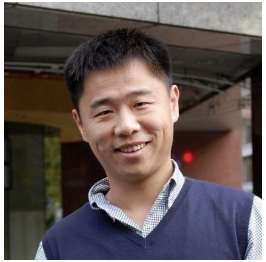
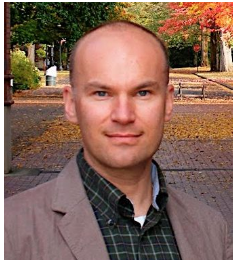
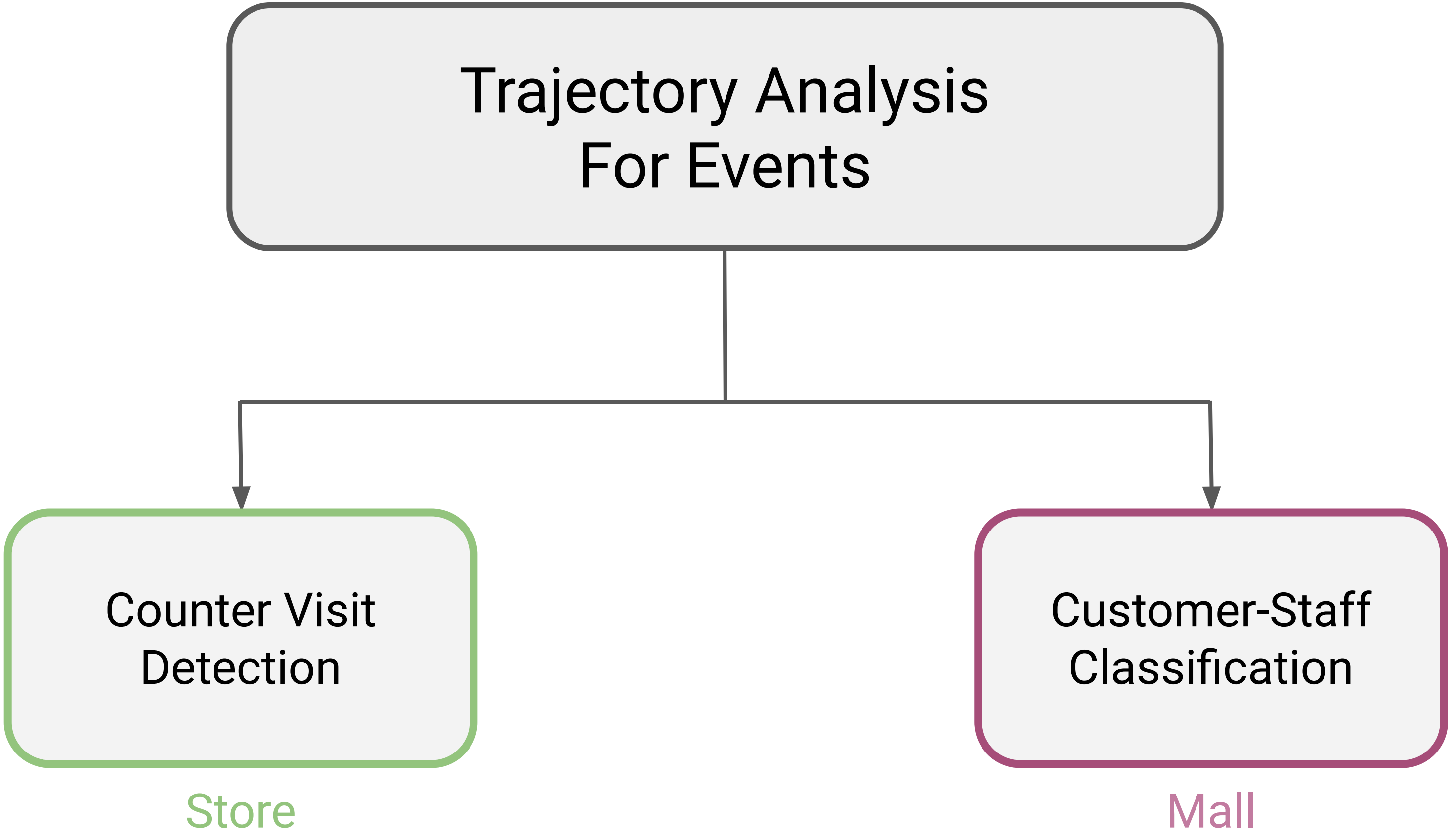
Trajectory-based Event Detection

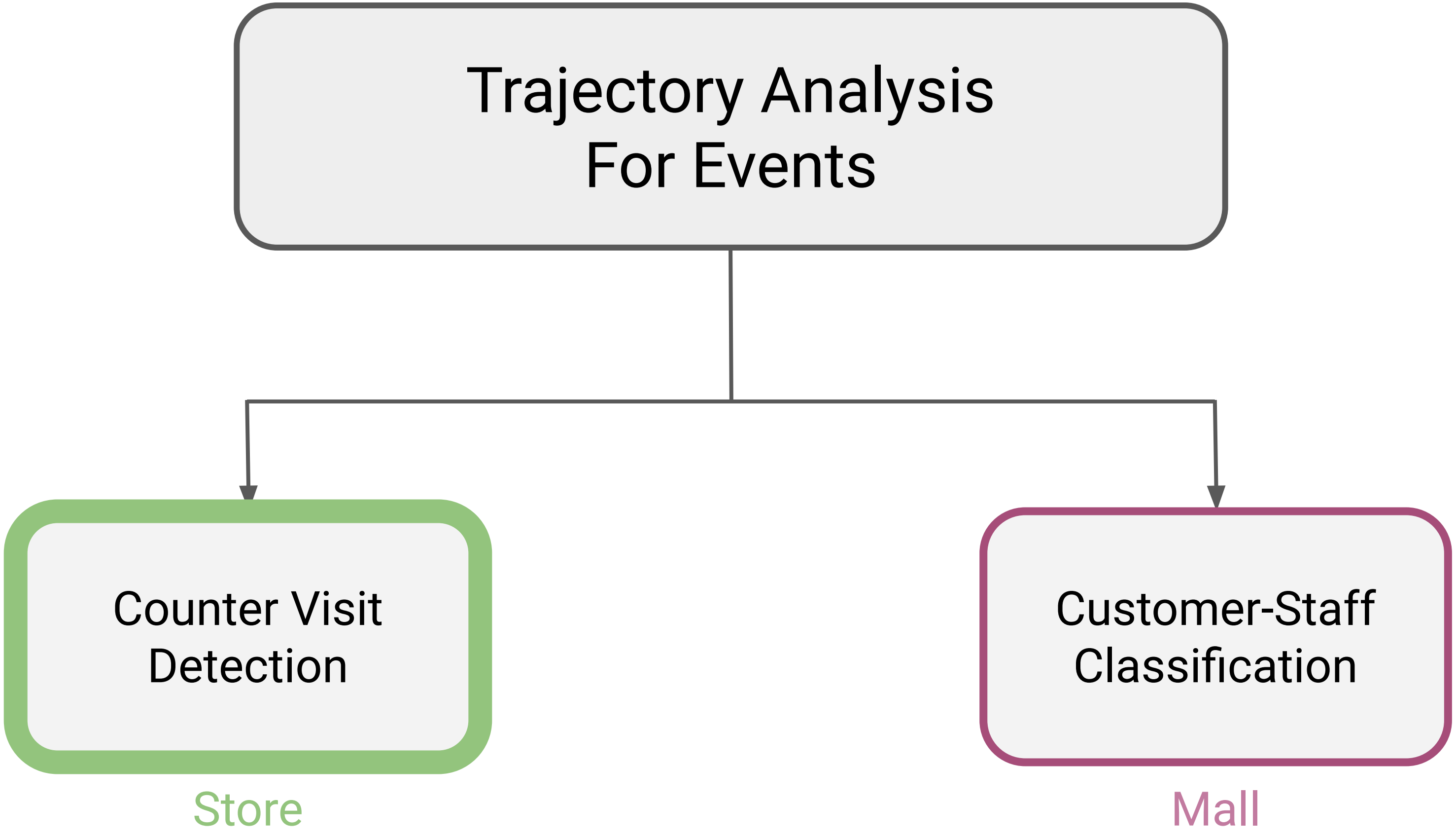

Purva Tendulkar


Chunhui Gu


Juan Carlos Niebles


Sinisa Todorovic


Trajectory Analysis
For Events
Counter Visit
Detection
Store
Customer-Staff
Classification
Mall

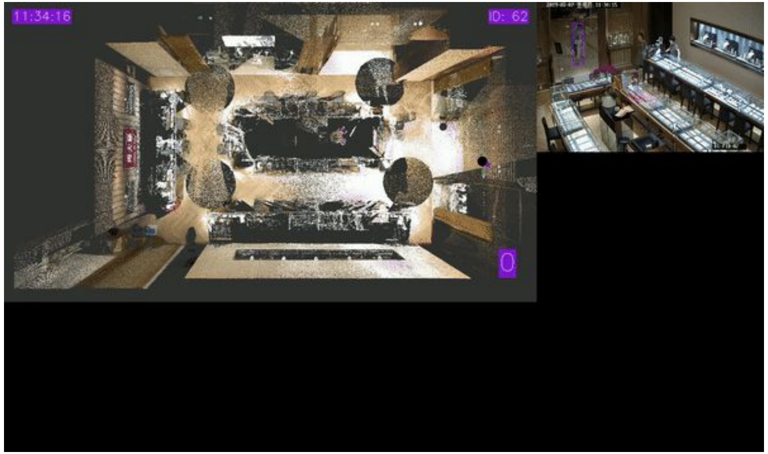

Trajectory Analysis
For Events
Counter Visit
Detection
Store
Customer-Staff
Classification
Mall

Counter Visit Detection : Task

"In a given store, for a customer, detect a counter-visit event."

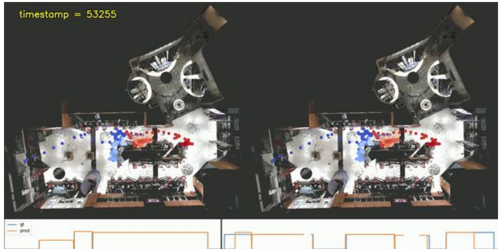
A counter-visit event is defined as <Ts, Te, C>, where

- Ts : start timestamp
- Te : end timestamp
- C : counter ID

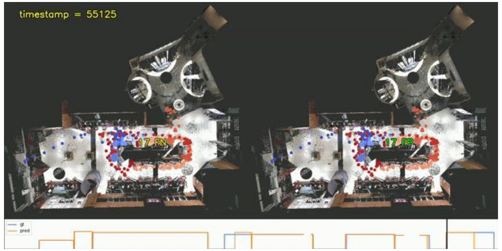


Counter Visit Detection : Challenges

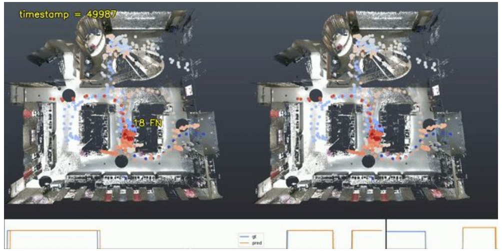
Multiple entries - ambiguity between event start and end timestamps



Localization errors - noise in trajectory

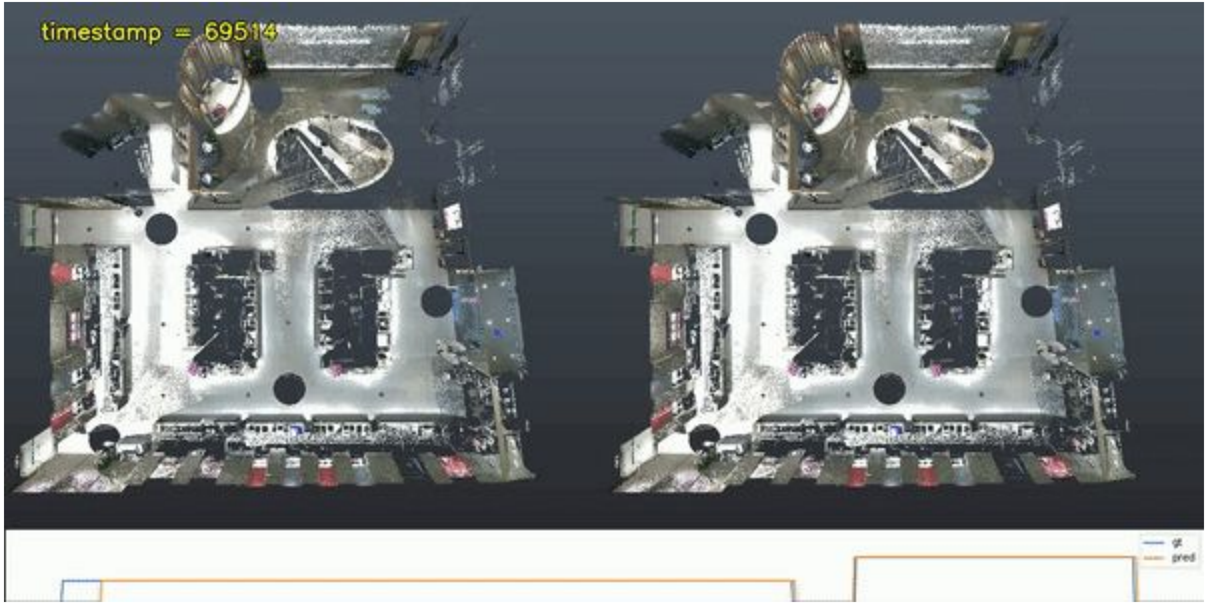


Confusion between nearby counters



Data

(x, y, t) trajectories only



Methods

Rule-based

- State Machine
- Input : x.y trajectory
- Output : 4 states (out, in, enter, exit)
- Rules : Determine events from output

Two-stage model

- 3-layer hand-designed Conv nets
- Input : Fixed length x,y trajectory
- Output :
  - Stage 1 : 4-class output (in, out, stay, background)
  - Stage 2 : Pairing in and out events for the same counter

Temporal Convolution

- Dilated Convolutional Layers
- Input : Fixed length x,y trajectory
- Output : Probability of event

RNN

- GRU
- Input : Varying length x,y trajectories
- Output : Score for presence at different counters

Methods

Rule-based

- State Machine
- Input : x.y trajectory
- Output : 4 states (out, in, enter, exit)
- Rules : Determine events from output

Two-stage model

- 3-layer hand-designed Conv nets
- Input : Fixed length x,y trajectory
- Output :
  - Stage 1 : 4-class output (in, out, stay, background)
  - Stage 2 : Pairing in and out events for the same counter

Temporal Convolution

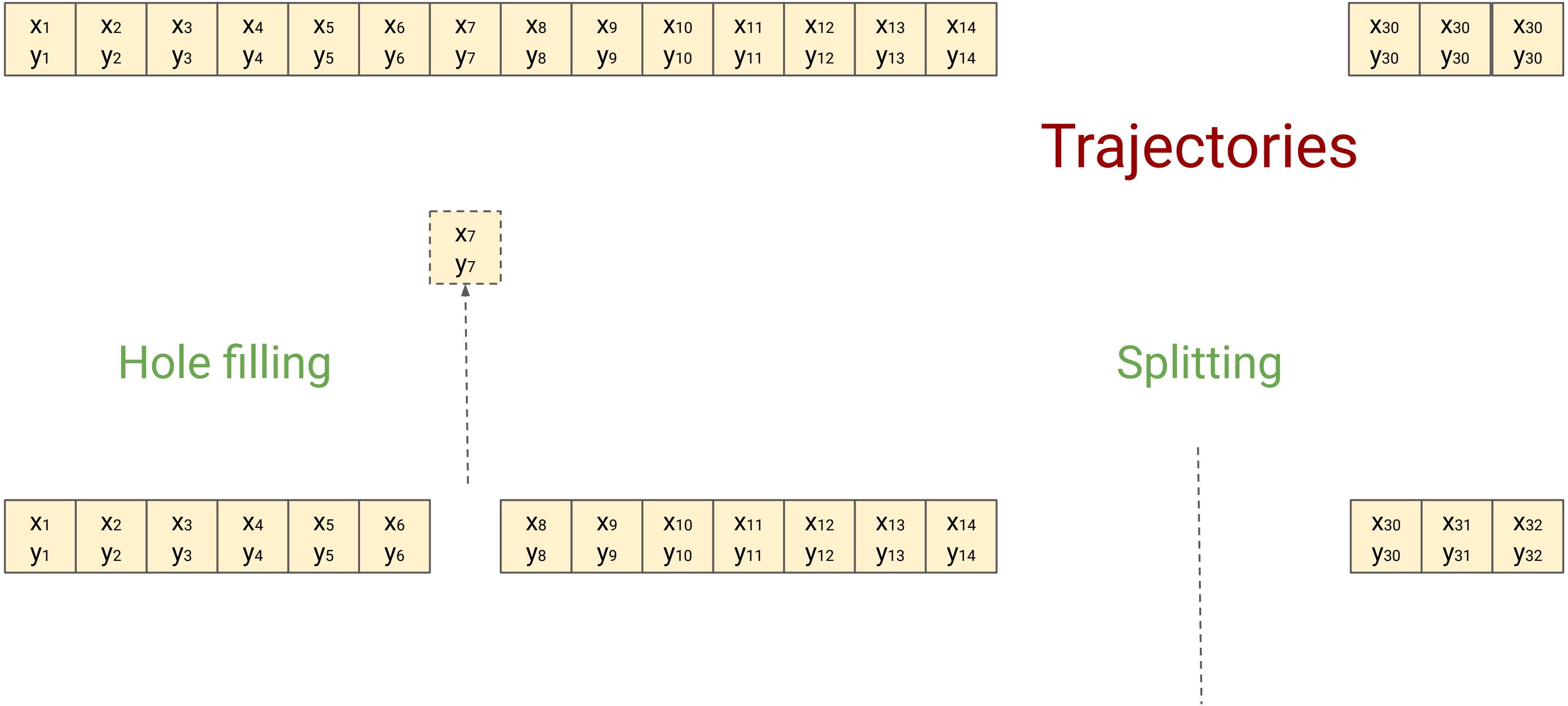
- Dilated Convolutional Layers
- Input : Fixed length x,y trajectory
- Output : Probability of event

RNN

- GRU
- Input : Varying length x,y trajectories
- Output : Score for presence at different counters

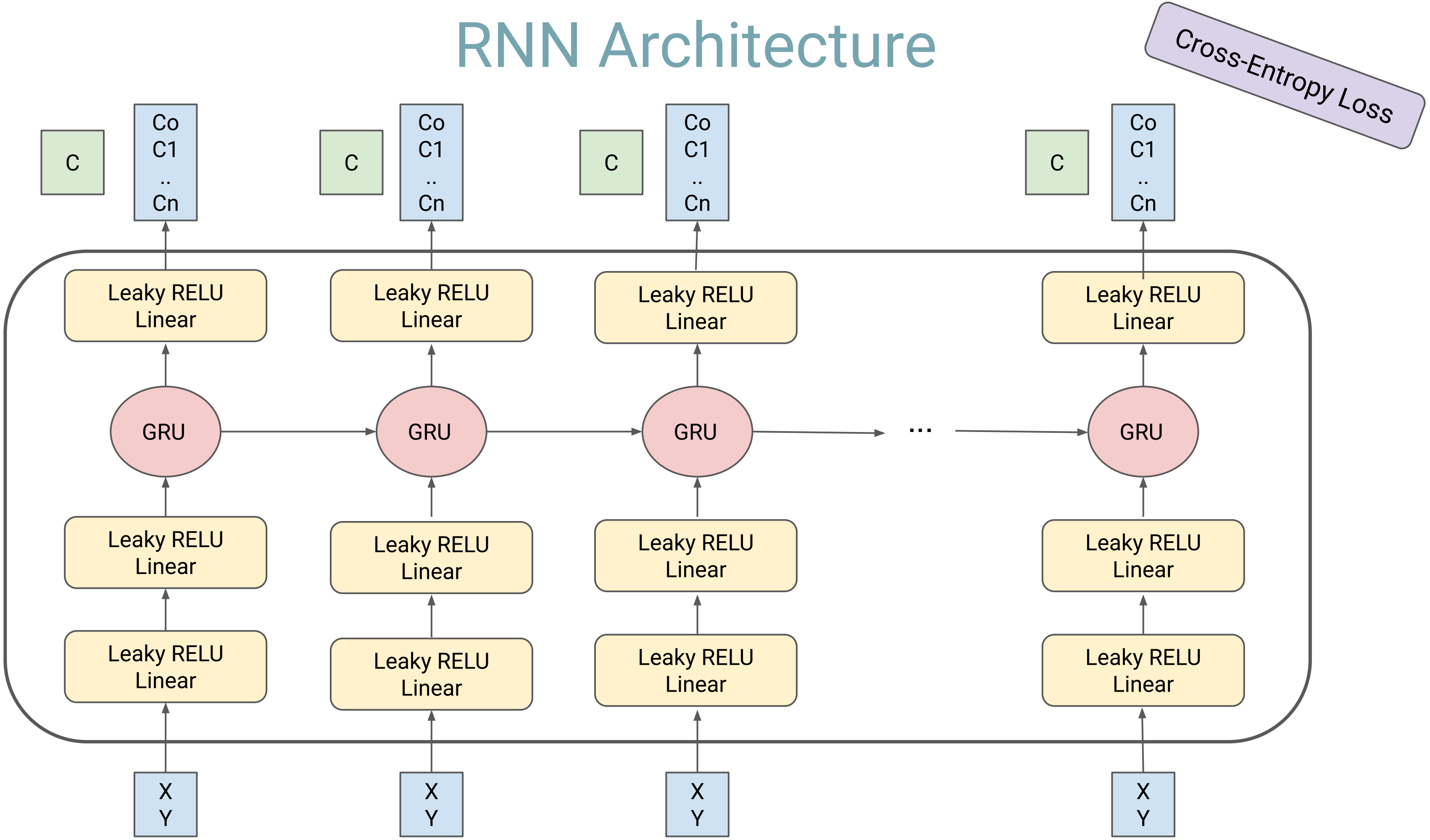
RNN for Event Detection

Data Pre-processing



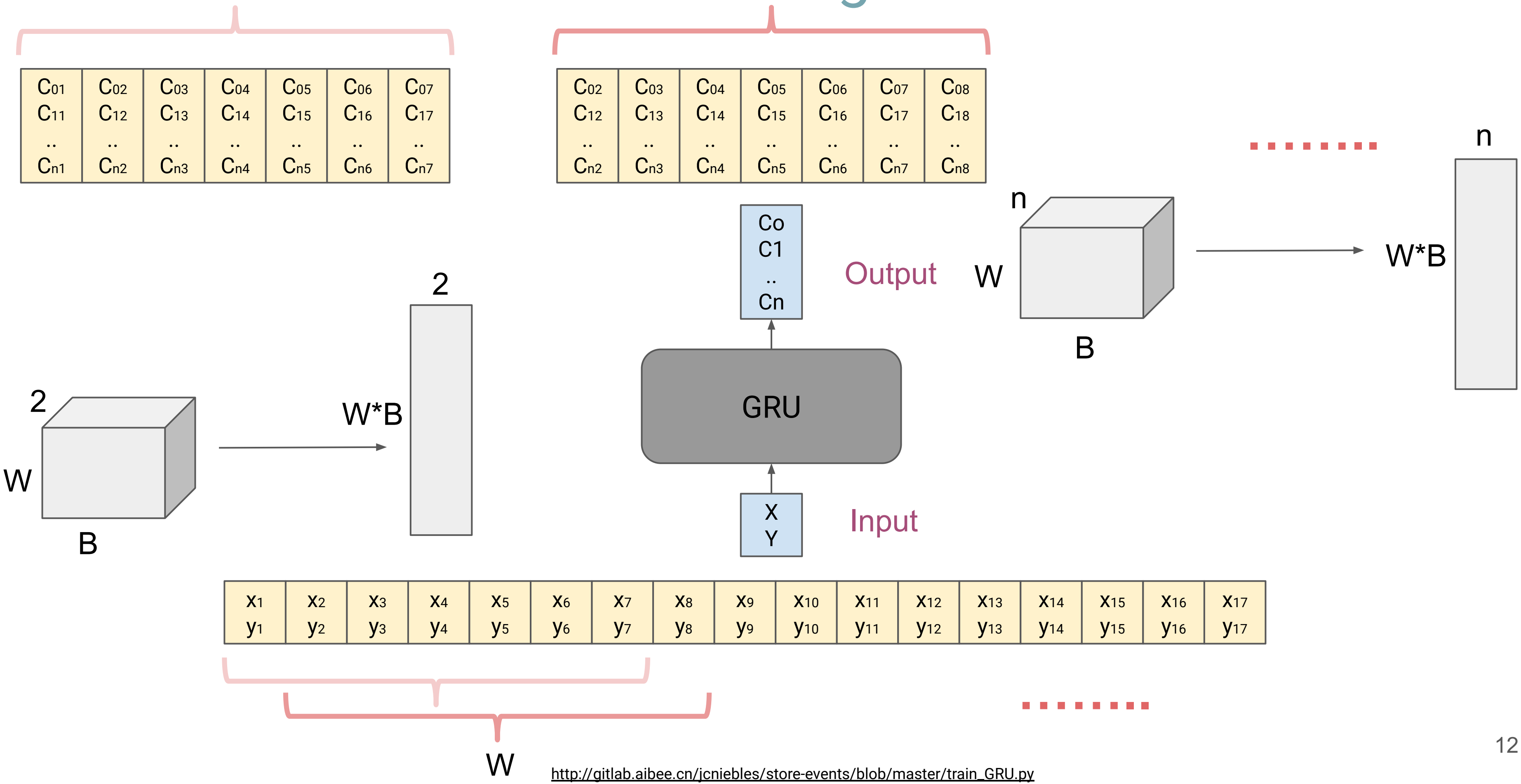
http://gitlab.aibee.cn/jcniebles/store-events/blob/master/data_loader.py

RNN Architecture



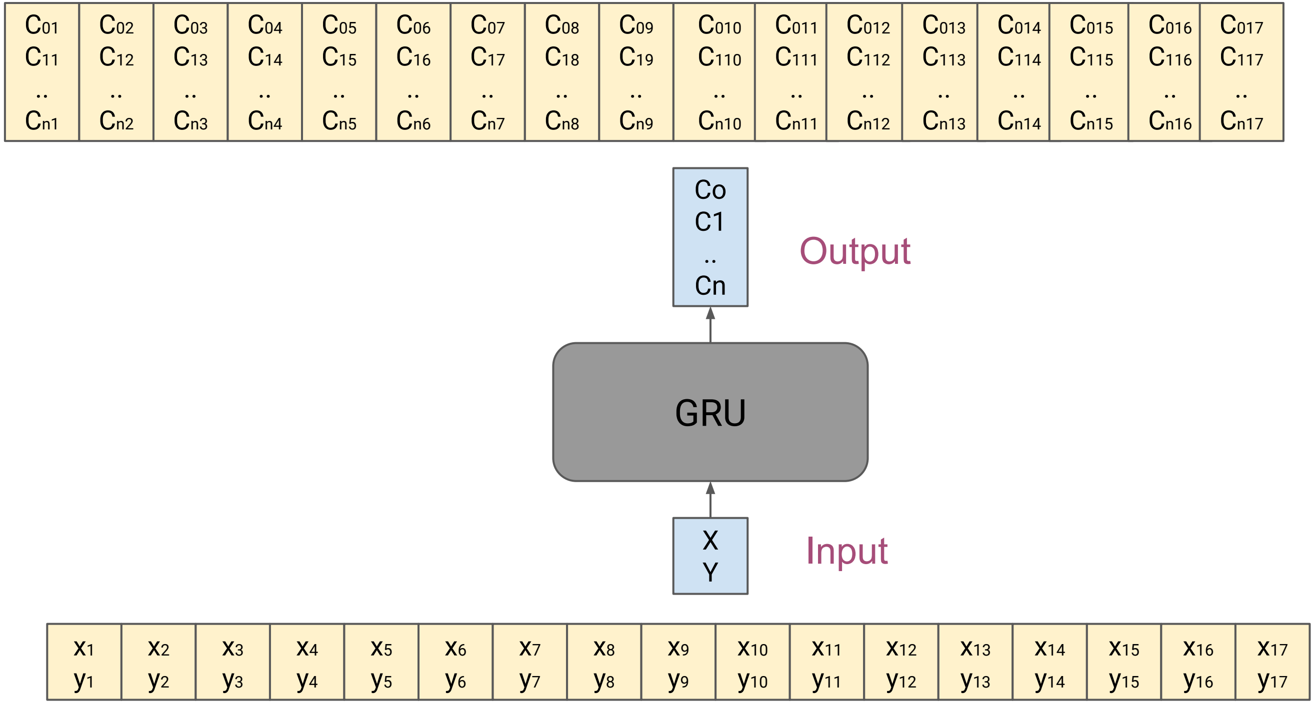
http://gitlab.aibee.cn/jcniebles/store-events/blob/master/models.py

RNN Training Data



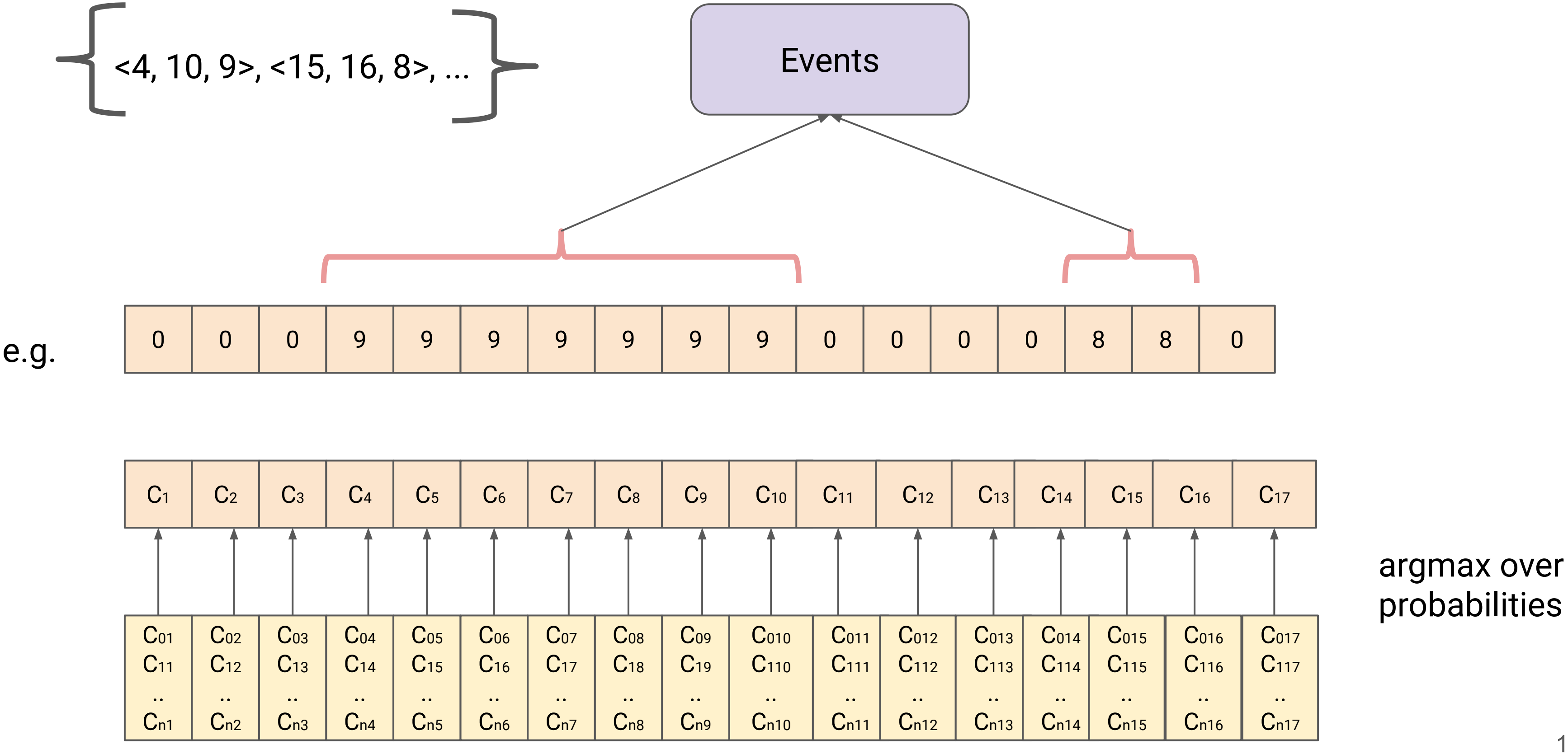
http://gitlab.aibee.cn/jcniebles/store-events/blob/master/train_GRU.py

RNN Inference



http://gitlab.aibee.cn/jcniebles/store-events/blob/master/test.py

RNN Post-processing



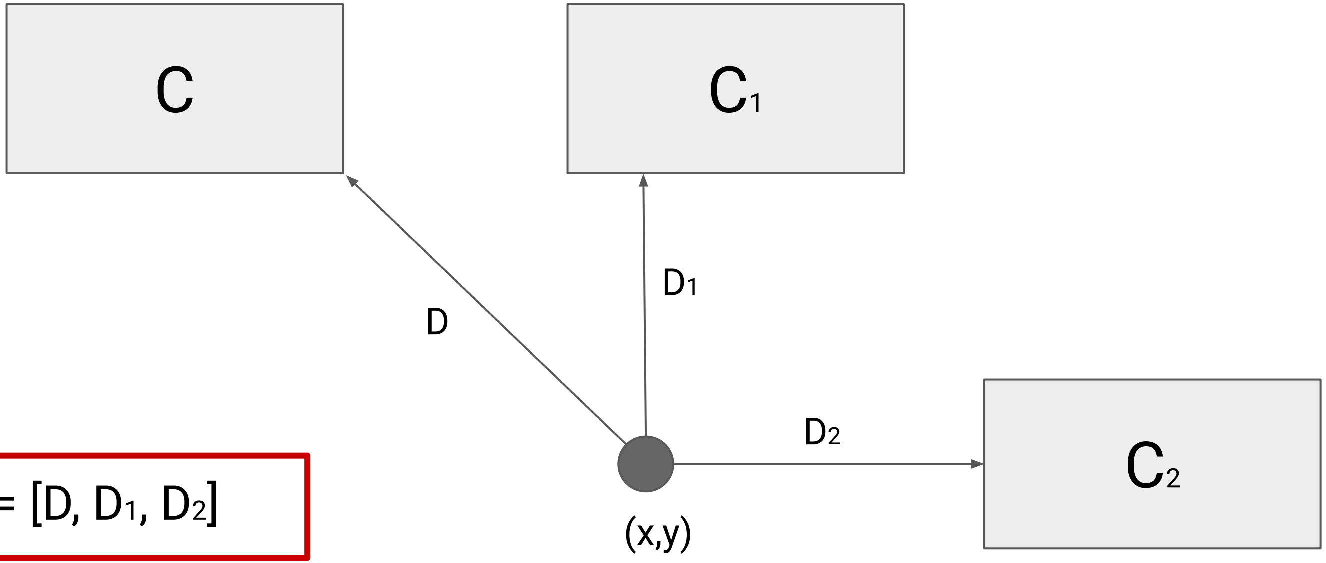
Results & Issues

Store Name	Benchmark Dates	Method	Counter Visit (tol=10s) (P/R)
CTF/beijing/xhm	20181207	Rule	0.83/0.84
		GRU	0.27/0.43

- Low performance on xy-based data
- Scalability issues
    - Dependency on store layout
    - Separate training required for each store

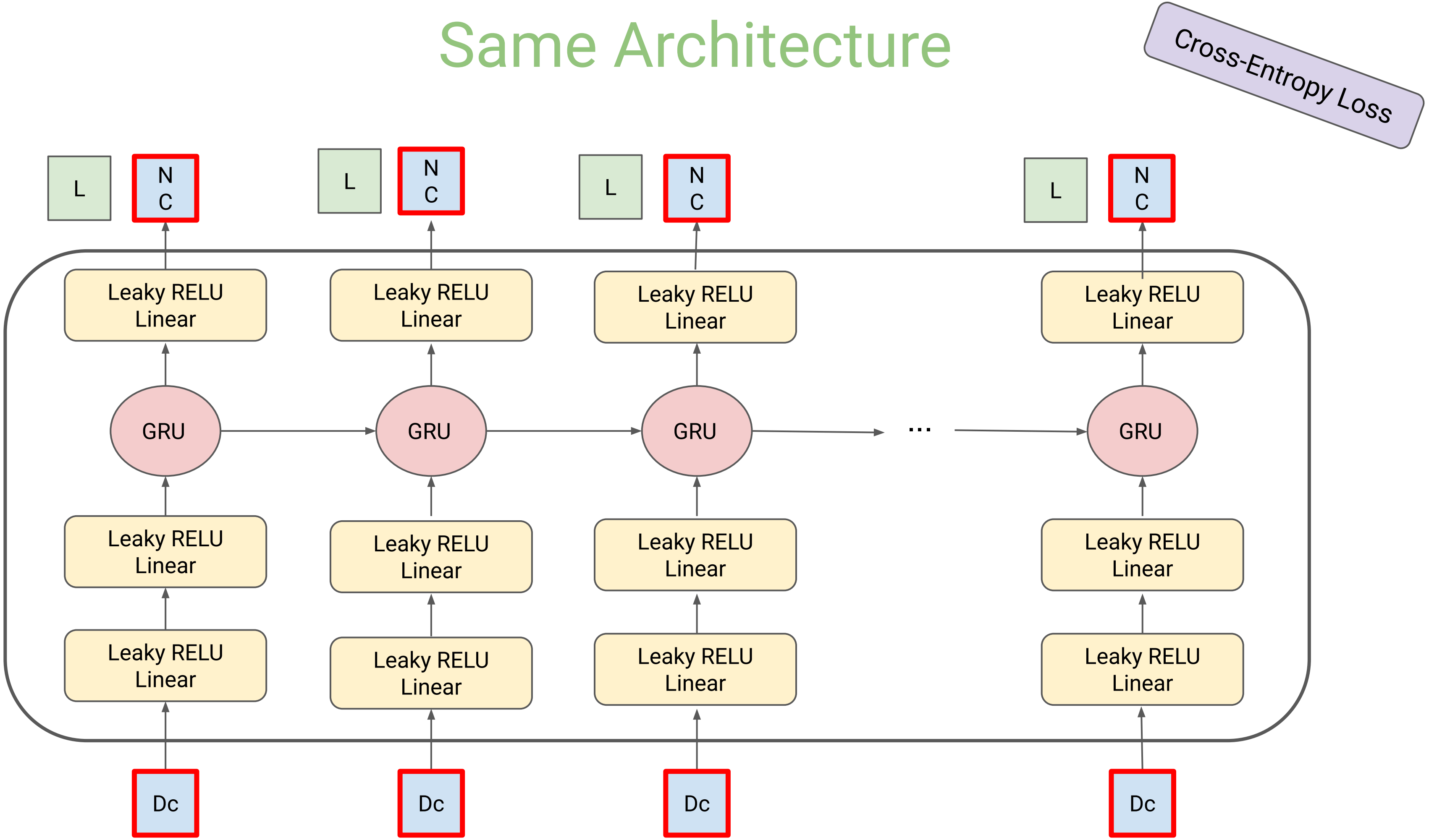
Distance-based RNN

From (x,y) to Distance



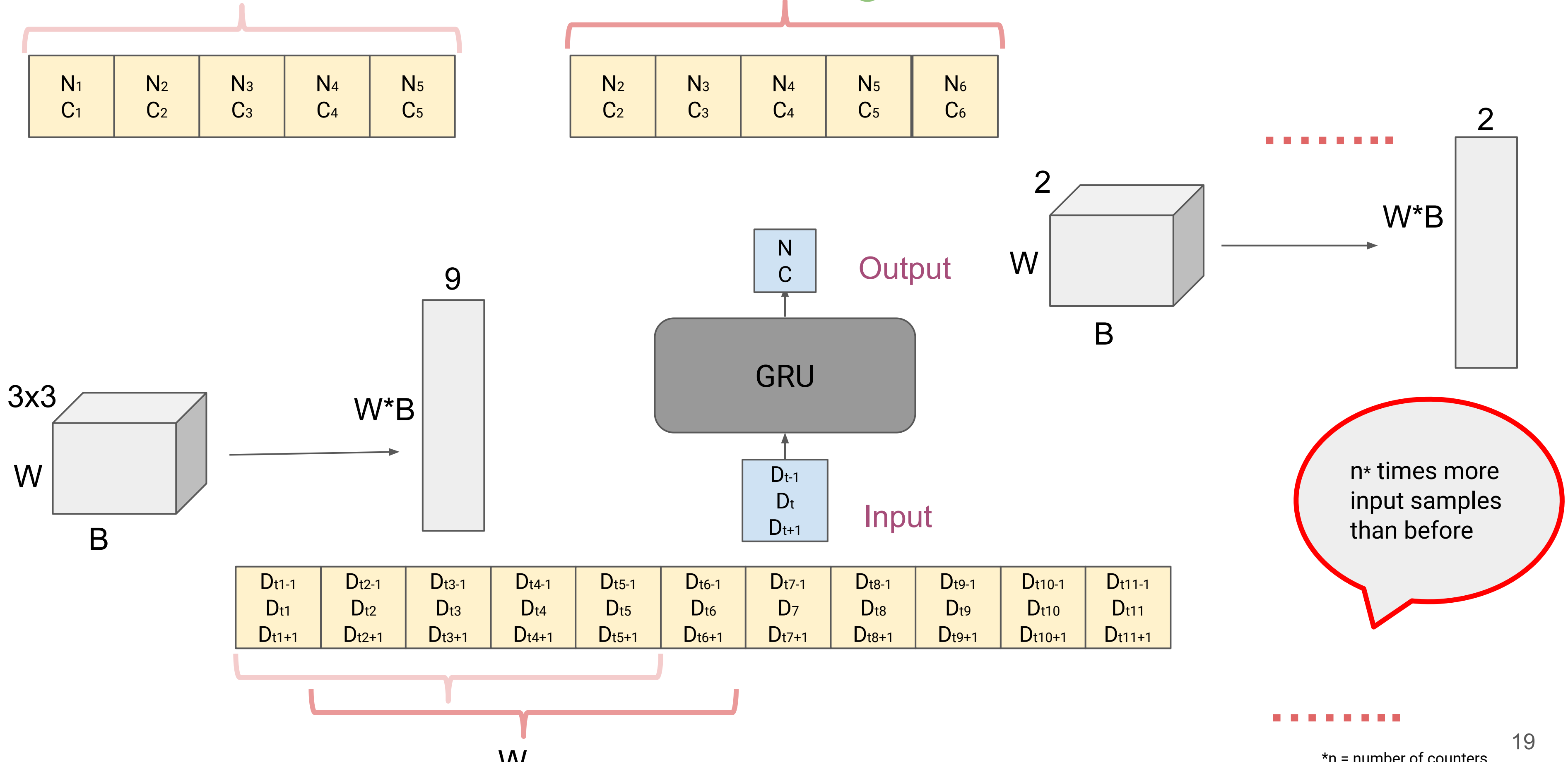
$D_c = [D, D_1, D_2]$

Same Architecture



http://gitlab.aibee.cn/jcniebles/store-events/blob/master/models.py

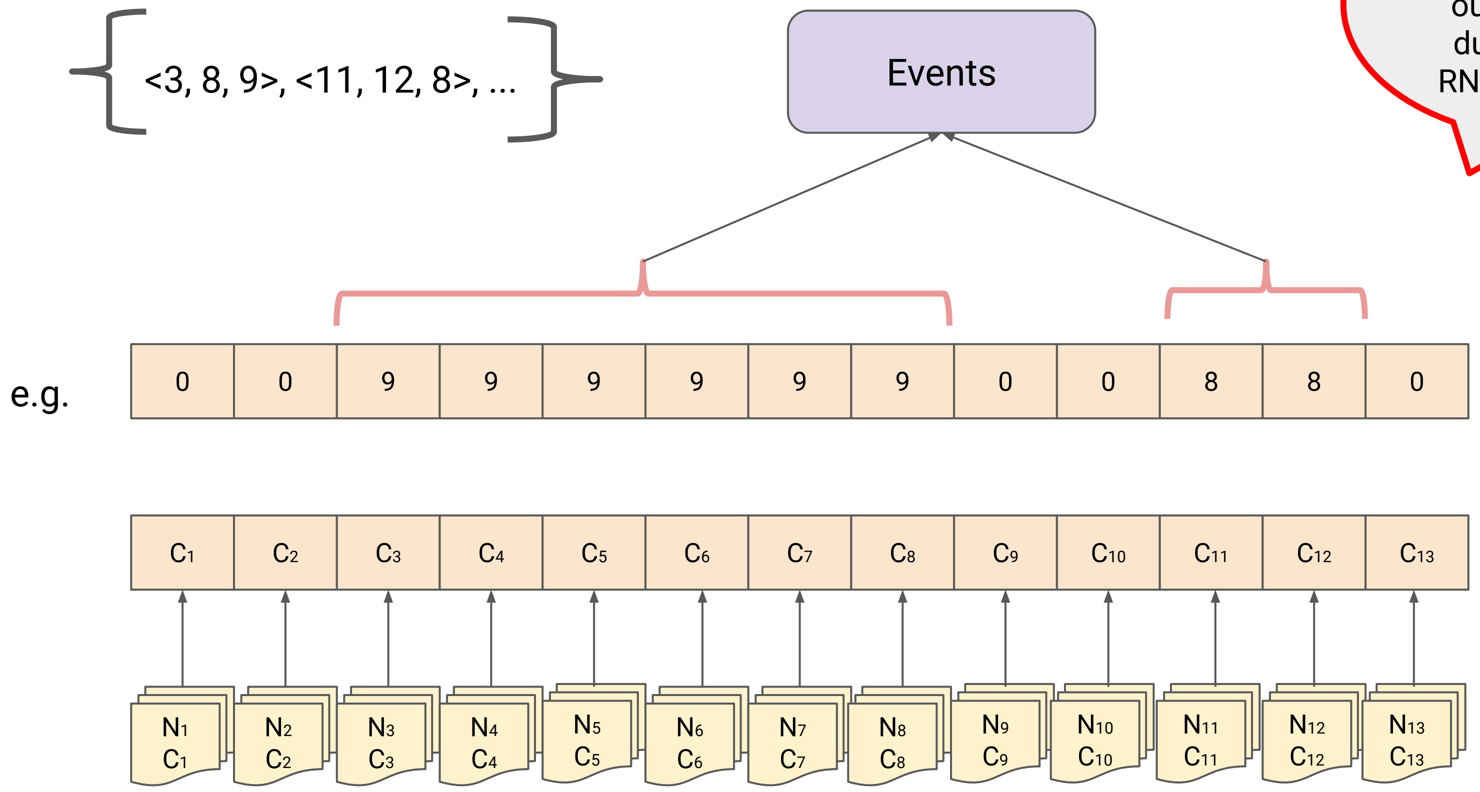
RNN Training Data



http://gitlab.aibee.cn/jcniebles/store-events/blob/master/train_GRU_dist.py

*n = number of counters

RNN Post-processing



No explicit smoothing of output required due to inherent RNN architecture

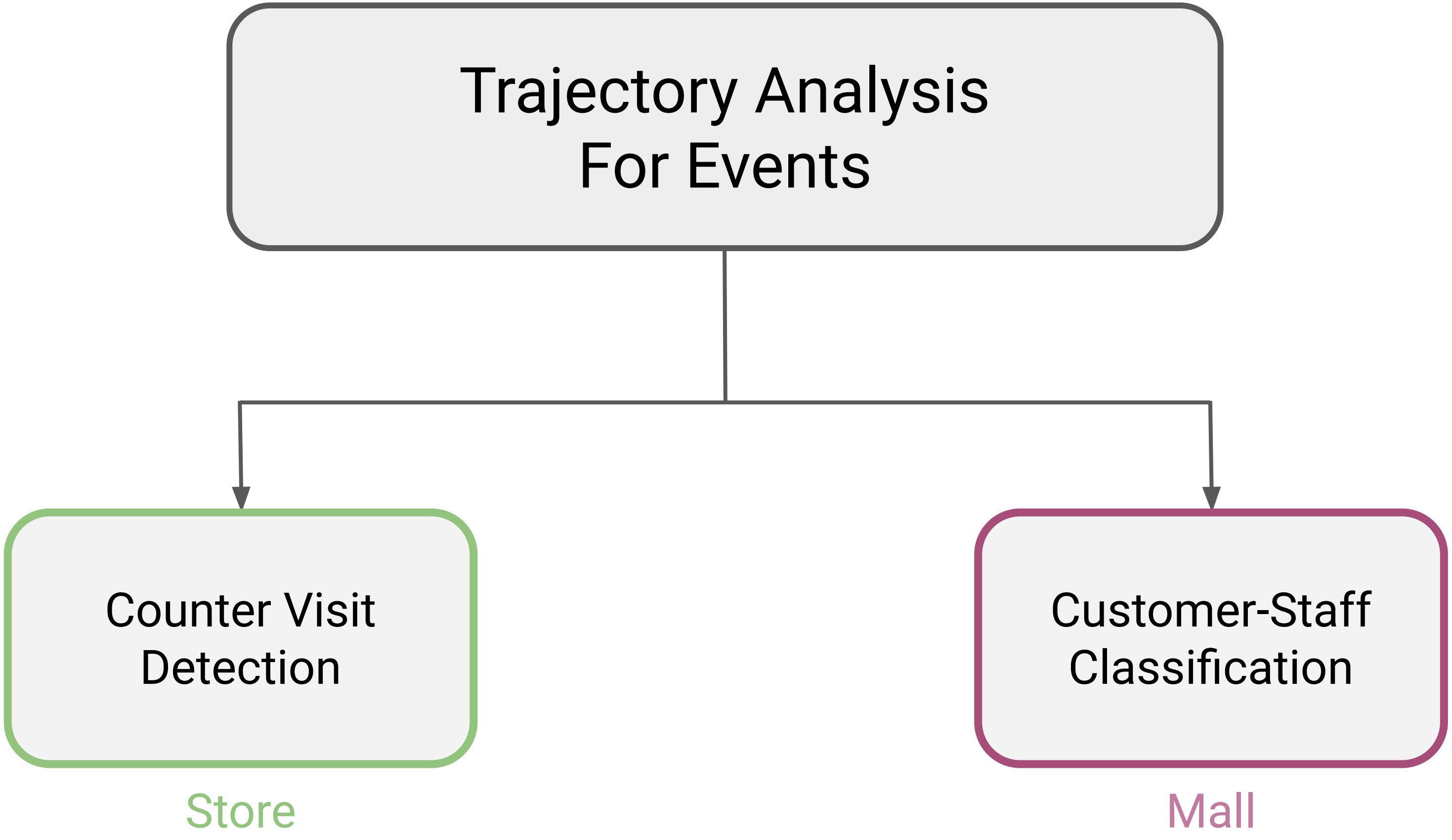
Normalize & argmax over Probabilities over all counters

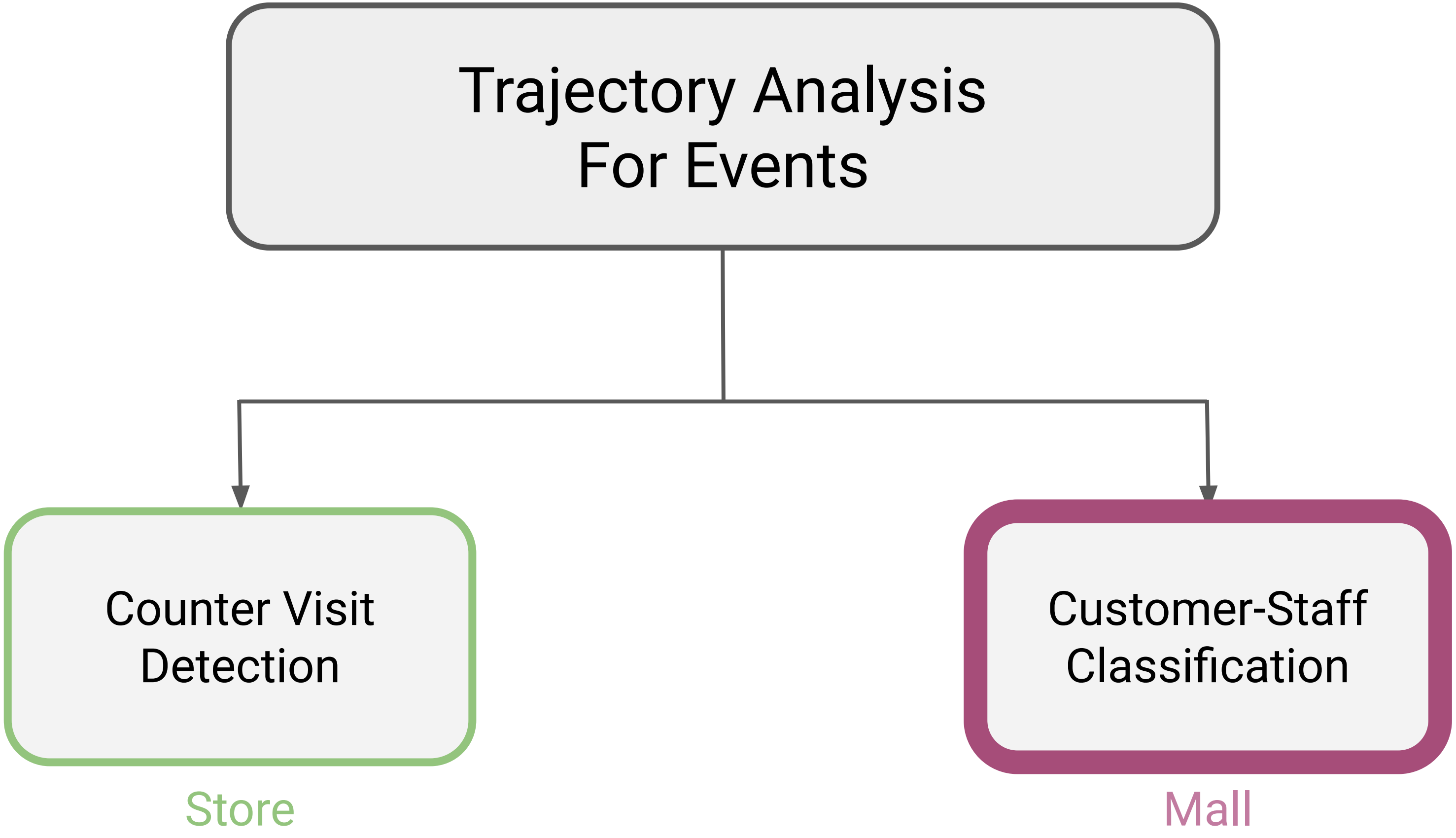
Results : xhm, wcc

Store Name	Benchmark Dates	Method	Counter Visit (tol=10s) (P/R)	Counter Visit (tol=30s) (P/R)
CTF/beijing/xhm	20181207	Rule	0.83/0.84	0.89/0.90
		GRU	0.88/0.91	0.95/0.98
	20181208	Rule	0.81/0.80	0.90/0.88
		GRU	0.82/0.88	0.89/0.94
	20190409	Rule	0.81/0.76	0.86/0.80
		GRU	0.83/0.81	0.87/0.88
CTF/beijing/wcc	20181006	Rule	0.79/0.78	0.86/0.85
		GRU (finetune)	0.93/0.96	0.96/0.97

Results : dfxtd, lnxsj

Store Name	Benchmark Dates	Method	Counter Visit (tol=10s) (P/R)	Counter Visit (tol=30s) (P/R)
CTF/beijing/dfxtd	20190409	Rule	0.86/0.86	0.91/0.90
		GRU (scratch)	0.71/0.73	0.78/0.81
		GRU (finetune)	0.73/0.66	0.80/0.74
		GRU (finetune + freeze)	0.79/0.78	0.87/0.86
CTF/guangzhou/lnxsj	20190409	Rule	0.76/0.79	0.85/0.89
		GRU (scratch)	0.65/0.53	0.74/0.63
		GRU (finetune)	0.76/0.68	0.82/0.74
		GRU (finetune + freeze)	0.82/0.71	0.86/0.76


Trajectory Analysis
For Events
Counter Visit
Detection
Store
Customer-Staff
Classification
Mall



Trajectory Analysis
For Events
Counter Visit
Detection
Customer-Staff
Classification
Store
Mall

Trajectory-based Intention Classification

Trajectory-based Intention Classification

"Classification of people based on intentions."

Understand Intentions

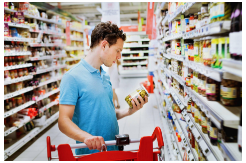
- Understanding clients
- Personalization

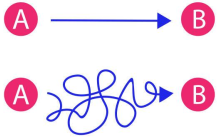
Learn Trends

- Mall business recommendations
- Research ideas

Motivation

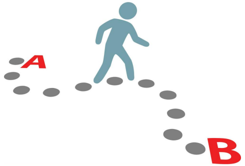
Strong intention vs browsing





Specific category shopper

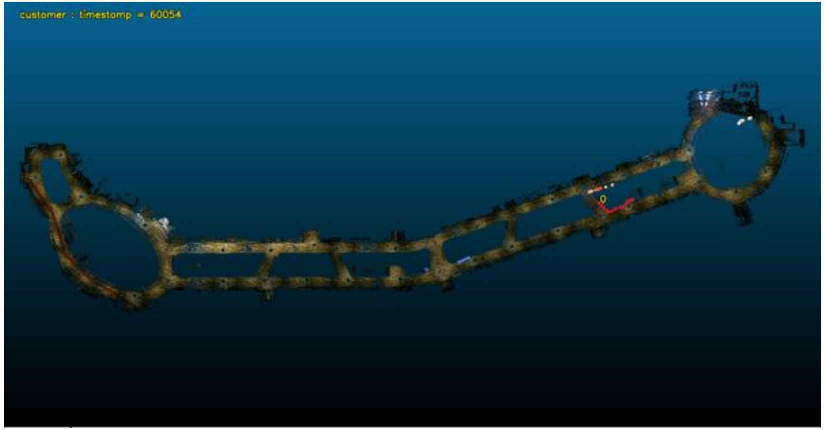
Intention based on previous store visits



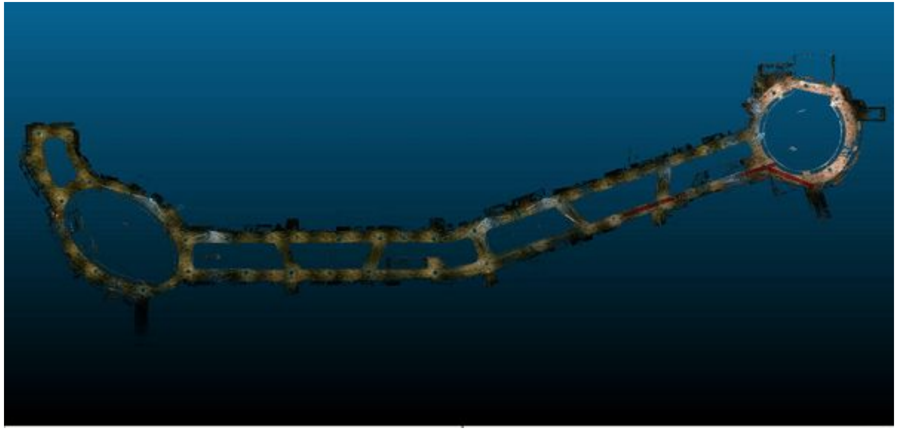
Customer vs Staff classification

Customer-Staff Classification : **Task**

"Given the mall and the person's xy-trajectory, classify whether this person is a customer or staff member"



Customer



Staff

Methods

Rule-based

- Adaboost
- Features (weak classifiers)
    - Hole count & percent
    - Hover count & percent
    - Retrace count & percent
    - Hole frequency
    - Hole duration
    - Total duration

RNN

- LSTM
- Features
    - Shifted (x,y)
    - Shifted t

Adaboost Features

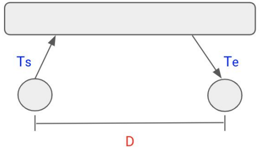
Total duration

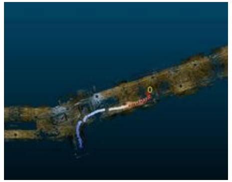
Holes

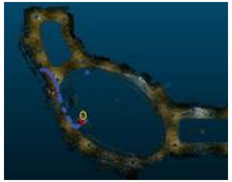
Hovering

Retracing

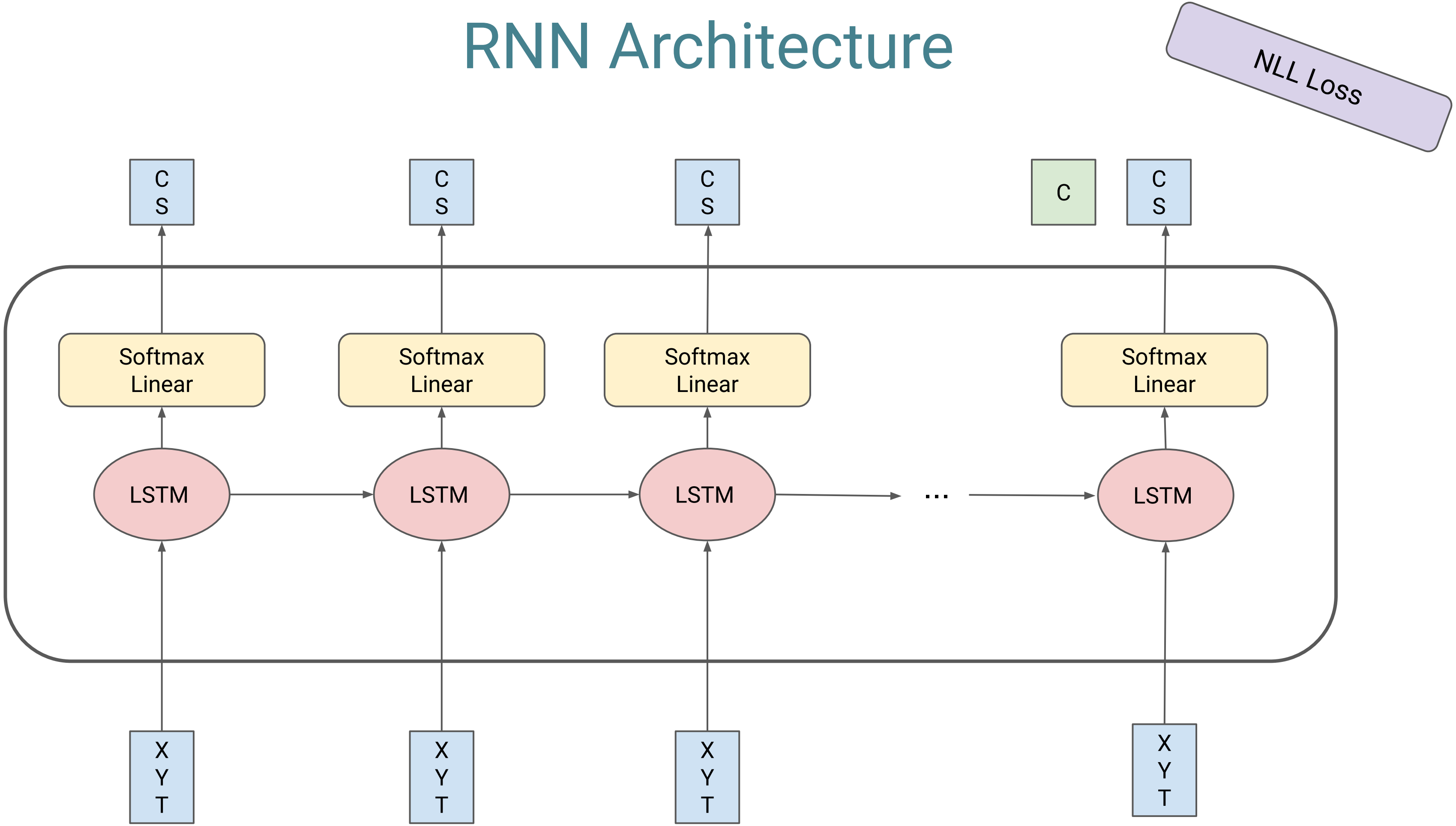
Te - Ts





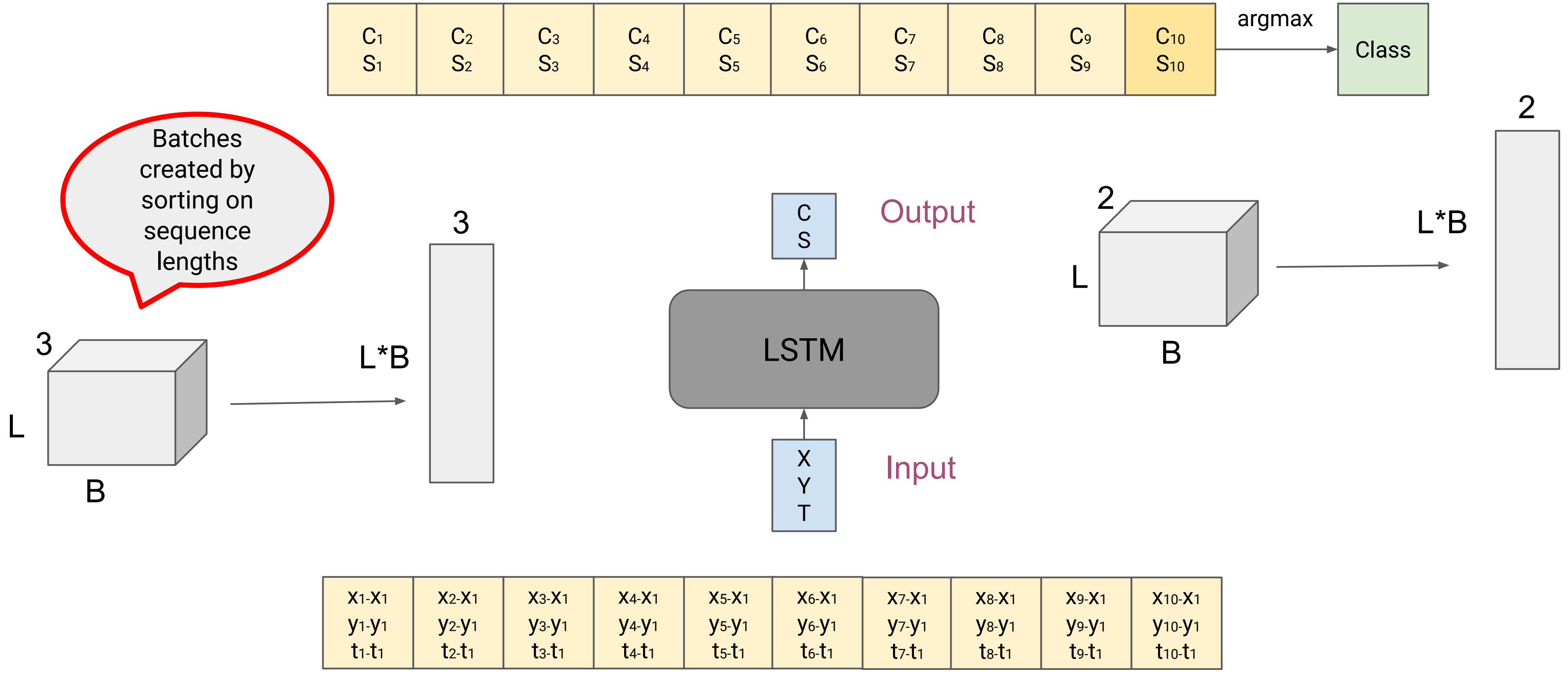


RNN Architecture



http://gitlab.aibee.cn/ptendulkar/customer-staff-classification/blob/master/code/lstm-code/models.py

RNN Training & Inference



L = length of longest sequence in batch

http://gitlab.aibee.cn/ptendulkar/customer-staff-classification/blob/master/code/lstm-code/train.py

Results

Methods	Metrics	Train	Val	Test
AdaBoost	P/R	0.79/0.67	0.88/0.76	0.68/0.41
	F	0.73	0.82	0.51
	Avg Accuracy	0.83	0.87	0.70
	All Accuracy	0.95	0.97	0.97
LSTM-2	P/R	0.80/0.65	0.65/0.61	0.50/0.34
	F	0.72	0.63	0.40
	Avg Accuracy	0.82	0.79	0.66
	All Accuracy	0.97	0.93	0.97

Issues & Next Steps

- Issues
  - Numbers not reliable; too much variation
  - LSTM may be too hard to train from such less data
  - Error cases not showing any trends
- Next Steps
  - Add some better features (e.g. visual)
  - Add more training data
  - Improve model or use model needing less data

Other Methods Tried (and not working well ..)

- Different Architecture
  - Classifier on final frame of visualization
  - GRU-based
    - Making a customer/staff prediction at every timestep
- Loss Function
  - Exponential loss
- Features
  - Shifted x, y : $(X_t - X_0)$
  - Velocity : $(X_t - X_{t-1})$
  - Smoothened velocity : $(X_t - X_c)$; c is a constant
  - Hovering : min $(X_t - X_{t-k})$; k E K

Intention Classification : **Next Directions**

- Explore trajectories, and build on current setup
- Add visual information
- Use semantic information of store types
- Use layout of mall for mall-specific recommendations
- Understand intentions
    - Strong intention vs browsing
    - Specific category shopper
    - Intention based on previous store visits
    - Understanding preferences
    - Understanding typical behaviors of customers in different stores

Summary

Thank You!

"Set-up model architectures for handling trajectories, and set some baselines to explore interesting events."

Code

http://gitlab.aibee.cn/jcniebles/store-events

http://gitlab.aibee.cn/ptendulkar/customer-staff-classification

Wiki

http://wiki.aibee.cn/pages/viewpage.action?spaceKey=EV&title=Trajectory-based+Event+Detection

http://wiki.aibee.cn/pages/viewpage.action?pageId=16352152