Recurrent Space-time Graph Neural Network

Andrei Nicolicioiu[1], Iulia Duţă[1], Marius Leordeanu[1,2,3]

iduta@bitdefender.com, anicolicioiu@bitdefender.com, marius.leordeanu@imar.ro

[1]Bitdefender, Romania [2]University Politehnica of Bucharest, Romania [3]Institute of Mathematics of the Romanian Academy

1. Overview

Graph Neural Nets for Video Classification

Our approach:

- propose a neural graph model recurrent in space and time
- **extract features** using backbone model
- **create graph** with info from video features
- process video by message passing to get long range interactions: **Space an Time Stages**

Main Contributions:

- proposed a Graph model **recurrent and factorized**
- introduce a **synthetic dataset** involving space-time interactions
- obtain **state-of-the-art** results on Something-Something dataset

2. Graph Creation

- features from 2D / 3D **backbone** at multiple scales
- each node receives information **pooled** from a region
- the nodes are **connected** if they come from neighbouring or overlapping regions

3. Time Processing Stage

- node: current spatial info + previous time step info
- each node updates its spatial information using a **recurrent** function
- no exchange messages between nodes

$h_{i,time}^{t,k} = f_{time}(\mathbf{v}_{i,space}^{k}, \mathbf{h}_{i,time}^{t-1,k})$.

4. Space Processing Stage

- **Send**: message should represent pairwise spatial interaction

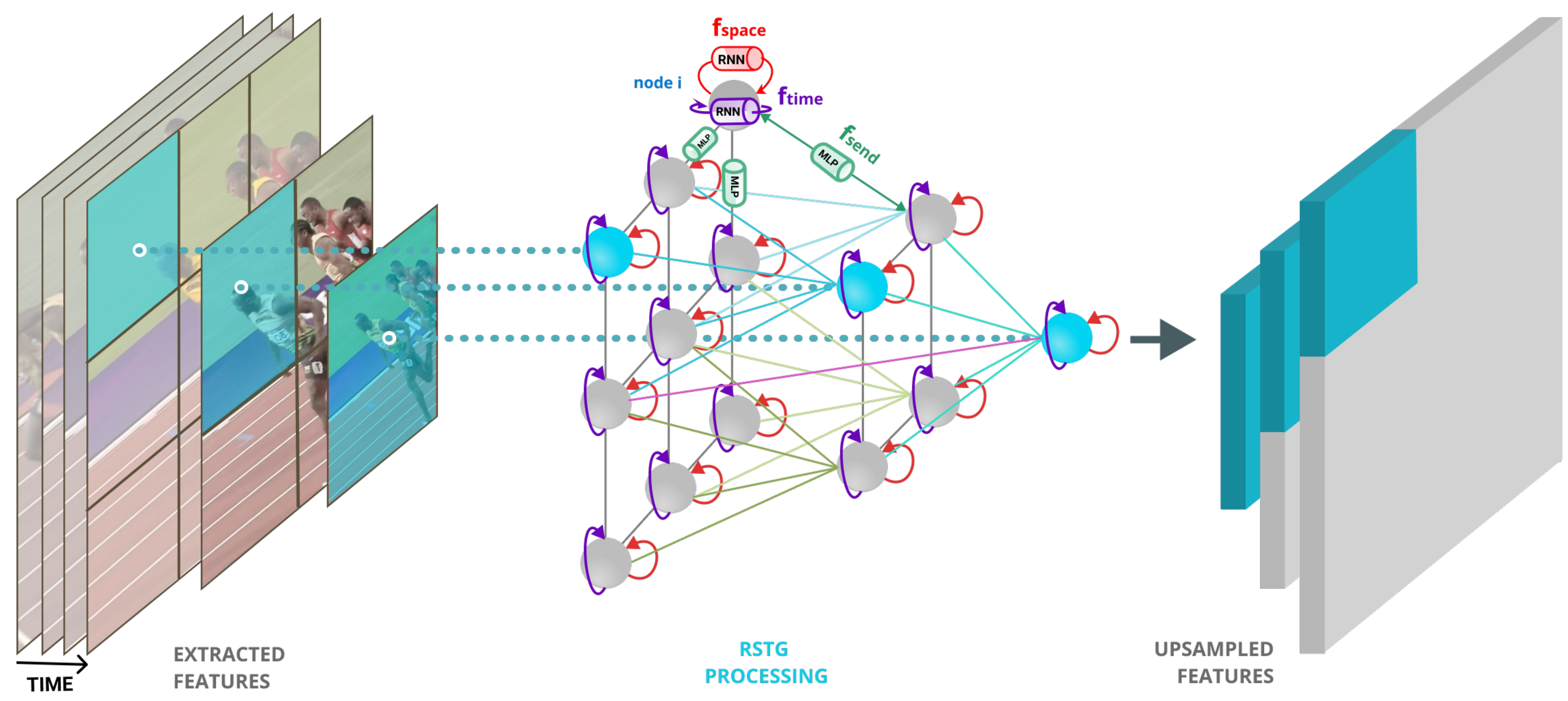
  $f_{send}(\mathbf{v}_j, \mathbf{v}_i) = MLP_s([\mathbf{v}_j|\mathbf{v}_i])$
- **Gather**: aggregate messages by an attention mechanism

  $f_{gather}(\mathbf{v}_i) = \sum_{j\in\mathcal{N}(i)} \alpha(\mathbf{v}_j, \mathbf{v}_i) f_{send}(\mathbf{v}_j, \mathbf{v}_i)$
- **Update**: incorporate global context into each local information

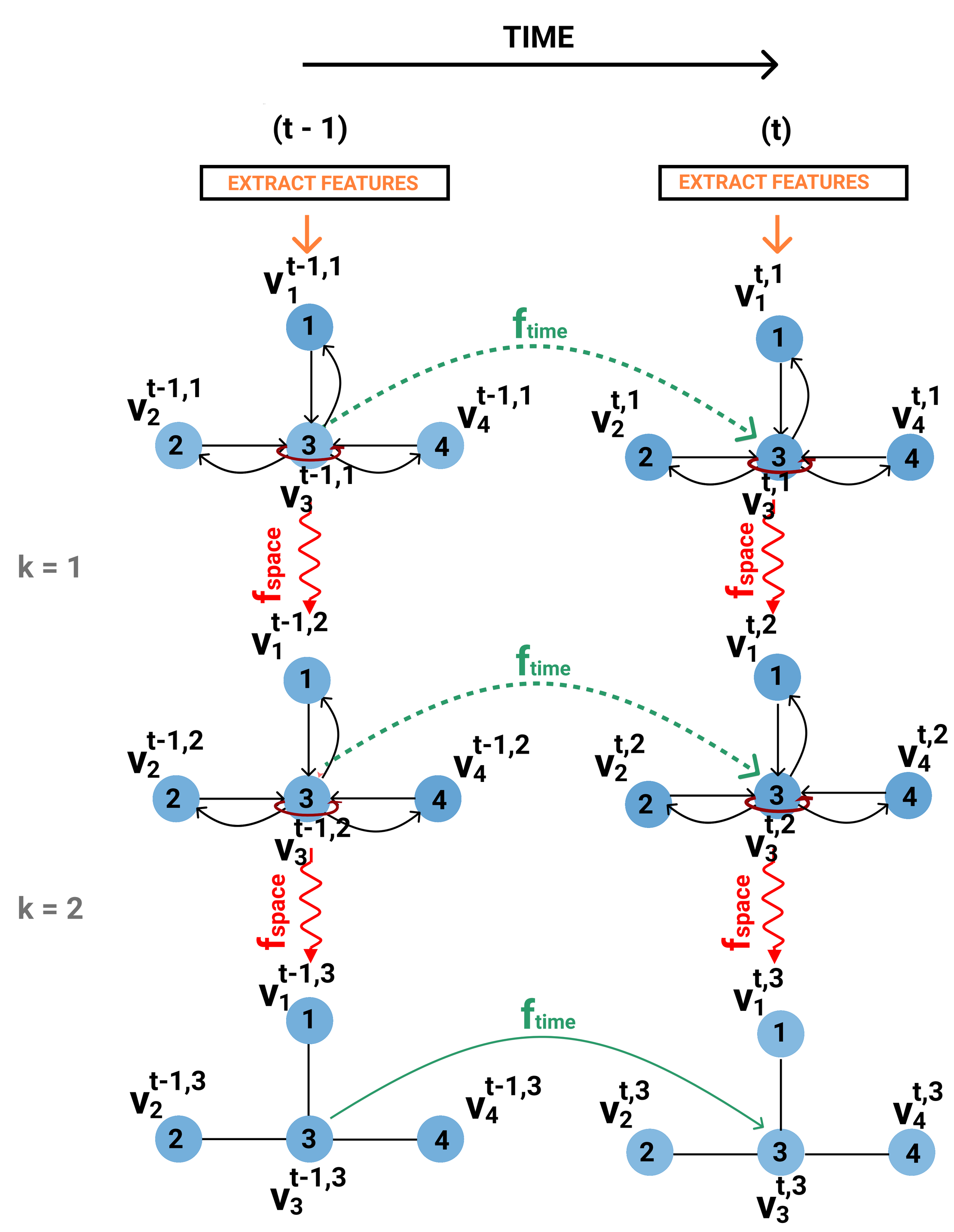
  $f_{space}(\mathbf{v}_i) = MLP_u([\mathbf{v}_i|f_{gather}(\mathbf{v}_i)])$

- applied at each time step
- **Positional Awarness**:
  - each source node should be **aware** of the destination node position
  - we concatenate the position of both nodes to the input of f_{send}
  - position is represented by gaussian centered in node's location

5. RSTG Architecture



6. Scheduler



- for more expressive power we alternate one Time Processing Stage with one Space Processing Stage
- K alternating stages + a final time stage

7. Something-Something Results

Table 1: Top-1 and Top-5 accuracy on Something-Something-v1.

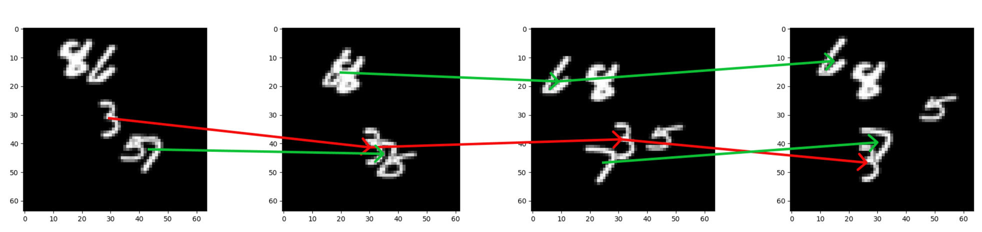
Model	Top-1	Top-5
C2D	31.7	64.7
TRN [1]	34.4	-
ours C2D + RSTG	**42.8**	**73.6**
MFNet-C50 [2]	40.3	70.9
I3D [3]	41.6	72.2
NL I3D [3]	44.4	76.0
NL I3D + Joint GCN [3]	46.1	76.8
$ECO_{Lite-16F}$[4]	42.2	-
MFNet-C101 [2]	43.9	73.1
I3D [5]	45.8	76.5
S3D-G [5]	48.2	78.7
RSTG-to-vec	47.7	77.9
RSTG-to-map res2	46.9	76.8
RSTG-to-map res3	47.7	77.8
RSTG-to-map res4	48.4	78.1
RSTG-to-map res3-4	49.2	78.8
ours I3D + RSTG	**49.2**	**78.8**

8. SyncMNIST Results

Table 2: Accuracy on SyncMNIST datasets

Model	3Sync	5Sync
Mean + LSTM	77.0	-
Conv + LSTM	95.0	39.7
I3D [6]	-	90.6
Non-Local [7]	-	93.5
RSTG: Space-Only	61.3	-
RSTG: Time-Only	89.7	-
RSTG: Homogenous	95.7	58.3
RSTG: 1-temp-stage	97.0	74.1
RSTG: All-temp-stages	**98.9**	94.5
RSTG: Positional All-temp	**-**	**97.2**

9. SyncMNIST Dataset



sync pair: (6,7)

10. References

[1] Zhou et al. ECCV 2018, [2] Lee et al. ECCV 2018,[3] Wang and Gupta ECCV 2018, [4] Zolfaghari et al. ECCV 0218, [5] Xie et al. ECCV 2018, [6] Carreira and Zisserman CVPR 2017, Wang et al. CVPR 2018