Discovering Dynamic Salient Regions for Spatio-Temporal Graph Neural Networks

Iulia Duta* [1], Andrei Nicolicioiu* [1], Marius Leordeanu [1] [2] [3]

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

Contributions

- **augment spatio-temporal GNNs** by learning to create localized nodes suited for spatial reasoning, that adapt to the input.
- the salient regions discovery **enhance relational processing** for video classification
- **DyReg-GNN discovers salient regions** that are well correlated with objects, **without object-level supervision**.

Our approach:

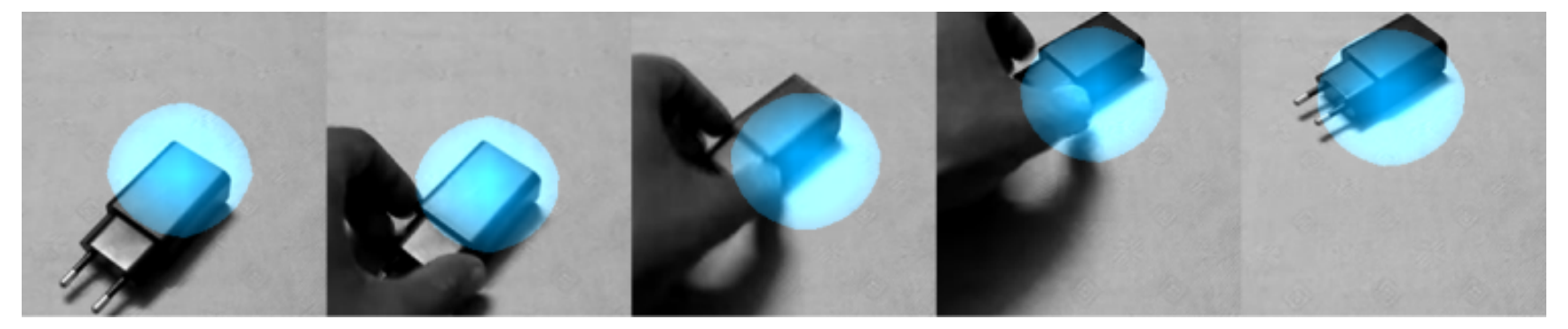
- generate location and size of N regions.
- define kernels that depend on regions parameters such that they could be learned from the high level supervision.
- create nodes by extracting features from these regions using bilinear interpolation.
- process the nodes with a spatio-temporal GNN and project each node into its initial location.

Results

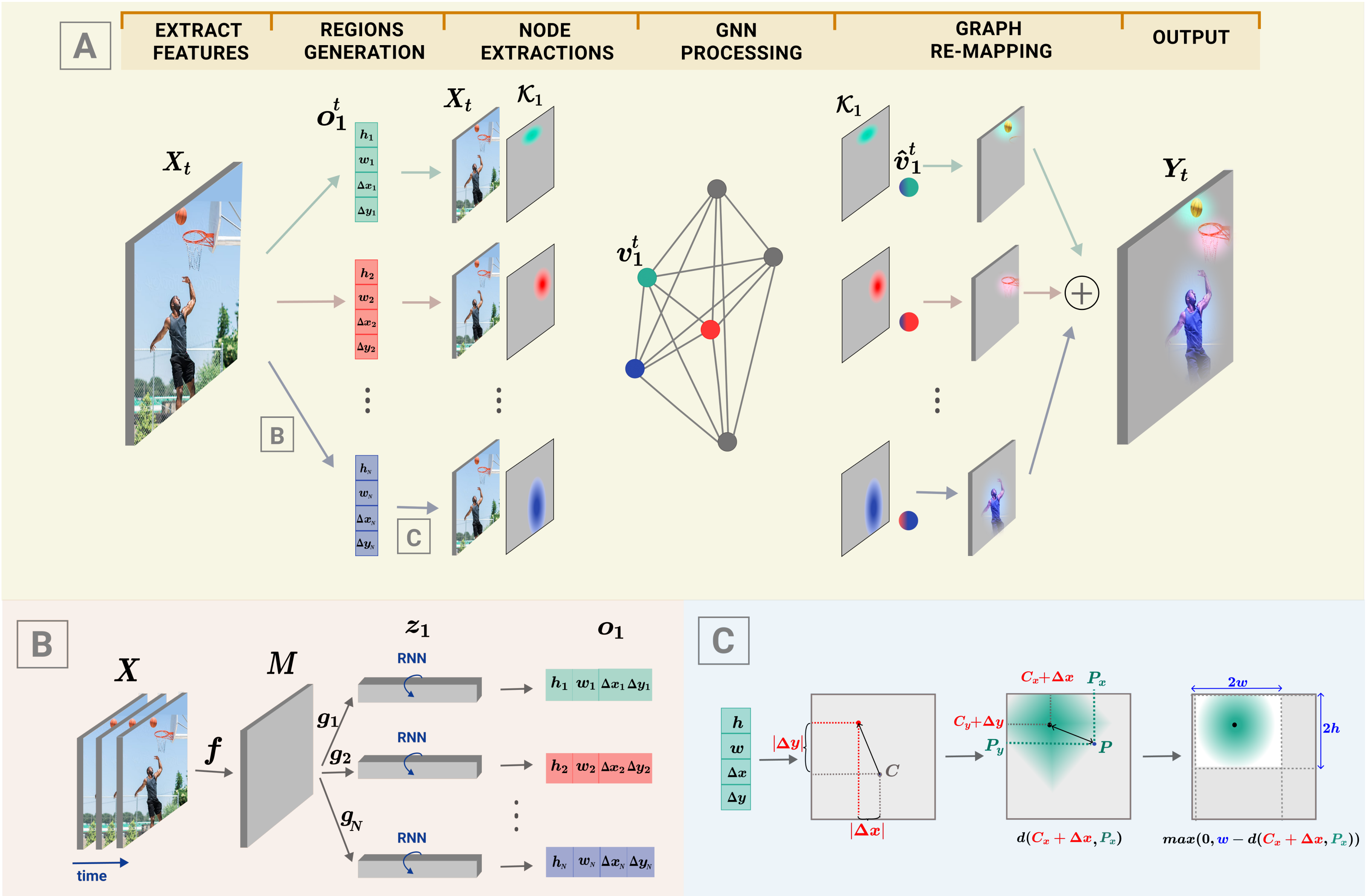
Something-Something is a real world dataset involving human-object interactions.

	Model	Top 1	Top 5
non-Graph	GST [1]	62.6	87.9
	TSM [2]	63.4	88.5
	STM [3]	64.2	89.8
	MSNet [5]	64.7	89.4
Graph	TRG [4]	59.8	87.4
	DyReG - r4	64.3	88.9
	DyReG - r3-4-5	64.8	89.4

Accuracy on Something-Something-V2 dataset.



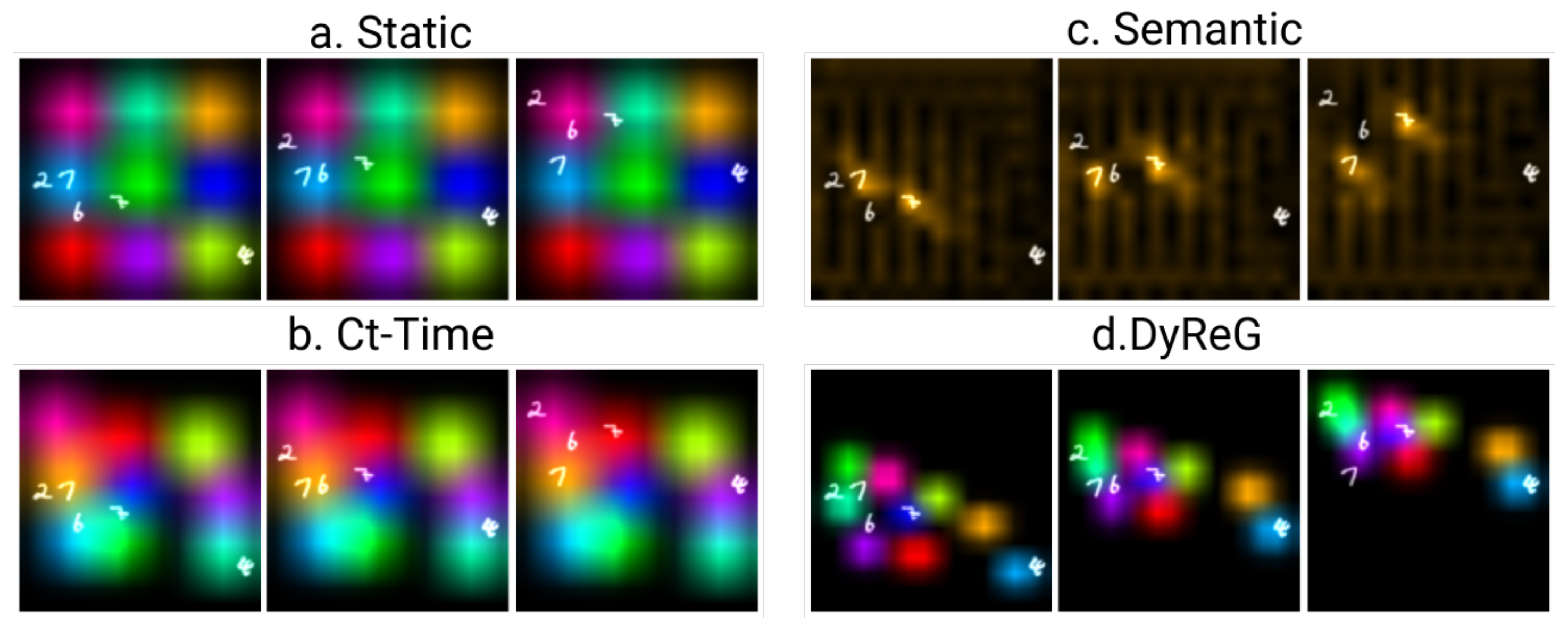
Visualisation of a single predicted kernel on Something-Something-V2.



Discover unsupervised salient regions, correlated with objects, that are useful for visual relational processing.

Model	Accuracy	Description
Static Nodes	81.48	Optimize regions across dataset
Ct-Time Nodes	86.77	Keep regions fixed in time
Semantic Nodes	82.41	Attend to all the input positions
DyReG Nodes	95.09	Full model with dynamic regions

Ablation on MultiSyncMNIST of different types of node extraction.



a. Static

c. Semantic

b. Ct-Time

d.DyReG

The goal of MultiSyncMNIST is to detect a group of digits that move synchronously.

Node Region Generation

- Produce the **most salient** $N = 9$ regions using a global processing.
  1. Each node is modeled by a function with its own set of parameters.
  2. A recurrent function is used to achieve consistency across time.
  3. Produce the **location and size** of each region.

$$M_t = f(X_t) \in \mathbb{R}^{H' \times W' \times C'}$$
$$\hat{\mathbf{m}}_{i,t} = g_i(M_t) \in \mathbb{R}^{C'}, \forall i \in \overline{1, N}$$
$$\mathbf{z}_{i,t} = \mathrm{GRU}(\mathbf{z}_{i,t-1}, \hat{\mathbf{m}}_{i,t}) \in \mathbb{R}^{C'}, \forall i \in \overline{1, N}$$
$$\mathbf{o}_{i,t} = (\Delta x_{i,t}, \Delta y_{i,t}, w_{i,t}, h_{i,t}) = \alpha(W_o \mathbf{z}_{i,t}) \in \mathbb{R}^4$$

Node Features Extraction

- Learn to generate region's parameters using the video-level supervision.
  - Make the node feature extraction **differentiable** w.r.t. region's parameters.
- Extract node features using an **interpolation** kernel.
  - The kernel decreases with the distance to the center and is non-zero up to a maximal distance of w_i.

$$\mathcal{K}^{(i)}(p_x, p_y) = k_x^{(i)}(p_x) k_y^{(i)}(p_y) \in \mathbb{R}$$
$$k_x^{(i)}(p_x) = \max(0, w_i - d(\Delta x_i, p_x))$$

Graph Processing

- Process the nodes with a **spatio-temporal GNN** similar to our previous work [6].
  - At each time step, send messages between nodes.
  - Across time, update each node independently using a RNN.

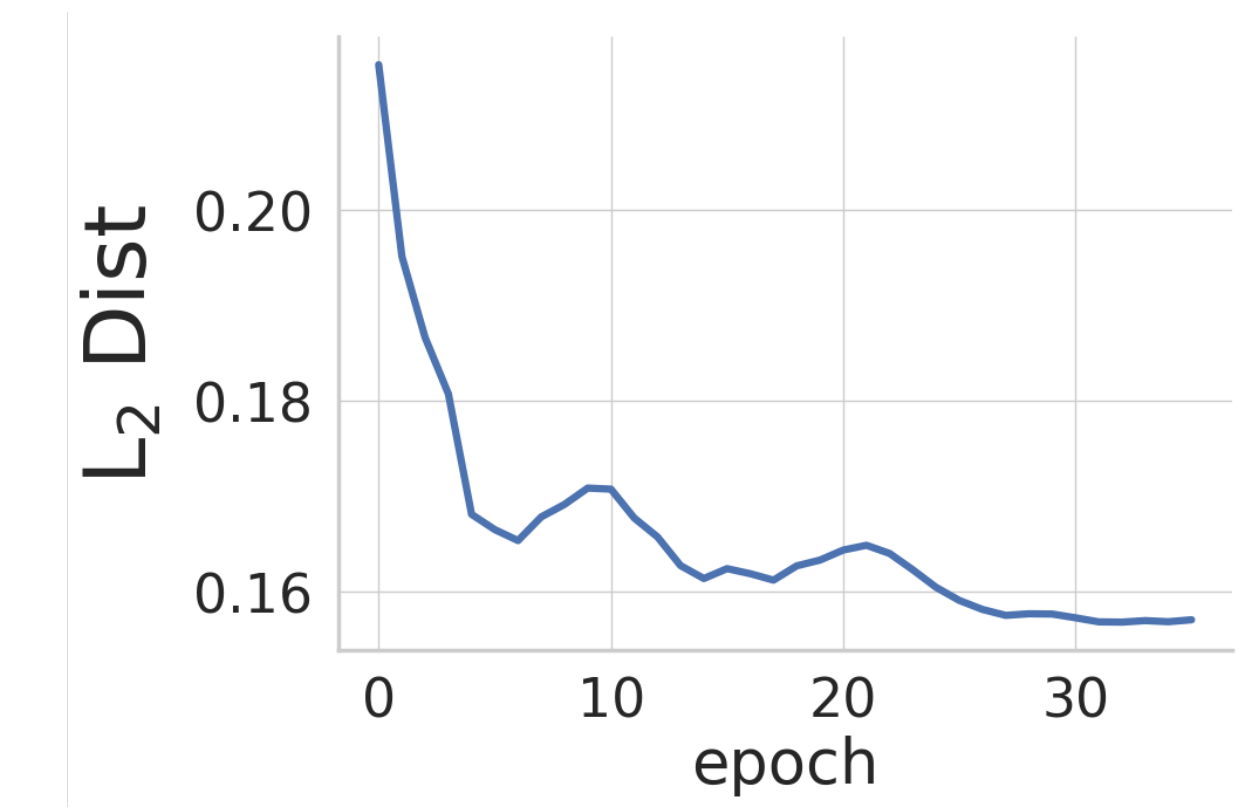
$$\mathbf{v}_{i,t} = \sum_{j=1}^{N} a(\mathbf{v}_{j,t}, \mathbf{v}_{i,t}) \mathrm{MLP}(\mathbf{v}_{j,t}, \mathbf{v}_{i,t}) \in \mathbb{R}^C$$
$$\hat{\mathbf{v}}_{i,t+1} = \mathrm{GRU}(\hat{\mathbf{v}}_{i,t}, \mathbf{v}_{i,t}) \in \mathbb{R}^C$$

Graph Re-Mapping

- The features of each updated node are sent to their initial region in the input, as indicated by their corresponding kernel.

$$\mathbf{y}_{p_x,p_y,t} = \sum_{i=1}^{N} \mathcal{K}_t^{(i)}(p_x, p_y) \hat{\mathbf{v}}_{i,t} \in \mathbb{R}^C$$

Object-centric representation



L_2 distance between our DyReG regions and ground-truth boxes on Smt-Smt-V2.

Model	FLOPS ↓	Dist ↓	Acc (%)↑
TSM-R50	65.8G	-	63.4
+ GNN+Fixed	+1.4G	0.170	64.1
+ GNN+ Detector	+41.1G	0.125	64.0
+ DyReg-GNN	+1.6G	0.129	**64.8**

Compute L_2 distance between ground-truth objects and predicted node regions.

- we do not optimize or enforce this metric in any way and learn **without object-level supervision**.
- distance decreases during training showing that **regions correlate with objects**.

Code and team homepage:

References

[1] Luo and Yuille ICCV 2019,
[2] Lin et al. ICCV 2019,
[3] Jiang et al. ICCV 2019,
[4] Zhang et al. TIP 2020,
[5] Kwon et al. ECCV 2020,
[6] Nicolicioiu et al. NeurIPS 2019