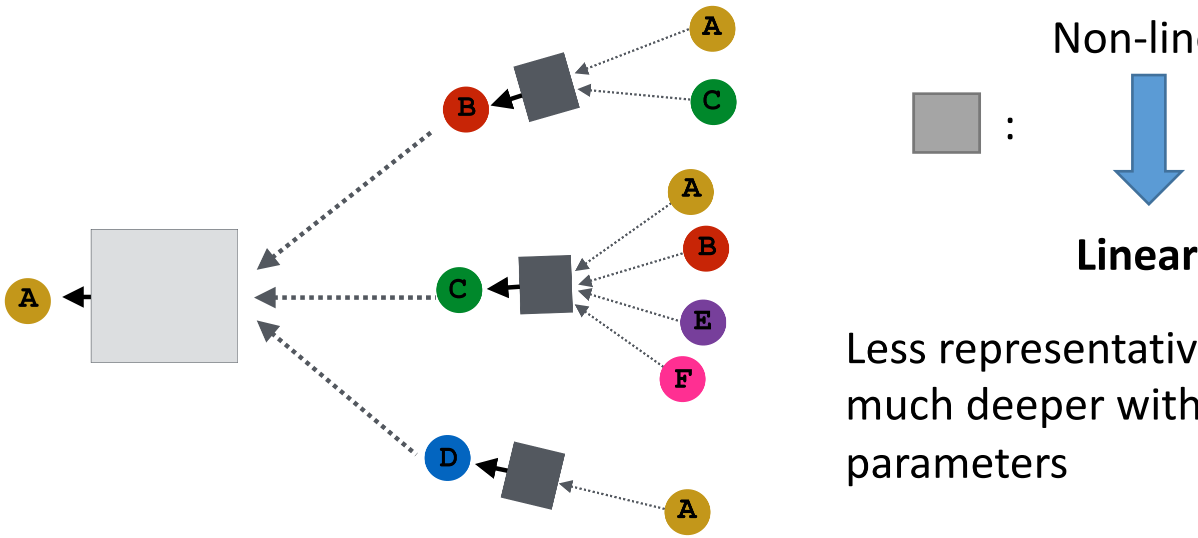
$$W' = D'^{-1}A'$$ Degree Normalization

- β holds certain self-explaining power.

Graph Diffusion --- II

- Modeling long-hop interweaving of highly dynamic node attributes and interactions with only a few labels





Less representative but much deeper with less parameters

Graph Diffusion --- III

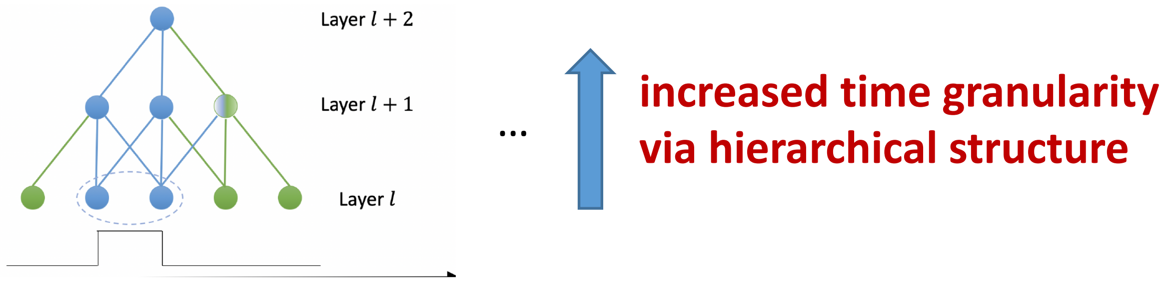
$$H_t = \sum_{k \geq 0} H_t^{(k)} \Gamma_k = \sum_{k \geq 0} (W_t'^T)^k H_t^{(0)} \Gamma_k, \quad H_t^{(0)} = f(X_t)$$

- Subscript *t* index the network screenshot along time
- X_t is **input node features** for the interacting group of people
- $f(\cdot)$ **pre-transformation** function to preprocess the features
- Γ_k is a diagonal matrix denoting **diffusion weights** at step *k*
- $H_t^{(k)}$ is the **hidden node embeddings** after *k*-hop diffusion
- W' is the **network (positive and negative edges)**
- Diffusion is a linear transformation done via the matrix multiplications

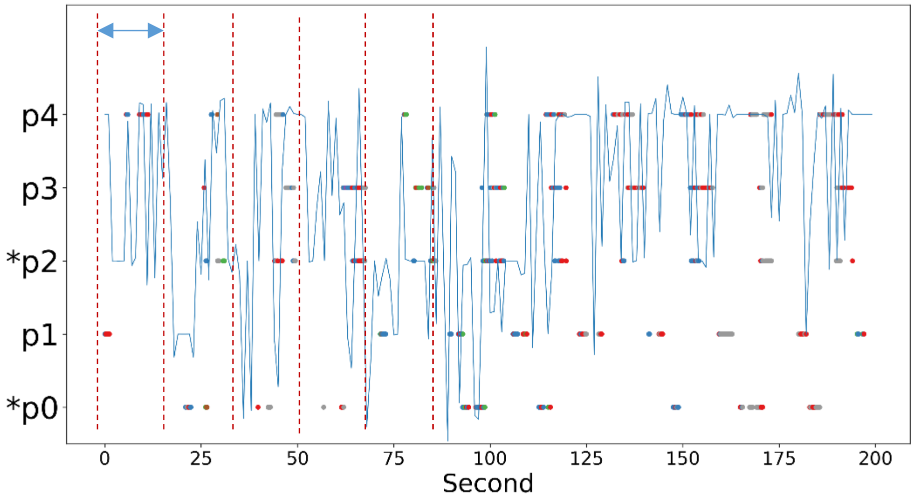
Set-Temporal Convolutional Newtork --- I

- The order (of interactions) is important at micro level
E.g., A looks at B and then B avoids the eye contact…

- Controllable time granularity via different *#layers*
→ details at different micro levels.



time granularity



Set-Temporal Convolutional Newtork --- II

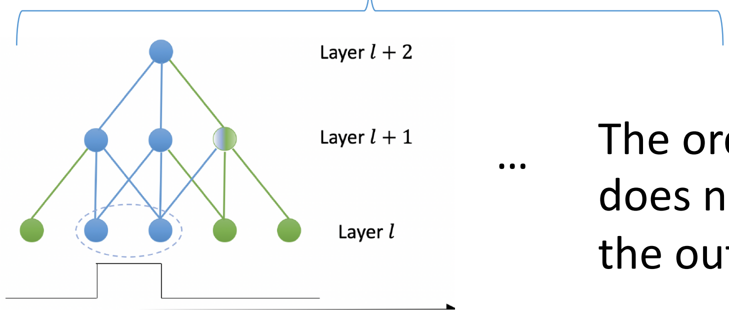
- The order (of interactions) is not important at macro level

E.g., A looks at B and then B avoids the eye contact…

(in the first 100s v.s. last 100s)

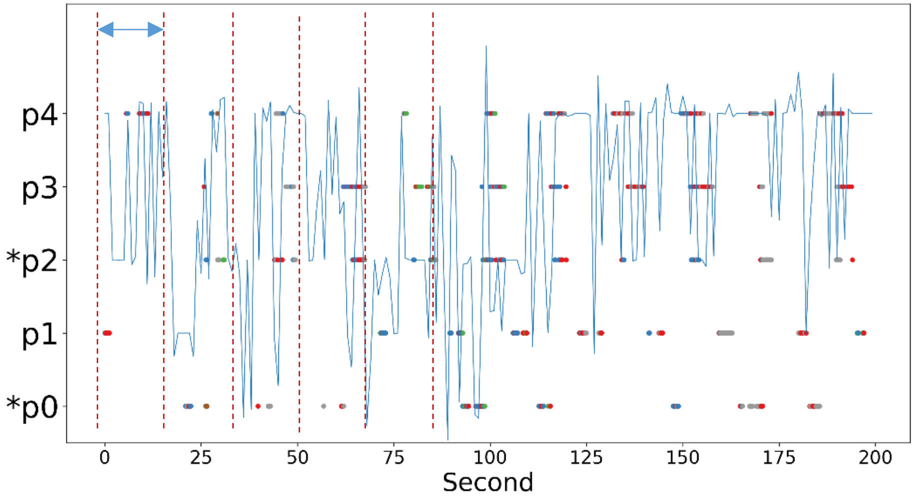
- Handle the long time span

Set – pooling {f1, f2, …, fk} -> f



The order of fi, fj does not impact the output

time granularity



Set-Temporal Convolutional Network --- III

1. Hierarchical Temporal Convolution interleaved with max pooling to capture behavioral patterns on different time scales

$$\bar{Z}_t^{(l)} = \mathrm{ReLu}(Z_t^{(l-1)} * C_t^{(l)}), \quad \{Z_t^{(0)}\}_{1 \le t \le T} = \{H_t\}_{1 \le t \le T}$$

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

*** : the convolution operator**

2. Mean-max Pooling over shortened interaction sequence $\{Z_t^{(L)}\}_{1 \le t \le T^{(L)}}$

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

$$Z_{\text{out}} = \text{mean-pool}_{1 \le t \le T^{(L)}}(\mathrm{ReLu}(Z'_t\Theta_1 + Z_{\max}\Theta_2))$$

Various Set Pooling to filter/collect signals scattered over time

Dataset & Task

	Task	Dataset	Classification	Networks	Avg. Time Steps*	Group Size	Interactions†
Ours	Dominance (R)	RESISTANCE-D	multiclass	956	2,514	$5 \sim 8$	4.007×10^6
	Dominance (E)	ELEA	binary	27	2,545	$3 \sim 4$	6.474×10^3
	Deception	RESISTANCE-S	binary	2,157	2,258	$5 \sim 8$	2.439×10^7
	Nervousness	RESISTANCE-N	multiclass	1,097	2,528	$5 \sim 8$	4.910×10^7
	Community	CIAW	multiclass	1	20	92	2.149×10^4

Our 3 datasets differ due to several practical reasons of data collection

Task No. and Task	Task Description
(1) Dominance Identification	Predict most/more dominant people in each interaction event
(2) Deception Detection	Predict all the lying people in each interaction event
(3) Nervousness Detection	Predict most nervous person in each interaction event
(4) Community Detection	Predict each person's community belonging

- The last task, community detection on a very different dataset CIAW, is included to show more general applicability of our framework.

Performance

Method		Dominance (R)	Dominance (E)	Deception	Nervousness
Knowledge-based	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
Dyn. GNNs	CD-GCN[30]	0.687±0.042	0.794±0.022	0.673±0.018	0.534±0.084
	GCRN[39]	0.587±0.096	0.795±0.032	0.643±0.045	0.336±0.104
	EvolveGCN[34]	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.

- **Expert knowledge-based approaches:**
  - Dominance: GDP-ML, GDP-RF, DELF, FacialCues, MKL, FacialCues
  - Deception: LiarRank, DDV, TGCN-L
  - Nervousness: FacialCues, LiarRanks
- **Dynamic GNN-based approaches:**
  - CD-GCN, GCRN, EvolveGCN

Ablation Study on Dominance Detection

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

Table 3: Ablation study on Task 1.

Note:

- **LSTM** is a type of Recurrent Neural Network widely used for capturing dynamics in sequential data
- **GCN** is a type of state-of-the-art convolutional network defined on graph-structured data, with *non-linearities* in between layers

Effect of time resolution of interaction sequence

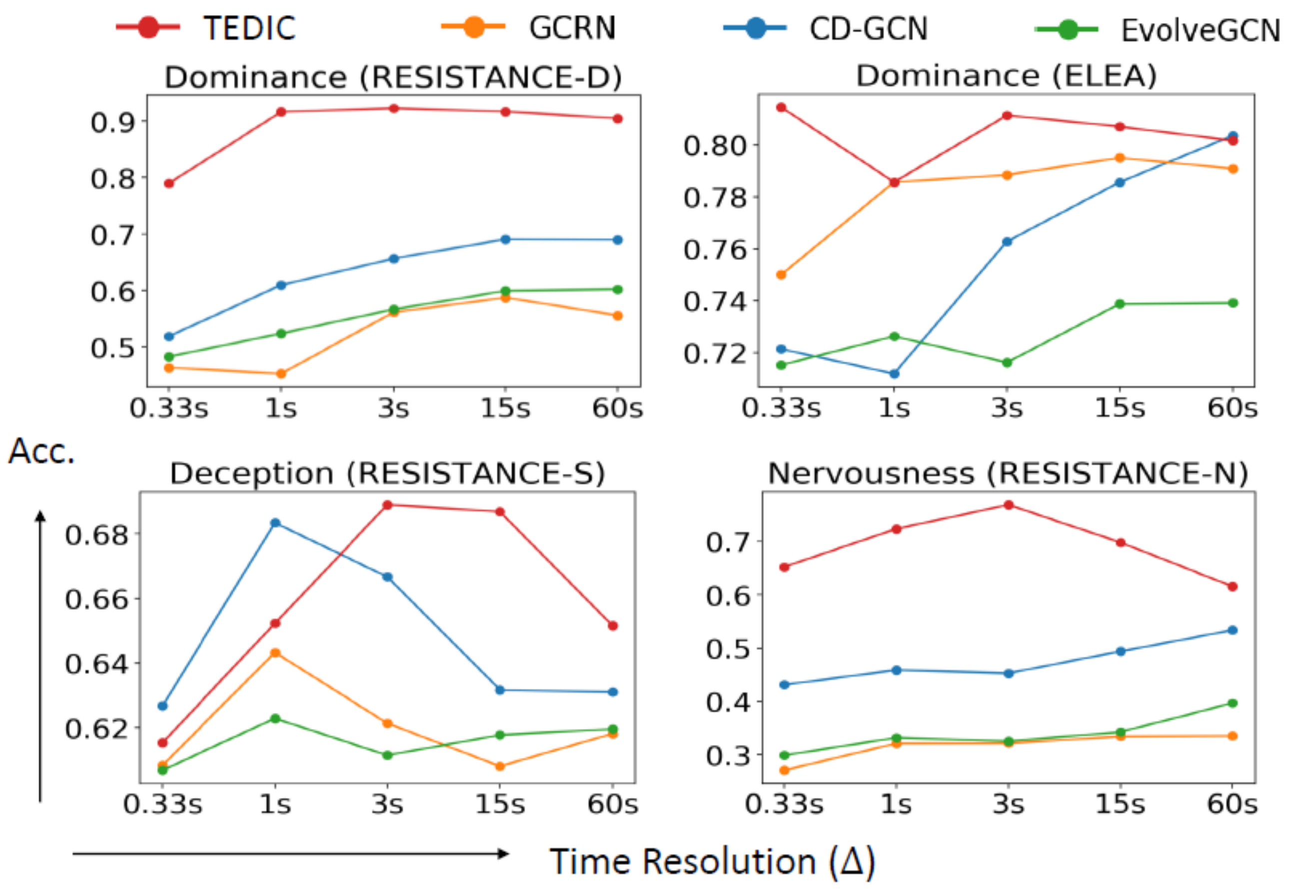


Figure 4: Accuracy as a function of time resolution of the interaction sequence.

Observations:

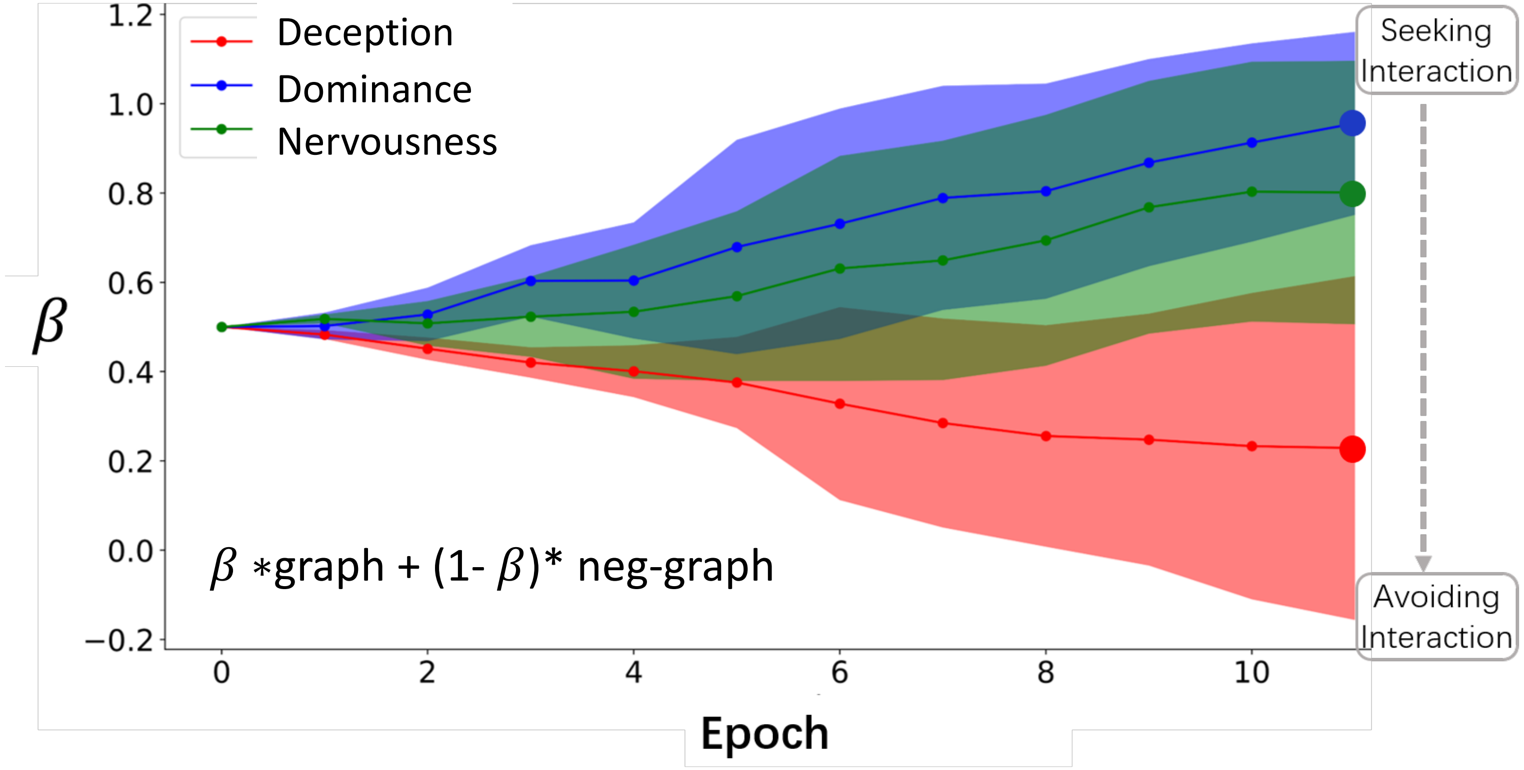
- Accuracy generally peaks at a certain time resolution:

→ both being too detailed and too rough hurts (for different reasons)

- TEDIC usually peaks at a finer resolution, with higher y values:

→ It is more capable of capturing fine details for prediction

Self-Explanation #I



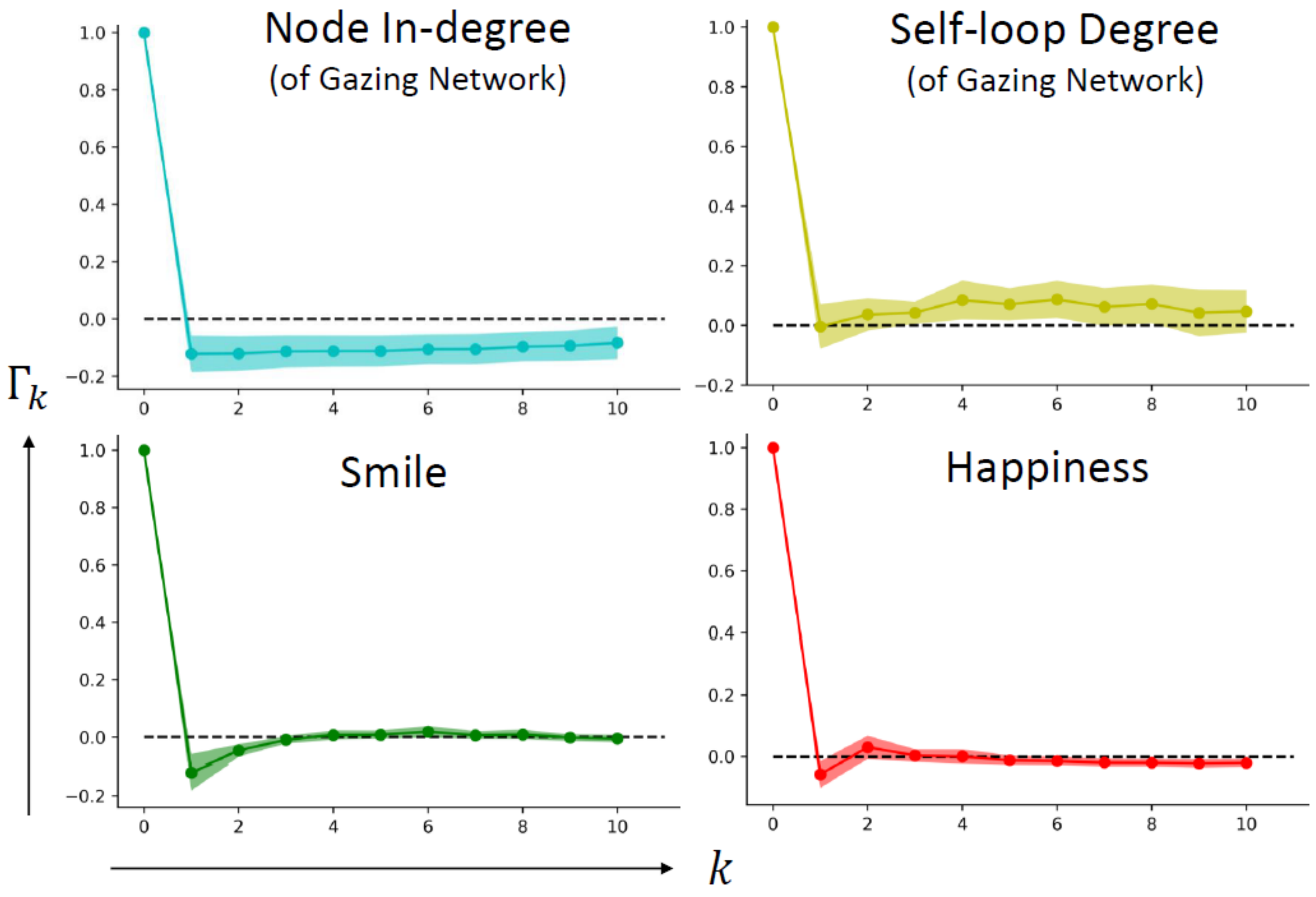
Deception Detection, Dominance Identification, Nervousness Detection run on RESISTANCE (our dataset)

Insights from Graph Weight β:

- **Deception Detection:** Avoiding interaction is more informative
- **Dominance Identification:** Seeking interaction is more informative
- **Nervousness Detection:** A mixture in between

--- Coincided with the findings of psychological theory [K. Rayner 1998]

Self-Explanation #II



Nervousness Detection. Γ_k weights for one features at k-hop.

Insights from Diffusion weightsΓ_k, k: #hops

- **Facial expression features:**

Do not impact others much in long-hop propagation

- **Others' eye focus (In-degree) and Ones' eye focus (self-loop degree)**

Impact much

--- When someone is gazed at by others, he tends to be nervous

Mini-study: Fairness of TEDIC

- Focus on deception detection task
- People from four different regions:
  - North America (USA)
  - Far East Asia (Hong Kong and Singapore, mainly Chinese ethnics)
  - Middle East Asia (Israel)
  - Africa (Zambia, Uganda)
- **Two research questions:**
  1. **Does our model learn knowledge regarding people's regional identity?**
  2. **To what extent does our model introduce bias to the decision process?**

Does our model learn knowledge about people’s regional identity?

- Extract embeddings from different layers
- Feed into a simple linear classifier to classify people’s regional identity, use **the accuracy to quantify** how much knowledge gets learned

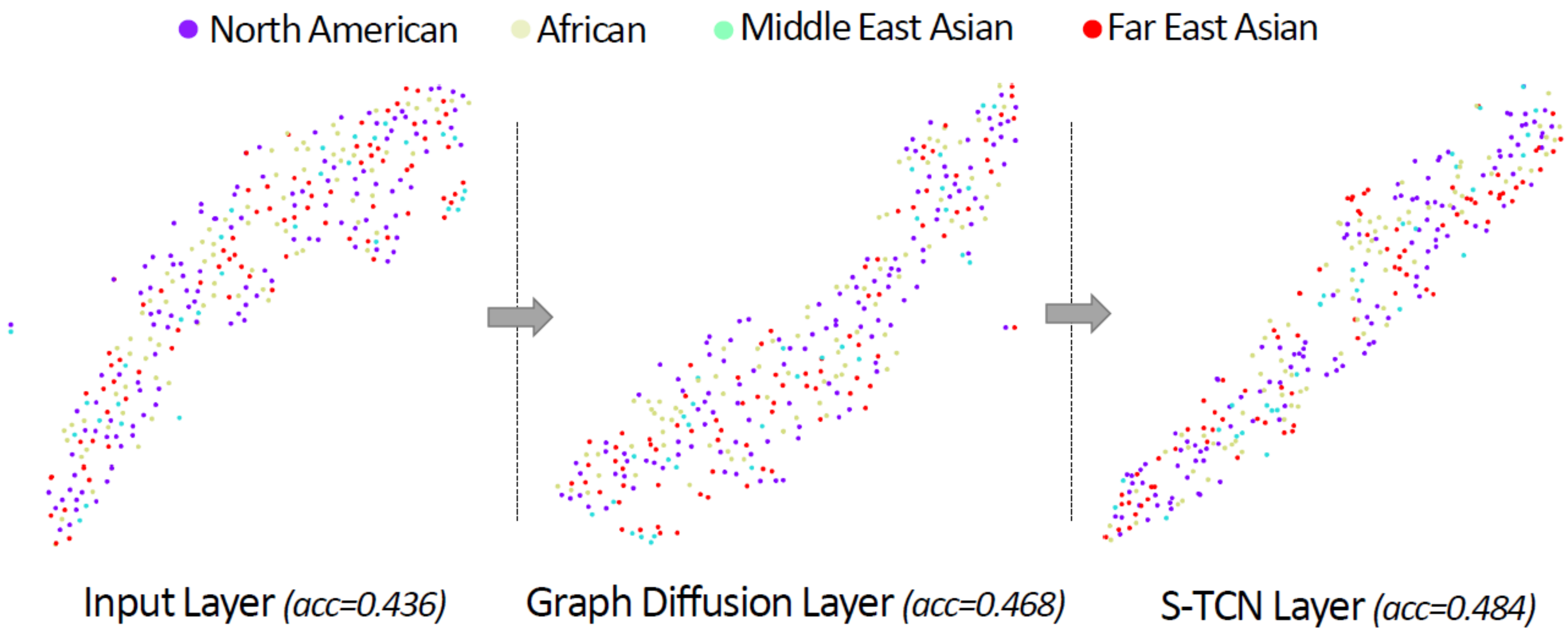


Figure 7: Projected embeddings output by different layers of our model with the corresponding regional identities.

To what extent does our model introduce bias to the decision process?

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.

- Statistically, no significant clue (i.e. $p < 0.05$) is found with our model being discriminatory towards people with different regional identities.

General applicability – A new dataset

- Is TEDIC applicable to a wider range of dynamic social interaction networks (beyond the "gazing network" in our data) ?
- We evaluate it on a very different social interaction network whose dynamics are less vibrant and the sequence is shorter

- **Dataset:** Contacts in a Workplace (CIAW)
- Contains 92 people's proximity information over two weeks collected by body sensors in an office building (one proximity network/day)
- **Task:** infer one's department identity

General applicability - Results

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.

- TEDIC can still perform well in such a very different scenario
- Though the accuracy gain to the strongest baseline is comparatively more marginal.
- We attribute the consistent performance to the high robustness of the S-TCN block to handle sequences with various length

Thanks for your time!

Questions / comments?