Vision and Cognitive Systems

SCQ1097939 - LM CS,DS,CYB,PD

Sequential data in Vision, Predictive Vision

Prof. Lamberto Ballan

University of Padova, A.Y. 2023/24 January 8, 2024

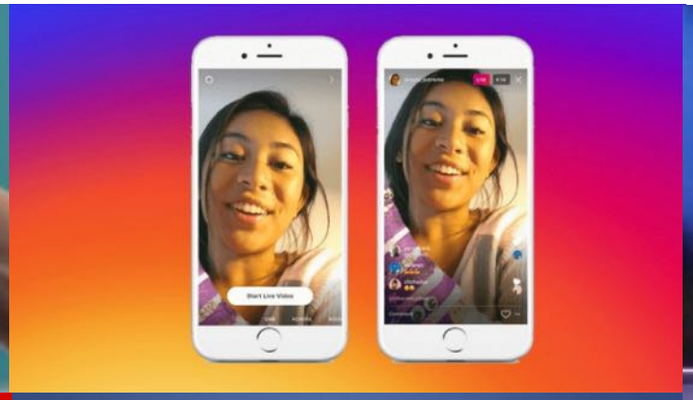
DIPARTIMENTO MATEMATICA
Department of Mathematics “Tullio Levi-Civita”

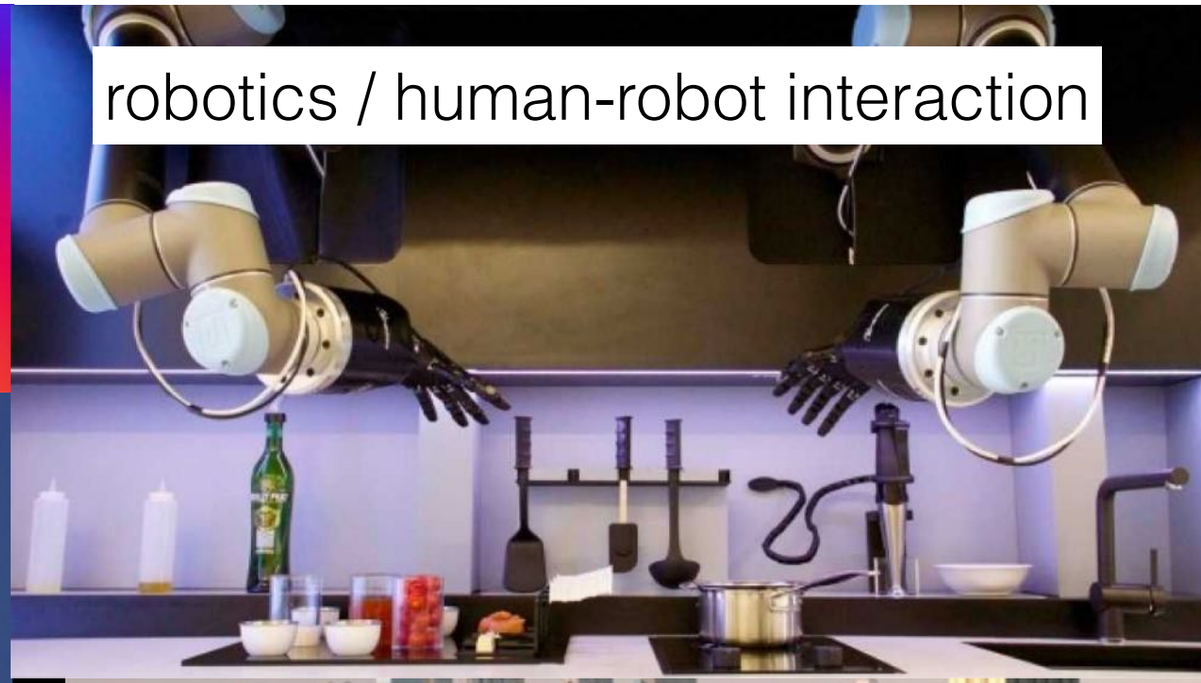
Beyond image understanding

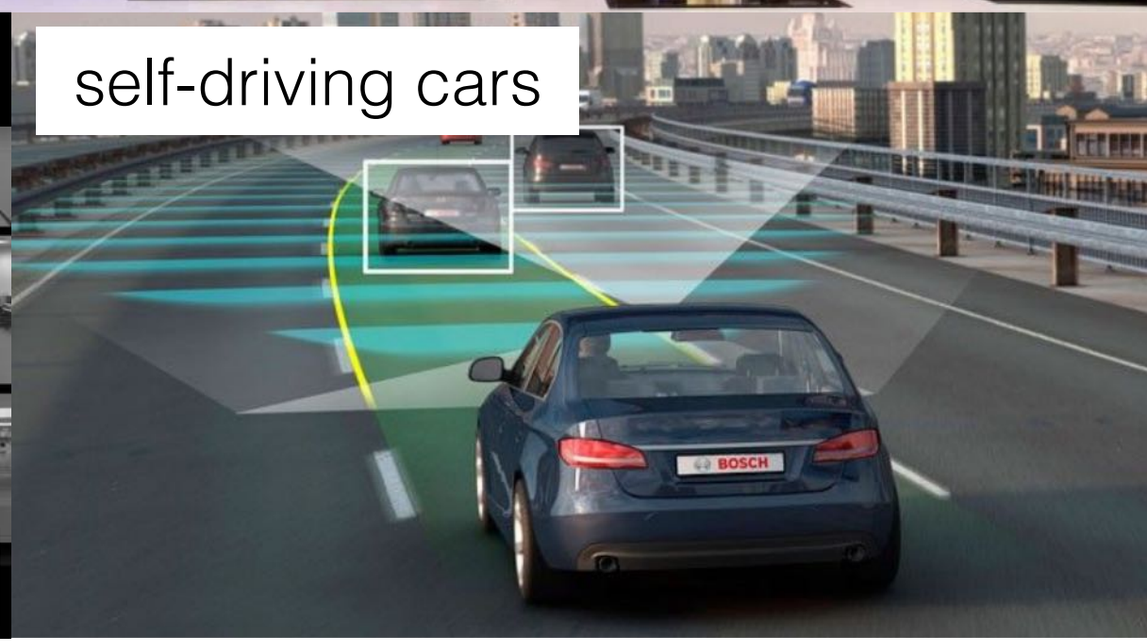
- Videos and sequential data in vision
  - RNNs for vision tasks (a few examples); vision and language (e.g. image captioning)
- Predictive vision
  - Focus on a specific task: trajectory prediction

Sequential data (in vision)

- Videos are everywhere







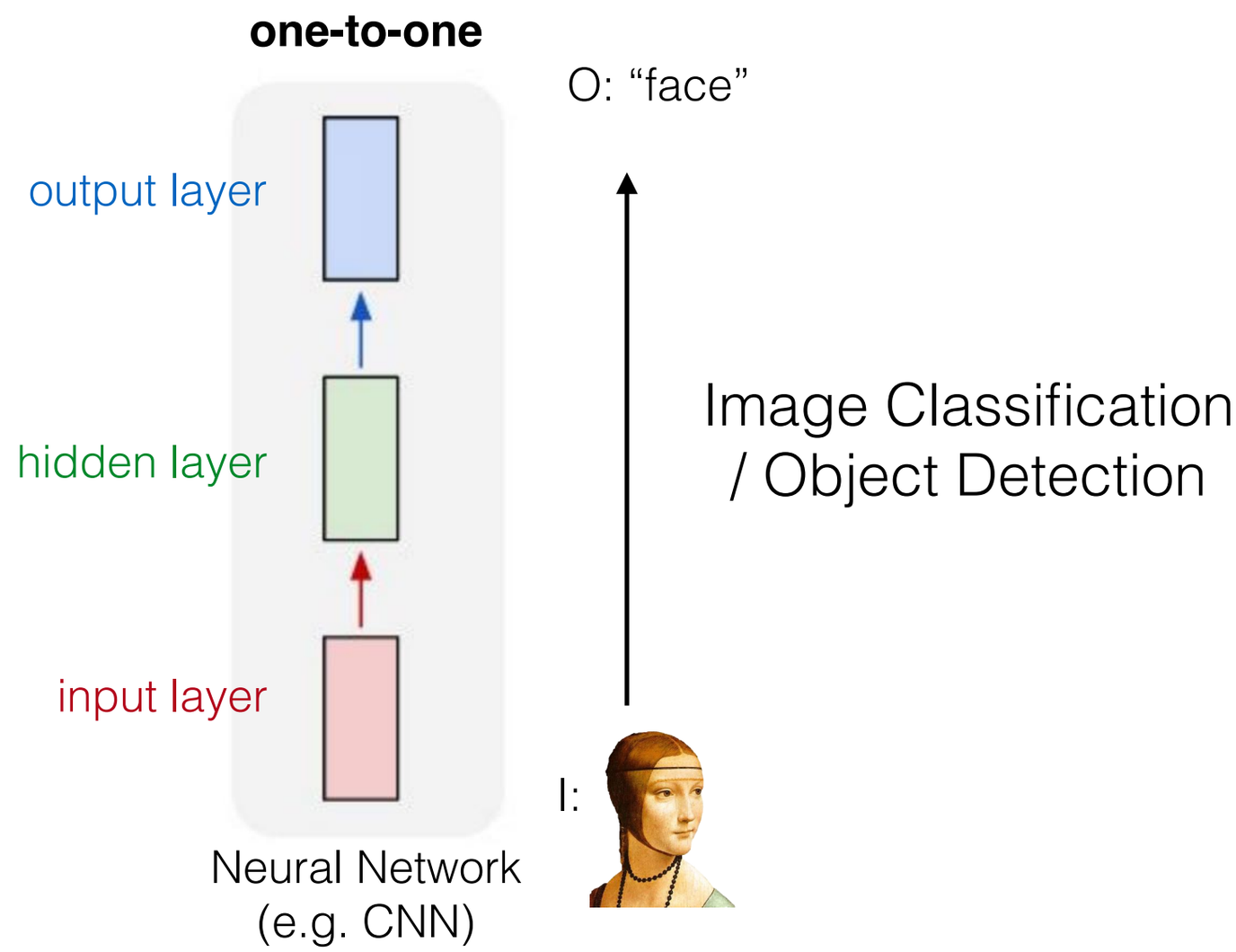
Sequential data (in vision)

- Datasets and benchmarks:

EPIC KITCHENS

Sequential data (in vision)

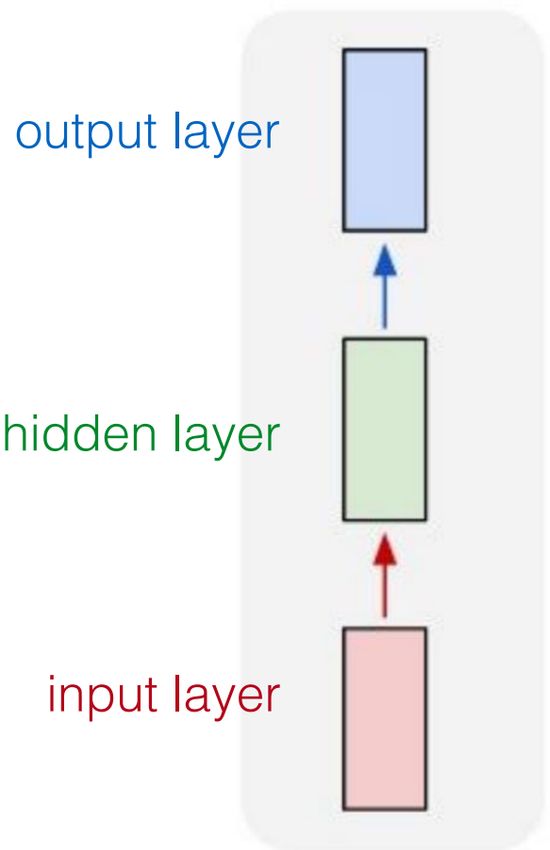
- How to process sequences?
  - Let's focus on neural network architectures



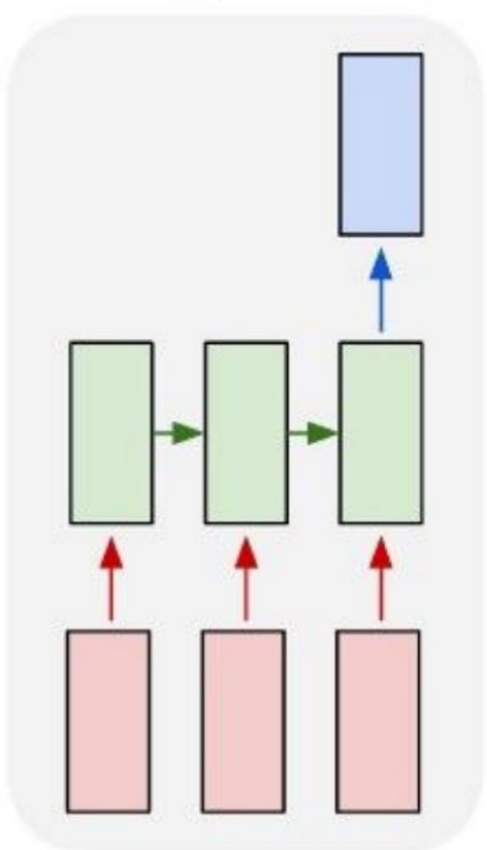
Sequential data (in vision)

- How to process sequences?
  - Let's focus on neural network architectures

one-to-one



many-to-one



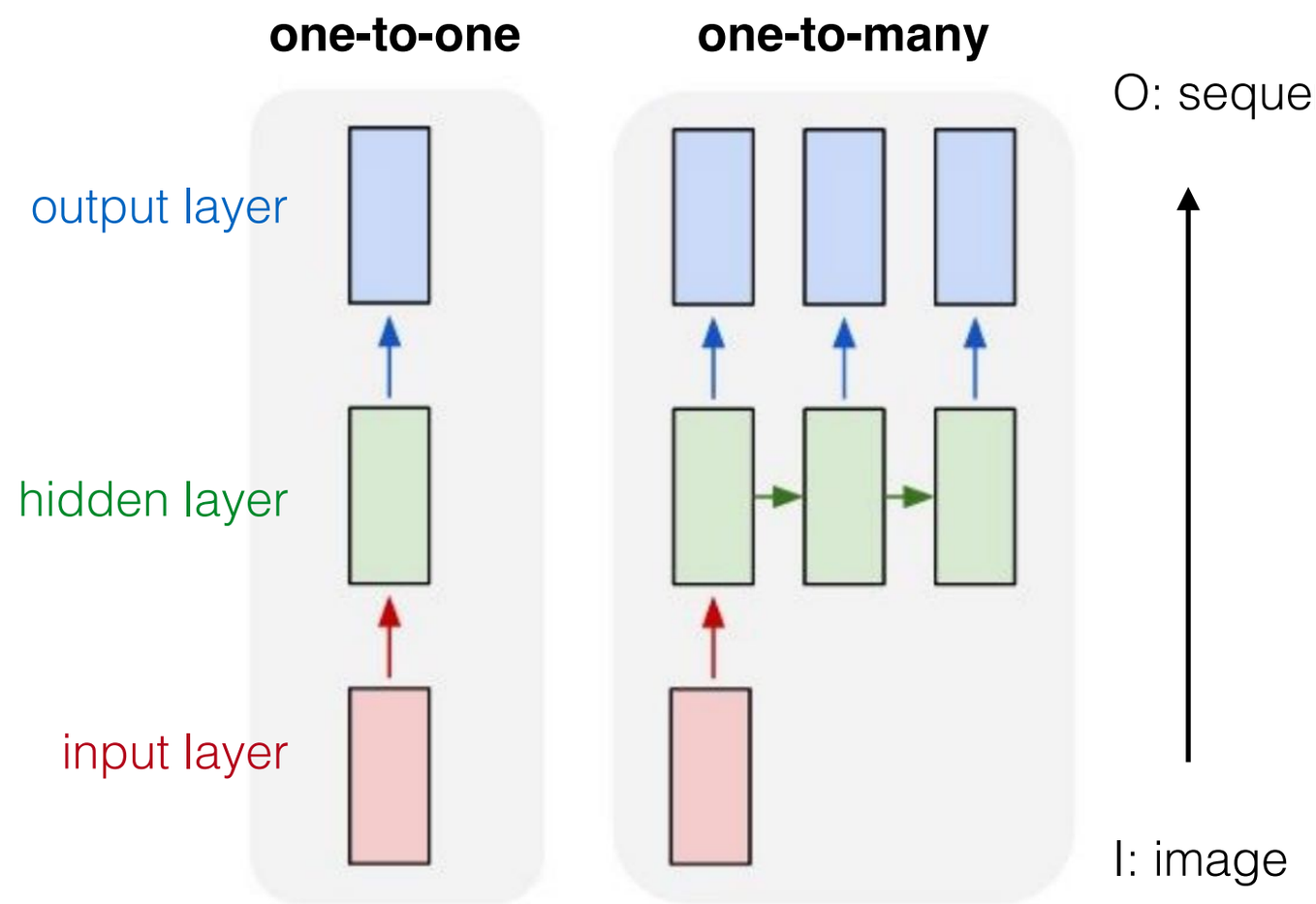
O: action class

e.g. Action Recognition

I: video frames

Sequential data (in vision)

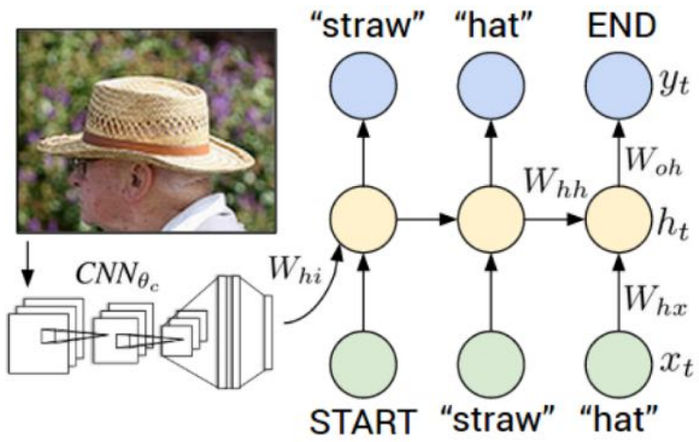
- How to process sequences?
  - Let's focus on neural network architectures



O: sequence of words

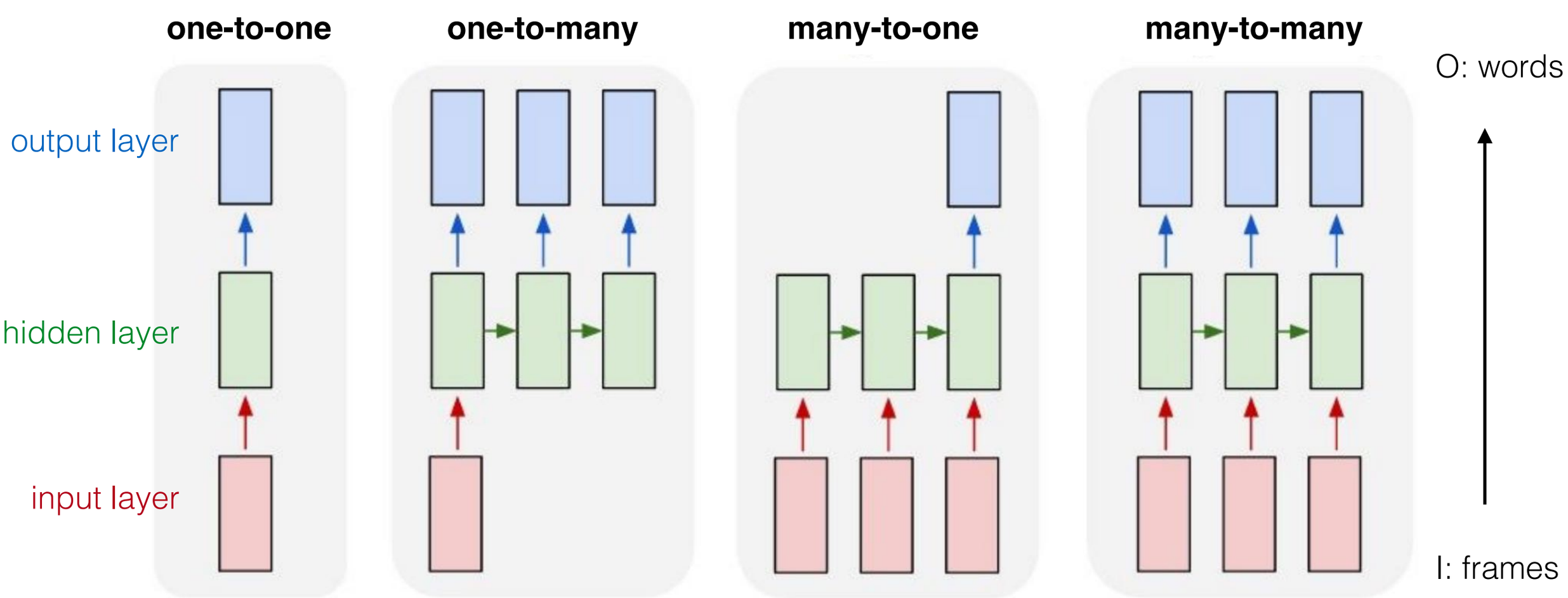
I: image

e.g. Image Captioning



Sequential data (in vision)

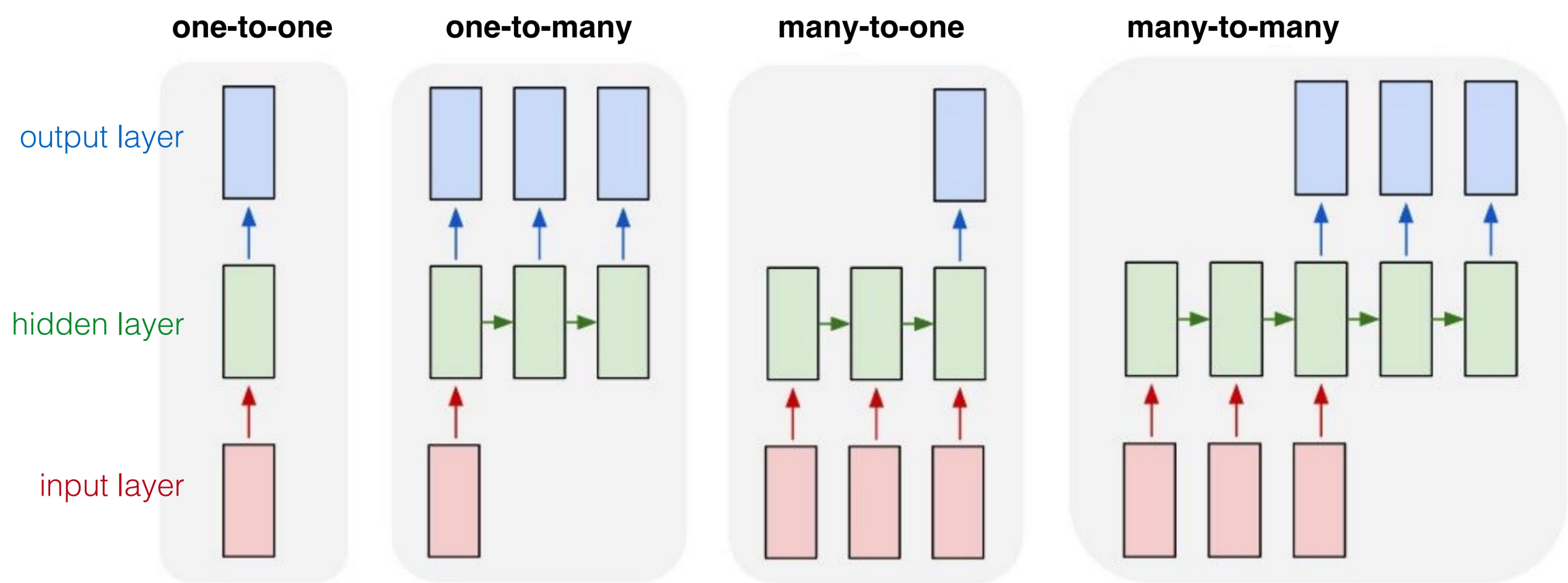
- How to process sequences?
  - Let's focus on neural network architectures



e.g. Video Classification on frames

Sequential data (in vision)

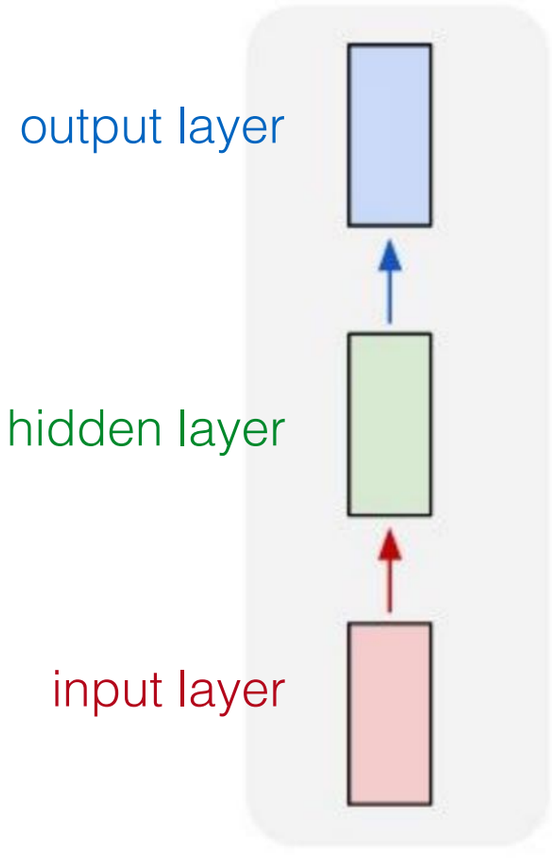
- How to process sequences?
  - Let's focus on neural network architectures



e.g. Video Captioning

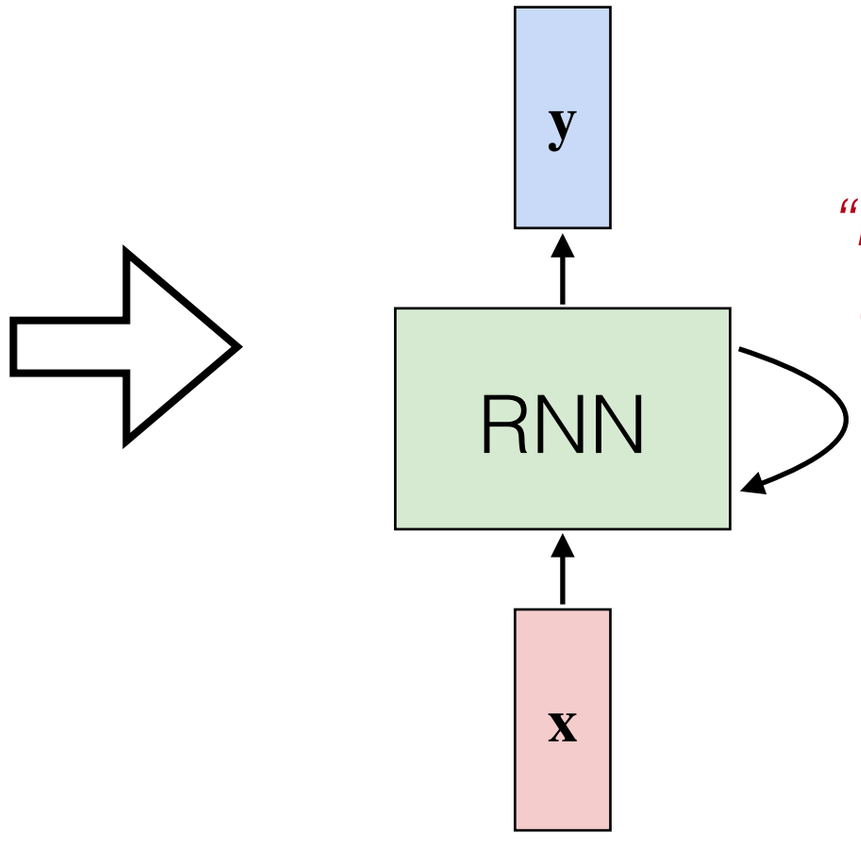
Sequential data (in vision)

- How to process sequences?
  - Let's focus on neural network architectures



Neural Network

Recurrent Neural Networks (RNN)



Key idea: RNNs have an "internal state" that is updated as a sequence is processed

Recurrent Neural Networks

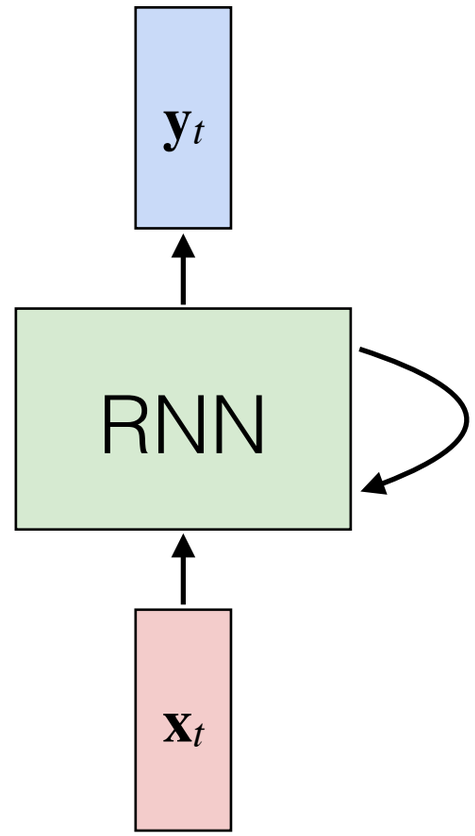
- We process a sequence of vectors $\mathbf{x}$ by applying a recurrence formula at every time step
  - Note: the same function and the same parameters W are used at every time step t

new state *old state*

$$\boldsymbol{h}_t = f_W(\boldsymbol{h}_{t-1}, \mathbf{x}_t)$$

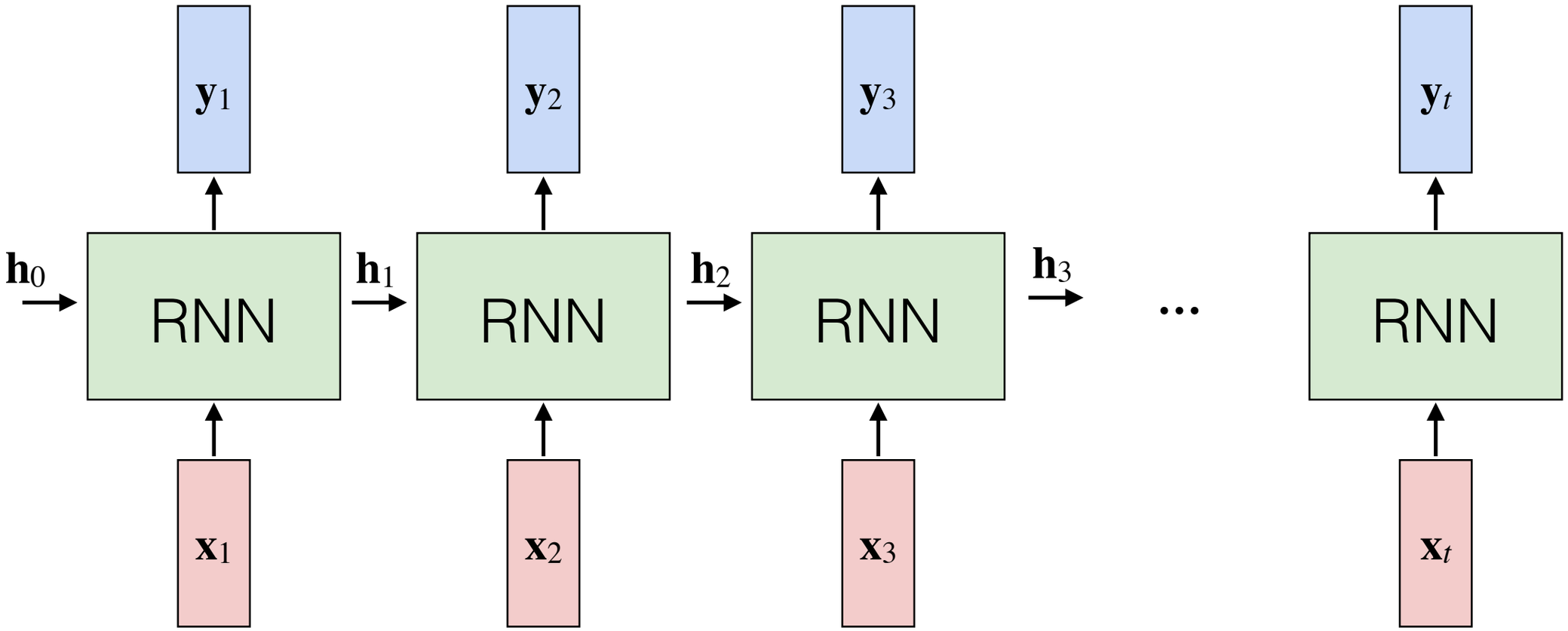
input vector at some time step t

some function parameterized by W



Recurrent Neural Networks

- We process a sequence of vectors $\mathbf{x}$ by applying a recurrence formula at every time step

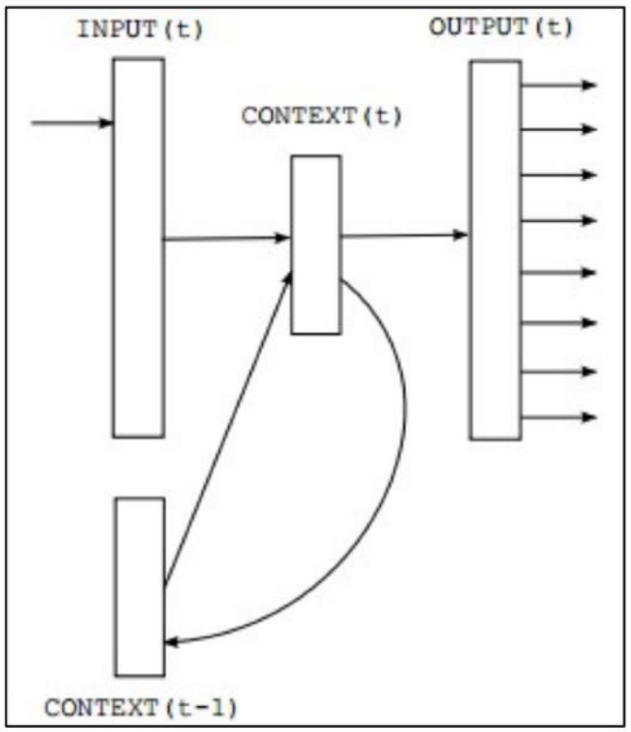


Recurrent Neural Networks

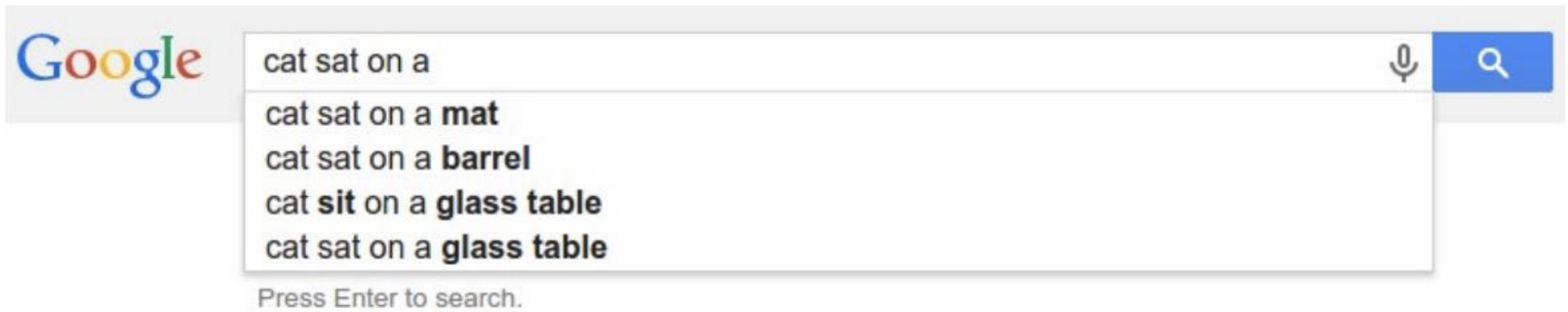
- RNNs come in many variants:
  - Hopfield networks
  - Elman networks: often referred as to Vanilla RNNs
  - Long Short-Term Memory (LSTM) networks
    - State-of-the-art model in many domains
  - Gated Recurrent Unit (GRU) networks
    - They are similar to LSTMs, but they have fewer parameters
  - Bi-directional RNNs
    - They combine the output of two RNNs: one processing the sequence from left to right, the other one from right to left

Recurrent Neural Networks & NLP

- RNNs are good at modelling sequences…



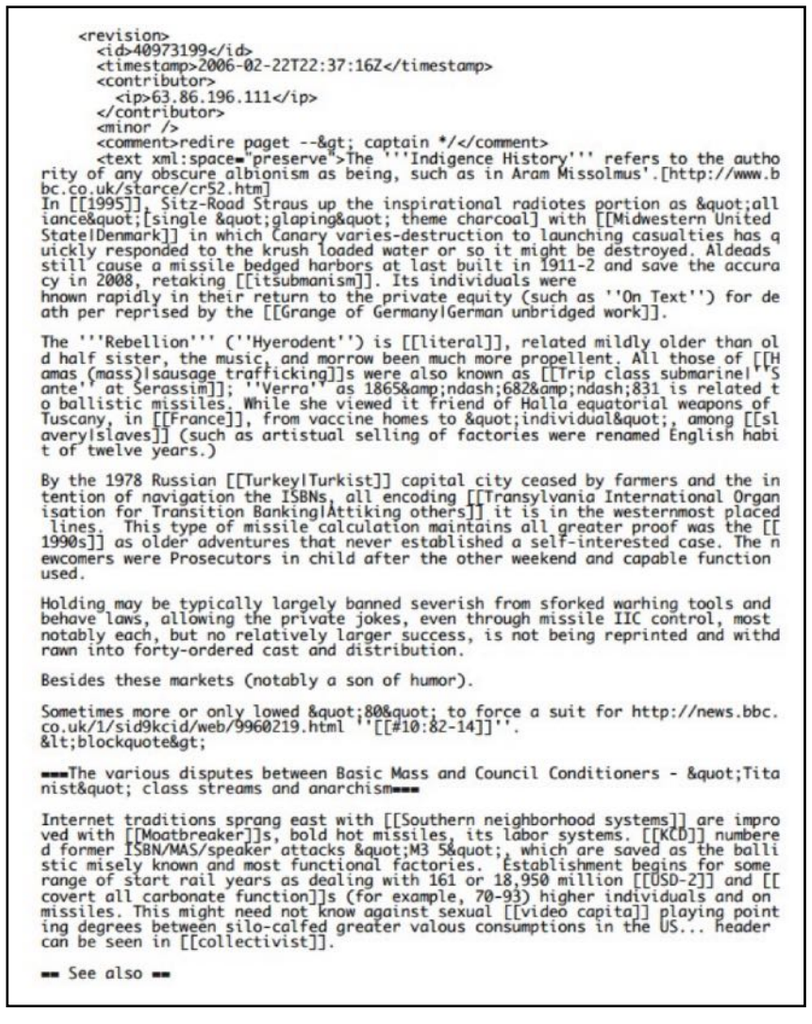
Word-level language model. Similar to:

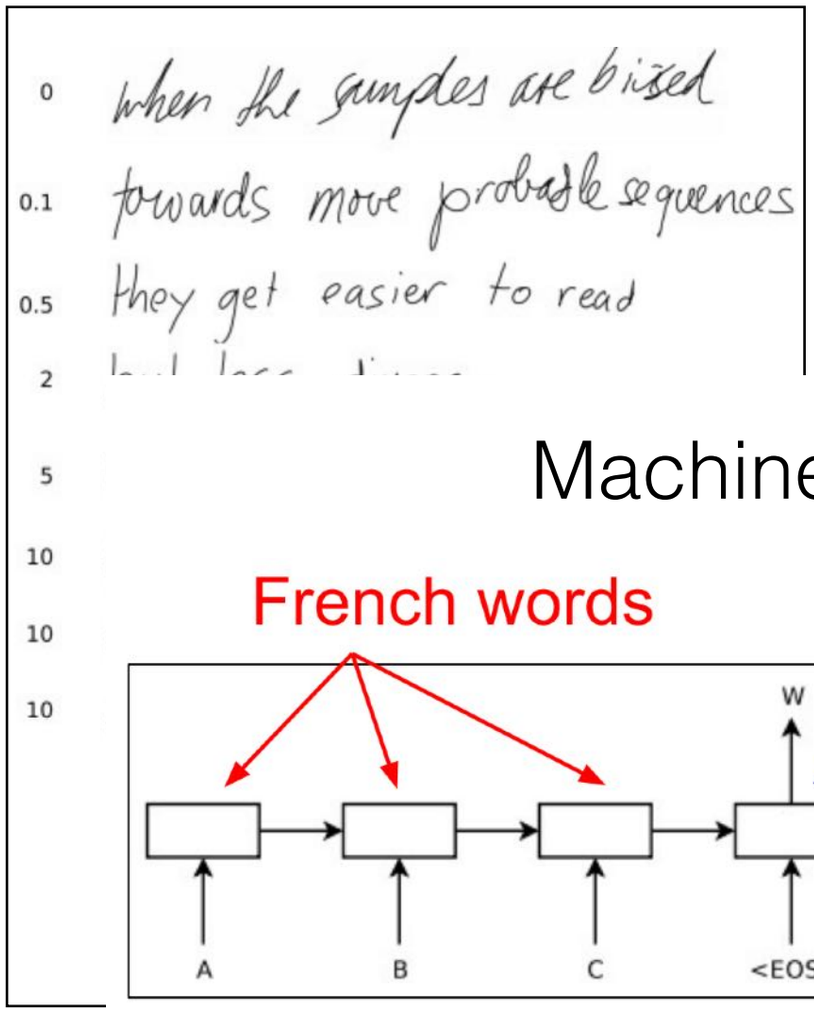


T.Mikolov et al., “Recurrent neural network based language model”, INTERSPEECH 2010

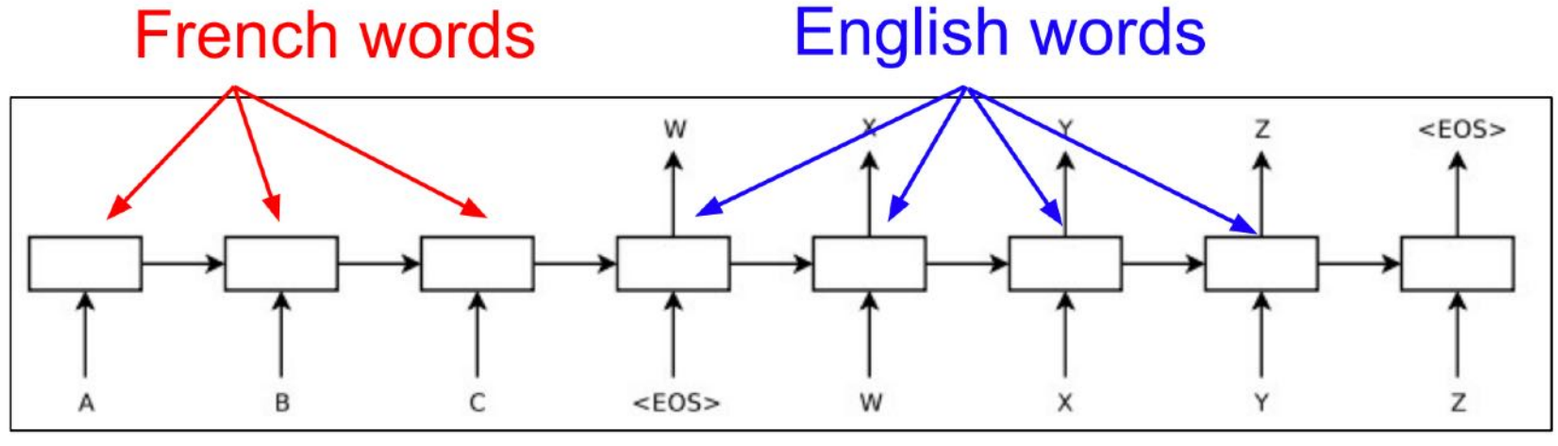
Recurrent Neural Networks & NLP

- RNNs are good at modelling sequences…





Machine Translation:



A.Graves, “Generating Sequences With Recurrent Neural Networks”, arXiv 2013

I.Sutskever, “Sequence to Sequence Learning with Neural Networks”, NeurIPS 2014

Recurrent Neural Networks & NLP

- Suppose we had the training sentence "cat sat on mat"
  - We want to train a language model:

$$P(\texttt{next word} \mid \texttt{previous word})$$

  - This means we want these to be high:

$$P(\texttt{cat} \mid [\texttt{<start>}])$$

$$P(\texttt{sat} \mid [\texttt{<start>}, \texttt{cat}])$$

$$P(\texttt{on} \mid [\texttt{<start>}, \texttt{cat}, \texttt{sat}])$$

$$P(\texttt{mat} \mid [\texttt{<start>}, \texttt{cat}, \texttt{sat}, \texttt{on}])$$

Recurrent Neural Networks & NLP

- Suppose we had the training sentence "cat sat on mat"

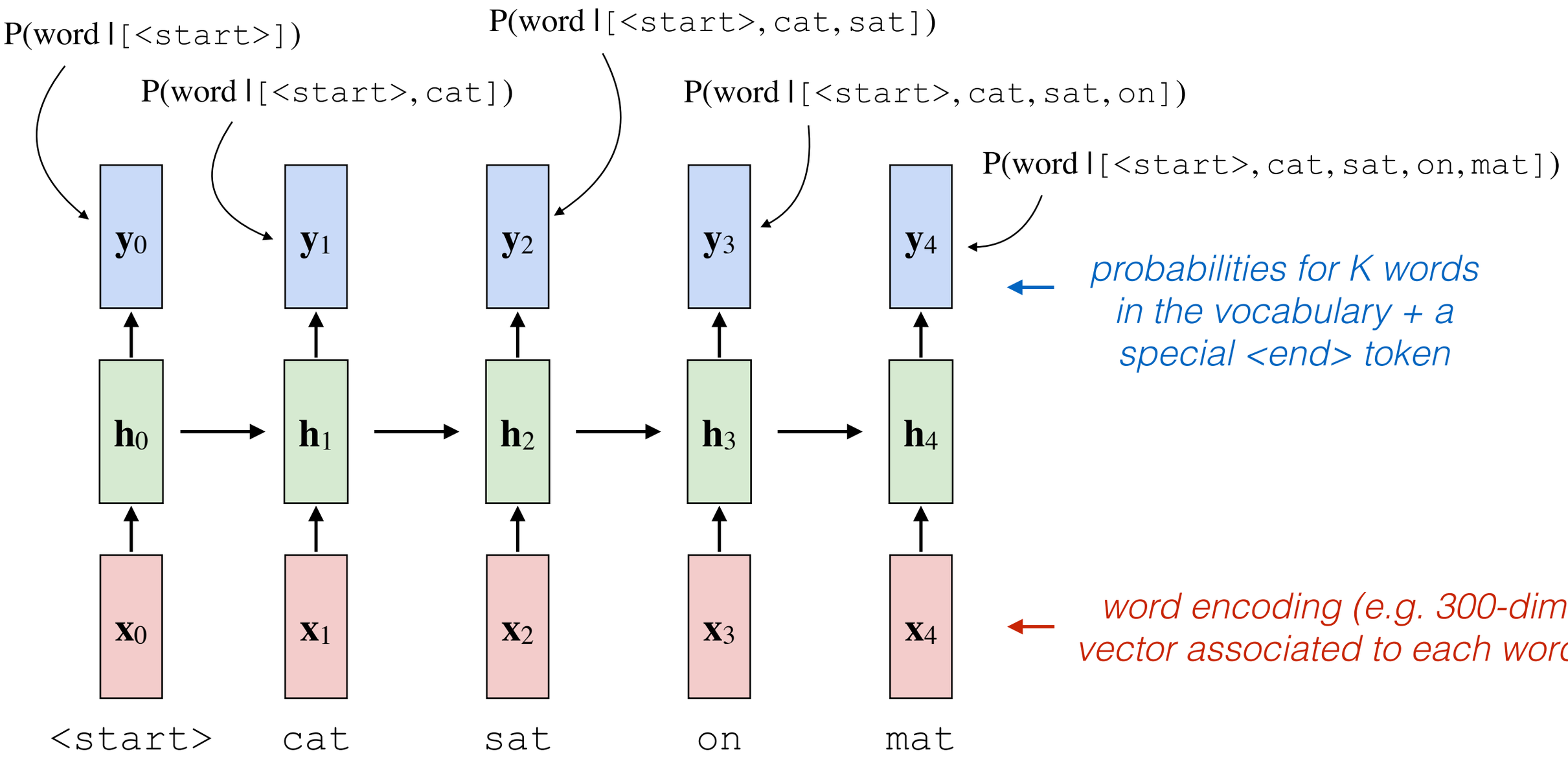
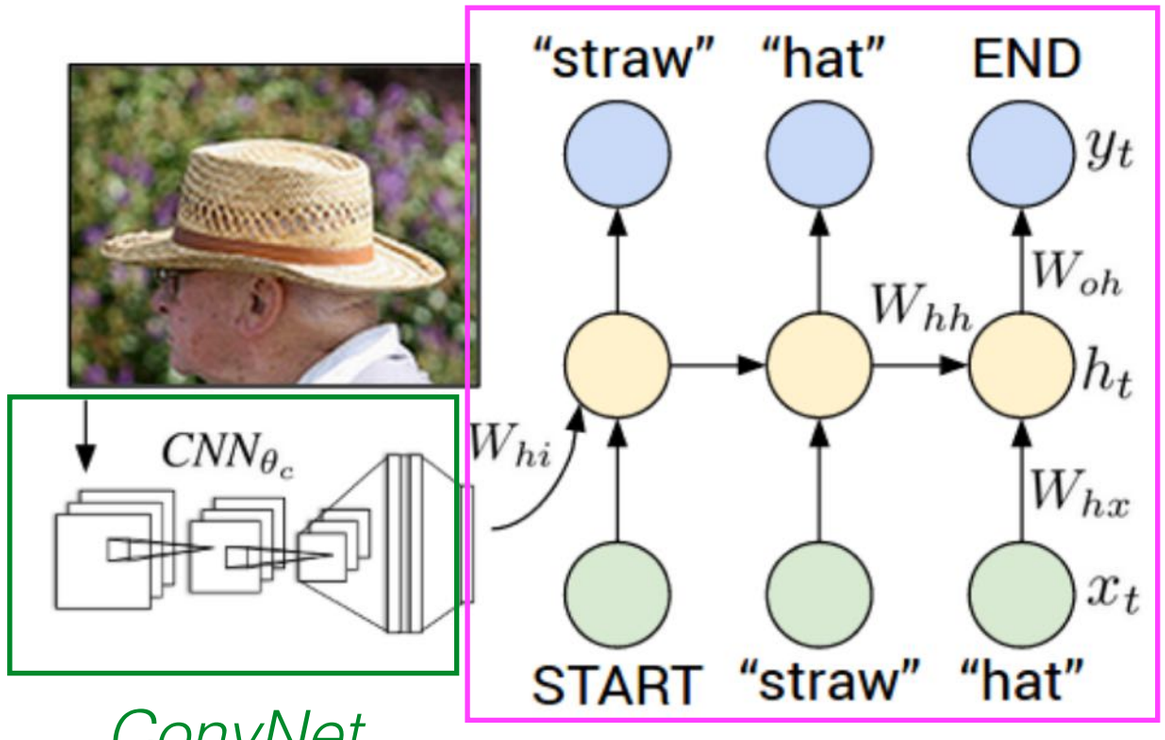


Image Captioning

- Goal: generate a (short) textual description of the image content



ConvNet

Recurrent Neural Network

A.Karpathy, L.Fei-Fei, "Deep Visual-Semantic Alignments…", CVPR 2015

O.Vinyals et al., "Show and tell: A neural image caption generator", CVPR 2015

J.Donahue et al., "Long-Term Recurrent Convolutional Networks…", CVPR 2015

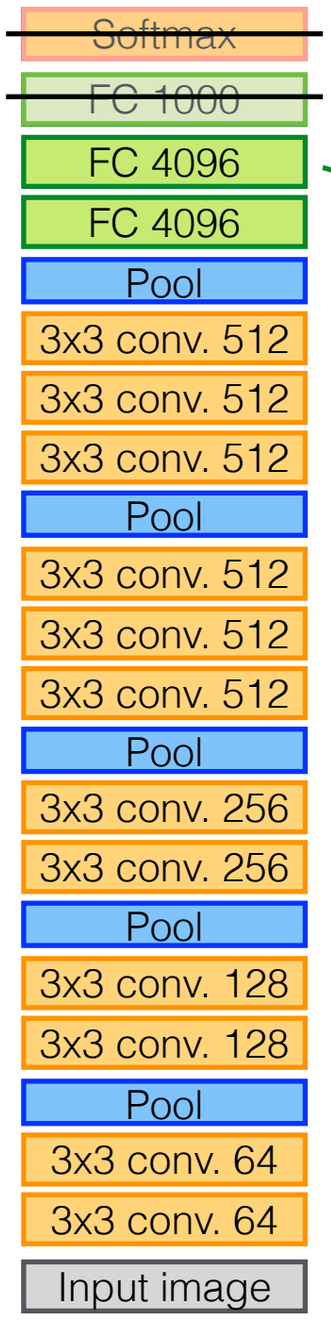
X.Chen, L.Zitnick, "Mind's Eye: A Recurrent Visual Representation…", CVPR 2015

Image credit: Karpathy, Fei-Fei

Image Captioning

Test Image

Pre-trained on ImageNet



h *is conditioned by the image representation*

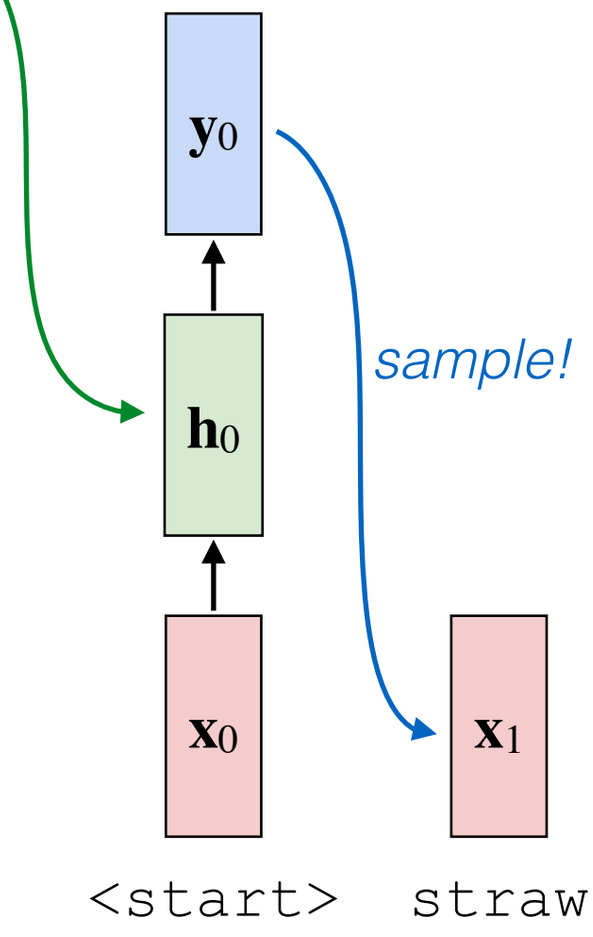
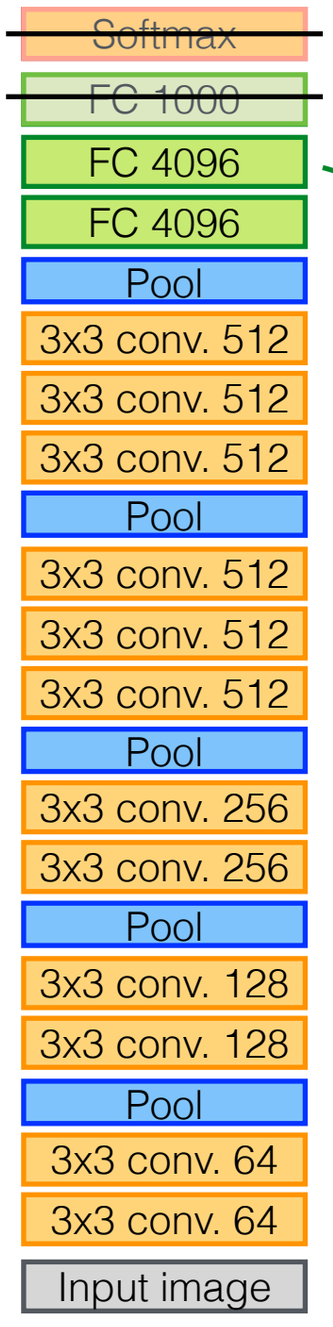


Image Captioning

Test Image

Pre-trained on ImageNet



h *is conditioned by the image representation*

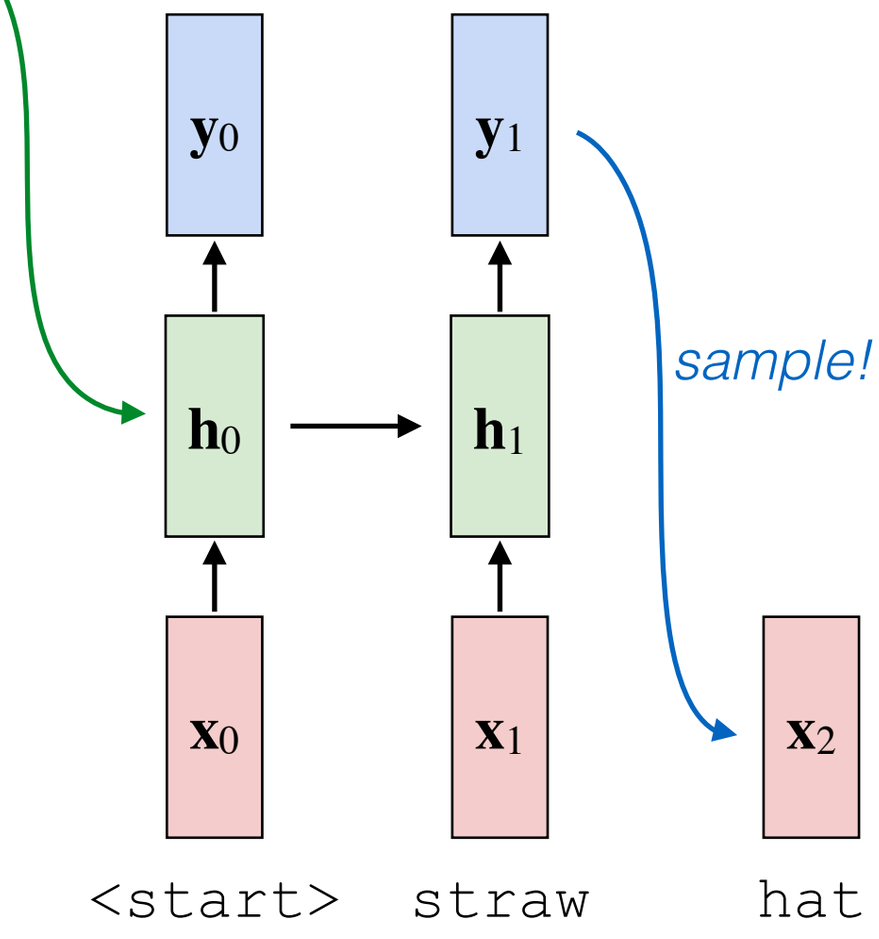
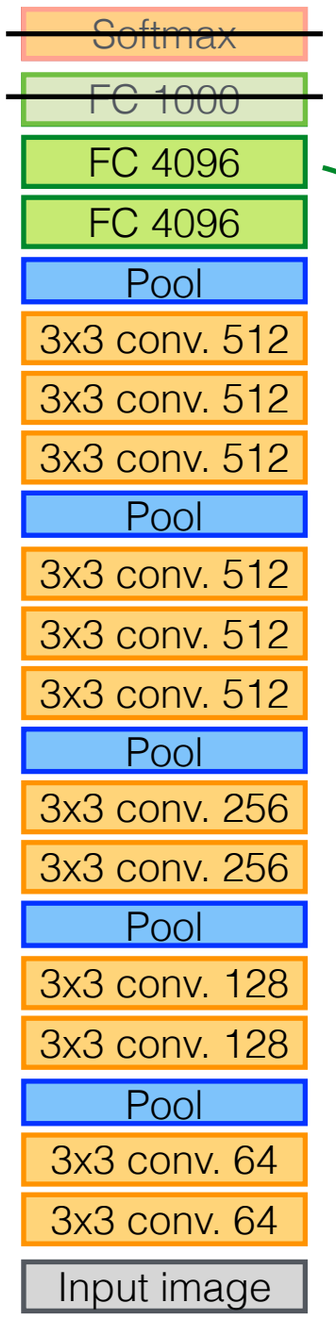


Image Captioning

Test Image

Pre-trained on ImageNet



h *is conditioned by the image representation*

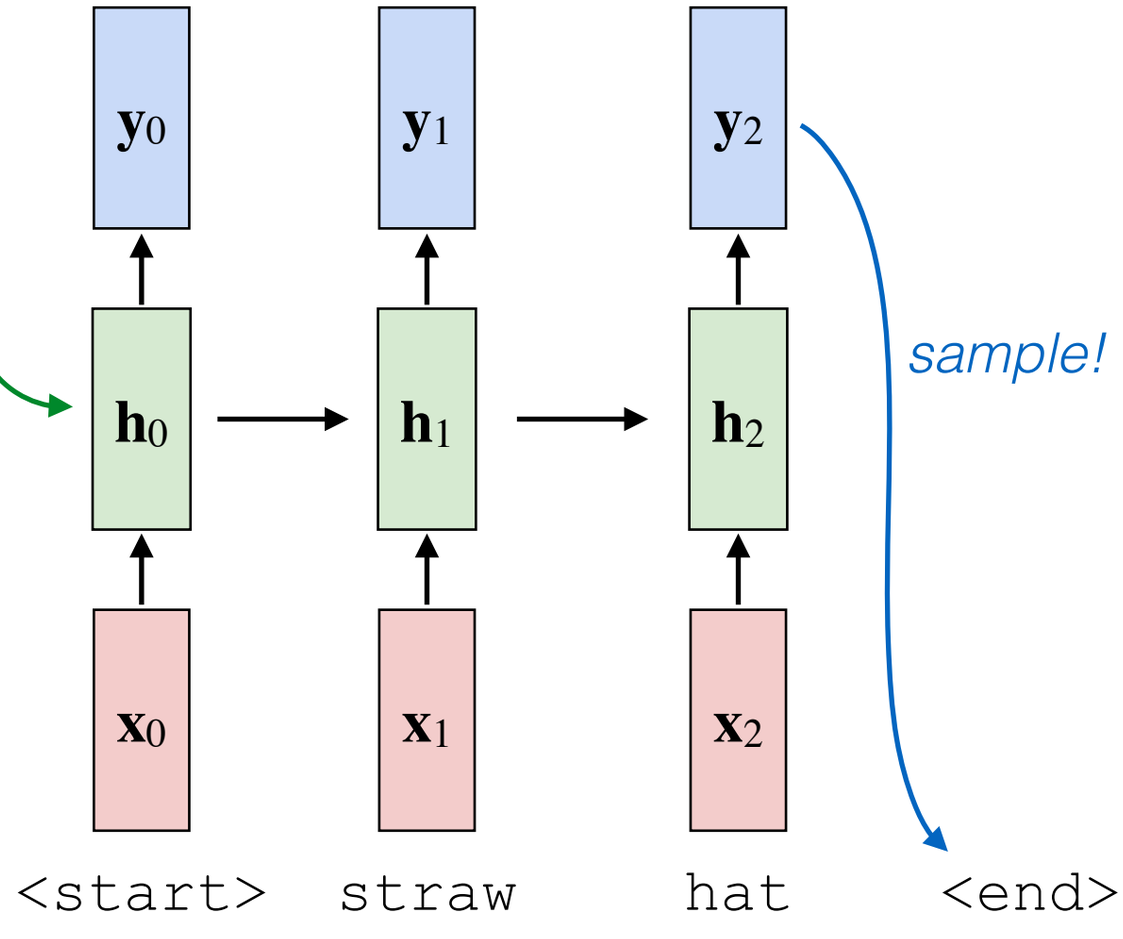


Image Captioning: results

(captions generated with neuraltalk2: https://github.com/karpathy/neuraltalk2)

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

Two people walking on the beach with surfboards

A tennis player in action on the court

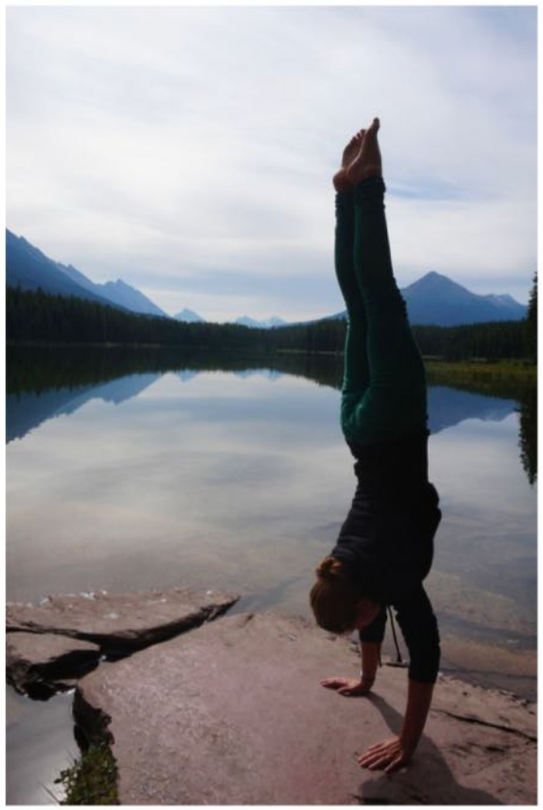
Two giraffes standing in a grassy field

Image Captioning: failure cases

(captions generated with neuraltalk2: https://github.com/karpathy/neuraltalk2)

A woman is holding a cat in her hand

A person holding a computer mouse on a desk

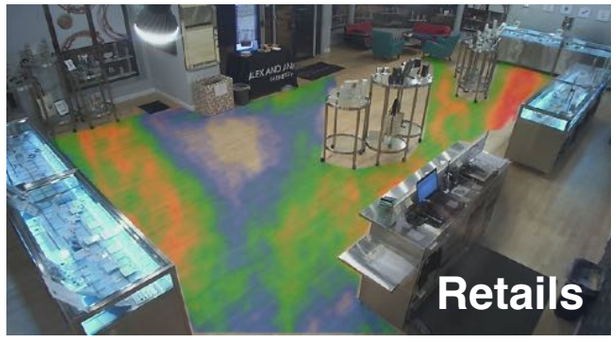


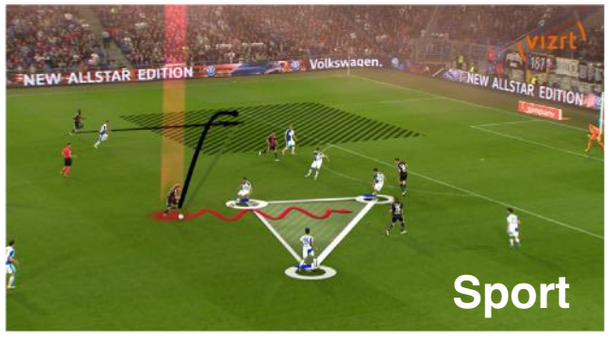
A woman standing on a beach holding a surfboard

A bird is perched on a tree branch

A man in a baseball uniform throwing a ball

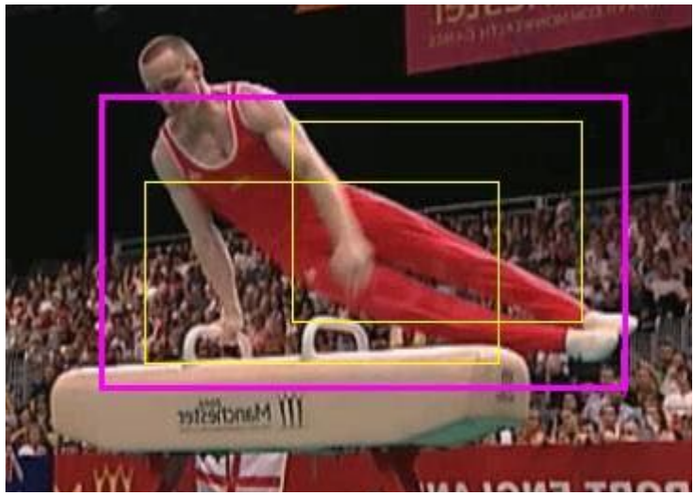
A brief intro to predictive vision and motion prediction

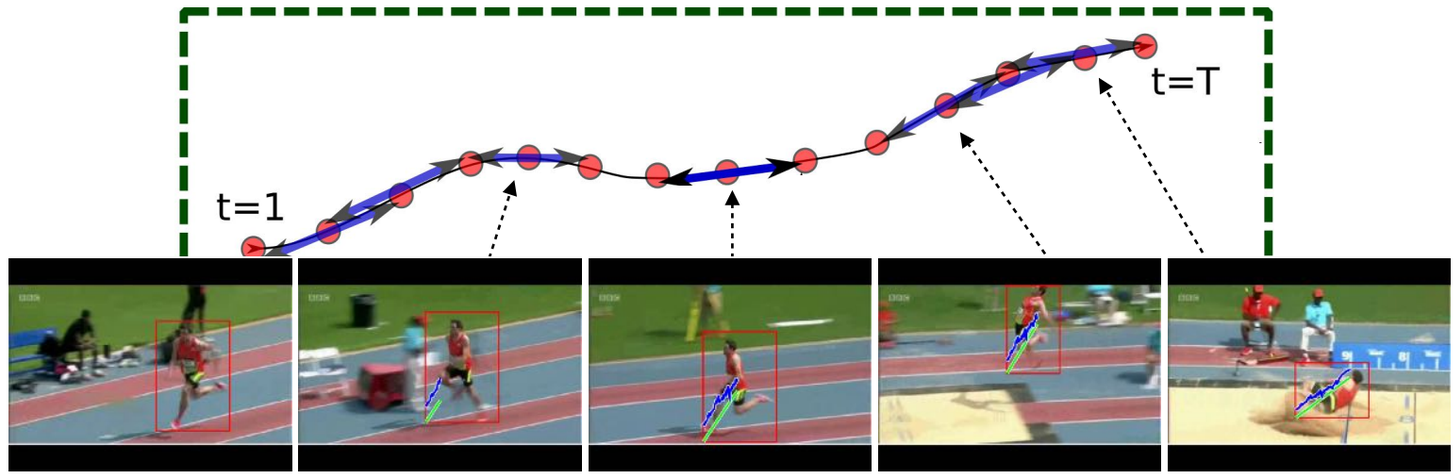




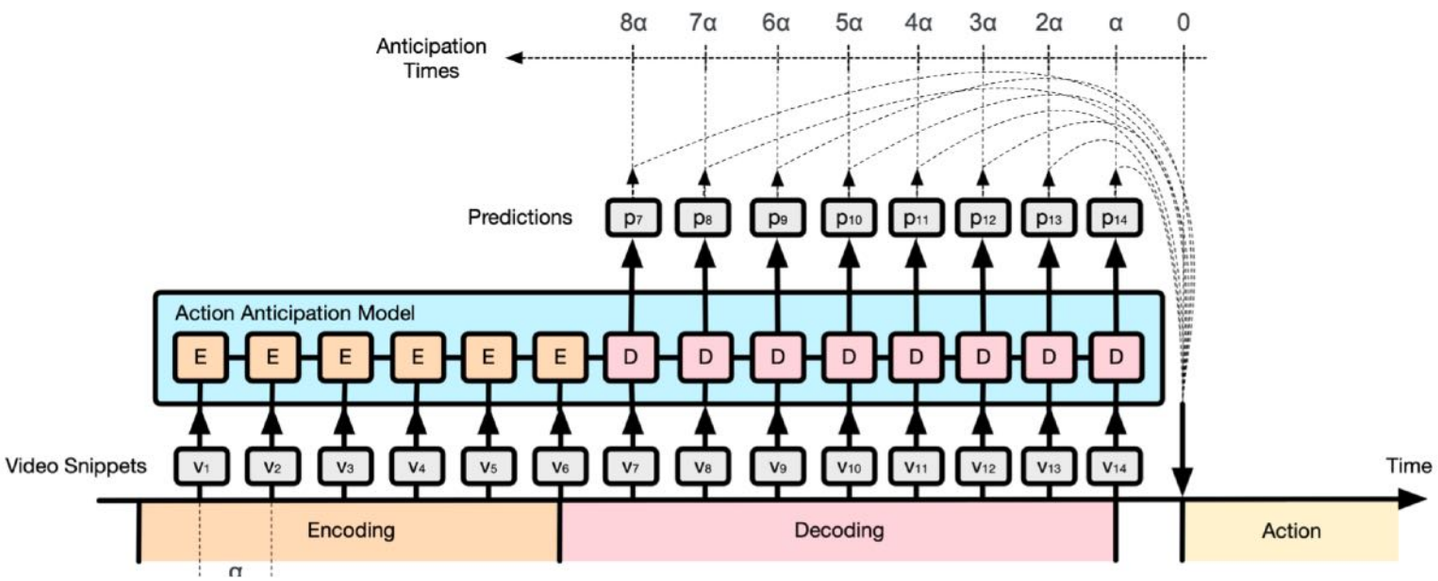
Predictive vision: main tasks

- Predicting action progress/completion in videos





- Action anticipation and early action detection



EPIC KITCHENS

55 hours

2513 actions

(125 verbs, 352 nouns)

[Camporese, Coscia, Furnari, Farinella, **Ballan** - ICPR'20]

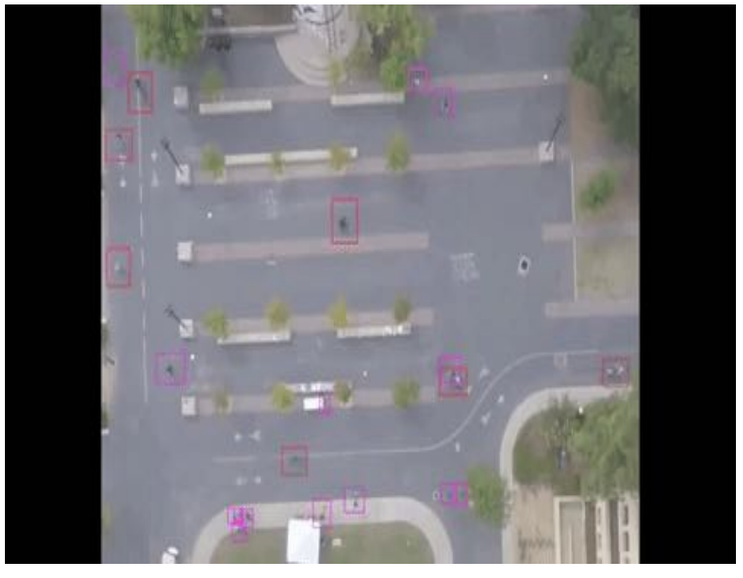
Humans in crowded spaces

- Humans motion in (a crowded) space is mostly influenced by the scene and the other active agents
  - E.g. Stanford Drone Dataset: videos of various agents that navigate a real world outdoor environment

ETH
(ETH and HOTEL)

+

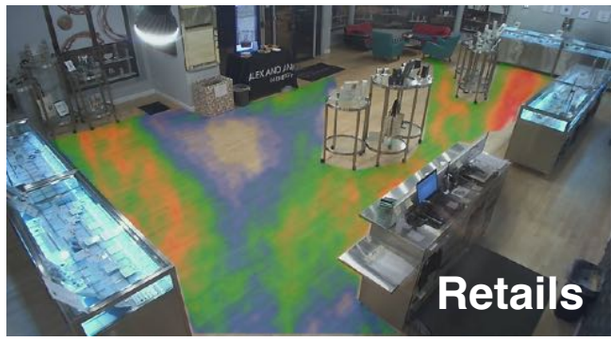
UCY
(ZARA1,ZARA2,UNIV)

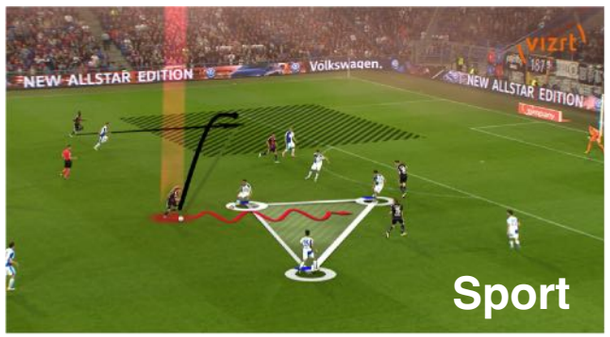


SDD

Goal: motion (*trajectory*) prediction

- Given a single picture and an observed agent, humans are able to predict the most likely future

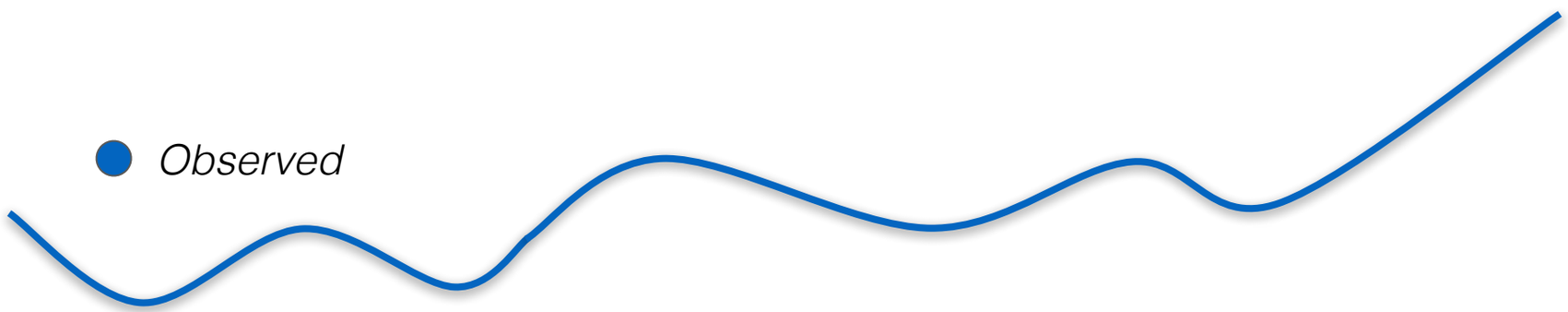




Goal: motion (*trajectory*) prediction

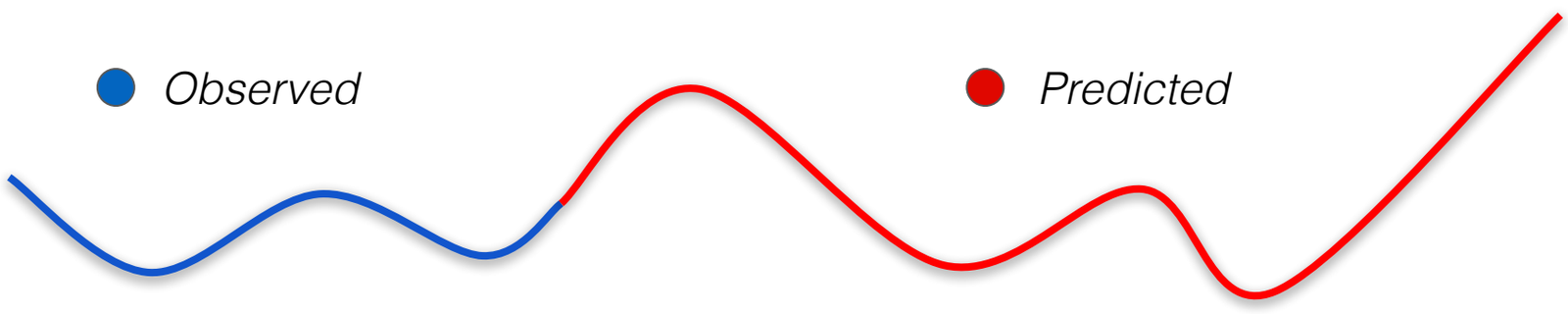
- Standard evolution protocol:
  - An agent is represented by its (x, y) coordinates

@Train time



E.g. 20 time-steps (8 sec) in ETH-UCY

@Test time

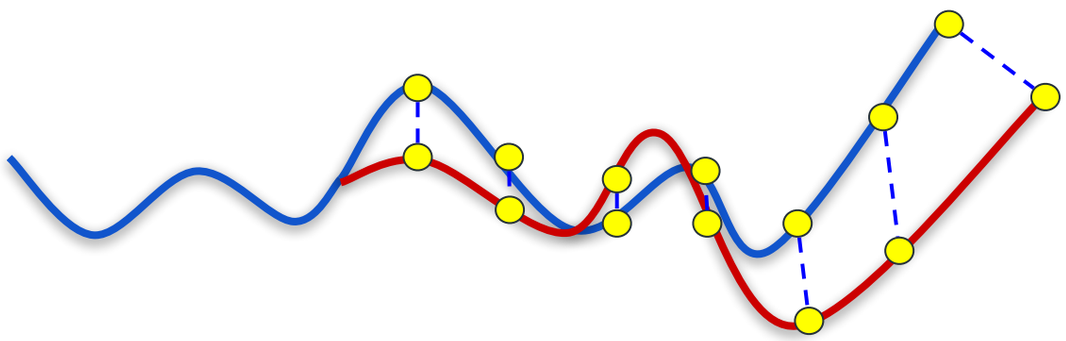


8 time-steps (~3.2 sec)

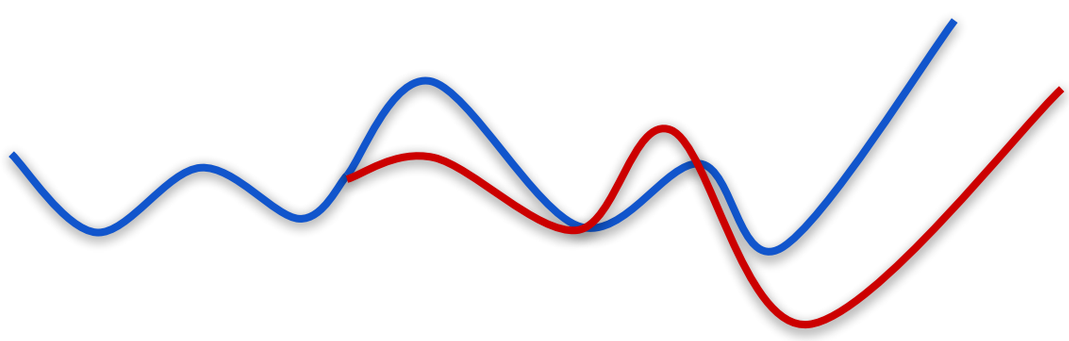
12 time-steps (~4.8 sec)

Goal: motion (*trajectory*) prediction

- Standard evolution protocol:
  - Given a predicted trajectory, we evaluate the quality of the prediction measuring the error w.r.t. the ground truth
  - Several Metrics: ADE, FDE (but also NLL)



Average Displacement Error (ADE)



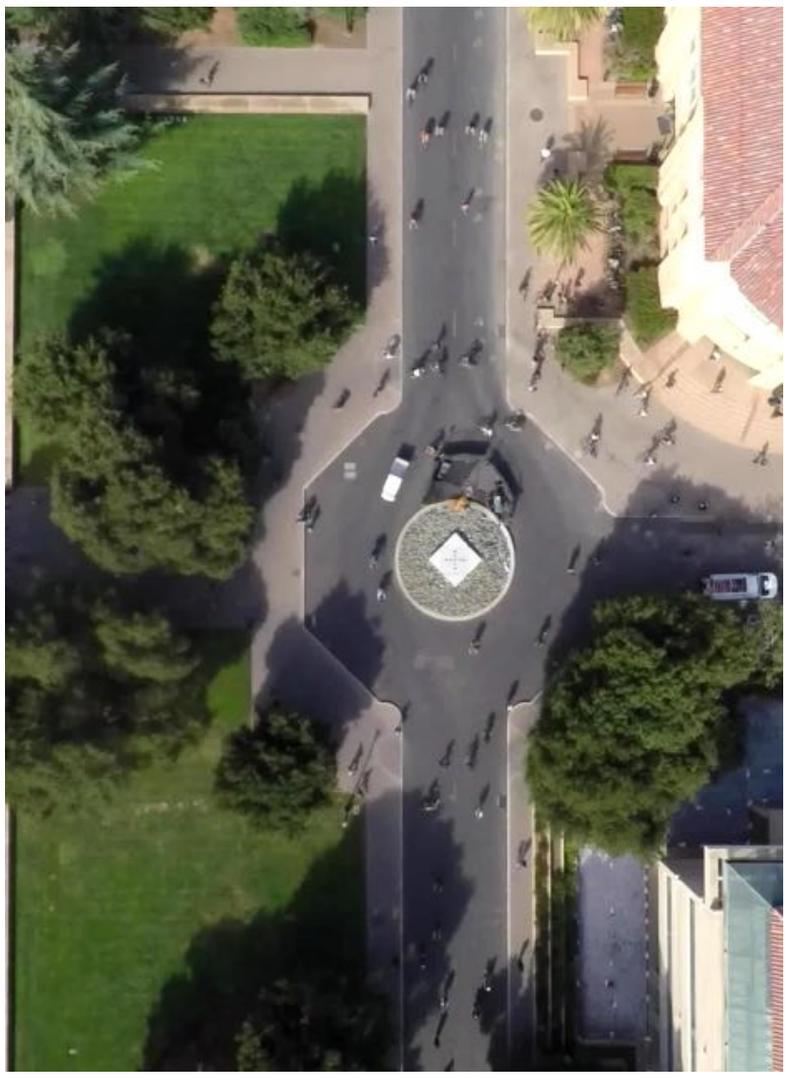
Final Displacement Error (FDE)

Negative Log-Likelihood (NLL): it measures the fit of a ground-truth sample to the predicted distribution

Motivation

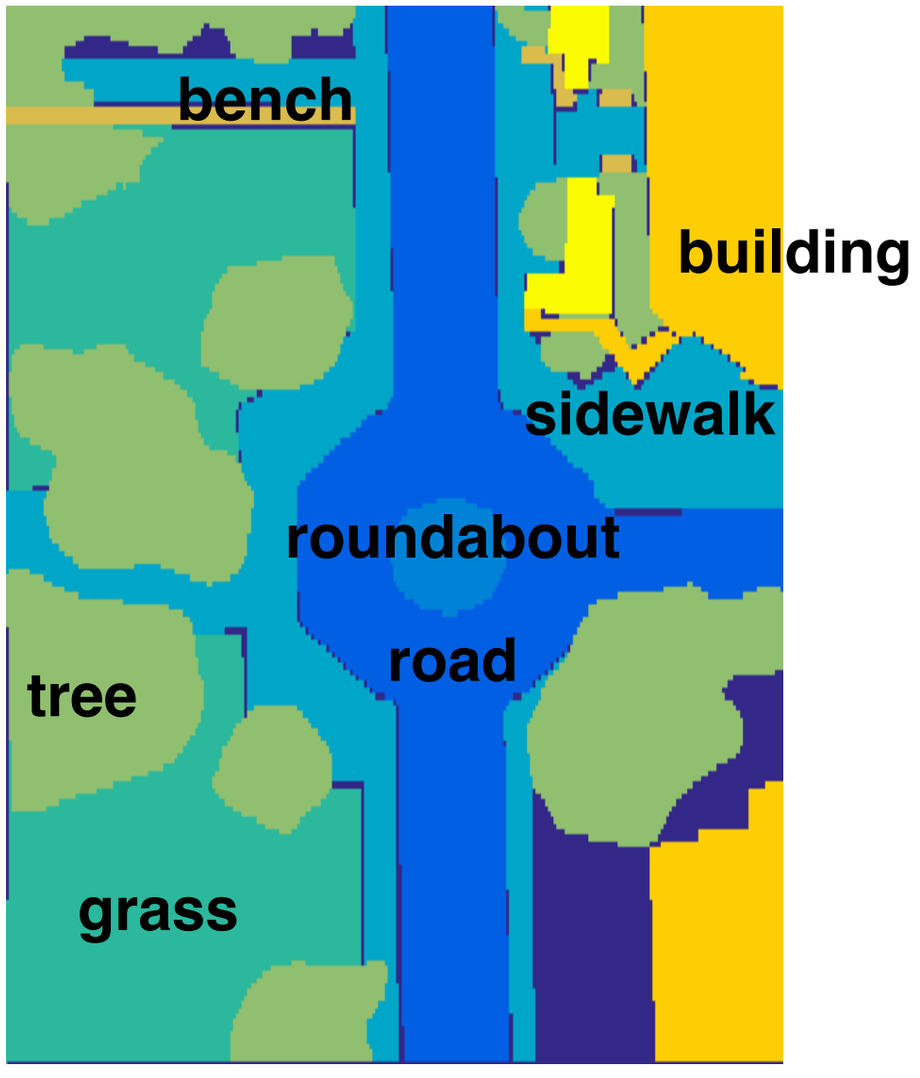
- We believe this ability is mostly driven by two factors

  - the *dynamics* of active agents



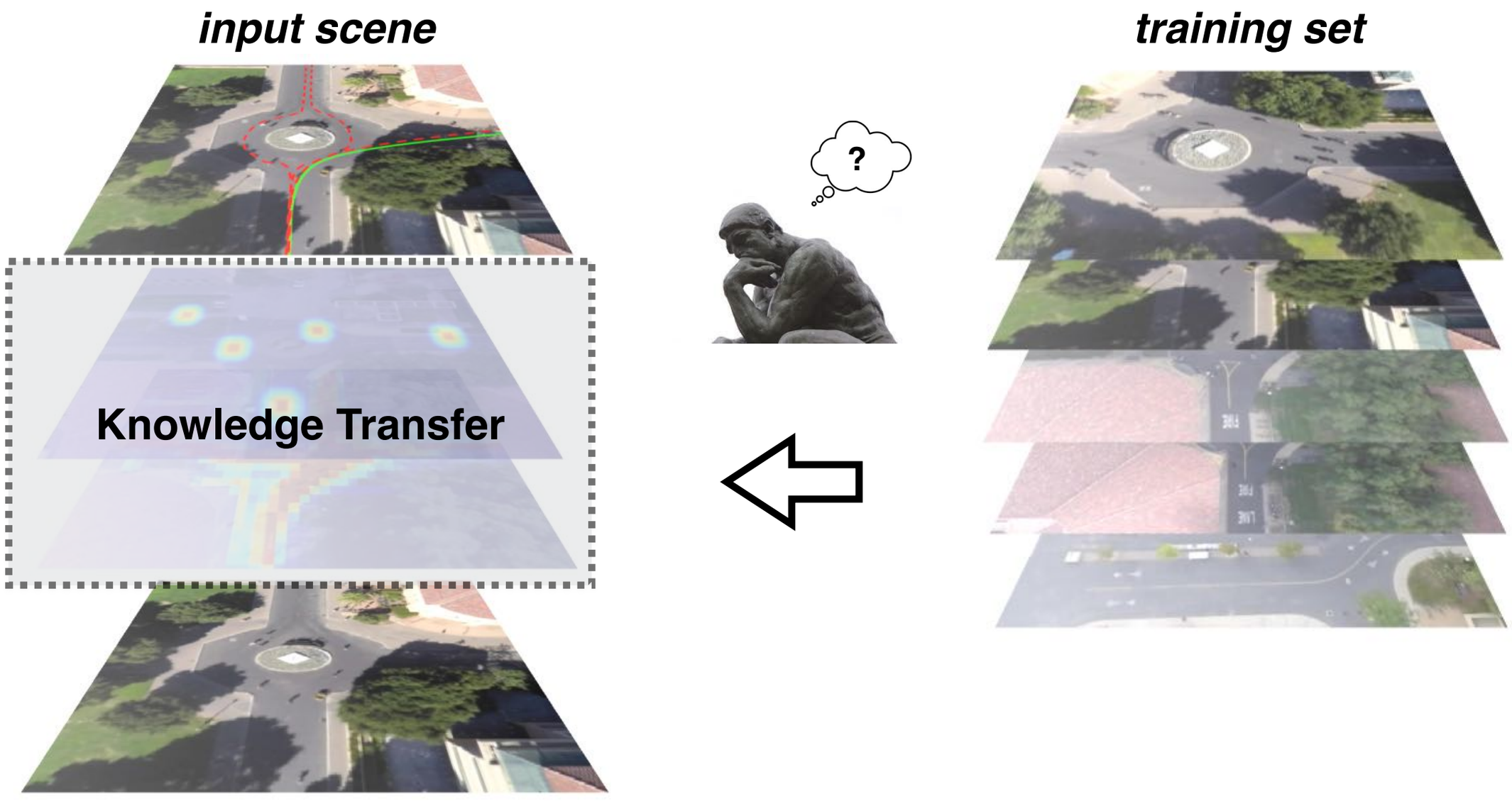
Motivation

- We believe this ability is mostly driven by two factors
  - the *dynamics* of active agents
  - the *semantic of the scene*



Approach

- This knowledge can be transferred to a new scene

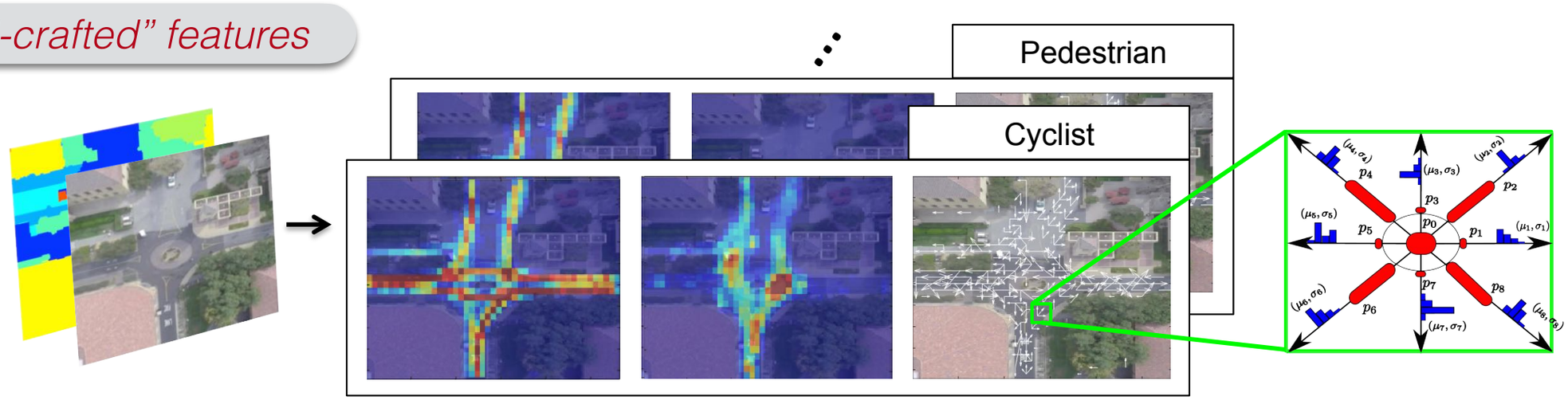


Approach

- Given an input scene we build a *"navigation map"* $\mathbf{M}$ which collects the navigation statistics

- In our early works, for each patch in $\mathbf{M}$ we collect:
  - Popularity score, Routing score, Histogram of Directions and Histogram of Speeds

"Hand-crafted" features

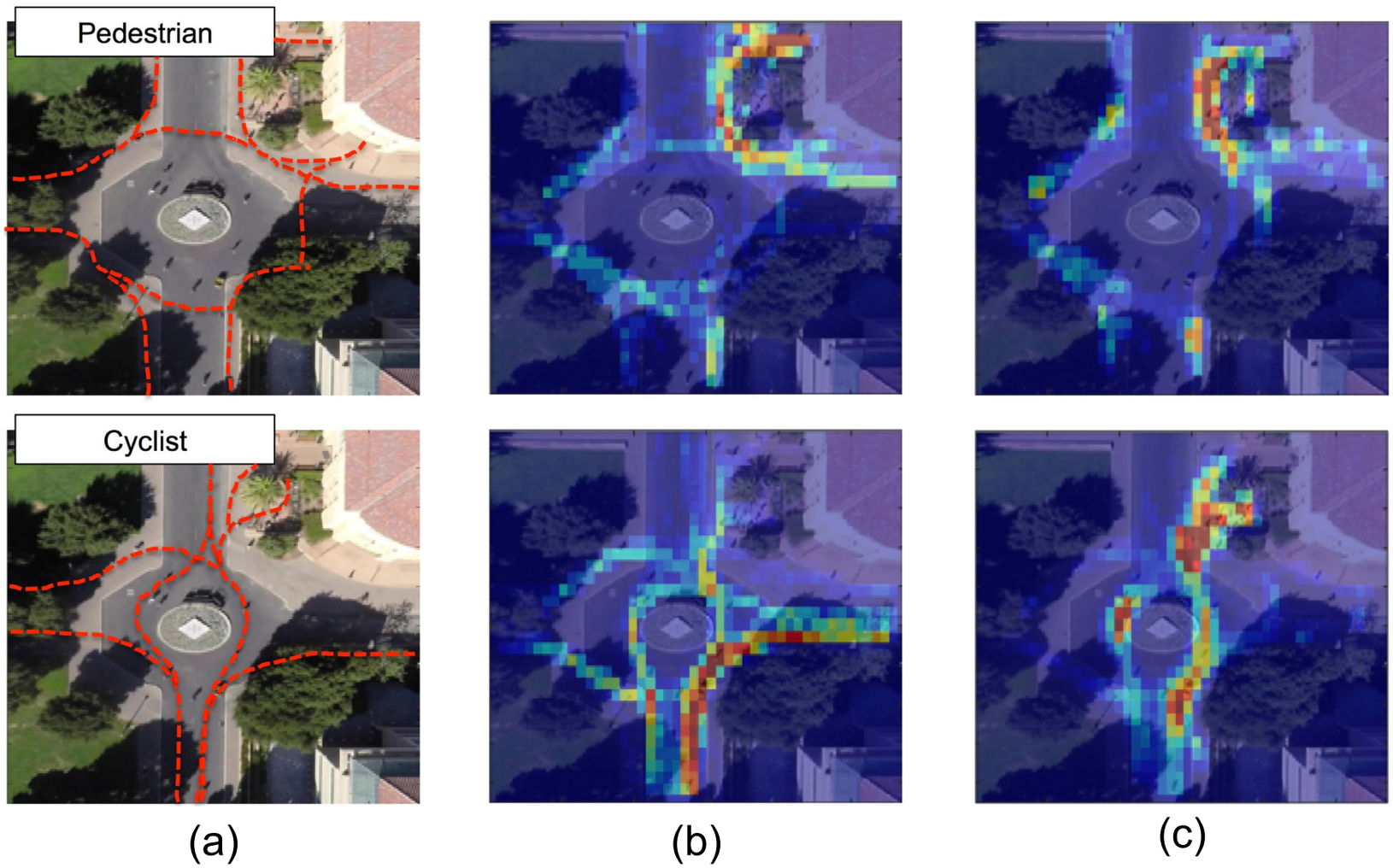


(a) Input scene (b) "Navigation map" (c) *HoD* and *HoS*

[**Ballan**, Castaldo, Alahi, Palmieri, Savarese - ECCV'16]

Prediction model

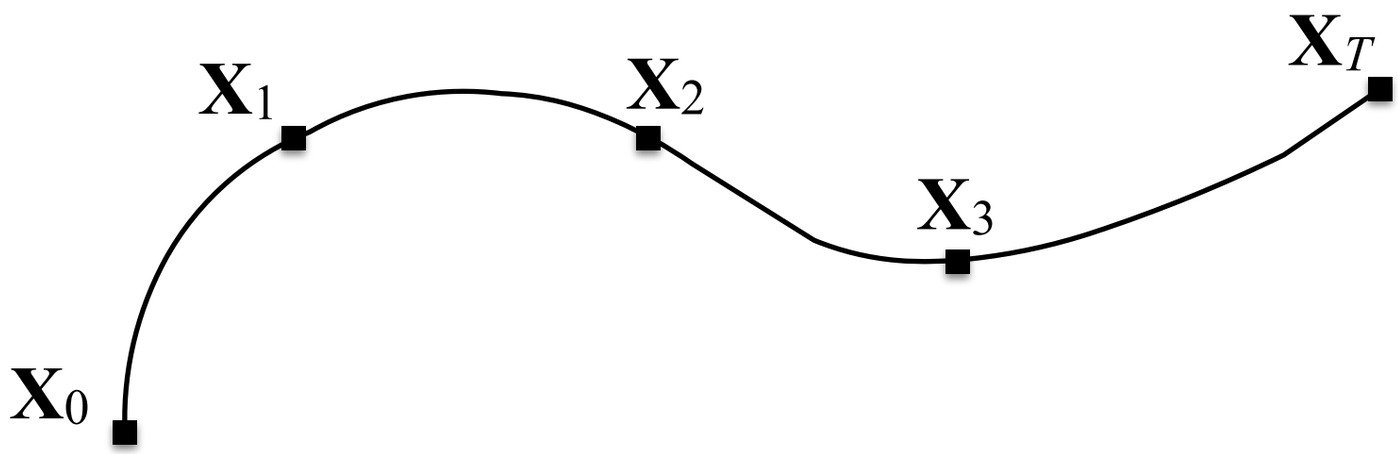
- Navigation Map: qualitative examples



Column (a) visualizes the most common paths for both classes.
Columns (b, c) show the corresponding popularity and routing maps.

Prediction model

- The target state variable is defined as $\mathbf{X}_k = (\mathbf{P}_k, \mathbf{V}_k)^T$
  - $\mathbf{P}_k = (X_k, Y_k)^T$ (position) and $\mathbf{V}_k = (\Omega_k, \Theta_k)^T$ (velocity)
- The target interacts with the map $\mathbf{M}$ by exploiting the navigation values for the patch he is occupying
- Given an initial state $\mathbf{X}_0$, our goal is to generate a sequence of future states $\mathbf{X}_1, \ldots, \mathbf{X}_T$ *(i.e., a trajectory)*



Prediction model

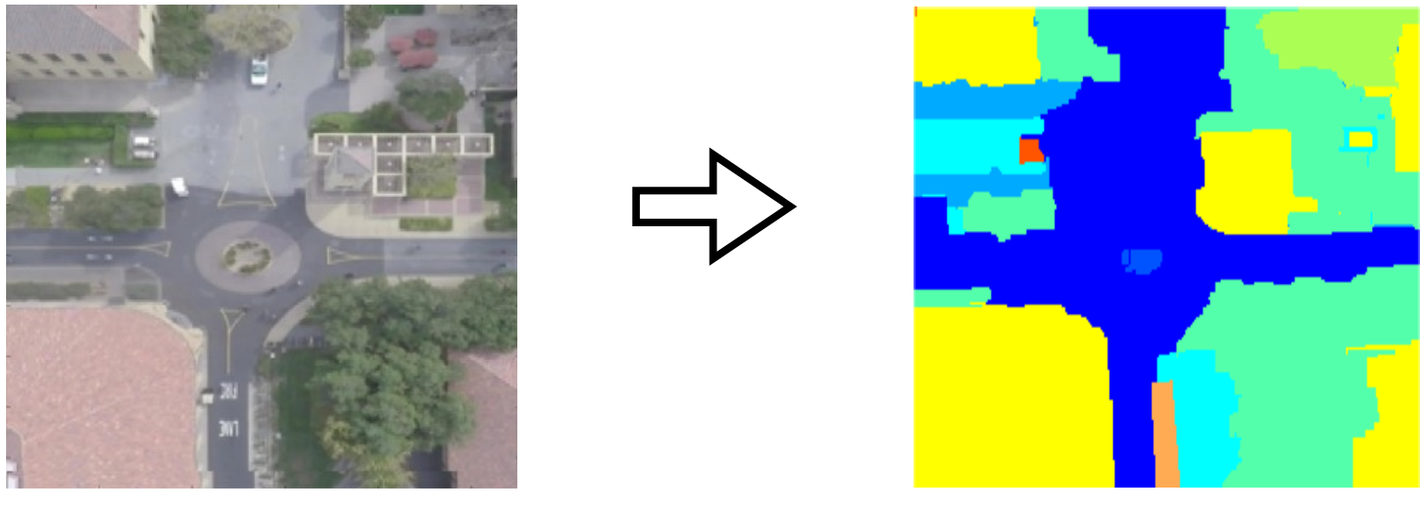
- The dynamic process describing the target motion is defined by:
  - $\mathbf{P}_{k+1} = \mathbf{P}_k + (\Omega_k cos\Theta_k, \Omega_k sin\Theta_k)' + \mathbf{w}_k$ (constant velocity)
  - $\mathbf{V}_{k+1} = \Phi(\mathbf{P}_k, \mathbf{V}_k; \mathbf{M})$
- The learned expected values in $\mathbf{M}$ allows our model to generate non-linear behaviors
- $\Phi(\cdot)$ is defined in probabilistic terms by means of a Dynamic Bayesian Network (DBN)

See also: [Coscia, Castaldo, Palmieri, Alahi, Savarese, **Ballan** - IVC'18]
(focus on long-term trajectory prediction) ***IVC Best Paper Award 2021***

Knowledge transfer

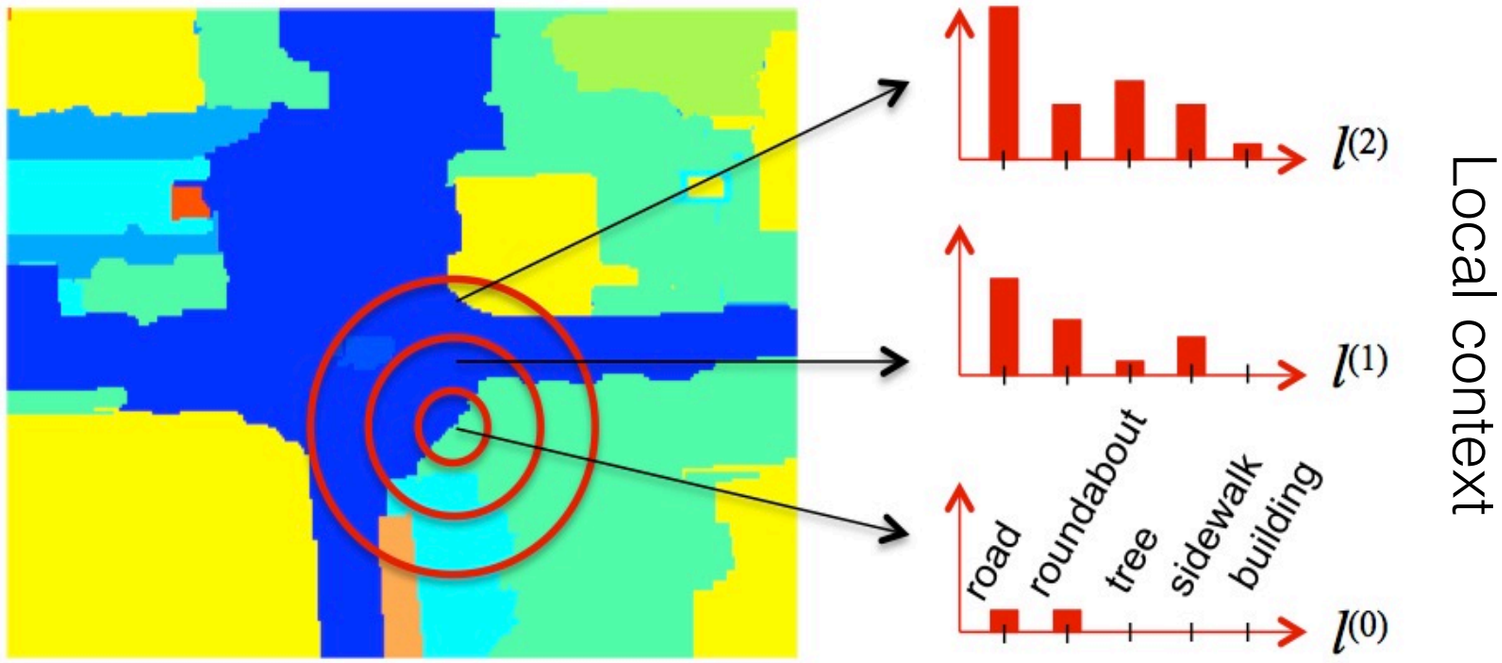
"The elements of the scene define a semantic context, and they might determine similar behaviors in scenes characterized by a similar context"

- Our data-driven approach uses scene similarity to transfer the functional properties to a new scene
- Scene parsing: we use a "non-parametric" algorithm (based on local features SIFT+LLC, GIST and MRF inference)



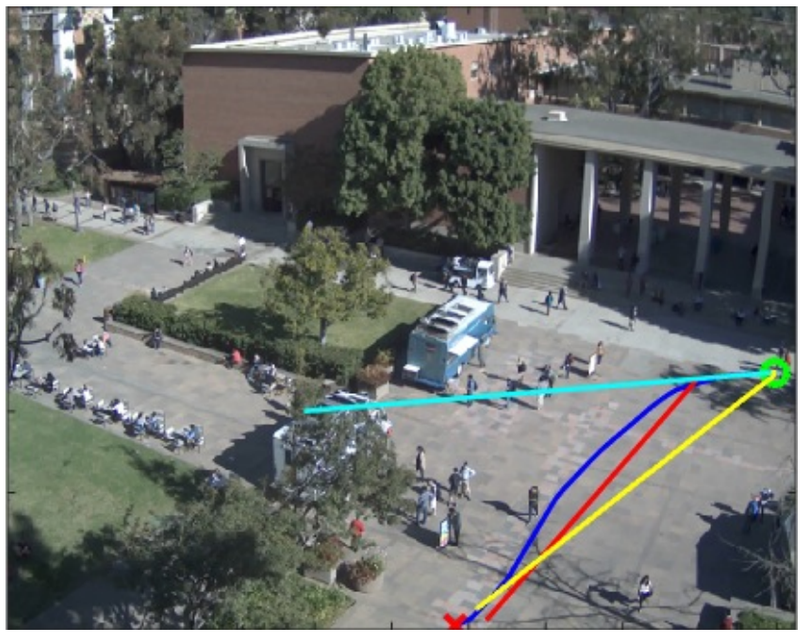
Knowledge transfer

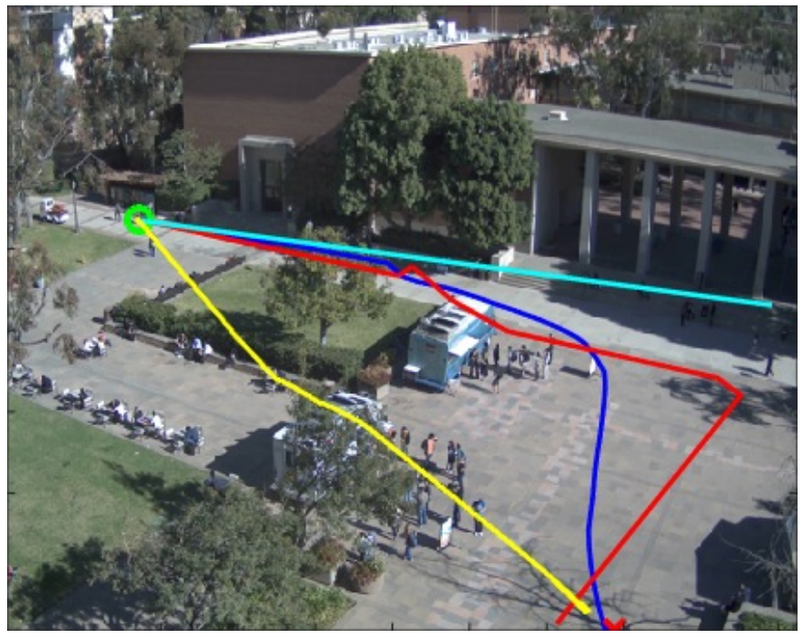
- Context Descriptors: a weighted concatenation of the *global* and *local* semantic context components
  - *global context*: vector of distances between classes
  - *local context*: encodes the spatial configuration of nearby patches at multiple levels

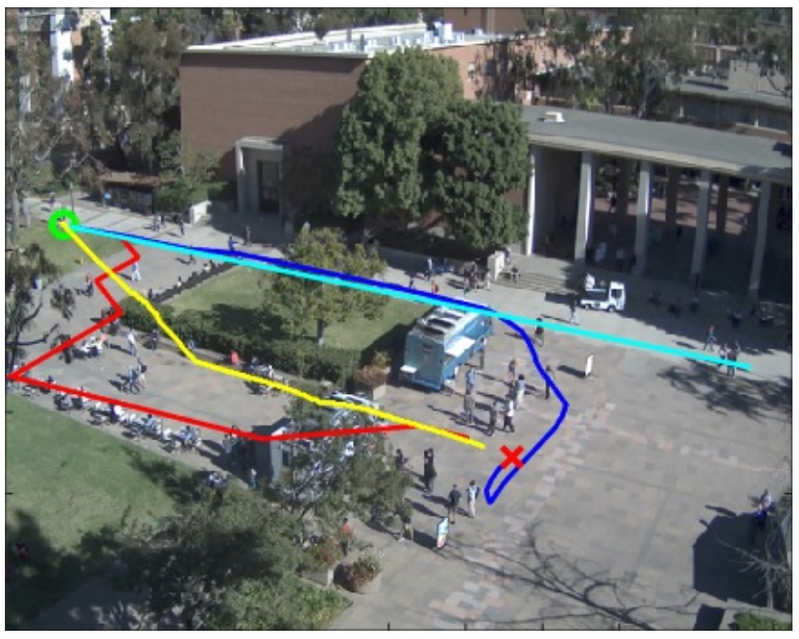


Results: path prediction

- Qualitative examples on the UCLA-courtyard scene (*blue* is ground-truth, *cyan* is LP, *yellow* is IOC, *red* is ours)



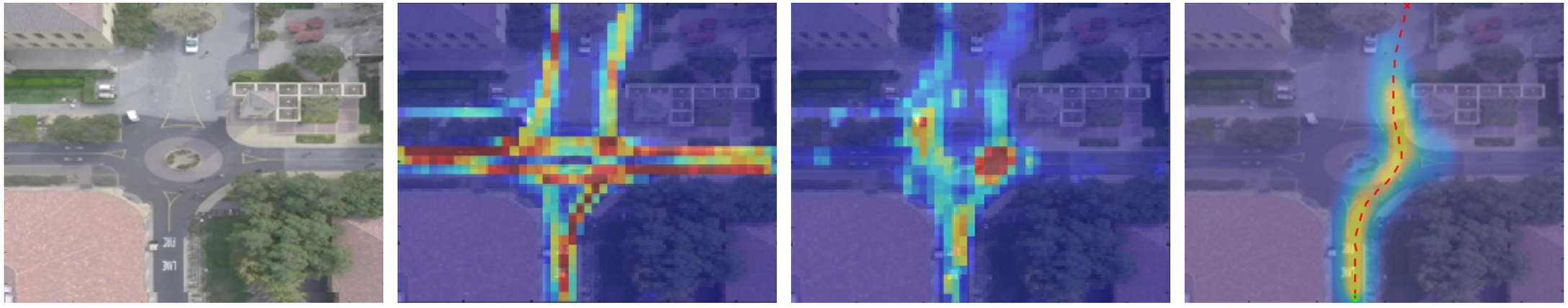




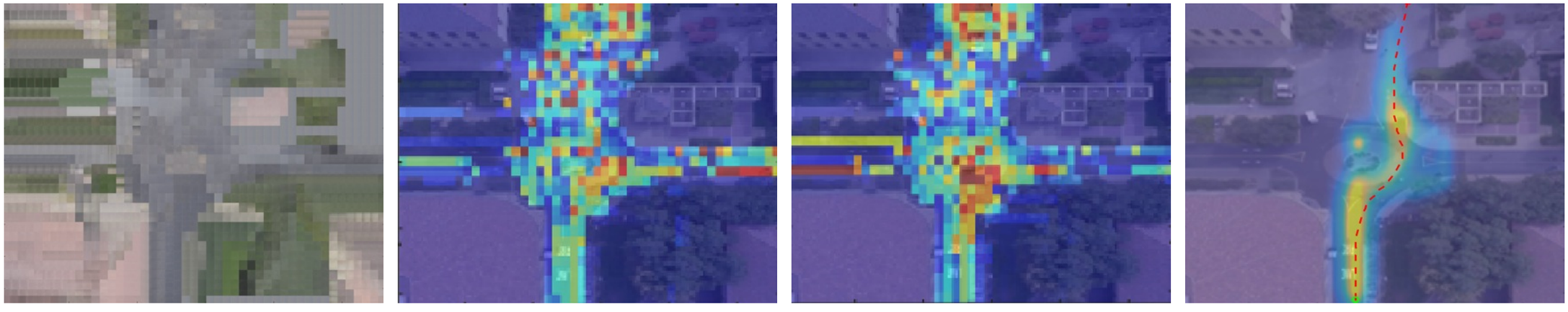
Results: knowledge transfer

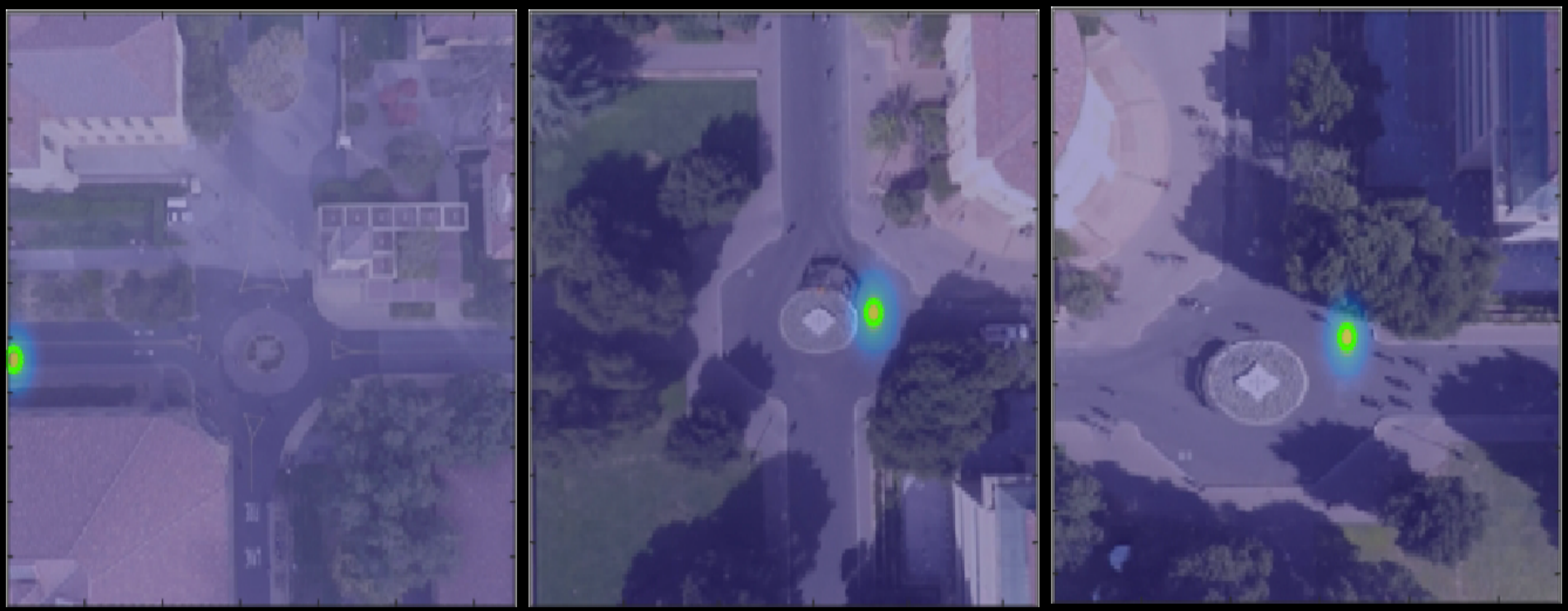
- Qualitative examples:
  - (1) path forecasting *vs* (2) knowledge transfer

(1)



(2)





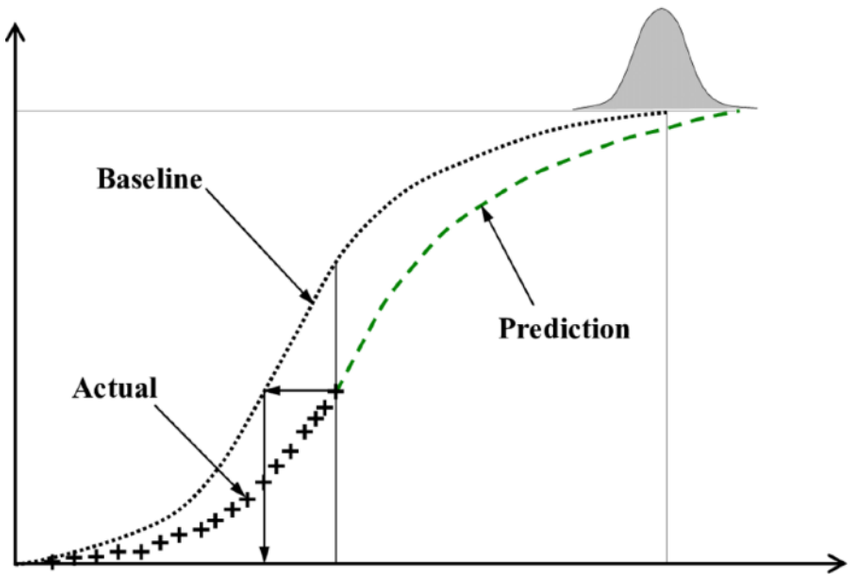
starting point

heatmap

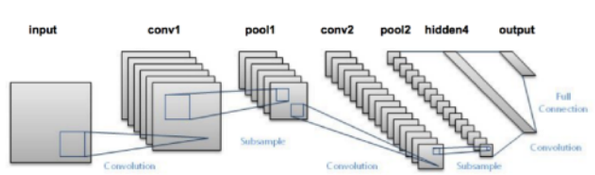
path prediction

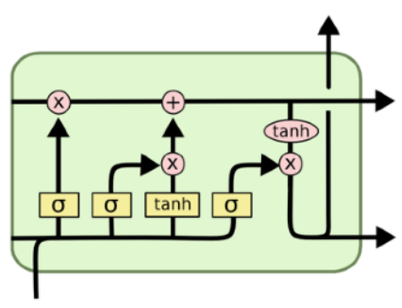
Trajectory prediction

- A broad range of approaches:



Based on Bayesian models / Kalman-based Filtering





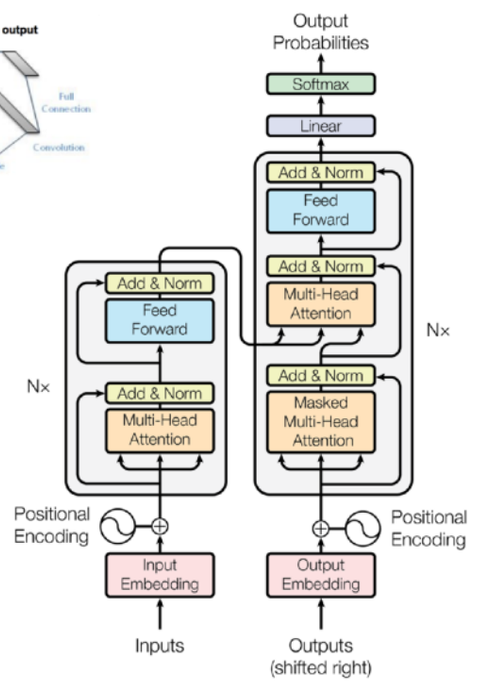


Figure 1: The Transformer - model architecture.

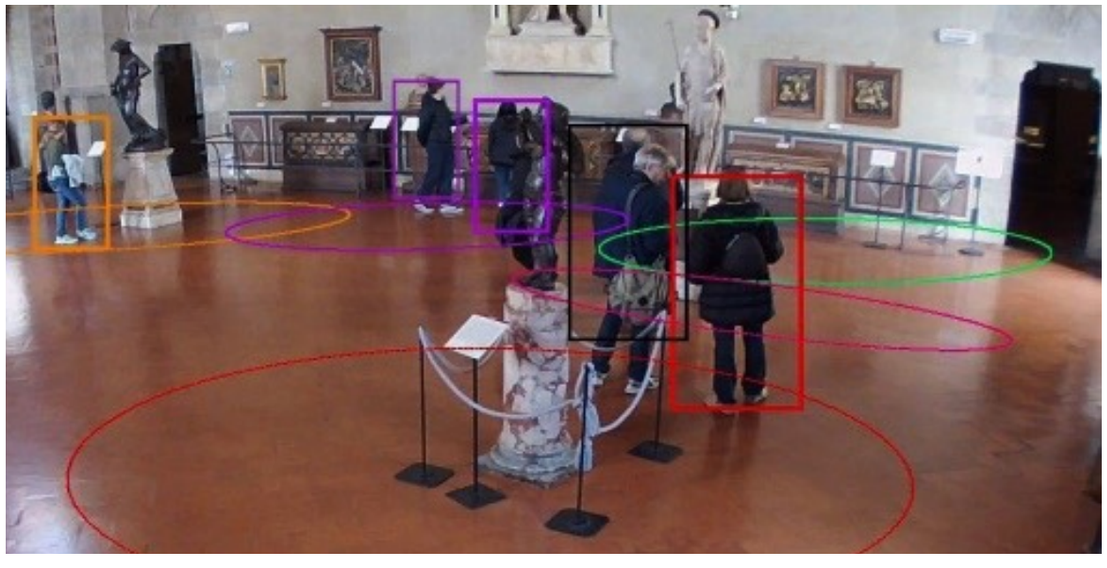
Based on Deep-NN Architectures

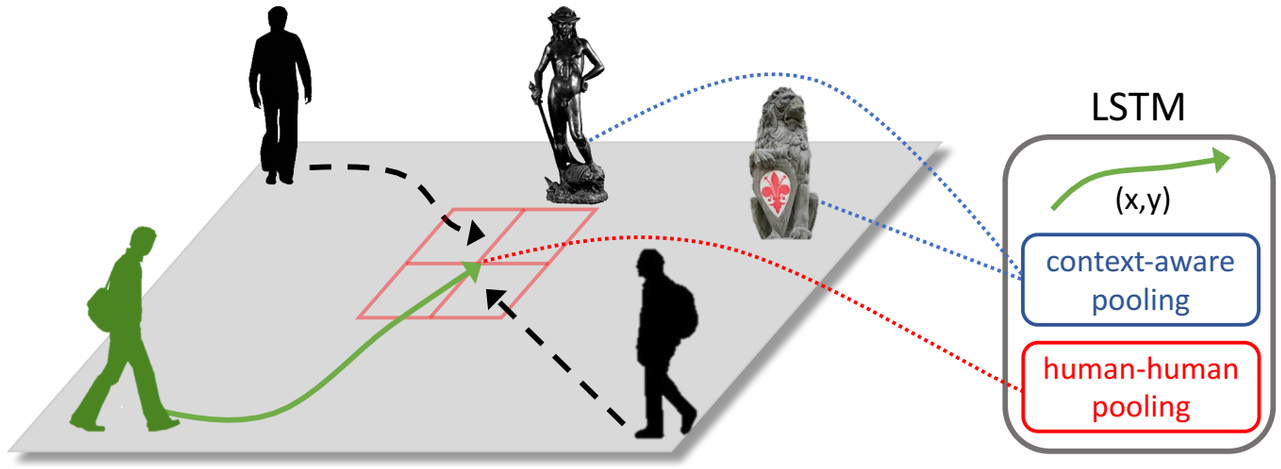
"Traditional approaches"

Current paradigm

Context-aware LSTM

- Trajectories are modelled using RNNs/LSTMs
- A context-aware pooling layer combining:
  - Human-human interactions
  - Human-space interactions

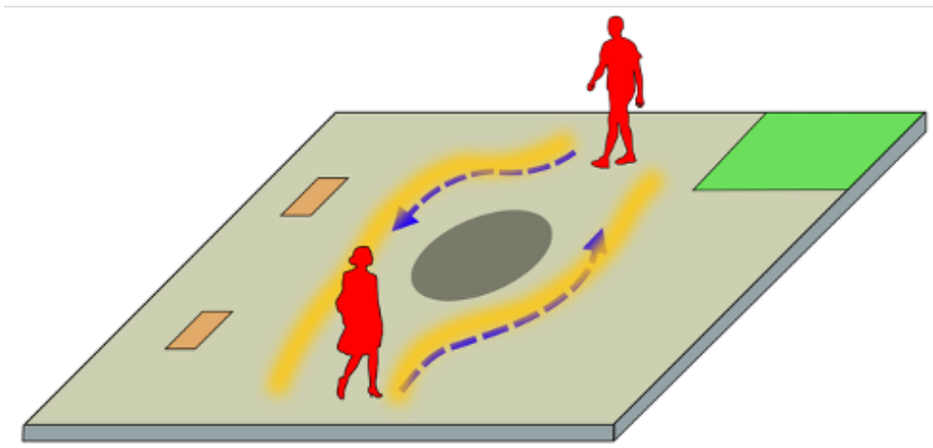




[Bartoli, Lisanti, **Ballan**, Del Bimbo - ICPR'18]

Context-aware LSTM

- Trajectories are modelled using RNNs/LSTMs
- A context-aware pooling layer combining:
  - Human-human interactions (*social* pooling)
    (+ *navigation* pooling)
  - Human-space interactions (*semantic* pooling)

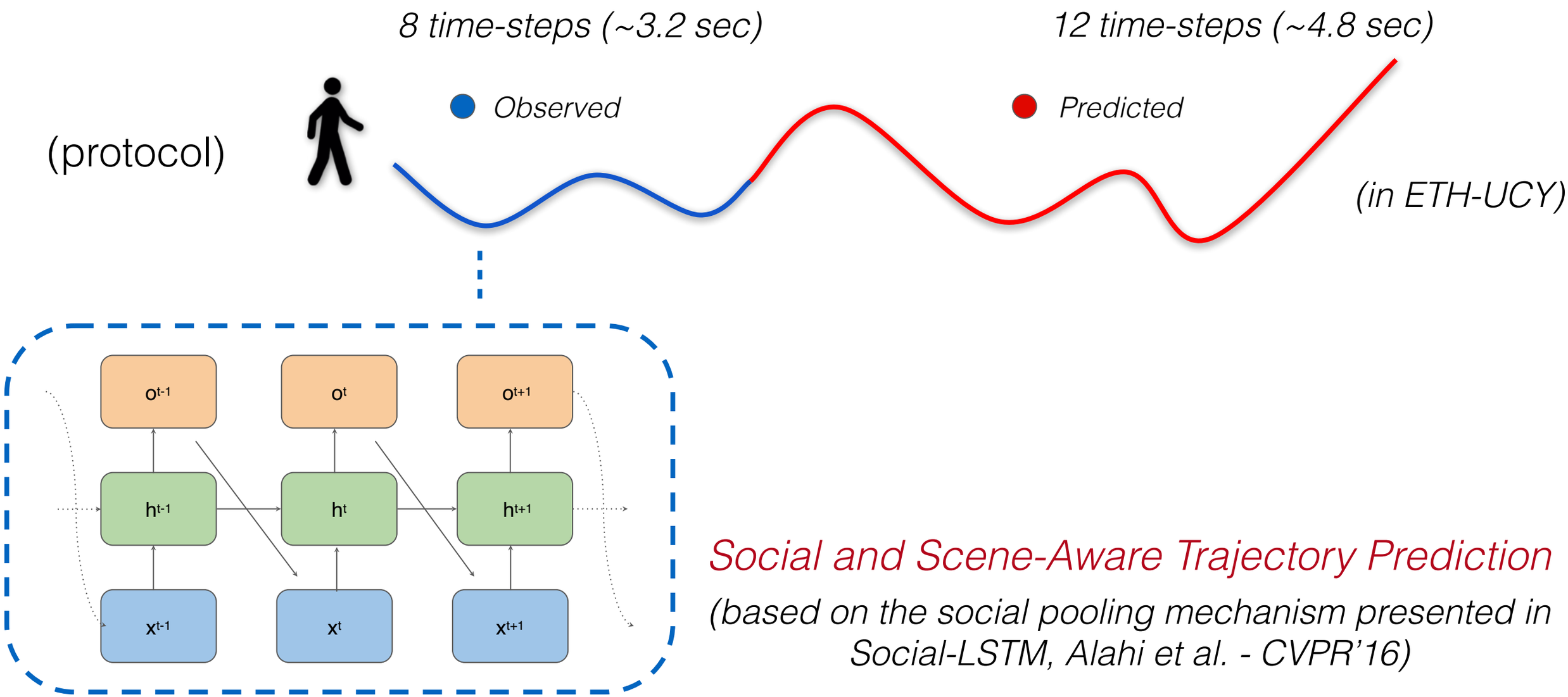


Social and Scene-Aware Trajectory Prediction

(based on the social pooling mechanism presented in Social-LSTM, Alahi et al. - CVPR'16)

Context-aware LSTM

- Trajectories are modelled using RNNs/LSTMs

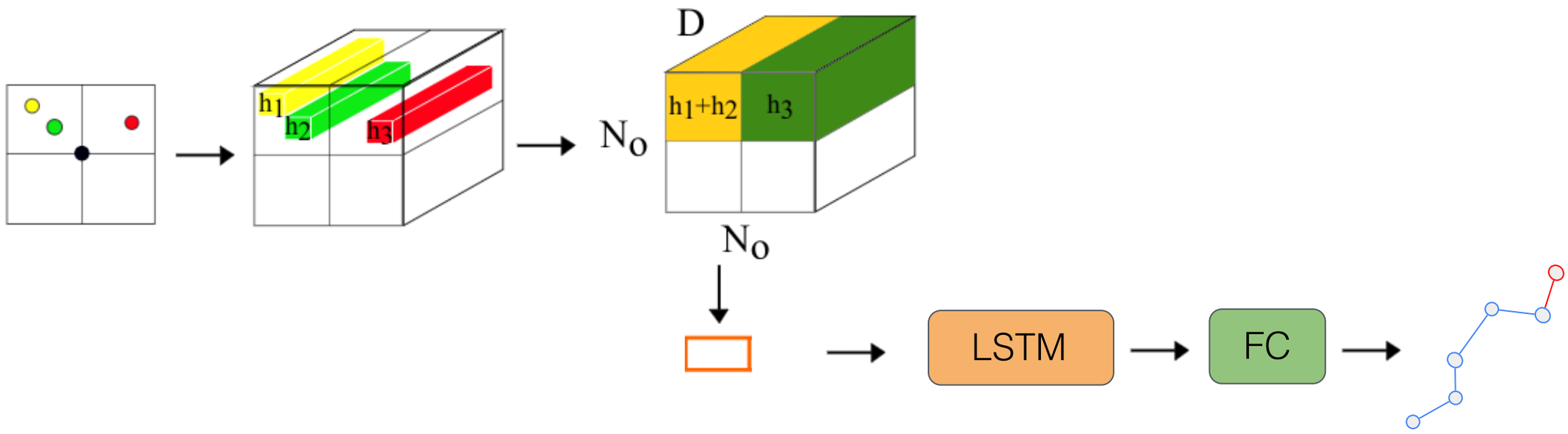


Social and Scene-Aware Trajectory Prediction

(based on the social pooling mechanism presented in Social-LSTM, Alahi et al. - CVPR'16)

Context-aware LSTM

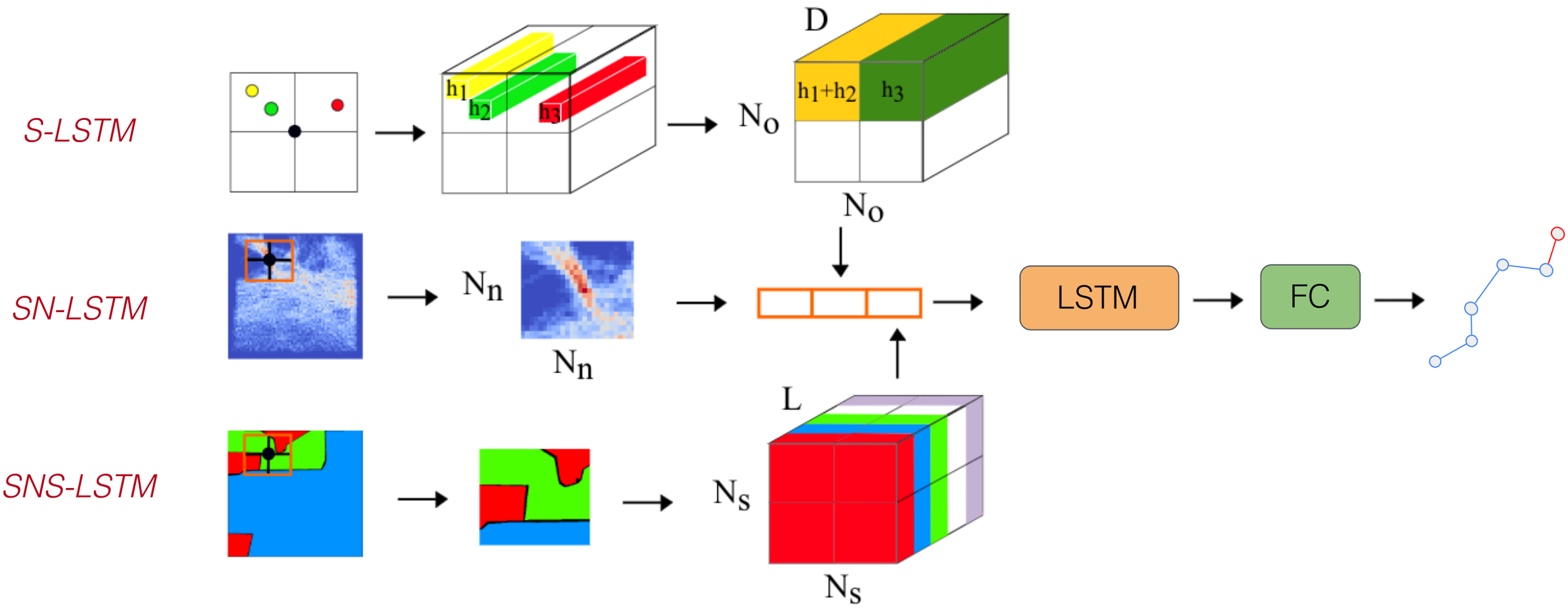
- Social and scene-aware trajectory prediction:



[Lisotto, Coscia, **Ballan** - ICCVw'19]

Context-aware LSTM

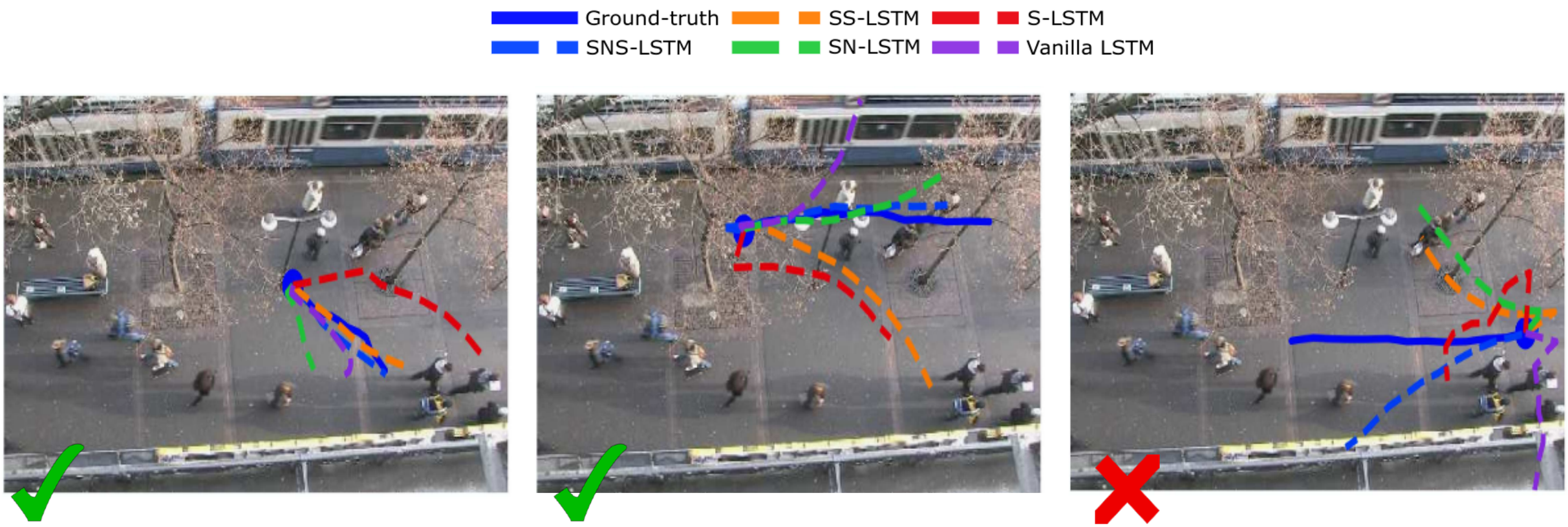
- Social and scene-aware trajectory prediction:



[Lisotto, Coscia, **Ballan** - ICCVw'19]

Experiments

- Qualitative results on ETH/UCY:



[Lisotto, Coscia, **Ballan** - ICCVw'19]

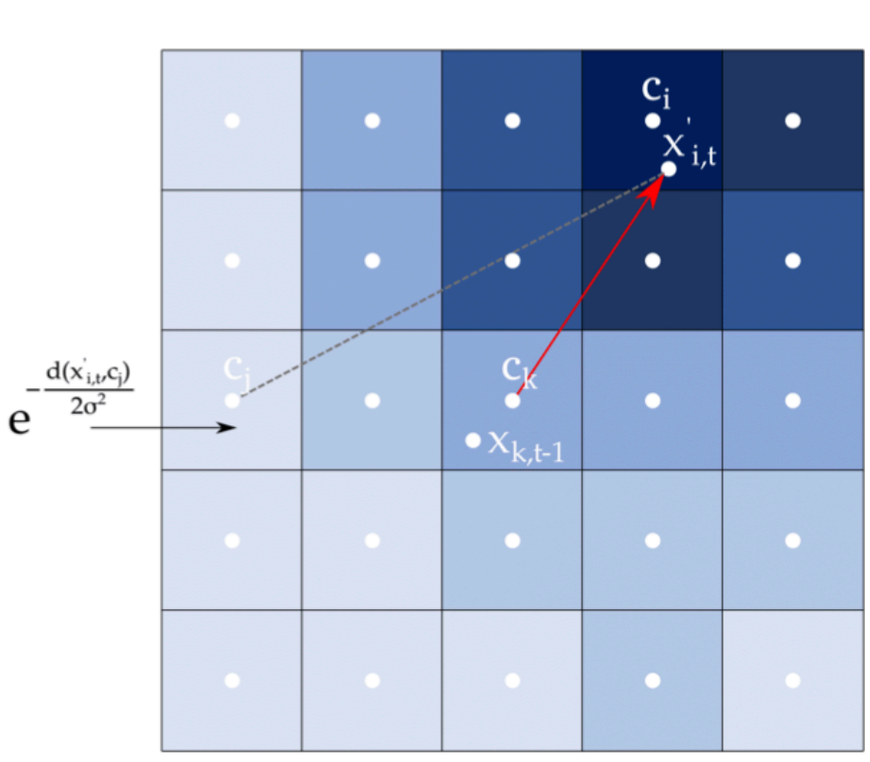
Experiments

- Quantitative results on ETH/UCY:
  - Baseline (backbone): “Vanilla” LSTM

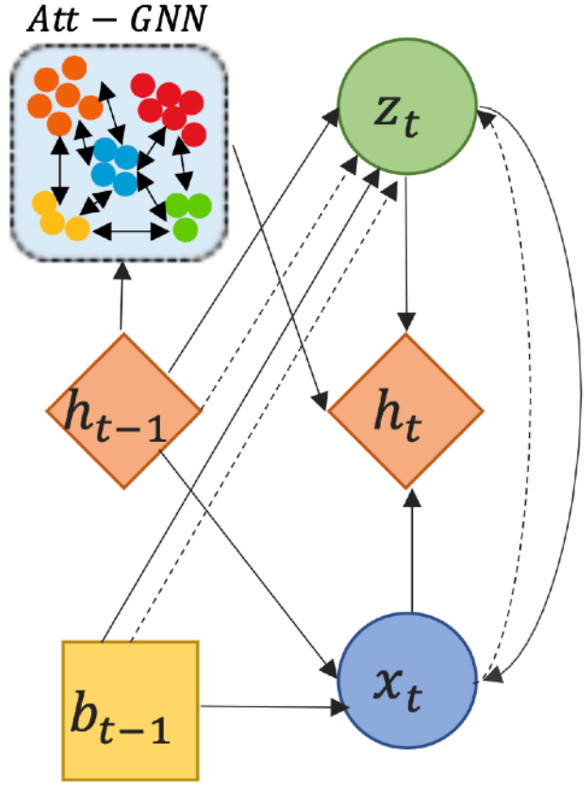
(ADE/FDE)	**ETH**	**HOTEL**	**UNIV**	**ZARA1**	**ZARA2**	**AVG**
“Vanilla” LSTM	0.52/2.84	0.33/1.90	0.52/2.92	0.41/2.35	0.27/1.48	0.41/2.30
S-LSTM	0.51/2.82	0.31/1.67	0.55/3.04	0.36/2.05	**0.25**/**1.42**	0.40/2.20
SN-LSTM (ours)	**0.47**/2.55	0.44/2.25	0.39/2.10	0.29/1.56	0.28/1.59	0.37/2.01
SS-LSTM (ours)	0.48/2.57	**0.24**/**1.38**	0.43/2.54	0.33/1.81	0.31/1.63	**0.36**/1.99
SNS-LSTM (ours)	0.58/**2.43**	0.30/1.58	**0.37**/**2.08**	**0.28**/**1.53**	0.26/1.44	**0.36**/**1.81**

Attentive Conditional-VRNN

- A recurrent variational autoencoder is conditioned on prior belief maps
  - The RNN hidden state is refined with an attention module
  - At inference time, it generates future displacements

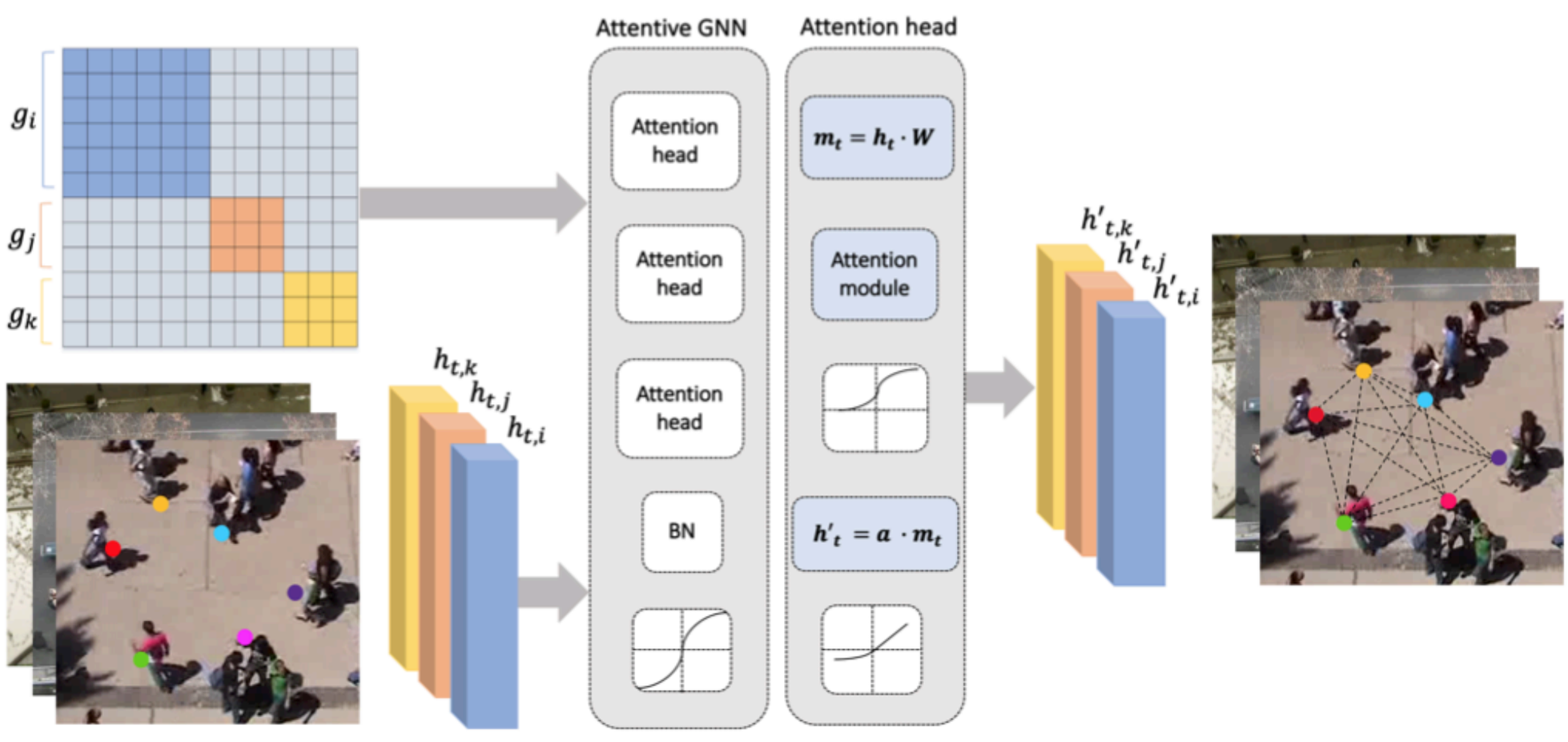


Belief Map



Attentive Conditional-VRNN

- Our model handles human interactions using an attentive hidden state refinement of our RNN
  - It's done by a graph NN based on an attention mechanism to learn relative weights between connected nodes



[Bertugli, Calderara, Coscia, **Ballan**, Cucchiara - CVIU'21]

Experiments

- Quantitative results on ETH/UCY:

(ADE/FDE)	**ETH**	**HOTEL**	**UNIV**	**ZARA1**	**ZARA2**	**AVG**
S-LSTM CVPR'16	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
S-GAN CVPR'18	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18
Trajectron ICCV'19	0.59/1.14	0.35/0.66	0.54/1.13	0.43/0.83	0.43/0.85	0.56/1.14
SoPhie CVPR'19	0.70/1.43	0.76/1.67	0.30/0.63	0.38/0.78	0.54/1.24	0.54/1.15
STGAT ICCV'19	0.78/1.60	0.30/0.54	0.51/1.08	0.33/0.72	0.29/0.63	0.44/0.91
AC-VRNN (ours)	0.61/**1.09**	**0.30**/**0.55**	0.58/1.22	0.34/**0.68**	0.28/**0.59**	**0.42**/**0.83**

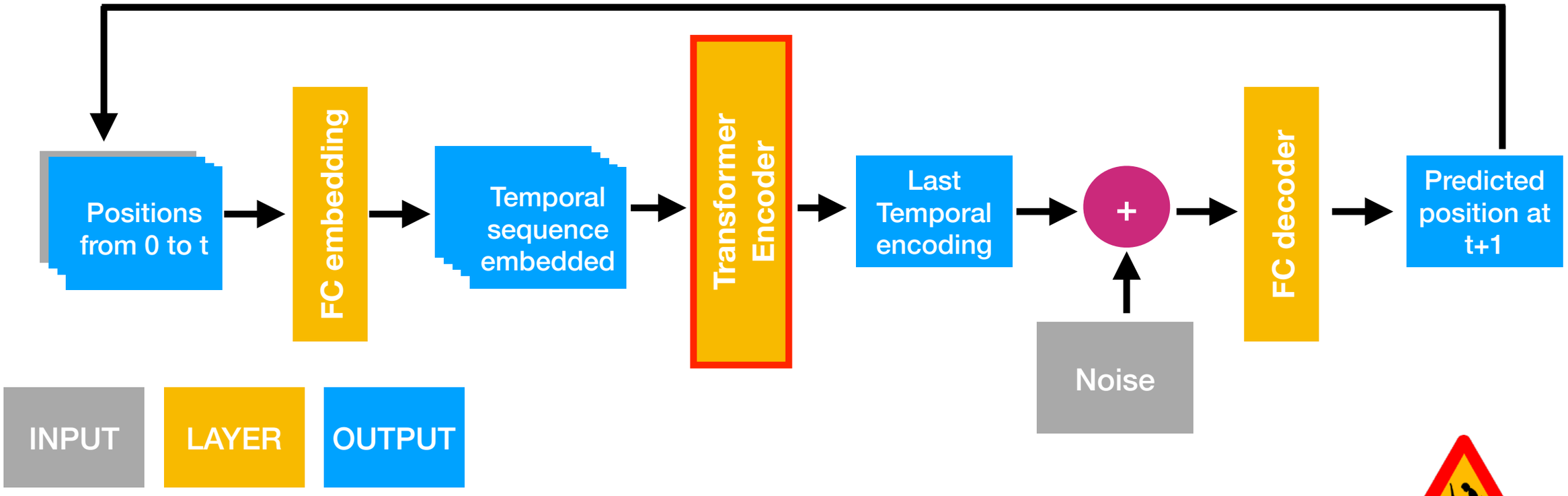
Experiments

- Quantitative results on ETH/UCY, SDD and IND:

(ADE/FDE)	**ETH**	**HOTEL**	**UNIV**	**ZARA1**	**ZARA2**	**AVG**	**SDD**	**IND**
S-LSTM CVPR'16	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54	-	-
S-GAN CVPR'18	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18	0.65/1.26	0.48/0.99
Trajectron ICCV'19	0.59/1.14	0.35/0.66	0.54/1.13	0.43/0.83	0.43/0.85	0.56/1.14	-	-
SoPhie CVPR'19	0.70/1.43	0.76/1.67	0.30/0.63	0.38/0.78	0.54/1.24	0.54/1.15	-	-
STGAT ICCV'19	0.78/1.60	0.30/0.54	0.51/1.08	0.33/0.72	0.29/0.63	0.44/0.91	0.57/1.09	0.48/1.00
AC-VRNN (ours)	0.61/**1.09**	**0.30/0.55**	0.58/1.22	0.34/**0.68**	0.28/**0.59**	**0.42/0.83**	**0.51/0.90**	**0.42/0.80**

Transformers and Self-Attention

- Recent approaches build on top of transformers
  - We are working on a Contextual Transformer for Trajectory Prediction (CONTRA) model
    - Several components: semantic, navigation, goal, …
  - Our “Vanilla transformer”:



[Coscia, Chiara, **Ballan** - *ongoing*]

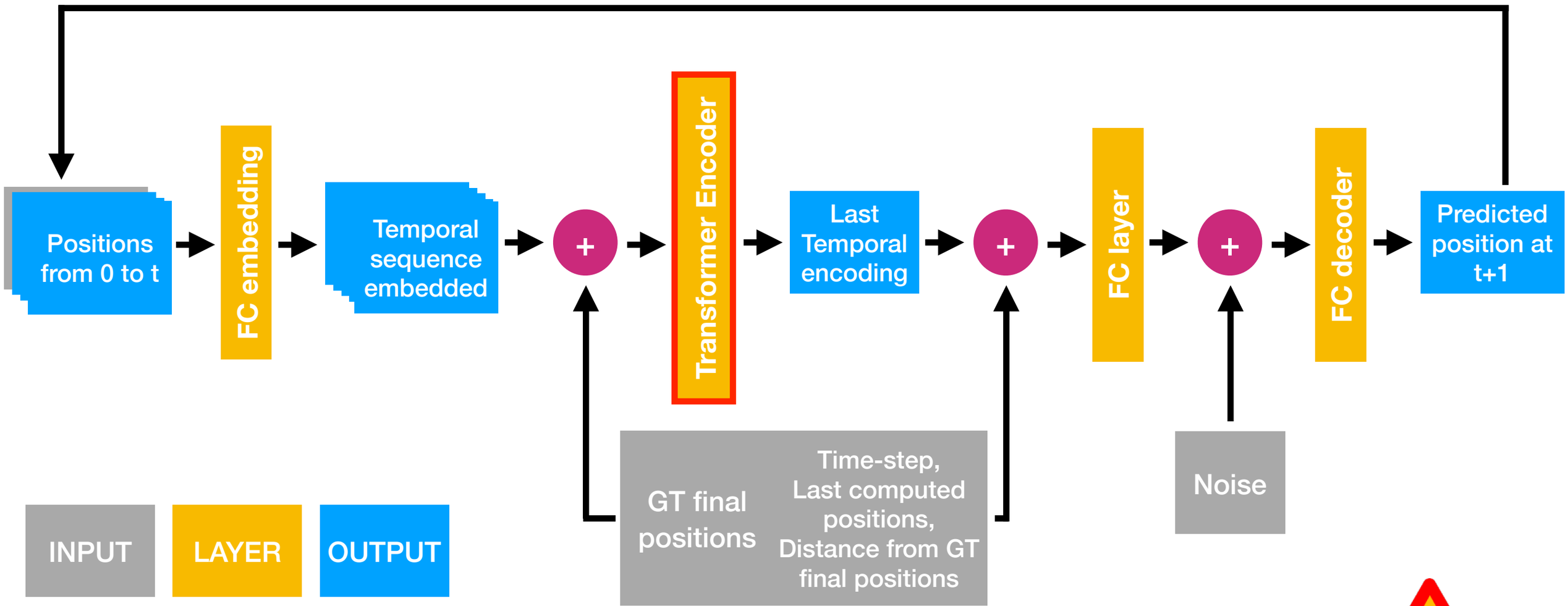
Experiments

- Quantitative results on ETH/UCY, SDD and IND:

(ADE/FDE)	**ETH**	**HOTEL**	**UNIV**	**ZARA1**	**ZARA2**	**AVG**	**SDD**	**IND**
S-GAN CVPR'18	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18	0.65/1.26	0.48/0.99
Trajectron ICCV'19	0.59/1.14	0.35/0.66	0.54/1.13	0.43/0.83	0.43/0.85	0.56/1.14	-	-
SoPhie CVPR'19	0.70/1.43	0.76/1.67	0.30/0.63	0.38/0.78	0.54/1.24	0.54/1.15	-	-
STGAT ICCV'19	0.78/1.60	0.30/0.54	0.51/1.08	0.33/0.72	0.29/0.63	0.44/0.91	0.57/1.09	0.48/1.00
AC-VRNN (ours)	0.61/1.09	0.30/0.55	0.58/1.22	0.34/0.68	0.28/0.59	0.42/0.83	0.51/0.90	0.42/0.80
STAR ECCV'20	0.36/**0.65**	0.17/0.36	**0.26/0.55**	**0.22/0.46**	0.31/0.62	0.26/0.53	**0.43**/0.85	**0.38/0.80**
CONTRA (vanilla)	**0.33**/0.66	**0.14/0.29**	0.33/0.68	0.24/0.50	**0.20/0.42**	**0.25/0.51**	0.44/**0.82**	0.40/0.86

Transformers and Self-Attention

- Conditioned (GT) goal transformer:
  - We tried several architectures/combinations based on a variety of early or late fusion



[Coscia, Chiara, **Ballan** - *ongoing*]

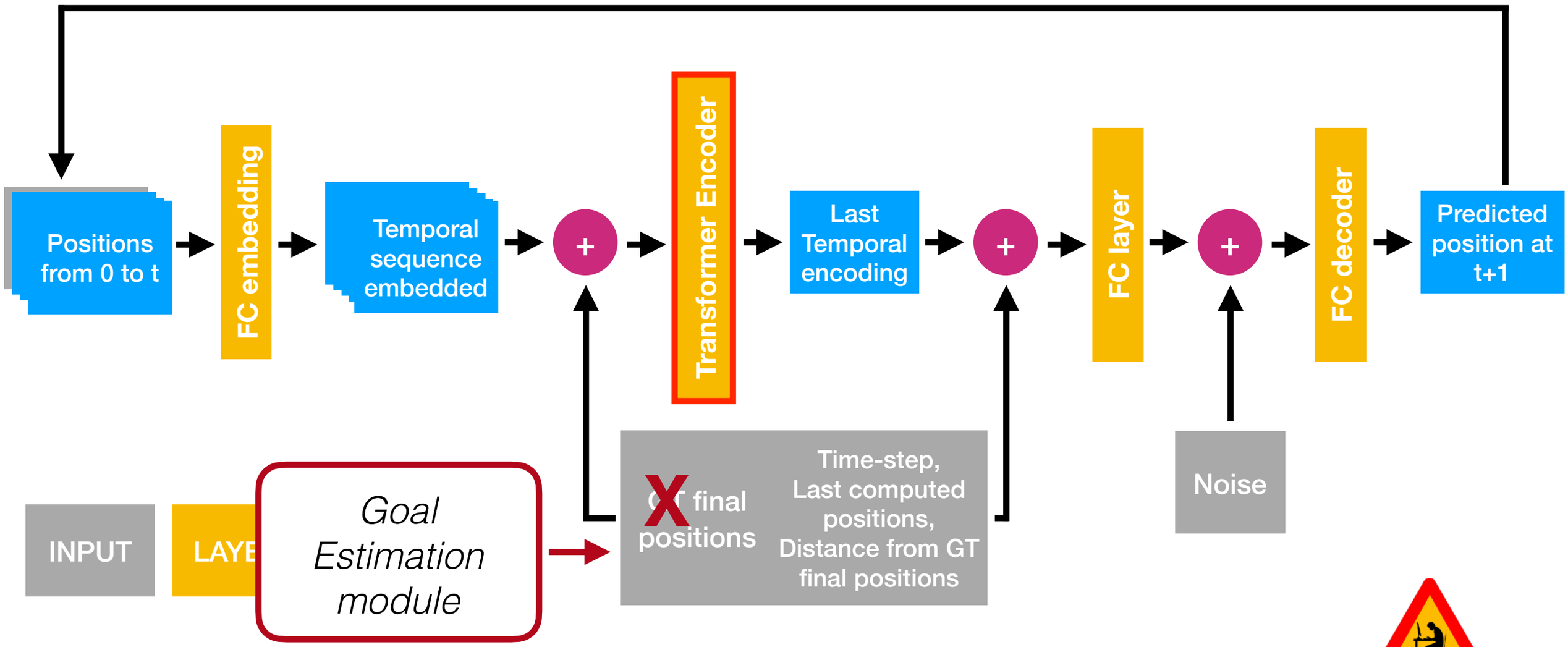
Experiments

- Quantitative results on ETH/UCY, SDD and IND:

(ADE/FDE)	**ETH**	**HOTEL**	**UNIV**	**ZARA1**	**ZARA2**	**AVG**	**SDD**	**IND**
S-GAN CVPR'18	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18	0.65/1.26	0.48/0.99
Trajectron ICCV'19	0.59/1.14	0.35/0.66	0.54/1.13	0.43/0.83	0.43/0.85	0.56/1.14	-	-
SoPhie CVPR'19	0.70/1.43	0.76/1.67	0.30/0.63	0.38/0.78	0.54/1.24	0.54/1.15	-	-
STAR ECCV'20	0.36/**0.65**	0.17/0.36	**0.26/0.55**	**0.22/0.46**	0.31/0.62	0.26/0.53	**0.43**/0.85	**0.38/0.80**
CONTRA (vanilla)	**0.33**/0.66	**0.14/0.29**	0.33/0.68	0.24/0.50	**0.20/0.42**	**0.25/0.51**	0.44/**0.82**	0.40/0.86
CONTRA (goal-GT)	0.16/0.06	0.06/0.03	0.13/0.12	0.09/0.05	0.07/0.03	0.10/0.06		

Transformers and Self-Attention

- Conditioned (GT) goal transformer:
  - We tried several architectures/combinations based on a variety of early or late fusion



[Coscia, Chiara, **Ballan** - *ongoing*]

Experiments

- Quantitative results on ETH/UCY, SDD and IND:

(ADE/FDE)	**ETH**	**HOTEL**	**UNIV**	**ZARA1**	**ZARA2**	**AVG**	**SDD**	**IND**
S-GAN CVPR'18	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18	0.65/1.26	0.48/0.99
Trajectron ICCV'19	0.59/1.14	0.35/0.66	0.54/1.13	0.43/0.83	0.43/0.85	0.56/1.14	-	-
SoPhie CVPR'19	0.70/1.43	0.76/1.67	0.30/0.63	0.38/0.78	0.54/1.24	0.54/1.15	-	-
STAR ECCV'20	0.36/**0.65**	0.17/0.36	**0.26/0.55**	**0.22/0.46**	0.31/0.62	0.26/0.53	**0.43**/0.85	**0.38/0.80**
CONTRA (vanilla)	**0.33**/0.66	**0.14/0.29**	0.33/0.68	0.24/0.50	**0.20/0.42**	**0.25/0.51**	0.44/**0.82**	0.40/0.86
CONTRA (goal-GT)	0.16/0.06	0.06/0.03	0.13/0.12	0.09/0.05	0.07/0.03	0.10/0.06		
CONTRA (goal-GT + 1m noise)	0.27/0.47	0.11/0.21	0.22/0.44	0.16/0.32	0.13/0.27	0.18/0.34		

Contact

- **Office:** Torre Archimede, room 6CD3
- **Office hours** (ricevimento)**:** Friday 09:00-11:00

✉ lamberto.ballan@unipd.it

🏠 http://www.lambertoballan.net

🏠 http://vimp.math.unipd.it

@ twitter.com/lambertoballan