Understanding Future Motion of Agents in Dynamic Scene using Deep-Learning

Masters Thesis Defense

Anuj Sharma

Supervised by:

Prof. Philip H. S. Torr and Dr. Puneet K. Dokania

Torr Vision Group, University of Oxford, UK

5^{th} September, 2017

OUTLINE

- Introduction
- Problem Definition
- Approach
- Background
- Model
- Experiments
- Results
- Conclusions

INTRODUCTION

- Understand the motion characteristics of agents (pedestrians, cyclists, cars etc.) in a dynamic traffic scenario

Stanford Drone Dataset

http://cvgl.stanford.edu/projects/uav_data/

PROBLEM DEFINITION

- To predict future motion of the agents, subject to their mutual interactions and scene, for the given past motion

$$f_\theta : X \mid (scene, interactions) \rightarrow Y$$

X : *past trajectory*

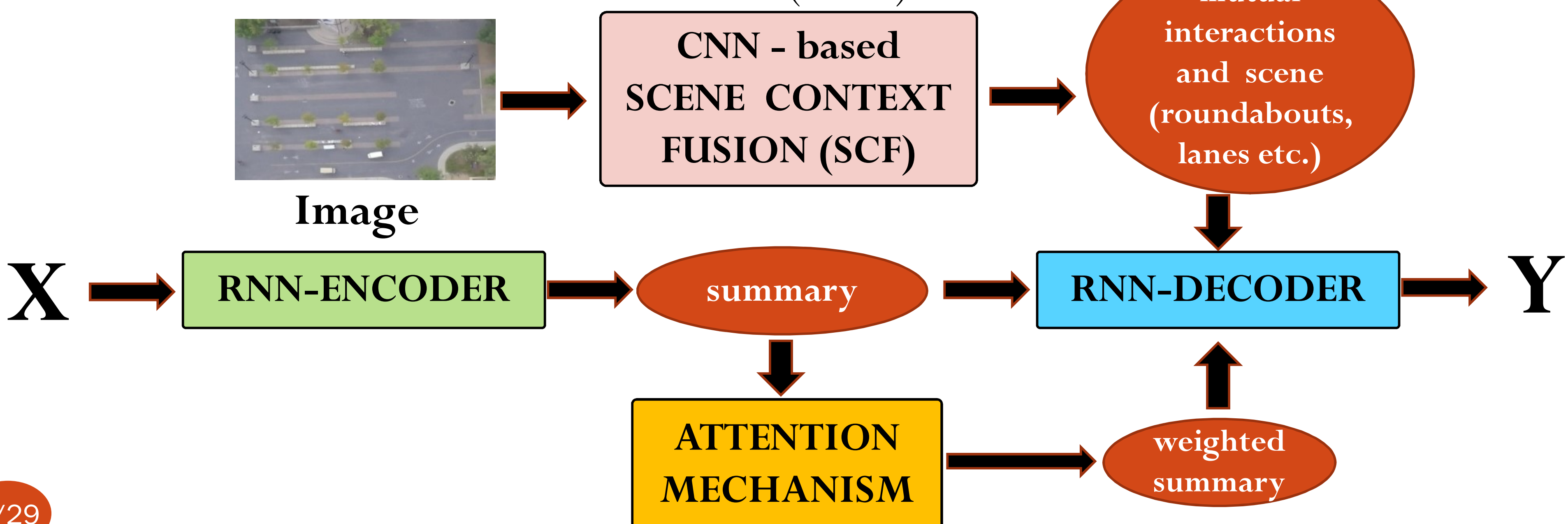
Y : *future trajectory*

θ : *parameters of the function* f

- Aim to learn the parameters θ through data-driven experience, like humans

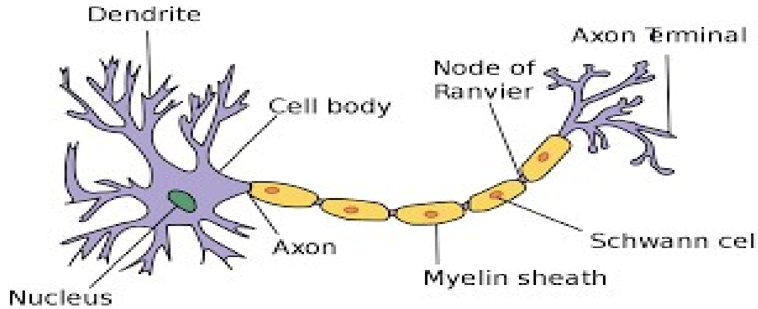
APPROACH

- Utilize concepts from Computer Vision and Deep Learning
  - Recurrent Neural Network (RNN)
  - Convolutional Neural Network (CNN)

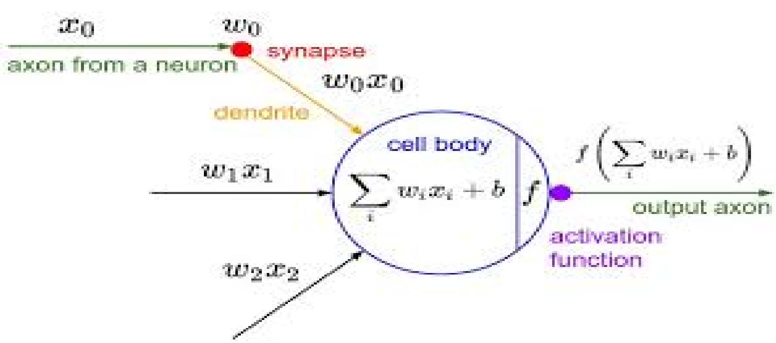


BACKGROUND – Neural Networks

- <u>Neurons</u> in human brain



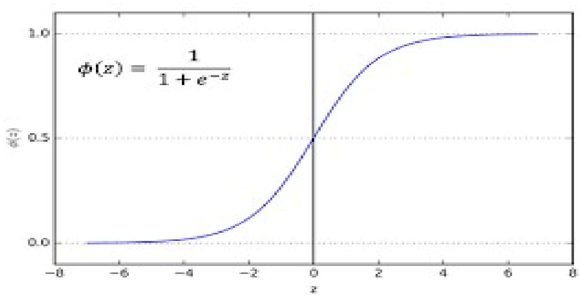
- Mathematically, expressed as (also known as fully-connected ***fc***):



BACKGROUND – Neural Networks

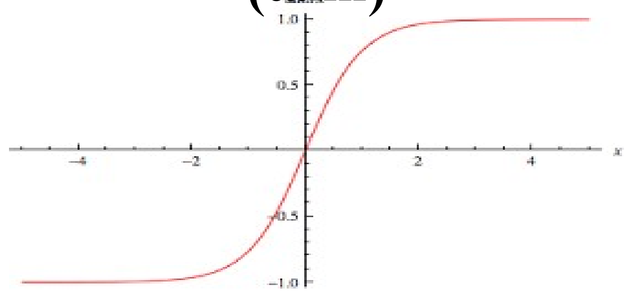
- <u>Activation Functions</u> – to introduce non-linearities

Sigmoid



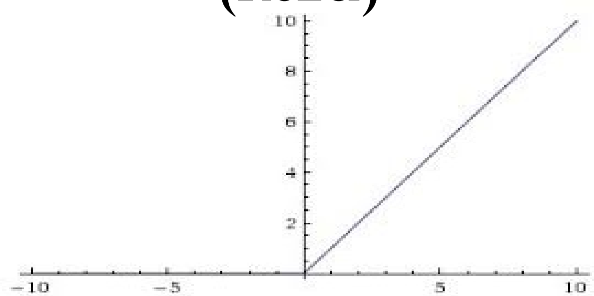
$$y = \sigma(x) = \frac{1}{1 + e^{-x}}$$

Hyperbolic tangent (tanh)



$$y = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

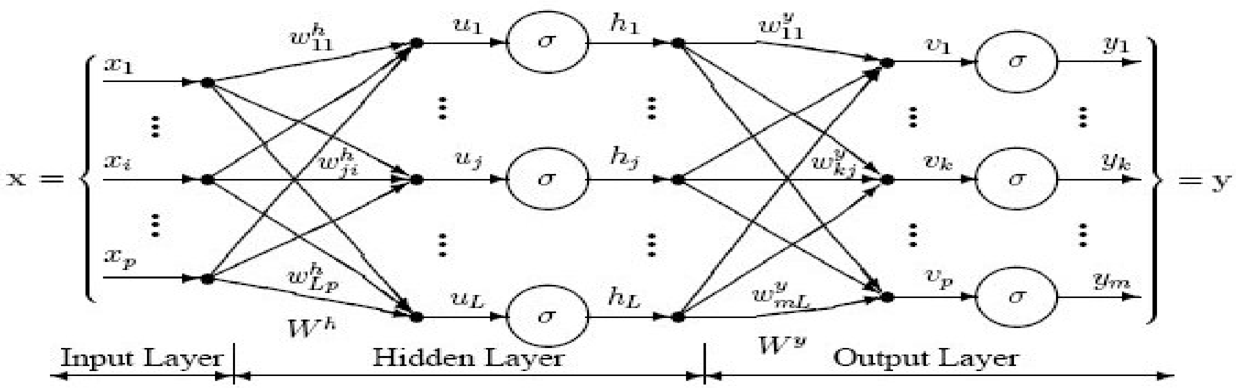
Rectified Linear Unit (ReLU)



$$y = \max(0, x)$$

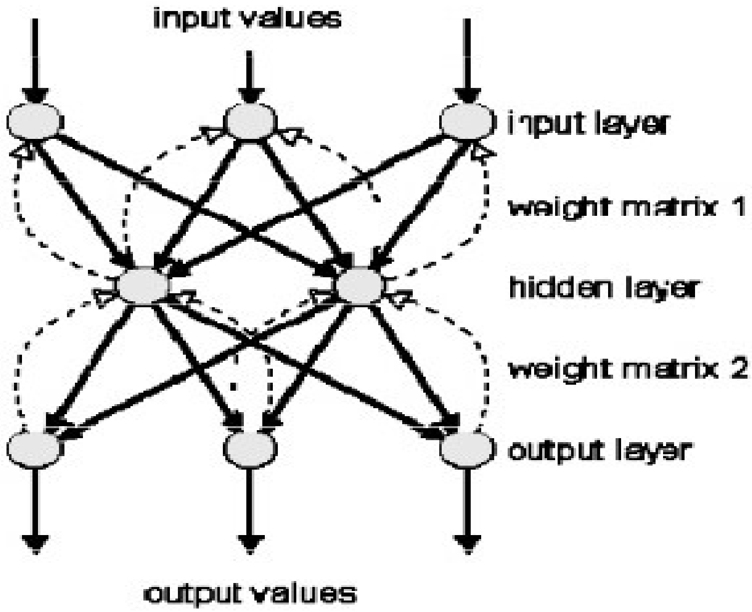
BACKGROUND – Neural Networks

- <u>Artificial Neural Network:</u>
  - Composition of functions



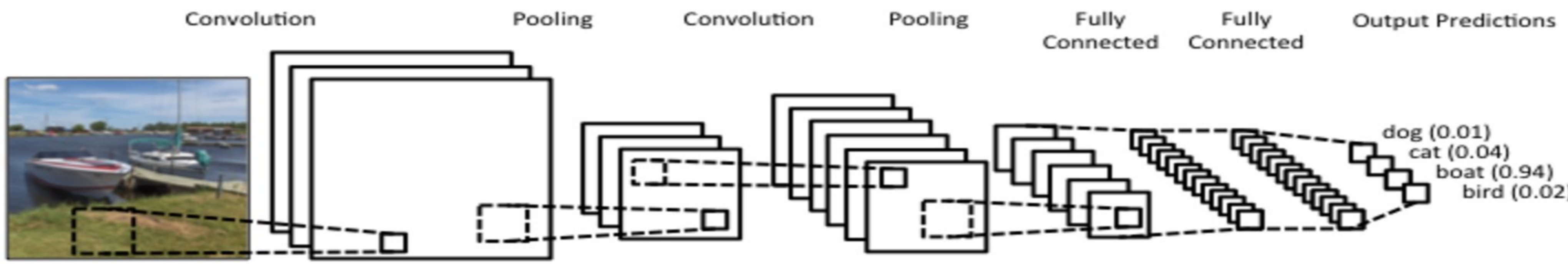
BACKGROUND – Neural Networks

- <u>Training:</u>
  - Back-propagation algorithm – backward flow of gradients



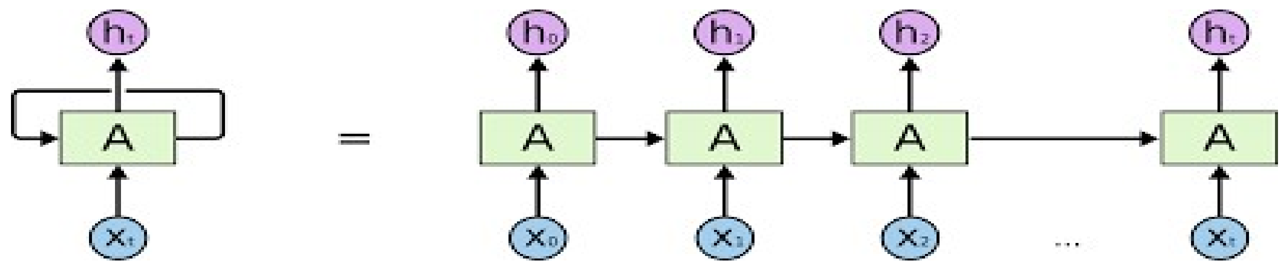
BACKGROUND - CNN

- Convolutional Neural Networks (CNNs)
  - Neural networks designed with layers of convolutions and pooling operations
  - Powerful enough to extract relevant features in image
  - Highly utilized in tasks, such as classification, segmentation, pose-estimation in images etc.



BACKGROUND - RNN

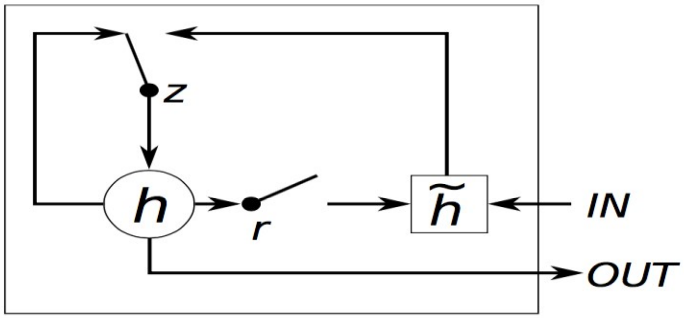
- <u>Recurrent Neural Networks</u>
  - Type of Neural Networks, designed with recurrent cells, to extract patterns in sequences
  - Capable to store and retrieve long-term memory
  - Highly utilized in tasks, such as, language translations, time series predictions, financial and weather forecasting etc.



A: RNN-cell,
X_t: input sequence
h_t: hidden features/ summary

BACKGROUND – RNN Cell

- Gated Recurrent Unit (GRU) type RNN-Cell
  - Input: x_t
  - Previous state: h_{t-1}
  - Update gate: z_t
  - Reset gate: r_t



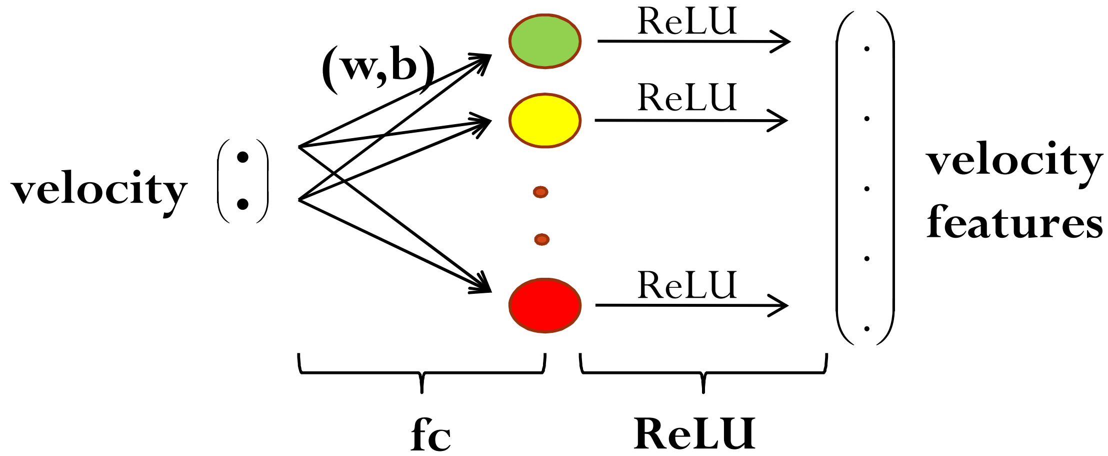
$$
\begin{aligned}
h_t &= z_t h_{t-1} + (1 - z_t)\tilde{h}_t \\
\tilde{h}_t &= \tanh\Big(W x_t + U(r_t \odot h_{t-1})\Big) \\
r_t &= \sigma\Big(W_r x_t + U_r h_{t-1}\Big) \\
z_t &= \sigma\Big(W_z x_t + U_z h_{t-1}\Big)
\end{aligned}
$$

MODEL - Nomenclature

Representation	Description
I_0	Image of the scene
N	Number of agents in the scene
$X = [X_1, X_2, \ldots, X_N]$	Past trajectory (ground truth) of the N agents
$Y = [Y_1, Y_2, \ldots, Y_N]$	Future trajectory (ground truth) of the N agents
$X_i = [X_{i,t-\nu+1}, X_{i,t-\nu+2}, \ldots, X_{i,t}]$	Past positions of i^{th} agent for ν steps
$Y_i = [Y_{i,t+1}, Y_{i,t+2}, \ldots, Y_{i,t+\delta}]$	Future positions of i^{th} agent for δ steps
$\dot{X} = [\dot{X}_1, \dot{X}_2, \ldots, \dot{X}_N]$	Past velocity of the N agents
$\dot{Y} = [\dot{Y}_1, \dot{Y}_2, \ldots, \dot{Y}_N]$	Future velocity of the N agents
$\hat{Y} = [\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_N]$	Predicted trajectory of the N agents

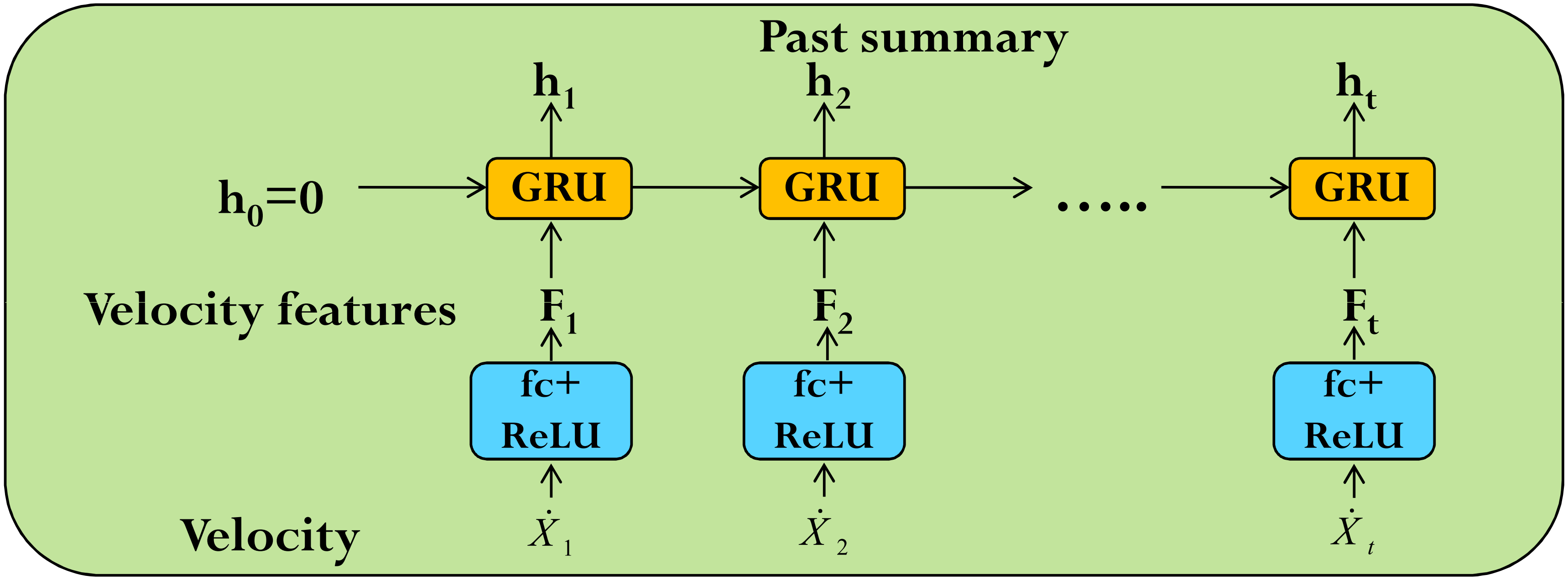
MODEL – RNN-Encoder

- Encodes the past motion into summary
- Takes velocity features as inputs instead of velocity



$$F_t = \max(0, w_v \dot{X}_t + b_v)$$

MODEL – RNN-Encoder

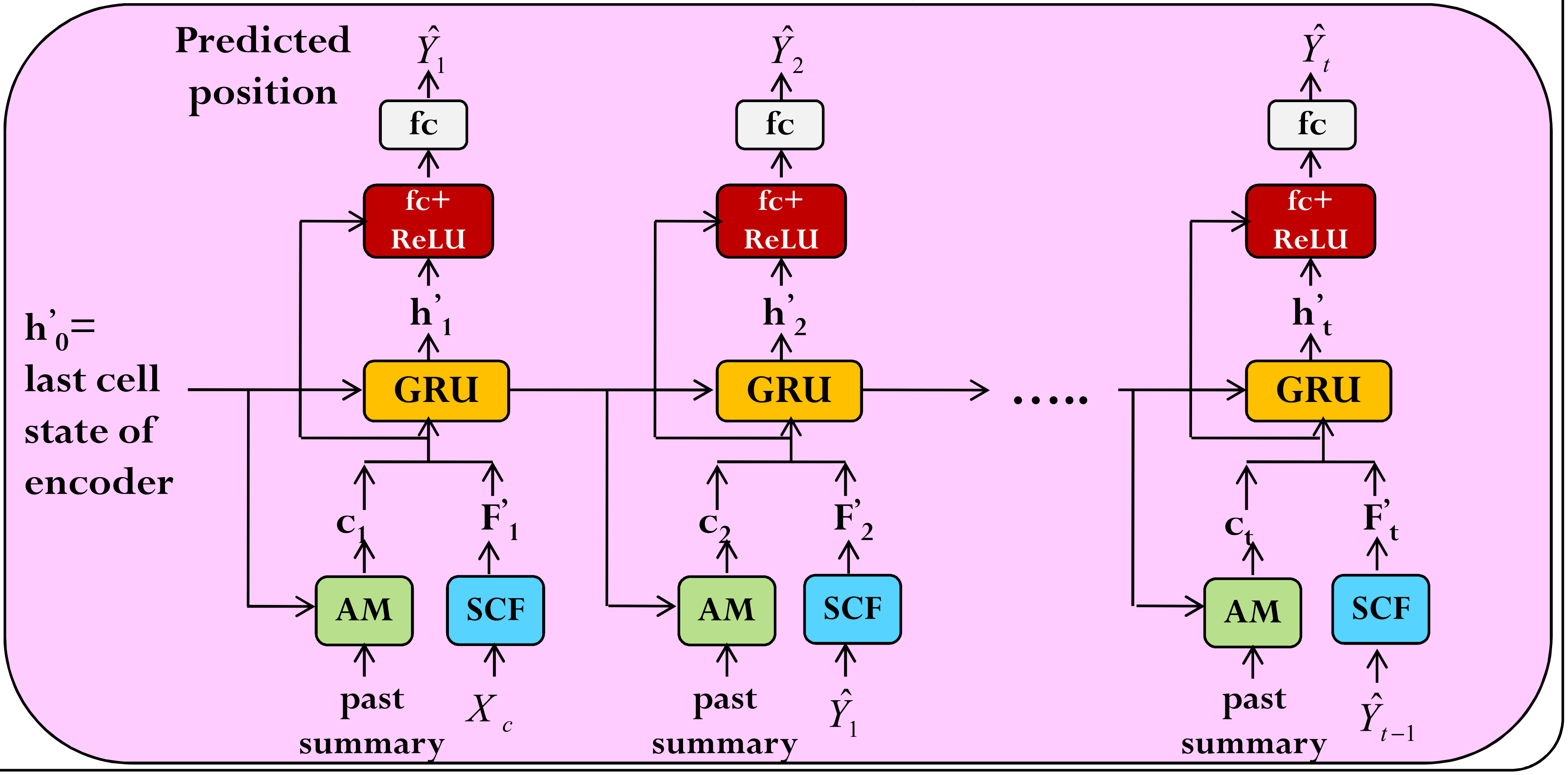


$$
\begin{aligned}
h_t &= z_t h_{t-1} + (1 - z_t)\tilde{h}_t \\
\tilde{h}_t &= \tanh\Big(W F_t + U(r_t \odot h_{t-1})\Big) \\
r_t &= \sigma\Big(W_r F_t + U_r h_{t-1}\Big) \\
z_t &= \sigma\Big(W_z F_t + U_z h_{t-1}\Big)
\end{aligned}
$$

MODEL – RNN-Decoder

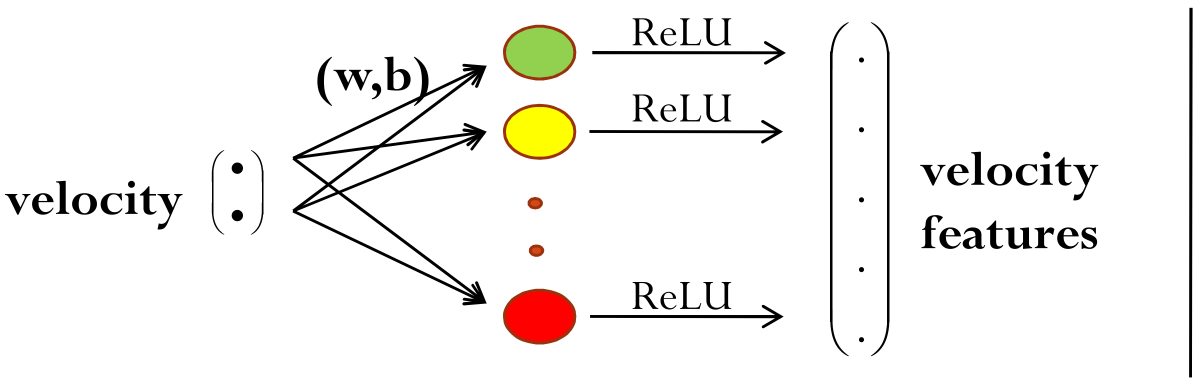
- Decodes the past summary conditioned on dynamic scene and interactions, to predict the future trajectory

AM: Attention Mechanism
SCF: Scene Context Fusion
c_t: weighted summary
F'_t: scene + interaction features
X_c: current position



MODEL – Scene Context Fusion (SCF)

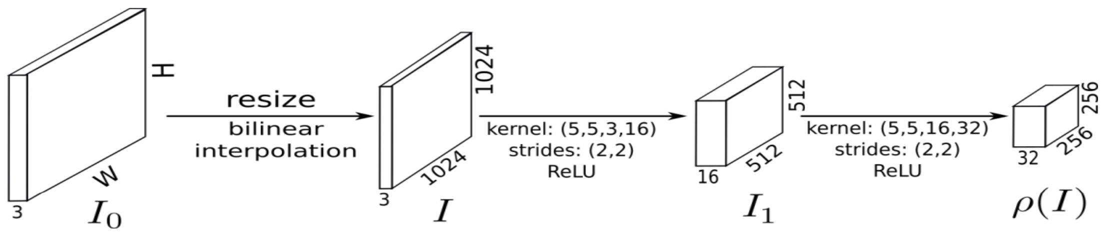
- Fuses the agent's motion context with features of scene and interactions among agents
- <u>Agent's motion context</u>
  - Map the velocity to high-dimensional feature representation



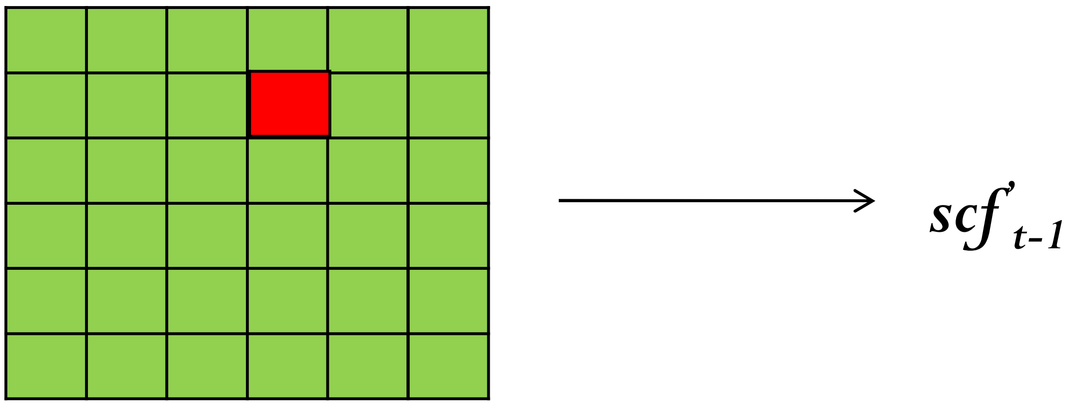
$$f'_{\hat{Y}_{t-1}} = max(0, w_v\dot{\hat{Y}}_{t-1} + b_v)$$

MODEL – Scene Context Fusion (SCF)

- Scene Features
  - Obtain scene features using CNN

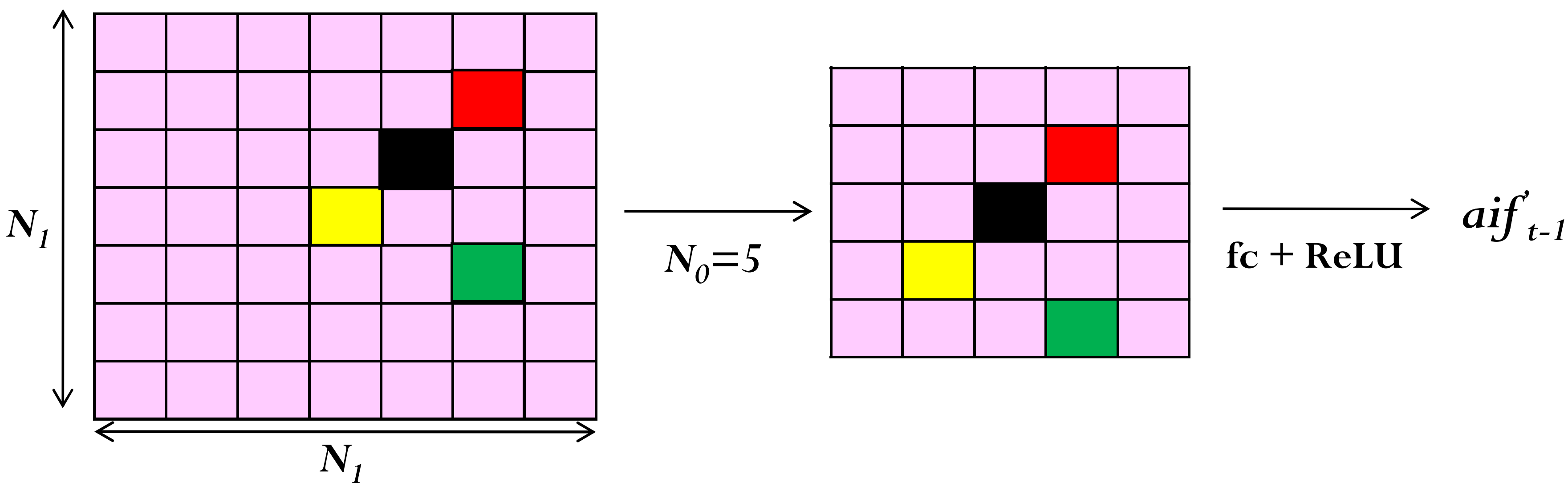


  - Pool scene features corresponding to the agent's position



MODEL – Scene Context Fusion (SCF)

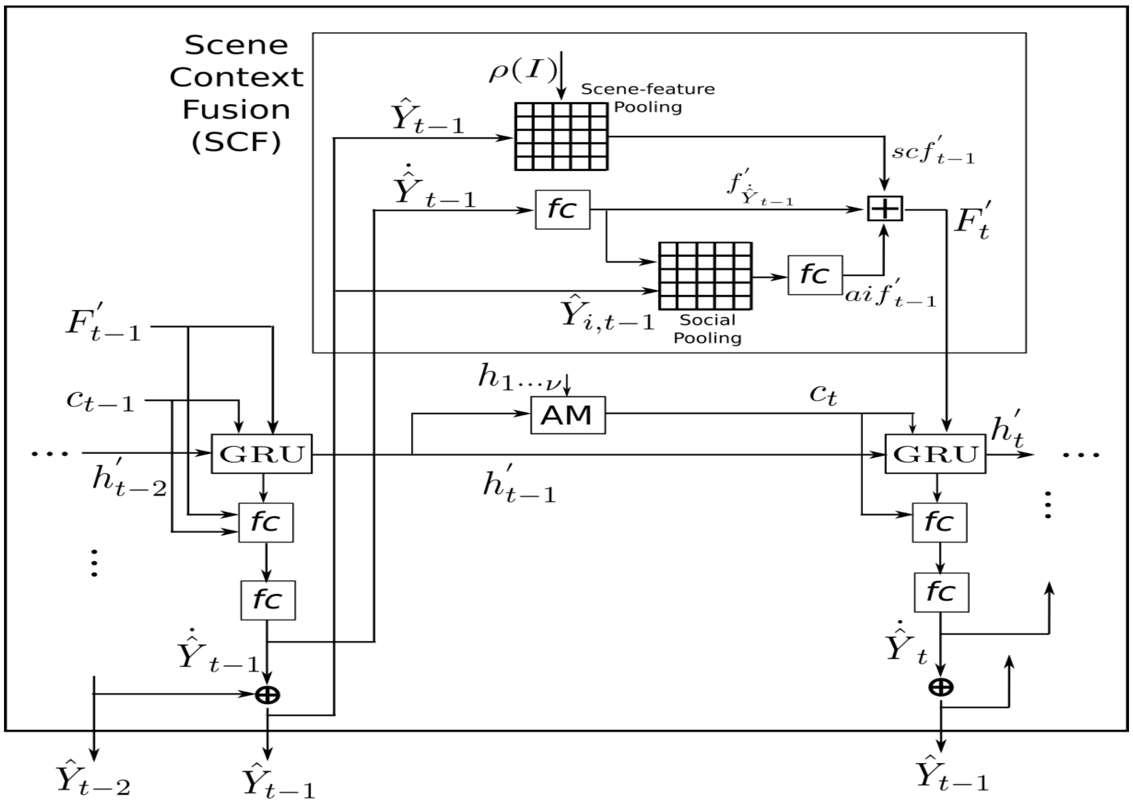
- <u>Interaction Features</u>
  - Velocity features of all agents placed at their respective positions in N_1xN_1 grid



  - Pool interaction features around the agent's position in N_0xN_0 grid

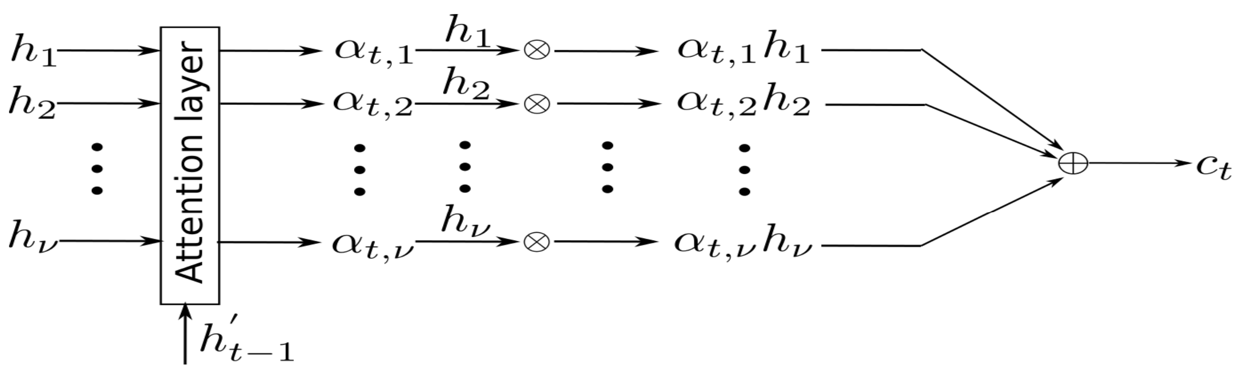
MODEL – Scene Context Fusion (SCF)

- <u>Overall architecture:</u>



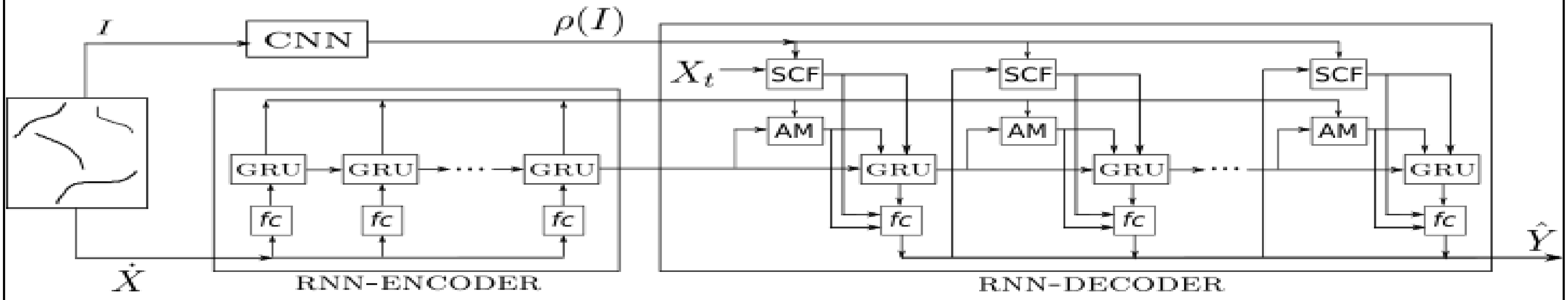
MODEL – Attention Mechanism (AM)

- Weighs all the past summaries w.r.t. the future scenarios



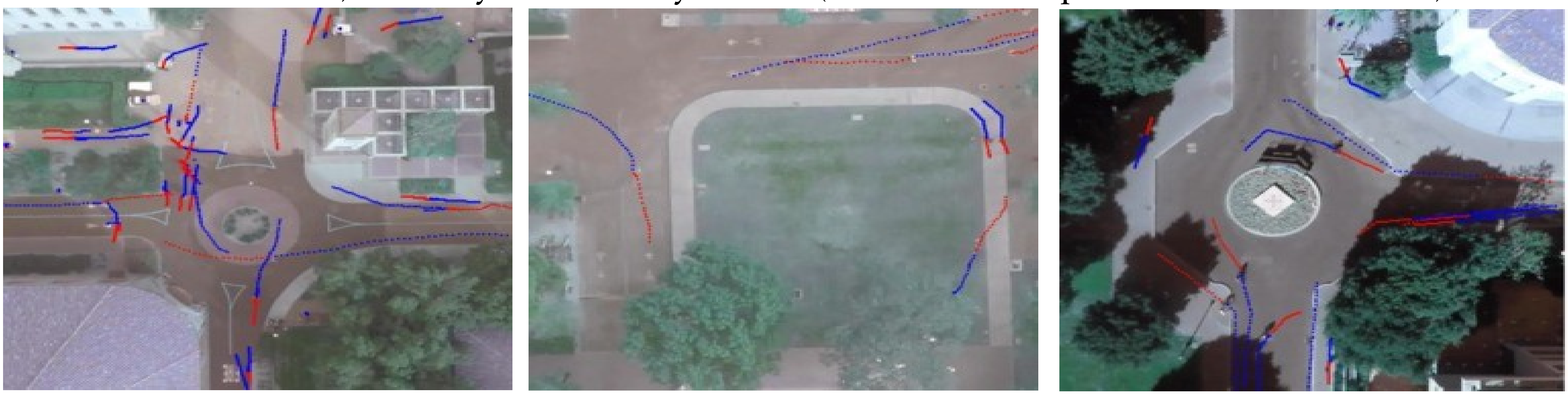
$$
\begin{aligned}
c_t &= \sum_{m=1}^{m=\nu} \alpha_{t,m} h_m \\
\alpha_{t,m} &= \frac{e_{t,m}}{\sum_{m=1}^{m=\nu} e_{t,m}} \\
e_{t,m} &= V_a^T \tanh\left(U'_a h'_{t-1} + W_a h_m\right) \cdots \forall m \in (1, \nu)
\end{aligned}
$$

MODEL – Architecture



EXPERIMENTS - Dataset

- **Stanford Drone Dataset**
  - Highly dynamic situations (roads, roundabouts, etc.) with many agents (pedestrians, cars, etc.) in many different dynamics (slow, fast, sharp maneuver, static, etc.)



  - Data split into 5-folds with usage of 4-folds for training and 1-fold for test performance evaluation

EXPERIMENTS – Model Parameters

Model	Parameters	Dimensions/Values
CNN	H, W	1024
	H_{CNN}, W_{CNN}	256
	w_{k_1}	$5 \times 5 \times 3 \times 16$
	w_{k_2}	$5 \times 5 \times 16 \times 32$
	b_{k_1}	16
	b_{k_2}	32
Encoder	w_v	16×2
	b_v	16
	W, W_r, W_z	48×16
	U, U_r, U_z	48×48
Attention Mechanism	U_a', W_a	48×48
	V_a	1×48
SCF	w_i	$16\times$ $(5 \times 5 \times 16)$
	b_i	16
	N_0	5
	N_1	32
Decoder	W', W_z', W_r', W_g'	48×64
	U', U_z', U_r', U_g'	48×48
	V', V_z', V_r', V_g'	48×48
	w_v'	2×48
	b_v'	2

Past: 2 seconds

Future: 4 seconds

Position in pixels

EXPERIMENTS – Evaluation Metrics

- Variants of model

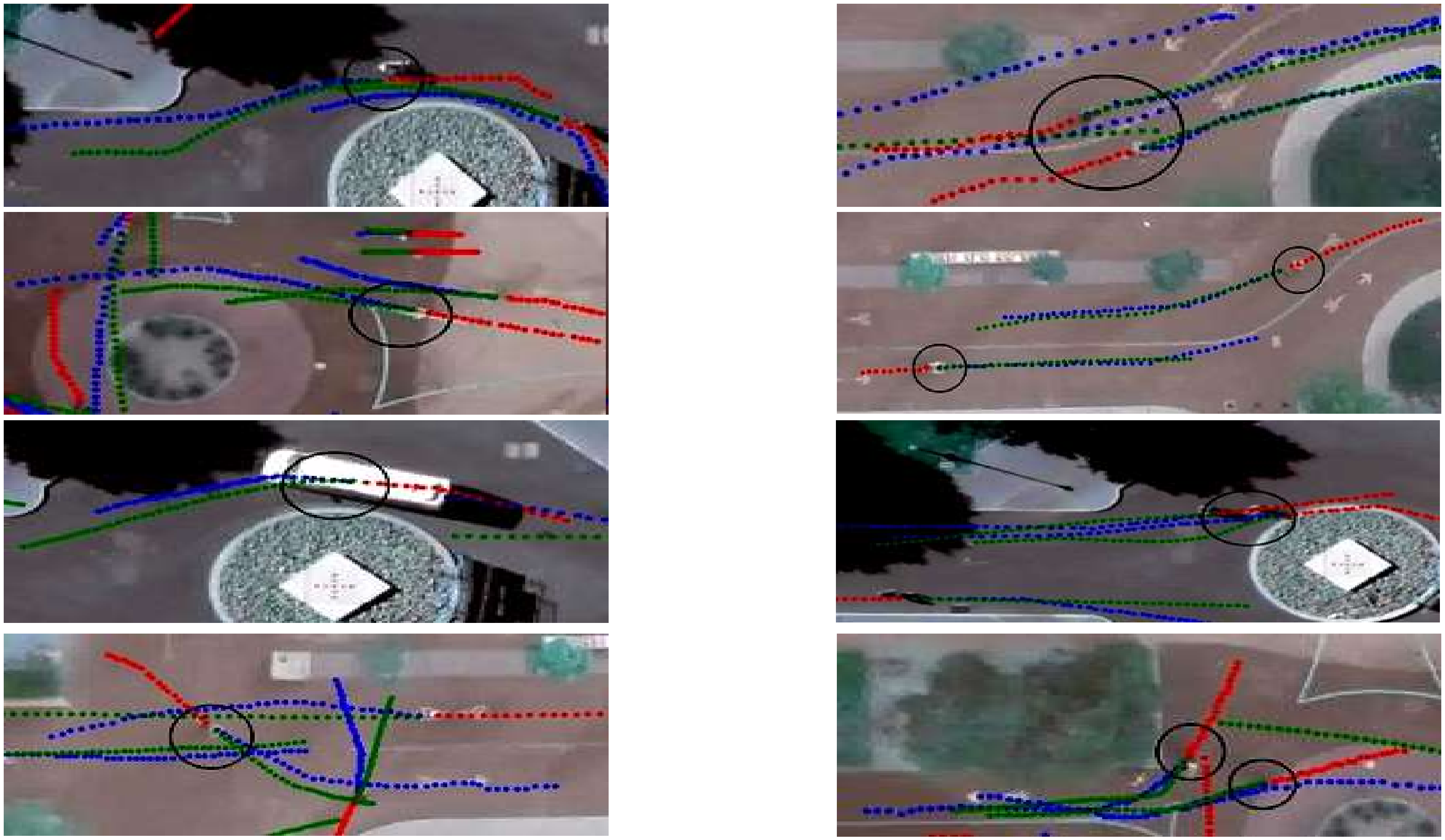
Model	Description
RNN-ED-DESIRE	RNN-Encoder-Decoder variant of DESIRE [1] without SCF
RNN-ED	RNN-Encoder-Decoder without AM and SCF
RNN-ED-VSI	RNN-Encoder-Decoder with SCF
RNN-ED-A	RNN-Encoder-Decoder with AM
RNN-ED-VSI-A	**Final Model: RNN-Encoder-Decoder with AM and SCF**

[1] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, Manmohan Chandraker, *'DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents'*; IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), 2017.

RESULTS - Summary

Model	Pixel Error (scaled by 1/5) at			
	1.0 seconds	2.0 seconds	3.0 seconds	4.0 seconds
RNN-ED-DESIRE	1.76	3.98	6.51	9.31
RNN-ED	1.75	3.94	6.47	9.26
RNN-ED-VSI	1.78	3.91	6.41	9.22
RNN-ED-A	1.70	3.84	6.32	9.08
RNN-ED-VSI-A	**1.70**	**3.79**	**6.22**	**8.92**

RESULTS - Figures



CONCLUSIONS

- Dynamic scene and interactions, with variable number of agents is taken care of by our model
- The final model RNN-ED-VSI-A predicts future trajectory quite well conditioned on the dynamic scene and interactions
- Attention Mechanism significantly improves the prediction accuracy

[1] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, Manmohan Chandraker, *'DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents'*; IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), 2017.

Thank You