Multi-Task Edge Prediction in Temporally-Dynamic Video Graphs

Osman Ülger[1], Julian Wiederer[2], Mohsen Ghafoorian, Vasileios Belagiannis[3], Pascal Mettes[1]
[1]University of Amsterdam, [2]Mercedes-Benz, [3]Friedrich-Alexander-Universität

University of Amsterdam · ATLAS LAB. · Mercedes-Benz

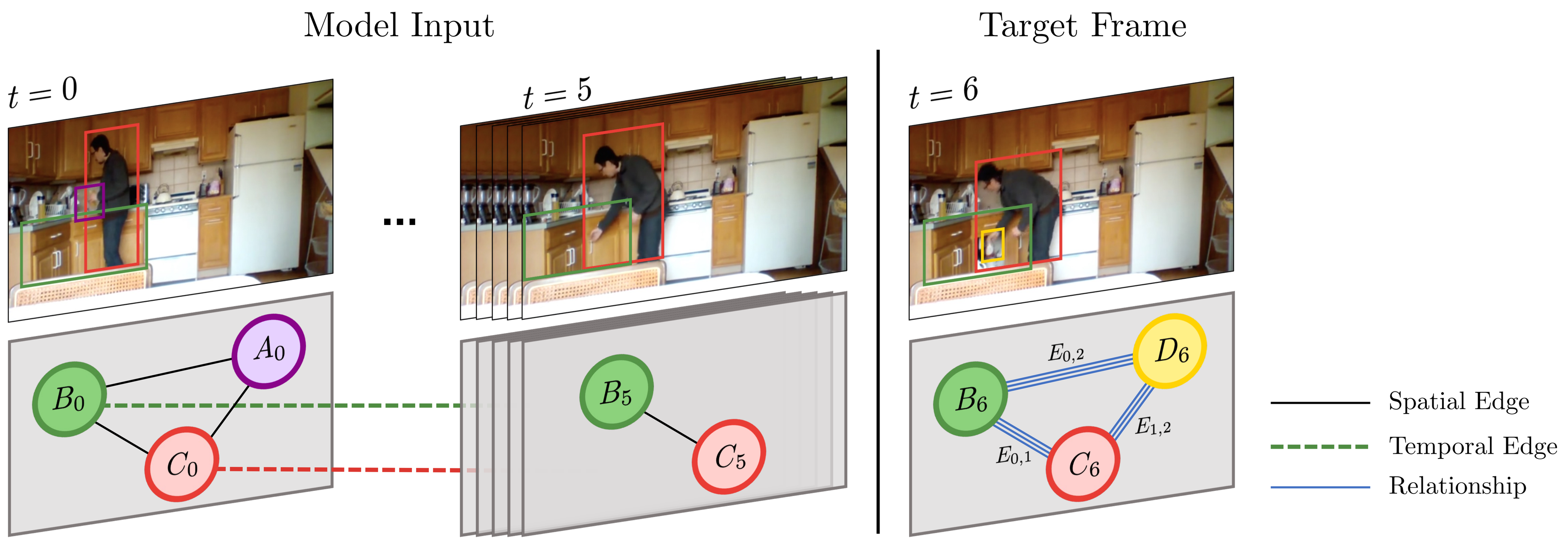
Why Temporally-Dynamic Video Graphs?

1. Objects (nodes) can enter and exit scenes over time
2. Visual relationships (edges) evolve dynamically

We need spatio-temporal video graphs **dynamic** in **space** and **time** – and the methods that can handle this type of data…

without padding the input!

Task

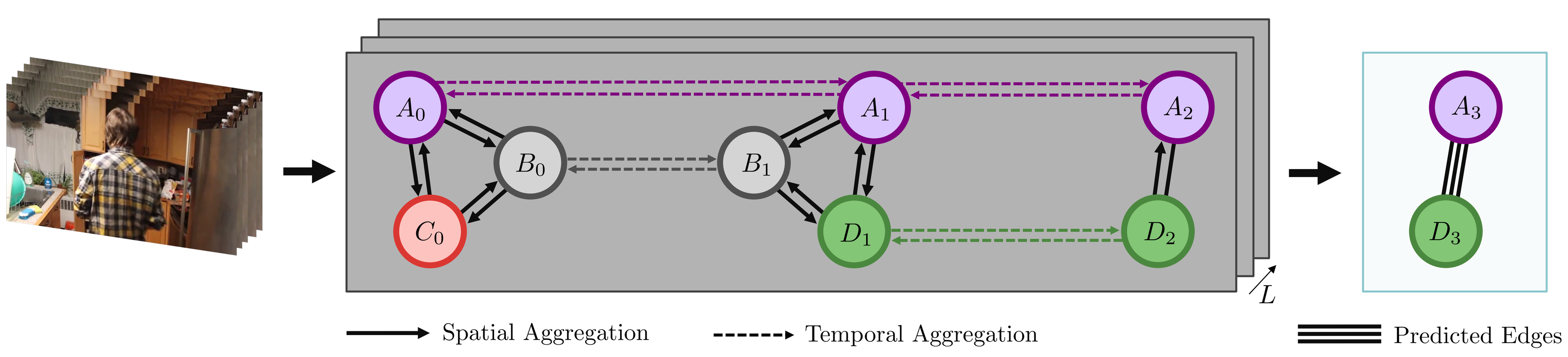


Build temporally-dynamic video graph from detections
- Spatial intra-frame links across different object detections
- Temporal inter-frame links between identical object detections across adjacent frames

Aggregate non-padded video graph
- Obtain *informed* feature representations
- Capture temporal and spatial dynamics of objects

Predict the future state of multiple relation types simultaneously
- Multi-label classification of relationships in target frame
- Target frame not part of spatio-temporal video graph

MTD-GNN | Approach



Temporally-dynamic scene graph creation
1. Obtain object feature representations of detections from pre-trained Mask-RCNN and Faster-RCNN backbones
2. Fully connect distinct object nodes in one frame
3. Link identical object nodes across adjacent using feature matching

Factorized spatio-temporal graph attention
- Multi-headed spatio-temporal graph attention layer
- Separation between relational **spatial** and **temporal** information

Multi-relational edge learning
- Use node representation outputs from graph attention layers
- Learning task-specific fully-connected layers for each relation

Prioritized Loss
- Emphasize the less frequent edge class in training set
- Normalize overrepresented class with #GT labels of larger class

Datasets
- CLEVRER
- Action Genome

Model Insights

Attention dimensionality
High dimensional attention layers are not a must. 256 dimensions for CLEVRER and 512 for Action Genome are sufficient

Attention heads
More heads consistently improve performance

Number of aggregations
Less aggregations are preferred to prevent over-smoothing

Multi-relational learning
Learning in multi-task setting benefits some, but not all edge types, likely caused by the difference in number of classes per type

Quantitative Results

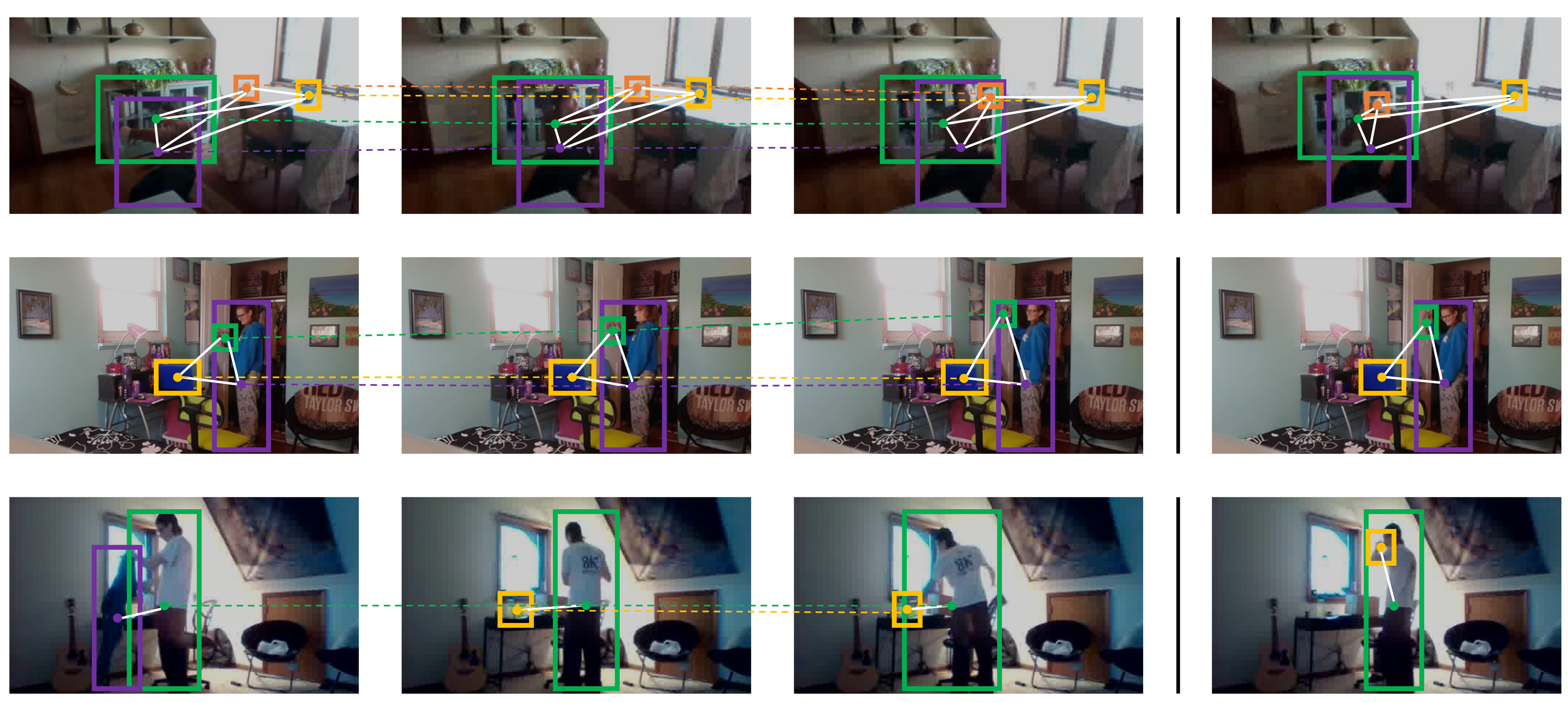
CLEVRER

	Collision prediction			Relative motion		
	F1↑	AP↑	AUC↑	F1↑	AP↑	AUC↑
Vanilla baselines						
RNN	0.247	0.322	0.527	0.733	0.710	0.514
LSTM	0.289	0.404	0.595	0.750	0.805	0.634
TCN	0.345	0.341	0.548	**0.839**	0.708	0.524
Graph attention (GA) baselines						
RNN + GA	0.168	0.410	0.597	0.835	0.758	0.591
LSTM + GA	0.283	0.389	0.604	0.796	0.783	0.606
TCN + GA	0.253	0.389	0.602	0.739	0.807	0.637
This paper						
MTD-GNN (Ours)	**0.594**	**0.607**	**0.768**	**0.839**	**0.820**	**0.688**

Action Genome

	Image		Video	
	R@20↑	R@50↑	R@20↑	R@50↑
With ground-truth detections				
VRD	24.92	25.20	24.63	24.87
Freq Prior	45.50	45.67	44.91	45.05
Graph R-CNN	23.71	23.91	23.42	23.60
MSDN	48.05	48.32	47.43	47.67
IMP	48.20	48.48	47.58	47.83
RelDN	49.37	49.58	48.80	48.98
MTD-GNN (Ours)	**50.09**	**50.09**	**49.54**	**49.54**
Without ground-truth detections				
MTD-GNN (Ours)	46.49	46.49	46.85	46.85

Qualitative Results



Conclusions

MTD-GNN can readily account for temporally-dynamic video graphs, making it more suitable for real-world scenarios with dynamic scenes

Experiments on CLEVRER and Action Genome show that our attention-based approach can model dynamic relations in graphs

Modelling multiple relations simultaneously can be beneficial when predicting individual relations

Code: