PopNet: Real-Time Population-Level Disease Prediction with Data Latency

Abhibha

Problem Definition

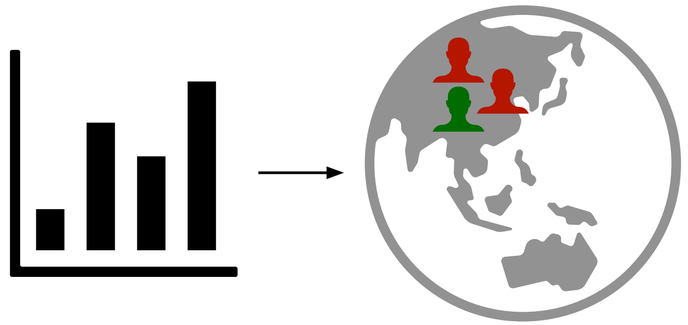
- Population - level disease prediction estimates the number of potential patients of particular diseases in some location at a future time based on (frequently updated) historical disease statistics.

- Problem: Data collection is often time consuming and has time delays with both historical and current disease statistics updated continuously.

- Solution: PopNet (population level disease prediction model)

  - Captures data latency

  - Incorporates updated data for improved predictions.

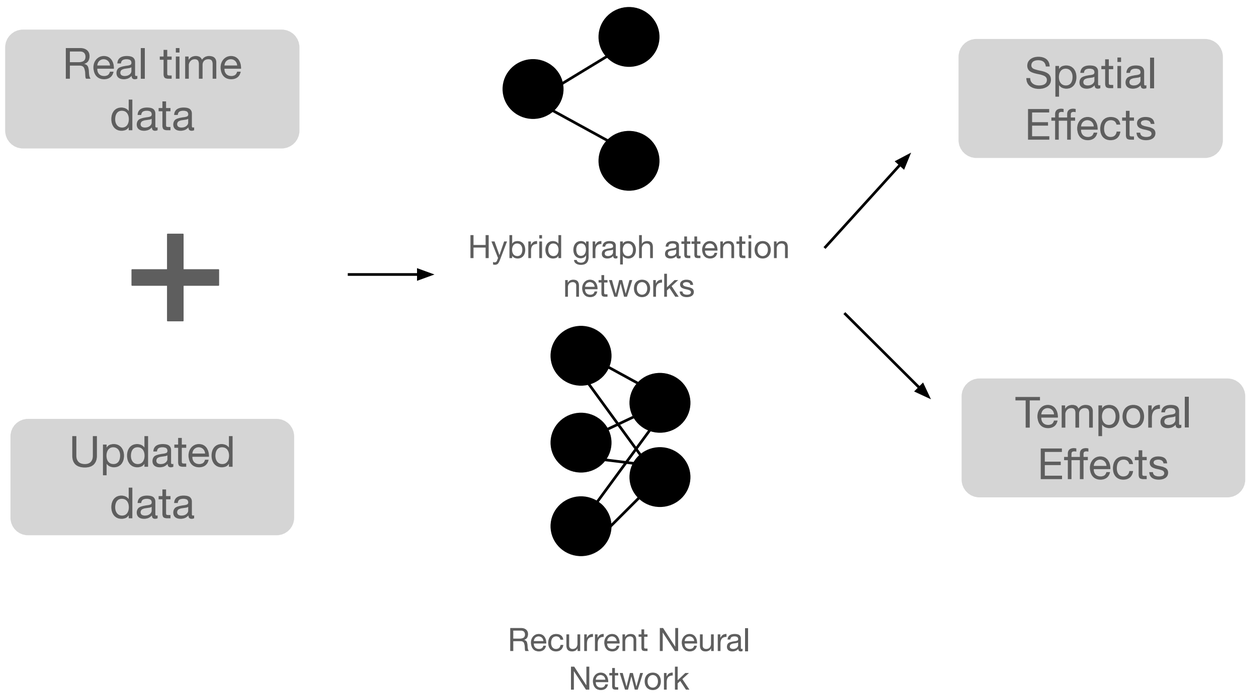


Related Work

- Spatio temporal models have been developed for different applications like traffic prediction, disease prediction, regional demand prediction and general time series prediction
- Deng and fellow researchers proposed a location attention mechanism and a graph message passing framework to predict influenza-like illness for different locations.
- Consider broader spatial-temporal prediction models in other fields such as traffic prediction, most works also utilize graph neural networks to extract spatial features and use RNNs or attention mechanisms to extract temporal features but those works also do not consider data latency.
- The traffic prediction model GMAN leverages the node2vec approach to preserve graph structure in node embeddings and then samples the neighboring nodes to obtain the embedding.

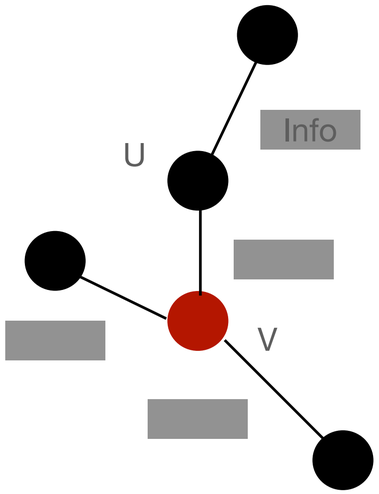
PopNet

- Models real time data and updated data using Graph attention networks (GAT) and Recurrent neural networks (RNN)
- Captures Spatial and Temporal Information from the data.
- Data updating patterns
  - Spatial Correlation: Geographically close locations have similar data updating patterns.
  - Seasonality: Data updating patterns are periodic
  - Disease correlation: Disease comorbidities may lead to similar updating patterns.



Graph Attention Networks

- Find target representation of red node $h_{N(V)}$
- Aggregate information from neighboring nodes -> capture features
- Source node: U and target Node: V



Attention function (Message importance). Calculated Embeddings of source and target node

Probabilities sum to 1

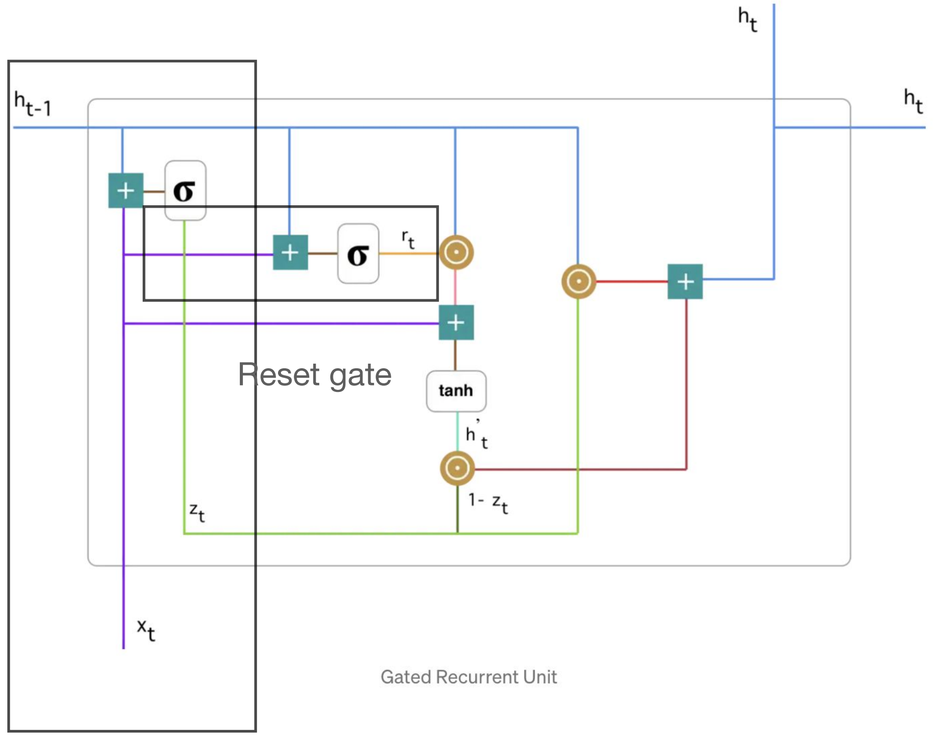
Final Equation:

Attention score

Learnable linear Projection matrix (Embeddings)

GRU

- RNN's suffer from vanishing gradient problem. GRU's help tackle that.
- GRU's use the update and reset gate. These vectors decide which information should be passed to the output.
- They can be trained to keep information from long ago, without it being lost through time or remove information which is irrelevant to the prediction.
- Update gate: Helps the model to determine how much of the past information needs to be passed along to the future.
- Reset gate: This gate is used from the model to decide how much of the past information to forget.



Update gate

H_{t-1} = previous hidden state

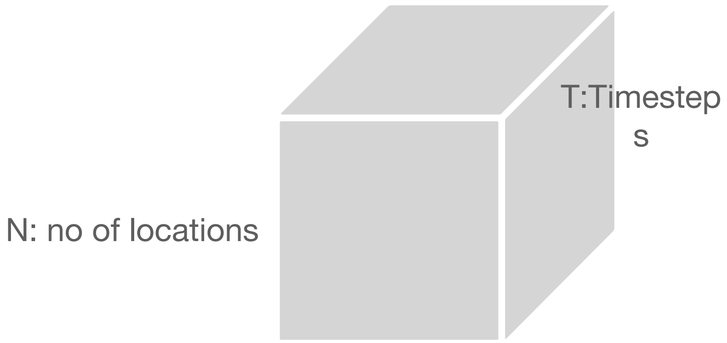
H_t= new hidden state

x_t= input at timestep t

Ref: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

Problem Formulation

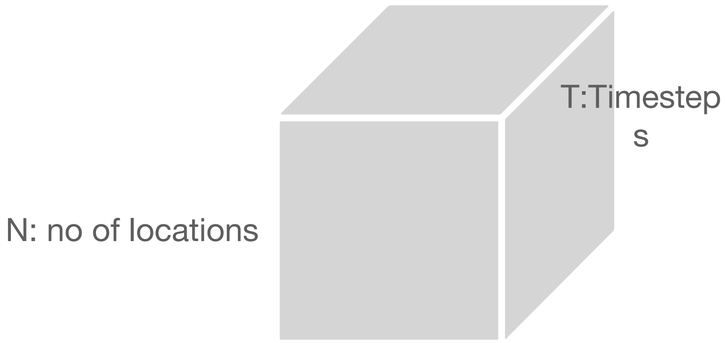
Disease statistics data X
Realtime data



3-D
Tensor

X_t = R (N x F) - time dimension
X_i = R (T x F) - location dimension
X_i^t = F - slice from X_t at i-th location

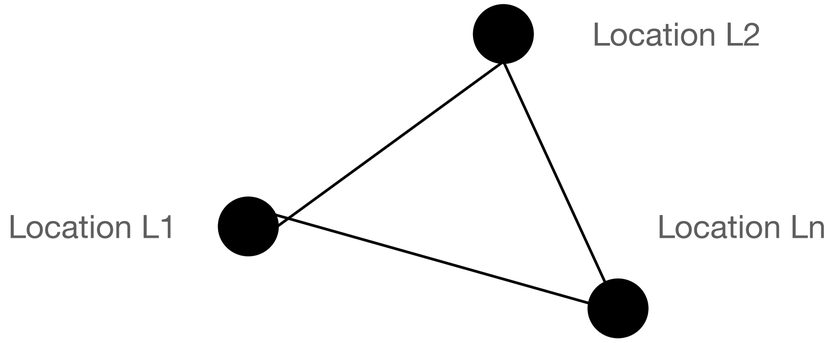
Disease statistics data U
Updated data



3-D
Tensor

U_t = R (N x F) - time dimension
U_i = (T x F) - location dimension
u_i^t = F -data updated for location I at time t

Location graph



Undirected graph

G = (V, E, A)
|V| = N location nodes
E = set of edges
A - adjacency matrix of graph

Dual Graph Attention Network

- Aim: Given original disease statistic X and updated disease data U, the task can be formulated as a regression task, which is to predict the future ground truth number of cases for a certain disease for all N locations at T + 1 timestep.

- For a time step T,

$$\mathbf{z}_i = \mathbf{W}_z \mathbf{x}_i,\ \mathbf{z}_i^u = \mathbf{W}_z^u \mathbf{u}_i$$

$$e_{ij} = \sigma(\mathbf{W}_a(\mathbf{z}_i | \mathbf{z}_j)),\ e_{ij}^u = \sigma(\mathbf{W}_a^u(\mathbf{z}_i^u | \mathbf{z}_j^u))$$

$W_z\ x_i$ = attention weight matrices
$W_z\ x_i$ = For realtime data
$W^u_z\ x_i$ = For updated data

e_{ij} = Attention score from node i to j
W_a = Attention weight matrices
| -> Concatenate operator (concatenating weights)

Applying softmax

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})},\quad a_{ij}^u = \frac{\exp(e_{ij}^u)}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^u)}$$

a_{ij} = Normalized attention score
N(i) = Set of one hop neighbors of node I

Calculating K independent attention scores

$$\mathbf{g}_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \mathcal{N}(i)} a_{ij}^k \mathbf{W}_g^k \mathbf{x}_j\right), \mathbf{g}_i^u = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \mathcal{N}(i)} a_{ij}^{u,k} \mathbf{W}_g^{u,k} \mathbf{u}_j\right)$$

W^k_g = Weight matrix for n^{th} attention head

For each node we obtain the node embeddings g_i (real time data X) and g^u_i (updated data U)

Capturing Spatial Latency

- Latency in data updating can vary between two nodes. Hence can't directly concatenate embeddings
- Spacial latency aware attention (S-Latt) to fuse spatial embeddings
- Idea: Aggregate spatial patterns from nearby nodes assuming they have similar data updating patterns.
- Learn spatial information embedding comprising of populations, #hospitals, ICU beds, longitude, etc.

SIE for node I

$$\mathbf{v}_i^s = MLP(\mathbf{S}_i)$$

V_i = spatial information embedding (SIE) for node I

S_i = Spatial info for node I

Attention score

$$e_{ij} = \sigma(\mathbf{W}_a(\mathbf{W}_g(\mathbf{g}_i|\mathbf{v}_i^s) + \mathbf{W}_u(\mathbf{g}_j^u|\mathbf{v}_j^s)))$$

J -> *N(I)* where *N(i)* neighboring nodes of node I

Time latency to normalize Attention score.

$$f(\Delta t_{ij}) = \frac{1}{\log(1 + \exp(\Delta t_{ij}))}.$$

Δ tij = temporal latency between node I and j

New attention weight a_{ij}

$$\hat{e}_{ij} = e_{ij} f(\Delta t_{ij}), \quad a_{ij} = \frac{\exp(\hat{e}_{ij})}{\sum_{k \in \mathcal{N}_u(i)} \exp(\hat{e}_{ik})},$$

Regularize attention score

Aggregated embedding g^u_i

$$\hat{\mathbf{g}}_i^u = \sigma\Big(\sum_{j \in \mathcal{N}_u(i)} a_{ij} \mathbf{g}_j^u\Big)$$

Concatenate node embedding gi
Aggregated updated embedding g^u_i and x_i

$$\hat{\mathbf{g}}_i = (\mathbf{g}_i|\hat{\mathbf{g}}_i^u|\mathbf{x}_i)$$

Capturing Temporal Latency

- Design temporal latency aware attention mechanism (T-Latt) to fuse two embeddings and deal with the latency between the real time an updated embeddings
- Enrich T-Latt with temporal information embedding (TIE) (c_t)
- Uses dilated convolution networks to extract temporal patterns from different time scales
- The larger the dilation (ϕ), Broader the time scale from which temporal patterns are extracted.

For location l

$$\mathbf{h}_t = GRU(\hat{\mathbf{g}}_1, \hat{\mathbf{g}}_2, ..., \hat{\mathbf{g}}_t)$$
$$\mathbf{h}_t^u = GRU_u(\mathbf{g}_1^u, \mathbf{g}_2^u, ..., \mathbf{g}_t^u)$$

Real time embeddings and Updated embeddings fed into GRU

Combination of Φ used to extract patterns in different time scales.
c_t = concatenated feature map vector

$$\mathbf{c}_t^{\phi} = \mathbf{m}(L, \phi) * \mathbf{X},$$

X = input disease sequence
m(L, Φ) = 1D convolution filter
Filter size = L
Φ = dilation rate

Select most informative patterns

Calculate attention score
Sigmoid function used to generate score between 0 and 1

$$\mathbf{a}_t^c = \sigma(MLP(\mathbf{x}_m))$$

X_m = MeanPool(x) over time dimension

Score act used to recalibrate feature map vector c_t

$$\hat{\mathbf{c}}_t = \mathbf{c}_t \odot \mathbf{a}_t^c$$

Capturing Temporal Latency

Similar to before we use TIE to enrich Attention. Also regularize using Time latency

$$e_{ti} = f(\Delta t_{ti}) * \sigma(\mathbf{W}_a(\mathbf{W}_{h1}(\mathbf{h}_t|\hat{\mathbf{c}}_t) + \mathbf{W}_{h2}(\mathbf{h}_i^u|\hat{\mathbf{c}}_t^u)))$$

$$a_{ti} = \frac{\exp(e_{ti})}{\sum_{j=1}^{t} \exp(e_{tj})}$$

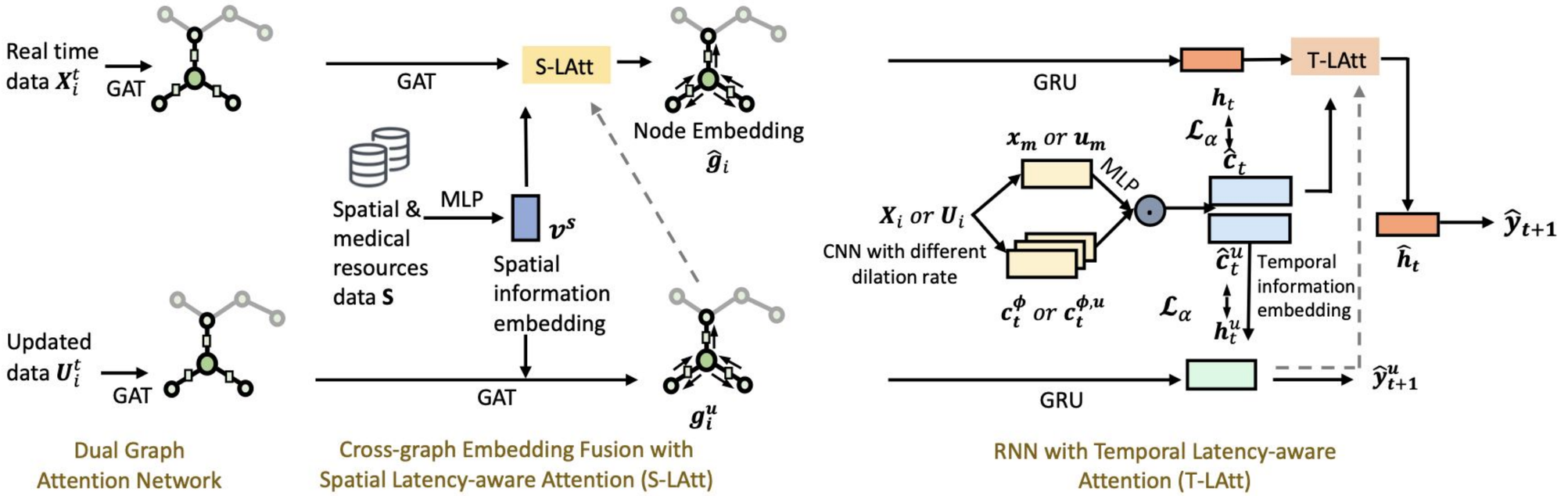
Aggregated temporal embeddings

$$\hat{\mathbf{h}}_t = (\mathbf{h}_t|\hat{\mathbf{h}}_t^u|\hat{\mathbf{c}}_t).$$

Concatenate aggregated temporal embedding h^u_t + Original temporal embedding + TIE = final temporal embedding

$$\hat{\mathbf{h}}_t^u = \sum_{i=1}^{t} a_{ti}\mathbf{h}_i^u.$$

Workflow


Real time data X_i^t
GAT
Updated data U_i^t
GAT
Dual Graph Attention Network
GAT
S-LAtt
Node Embedding $\hat{g}_i$
Spatial & medical resources data S
MLP
v^s
Spatial information embedding
GAT
g_i^u
Cross-graph Embedding Fusion with Spatial Latency-aware Attention (S-LAtt)
GRU
T-LAtt
h_t
$\mathcal{L}_\alpha$
x_m or u_m
MLP
$\hat{c}_t$
X_i or U_i
CNN with different dilation rate
c_t^ϕ or $c_t^{\phi,u}$
$\hat{c}_t^u$
Temporal information embedding
$\mathcal{L}_\alpha$
h_t^u
$\hat{h}_t$
$\hat{y}_{t+1}$
GRU
$\hat{y}_{t+1}^u$
RNN with Temporal Latency-aware Attention (T-LAtt)

Loss Function

- For model updating with new data, retraining is time consuming and directly fine tuning on new data discards historical patterns

- Solution: Alignment module that initializes hidden states in a better way. We use convolutional features as they are not sequence dependent.

- This is achieved by learning a mapping function between the TIE c_t, c^u_t and the hidden states of RNN h_t, h^u_t at each timestep respectively.

- Alignment loss is defined using the Kullback - Leibler Divergence. It measures how different two probability distributions are.

- The goal is to learn a close estimation of h_t using c_t.

Map c_t to a latent space using Mapping function

$$\hat{\boldsymbol{c}}_t^m = m_\theta(\hat{\boldsymbol{c}}_t)$$

M_θ= mapping function

Calculate probability distribution

$$p(\hat{\boldsymbol{c}}_t^m) = softmax(\hat{\boldsymbol{c}}_t^m); \quad q(\boldsymbol{h}_t) = softmax(\boldsymbol{h}_t)$$

Alignment loss function Using the Kullback Leibler Divergence

$$\mathcal{L}_a = \sum_{i=1}^{n} p(\hat{\boldsymbol{c}}_t^m) \log\left(\frac{p(\hat{\boldsymbol{c}}_t^m)}{q(\boldsymbol{h}_t)}\right)$$

$$\mathcal{L}_r = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_{i+1} - y_{i+1})^2; \quad \mathcal{L}_u = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}^u_{i+1} - y^u_{i+1})^2$$

Squared Loss for realtime and Updated data

Final loss function

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_u + \mathcal{L}_a$$

Training Algorithm

Algorithm 1 The PopNet model

Input:

Real-time disease statistics $\mathcal{X}$, updated disease statistics $\mathcal{U}$, update intervals $\Delta \boldsymbol{t} = [\Delta \boldsymbol{t}_1, \Delta \boldsymbol{t}_2, ..., \Delta \boldsymbol{t}_T]$, prediction targets $\boldsymbol{Y} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, ..., \boldsymbol{Y}_T]$ and location graph $\mathcal{G}$.

Training:

for $i = 1$ to T **do**

Input $\mathcal{X}$ and $\mathcal{U}$ and get node embeddings using Eq. 4;

Aggregate $\hat{\mathbf{g}}^{\boldsymbol{u}}$ to $\hat{\mathbf{g}}$ using Eq. 9 and 10; → Incorporate spatial latency

Input spatial embeddings to GRU networks using Eq. 11; → Calculate h_t and h^u_t

Calculate temporal information embeddings using Eq. 14; → Calculate c_t

Aggregate $\mathbf{h}_i^{\boldsymbol{u}}$ to $\mathbf{h}_i$ using Eq. 16 and 17; → Incorporate temporal latency

Make predictions for $i + 1$ timestep;

end for

Generate distributions using Eq. 18 and 19; → Generate c_t and h_t probability distributions using softmax

Calculate KL divergence using Eq. 20;

Optimize model parameters by minimizing loss in Eq. 22.

Iterative training:

Calculate TIE for the input sequence using Eq. 14;

Use learned m_θ to generate the initial hidden state of GRU;

Repeat the normal training steps.

Experimental Setup

Dataset

- Experimental Disease dataset: Respiratory diseases (ICD10 codes J00-J99) are common and most of them are contagious. The number of cases is larger than most other diseases. After filtering out locations that have less than 100 cases, Finally the authors get 1,693 counties for respiratory diseases prediction.

- Tumors dataset: Compared to respiratory diseases, the tumors (ICD10 codes C00-D49) have fewer cases, and the data update period is also shorter. After filtering out locations that have less than 10 cases, Finally the authors get 1,829 counties for respiratory diseases prediction.

ICD code	Category description	Avg. cases
A00-B99	Certain infectious and parasitic diseases	314.9
B00-D49	Tumors	966.1
D50-D89	Diseases of the blood, blood-forming organs and immune mechanism	344.8
E00-E89	Endocrine, nutritional and metabolic diseases	1508.8
F01-F99	Mental, Behavioral and mental disorders	1470.1
G00-G99	Diseases of the nervous system	744.2
H00-H59	Diseases of the eye and adnexa	385.8
H60-H95	Diseases of the ear and mastoid process	192.2
I00-I99	Diseases of the circulatory system	1717.2
J00-J99	Diseases of the respiratory system	1774.7
K00-K95	Diseases of the digestive system	653.1
L00-L99	Diseases of the skin and subcutaneous tissue	496.9
M00-M99	Diseases of the musculoskeletal system and connective tissue	2226.7
N00-N99	Diseases of the genitourinary system	897.0
O00-O99	Pregnancy, childbirth and the puerperium	152.6
P00-P96	Certain conditions originating in the perinatal period	45.9
Q00-Q99	Congenital malformations, deformations and chromosomal abnormalities	80.2
R00-R99	Symptoms, signs and abnormal clinical and laboratory findings	2480.2
S00-T88	Injury, poisoning and certain other consequences of external causes	683.3
V00-Y99	External causes of morbidity	12.4
Z00-Z99	Factors influencing health status and contact with health services	2653.2

Table 5: ICD codes and disease category descriptions

Baseline and Metrics

- Baselines

The authors evaluated PopNet against the following spatio-temporal prediction baselines: SARIMAX, GRU, ASTGCN, GMAN, EvolveGCN, ColaGNN and STAN. They also compare PopNet with the reduced version as the ablation study.

- **PopNet**-LAtt: Reduce both S-LAtt and T-LAtt mechanisms from PopNet.
- **PopNet**-SLAtt: Reduce the spatial latency-aware atten- tion from PopNet.
- **PopNet**-TLAtt: Reduce the temporal latency-aware at- tention from PopNet.
- **PopNet**-L$_a$ We reduce the alignment module and the loss term L$_a$ from PopNet.

- Metrics

The root mean squared error (RMSE), mean absolute error (MAE) and the mean absolute percentage error (MAPE) have been used to gauge performance of model

- Training Procedure

Training set - 60 weeks

Validation set - 20 weeks,

Testing set - 20 weeks

To be fair, the training, validation and test data for all models are the same.

Results

Performance on Disease Prediction

PopNet outperforms all baseline methods on all metrics. On respiration disease dataset, PopNet achieves 47% lower RMSE, 23.2% lower MAE, and 29.4% lower MAPE, and on tumors dataset

PopNet achieves 29% lower RMSE, 24% lower MAE, and 13.2% lower MAPE, both compared with the best baseline ColaGNN.

Tumors Prediction				
Model	**RMSE ($\times 10^5$)**	**MAE**	**MAPE**	**P-value**
SARIMAX	21.41	426.0	69.8	0.0
GRU	16.20	313.3	60.7	0.0
GMAN	13.12	315.7	53.4	7e-25
ASTGCN	15.59	332.5	56.2	0.0
EvolveGCN	8.94	269.0	50.4	5e-11
STAN	6.56	215.8	48.7	7e-9
ColaGNN	4.75	172.5	42.3	9e-5
PopNet-LAtt	8.92	283.4	51.5	4e-10
PopNet-TLAtt	3.25	142.9	38.1	3e-4
PopNet-SLAtt	4.50	165.8	40.1	5e-5
PopNet	**2.90**	**131.6**	**36.7**	-

Table 1: Disease Prediction Performance

Respiratory Diseases Prediction				
Model	**RMSE ($\times 10^5$)**	**MAE**	**MAPE**	**P-value**
SARIMAX	15.54	542.5	51.3	0.0
GRU	10.89	340.2	43.2	5e-20
GMAN	9.10	329.8	37.4	4e-15
ASTGCN	10.33	303.6	39.9	4e-13
EvolveGCN	9.85	312.4	36.9	8e-9
STAN	9.54	305.7	36.8	4e-9
ColaGNN	8.06	291.3	33.7	5e-5
PopNet-LAtt	9.82	311.6	39.2	4e-10
PopNet-TLAtt	6.32	271.3	29.9	5e-4
PopNet-SLAtt	8.85	297.5	31.3	7e-8
PopNet	**4.29**	**223.8**	**23.8**	-

Results

Performance at Different Locations

We report the number of locations that each model has the best performance. Here ‘Others’ sums up the results of SARIMAX, GMAN, GRU, ASTGCN and EvolveGCN.

It is easy to see PopNet achieves the best performance on over 90% of locations for both tasks.

For respiratory diseases, PopNet has 12.4% improved MAPE on average than the best baseline. For tumors prediction, PopNet achieves the best performance on 1645 locations, and the average MAPE improvement is 10.7%

Table 2: # of locations where each model performs the best.

Respiratory Diseases			
Model	# of Locations	% of Locations	Mean $\Delta MAPE$
STAN	42	2.5%	3.5%
ColaGNN	33	1.9%	2.7%
Others	6	0.4%	4.1%
PopNet	**1612**	**95.2%**	**12.4%**
Tumors			
Model	# of Locations	% of Locations	Mean $\Delta MAPE$
STAN	40	2.2%	8.5%
ColaGNN	122	6.7%	4.6%
Others	22	1.2%	5.5%
PopNet	**1645**	**89.9%**	**10.7%**

Results

Train on the original dataset (week 1-50), and deploy into practice (week 50-80). Then refresh the model using newly collected data during the deployment phase (week 60-80). Finally, we re - deploy the model to test the performance (week 80-100).

PopNet outperforms baselines, achieving 39% lower RMSE, 34% lower MAE, and 25.6% lower MAPE on respiratory disease prediction and 70% lower RMSE and 49% lower MAE on tumor prediction, compared with the best baseline.

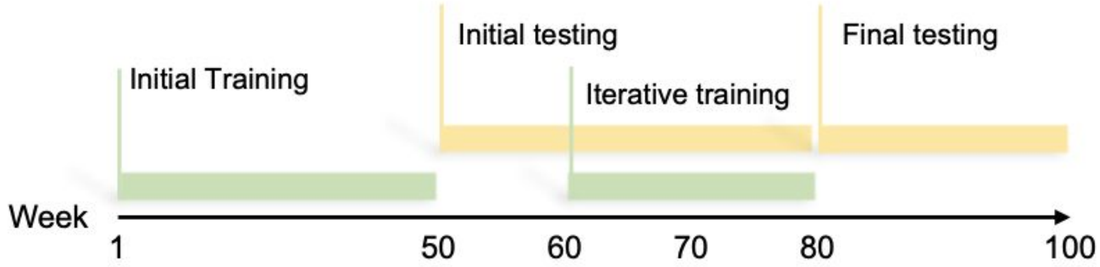


Table 3: Prediction performance with iterative traing

Respiratory diseases				
Model	**RMSE ($\times 10^5$)**	**MAE**	**MAPE**	**P-value**
GRU	14.77	495.4	52.5	0.0
GMAN	14.17	450.5	47.3	0.0
ASTGCN	12.45	432.7	46.2	3e-20
EvolveGCN	12.18	429.7	43.1	1e-18
STAN	13.90	448.5	47.0	0.0
ColaGNN	11.82	416.1	39.8	9e-17
PopNet-$\mathcal{L}_\alpha$	11.43	355.3	33.5	5e-10
PopNet	**7.23**	**275.6**	**29.6**	-
Tumors				
Model	**RMSE ($\times 10^5$)**	**MAE**	**MAPE**	**P-value**
GRU	18.95	397.6	52.9	8e-23
GMAN	18.99	401.6	53.2	0.0
ASTGCN	19.72	401.3	55.6	0.0
EvolveGCN	17.26	385.1	50.0	0.0
STAN	10.21	304.7	40.8	6e-15
ColaGNN	8.75	264.4	35.3	1e-8
PopNet-$\mathcal{L}_\alpha$	4.92	182.7	35.8	3e-4
PopNet	**2.60**	**135.4**	**34.7**	-

Results

Performance with Longer Prediction Window

Long-term prediction is also significant for disease prediction in practice. The output size is changed to make PopNet and other baseline models predict for the future 5 weeks.

The results show that the MAE rises as the prediction window increases since it becomes more difficult to predict longer future trends.

For tumors, the MAE of STAN and ASTGCN increase 11% and 24%, respectively, while PopNet only increases 6%.

For respiratory diseases, the MAE of PopNet increases 14% as the prediction window length increases from 1 to 5, while the baseline model STAN increases 24% and ColaGNN increases 31%.

Table 4: Long-term prediction (window = 5 weeks)

Respiratory Diseases				
Model	**RMSE ($\times 10^5$)**	**MAE**	**MAPE**	**P-value**
GRU	16.90	491.4	34.7	7e-21
GMAN	17.76	440.3	35.2	6e-23
ASTGCN	15.23	410.3	33.4	2e-15
STAN	9.30	380.5	31.5	4e-8
ColaGNN	9.78	384.1	31.7	8e-8
PopNet	**5.71**	**255.4**	**27.8**	-
Tumors				
Model	**RMSE ($\times 10^5$)**	**MAE**	**MAPE**	**P-value**
GRU	21.56	452.6	38.8	0.0
GMAN	19.35	420.1	35.9	5e-20
ASTGCN	19.78	425.1	36.0	6e-23
STAN	6.72	240.3	34.1	3e-9
ColaGNN	5.37	231.9	33.5	5e-7
PopNet	**2.79**	**138.4**	**31.5**	-

Conclusion and Future Work

The authors propose PopNet for real-time population-level disease prediction which considers data latency.

PopNet uses two separate systems to model real-time and updated disease statistics data, and then adaptively fuses the two systems using both spatial and temporal latency-aware cross-graph attention. The latency-aware attention is augmented with spatial and temporal information embeddings to adaptively extract and utilize geographical and temporal progression features.

The researchers conducted extensive experiments across multiple real-world claims datasets. PopNet outperforms leading spatial-temporal models in all metrics and shows the promising utility and efficacy in population-level disease prediction.

In future work, the authors aim to use a more flexible way to generate better location graphs instead of using hard defined edge weights, which is the major limitation of this work.

Thank you