DAG-Net: Double Attentive Graph Neural Network for Trajectory Forecasting

Alessio Monti, Alessia Bertugli, Simone Calderara, Rita Cucchiara

{name.surname}@unimore.it

AImageLab
University of Modena and Reggio Emilia, Modena, Italy

Introduction

Problem definition

Trajectory forecasting is the field of research that deals with predicting the future trajectory of agents (*e.g.* people, vehicles, etc.) given a brief history of their past positions.

Applications

- **Autonomous driving**: self-driving vehicles and autonomous agents can take advantage of likely predictions to better plan their moves and avoid collisions
- **Surveillance**: surveillance systems can leverage accurate predictions to increase the quality of tracking and improve the crowd control
- **Sports**: in competitive settings such as sports, predicting the next moves of the opposing team can represent a concrete advantage in tactical analysis

Datasets

Stanford Drone Dataset

Stanford Drone Dataset[1] major highlights:

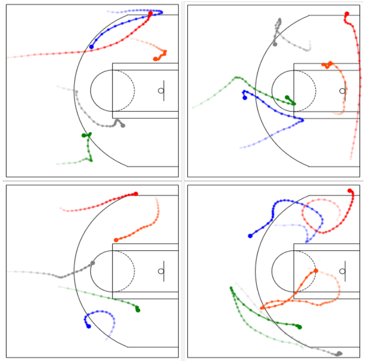
- Top-down videos recorded by a hovering drone
- 8 different scenarios from a college campus
- Several interacting agents (pedestrians, bikes, cars, *etc.*)
- Trajectory composed of a series of (x, y) absolute coordinates
- Real-world reference system (*i.e.* meters)

[1] Robicquet et al. "Learning Social Etiquette: Human Trajectory Prediction In Crowded Scenes", *In ECCV, 2016*

SportVU NBA Dataset

NBA dataset[1] major highlights:

- Top-down view of the court
- 10 agents per scene: 5 defenders, 5 attackers
- Complex dynamics: rules, tactics, opponents
- Trajectory composed of a series of (x, y) absolute coordinates
- Real-world reference system (*i.e.* feet)

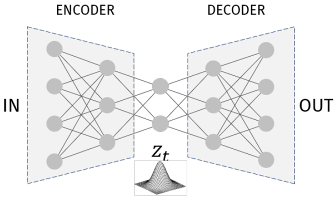


[1] SportVU - STATS Perform, https://www.statsperform.com/team-performance/basketball/optical-tracking/

Baseline

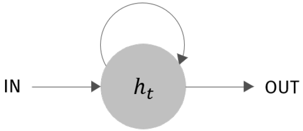
Variational Recurrent Neural Network

Our baseline is represented by the **Variational Recurrent Neural Network**[1]: the architecture comes from the union of a Recurrent Neural Network[2] and a generative model (the Variational Auto-Encoder[3])



Variational Auto-Encoder

+



Recurrent Neural Network

[1] Chung et al. "A Recurrent Latent Variable Model for Sequential Data", *In NIPS, 2015*

[2] Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", *In EMNLP, 2014*

[3] Kingma and Welling. "Auto-Encoding Variational Bayes", *In ICLR, 2014*

Variational Recurrent Neural Network

VAEs are excellent **generative models** that allow us to produce multiple future outcomes given the same input, in perfect compliance with the **multi-modal nature** of human motion.

However, without further solutions, consecutive generations are totally independent: in our case, the predicted positions of a same trajectory would not be correlated.

Variational Recurrent Neural Network

By **conditioning the generation on the hidden state of a recurrent cell** that watches the sequence of inputs that are given to the network. This way, the VRNN can correlate the future samples predicted by the VAE along the temporal axis, thus recovering the useful patterns that characterise the input data.

Variational Recurrent Neural Network

Pros:

- Captures the multi-modal nature of human movement
- Produces future generations in coherence with past positions

Cons:

- Predicts every trajectory independently
- It does not consider agents that share the same scene

Variational Recurrent Neural Network

Pros:

- Captures the multi-modal nature of human movement
- Produces future generations in coherence with past positions

Cons:

- Predicts every trajectory independently
- It does not consider agents that share the same scene

Attentive VRNN

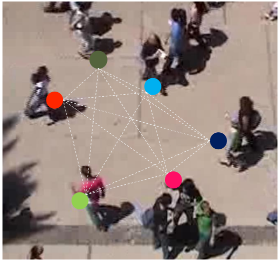
Attentive VRNN (A-VRNN)

By sharing pedestrians' information across the agents in the same scene, every agent becomes aware of the others.

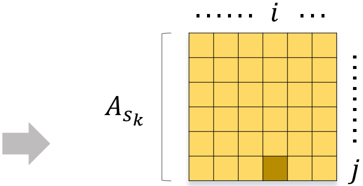
$\Rightarrow$ we choose to share the hidden states $\mathbf{h}_t$ of the different agents

Attentive VRNN (A-VRNN)

We treat the **pedestrian space as an undirected fully-connected graph** where every node (agent) is described by its hidden state $\mathbf{h}_t$.



$scene_k$

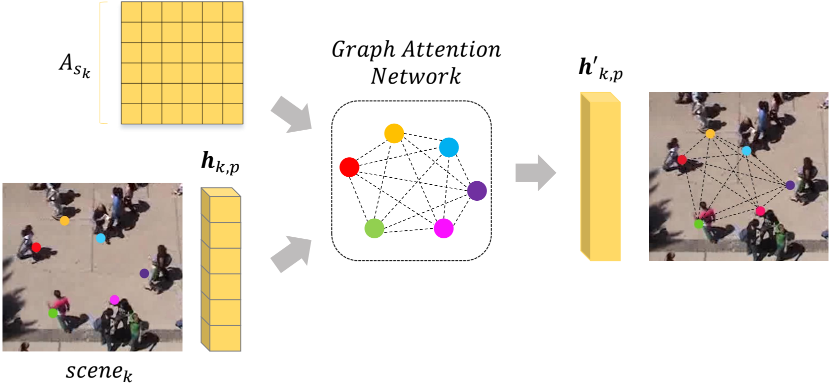


$$A_{ij} = exp\left(-\frac{d(i,j)}{2\sigma^2}\right)$$

To describe the (spatial) relationships between the nodes, we employ a **similarity-based adjacency matrix**.

Attentive VRNN (A-VRNN)

This way, we can exploit a **Graph Attention Network**[1] to recombine the hidden states at each node, thus providing the subjects with neighbourhood information.



[1] Veličković et al. "Graph Attention Networks", *In ICLR, 2018*

Attentive VRNN (A-VRNN)

Pros:

- The agent becomes aware of what the others have done
- By conditioning the network generations on the refined hidden states, we exploit community information and increase its precision

Cons:

- The model exploits only *past* information
- It lacks a longer-term view on what could happen in the future

Attentive VRNN (A-VRNN)

Pros:

- The agent becomes aware of what the others have done
- By conditioning the network generations on the refined hidden states, we exploit community information and increase its precision

Cons:

- The model exploits only *past* information
- It lacks a longer-term view on what could happen in the future

DAG-Net

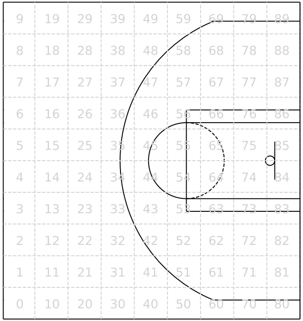
Future information: goals (1)

To jointly exploit future information, we additionally condition the VRNN generation on a new information extracted from ground-truth data: **the long-term objective** of the agent.

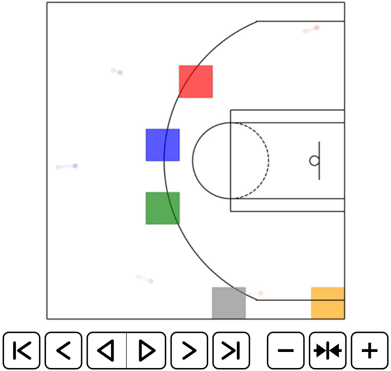
To this end, we firstly divide the pedestrian space in a regular grid of macro cells.

Future information: goals (2)

The goal $\mathbf{g}_t$ can be then expressed in spatial terms by one-hot encoding: the goal is the region of space (cell) that the agent will occupy after a given number of time-steps.

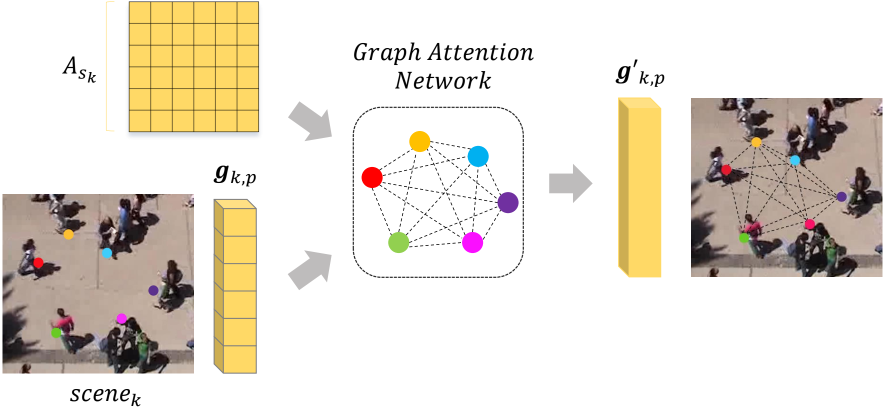


9	19	29	39	49	59	69	79	89
8	18	28	38	48	58	68	78	88
7	17	27	37	47	57	67	77	87
6	16	26	36	46	56	66	76	86
5	15	25	35	45	55	65	75	85
4	14	24	34	44	54	64	74	84
3	13	23	33	43	53	63	73	83
2	12	22	32	42	52	62	72	82
1	11	21	31	41	51	61	71	81
0	10	20	30	40	50	60	70	80



DAG-Net: Double Graph Attention Neural Network

To collect community information, in a similar fashion as before we employ a (second) **Graph Attention Network** that shares the single goals between the agents that are spatially close to each other.



Results

Metrics

Average Displacement Error

$$ADE = \frac{\sum_{i \in \mathcal{P}} \sum_{t=0}^{T_{pred}} \sqrt{((\hat{x}_t^i, \hat{y}_t^i) - (x_t^i, y_t^i))^2}}{\mid \mathcal{P} \mid \cdot \, T_{pred}} \tag{1}$$

⇒ average Euclidean distance over entire predicted sequence

Final Displacement Error

$$FDE = \frac{\sum_{i \in \mathcal{P}} \sqrt{((\hat{x}_{T_{pred}}^i, \hat{y}_{T_{pred}}^i) - (x_{T_{pred}}^i, y_{T_{pred}}^i))^2}}{\mid \mathcal{P} \mid} \tag{2}$$

⇒ Euclidean distance on the last predicted time-step

Results (1) - Ablation

Dataset	Model	Interact.	Goals	ADE	FDE
NBA (atk)	VRNN	✗	✗	9.58	15.83
	A-VRNN	✓	✗	9.67	15.96
	DAG-Net (Our)	✓	✓	**9.18**	**13.54**
NBA (def)	VRNN	✗	✗	7.07	10.62
	A-VRNN (Our)	✓	✗	7.01	10.42
	DAG-Net (Our)	✓	✓	**7.01**	**9.76**
SDD	VRNN	✗	✗	0.58	1.17
	A-VRNN (Our)	✓	✗	0.56	1.14
	DAG-Net (Our)	✓	✓	**0.53**	**1.04**

Jointly considering past and future information grants **more reliable generations** than considering single trajectories or relying only on agents' past interactions.

Results (2) - SOTA comparison

Model	**NBA (atk)**		**NBA (def)**		**SDD**	
	ADE	FDE	ADE	FDE	ADE	FDE
STGAT[1]	9.94	15.80	7.26	11.28	0.58	1.11
Social-Ways[2]	9.91	15.19	7.31	10.21	0.62	1.16
Weak-Supervision[3]	9.47	16.98	7.05	10.56	-	-
Our	**8.98**	**14.08**	**6.87**	**9.76**	**0.53**	**1.04**

Our double-graph solution also allows to **advance the current state of the art** in both the urban and the sports settings, proving the model strength in different environments.

[1] Amirian et al. ”Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs”, *In CVPR-W, 2019*

[2] Zhan et al. ”Generating Multi-Agent Trajectories using Programmatic Weak Supervision”, *In ICLR, 2019*

[3] HUang et al. ”STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction”, *In ICCV, 2019*

Thank you!

Source code: https://github.com/alexmonti19/dagnet