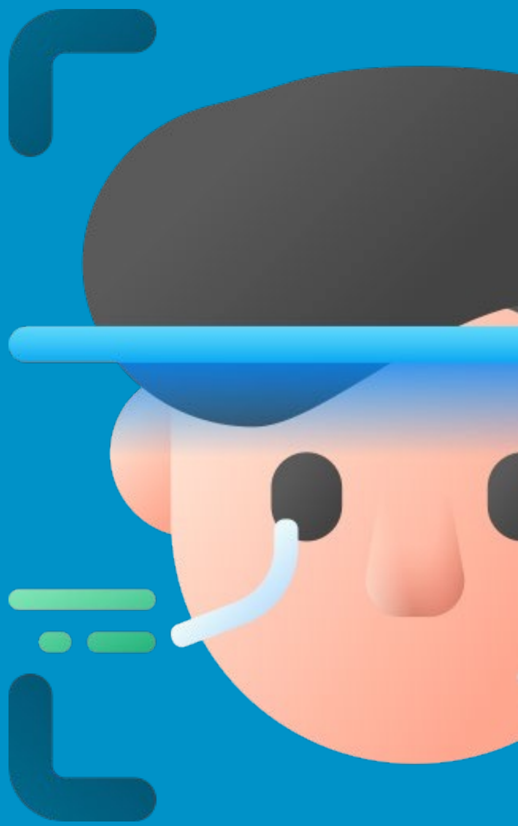
Computer Vision 2023-2024

# PBSC: LSTM-based model for Vehicle Trajectory Prediction



Authors: Fabrizio Italia, Stefano D’Urso and Vincenzo Crisà

SAPIENZA
UNIVERSITÀ DI ROMA

# Outline

- Introduction
- Proposed methods
- Waterloo Intersection Dataset
- Data preparation
- Implementation details
- Evaluation Metric
- Experimental results
- Conclusion and Future Works
- References

# Introduction

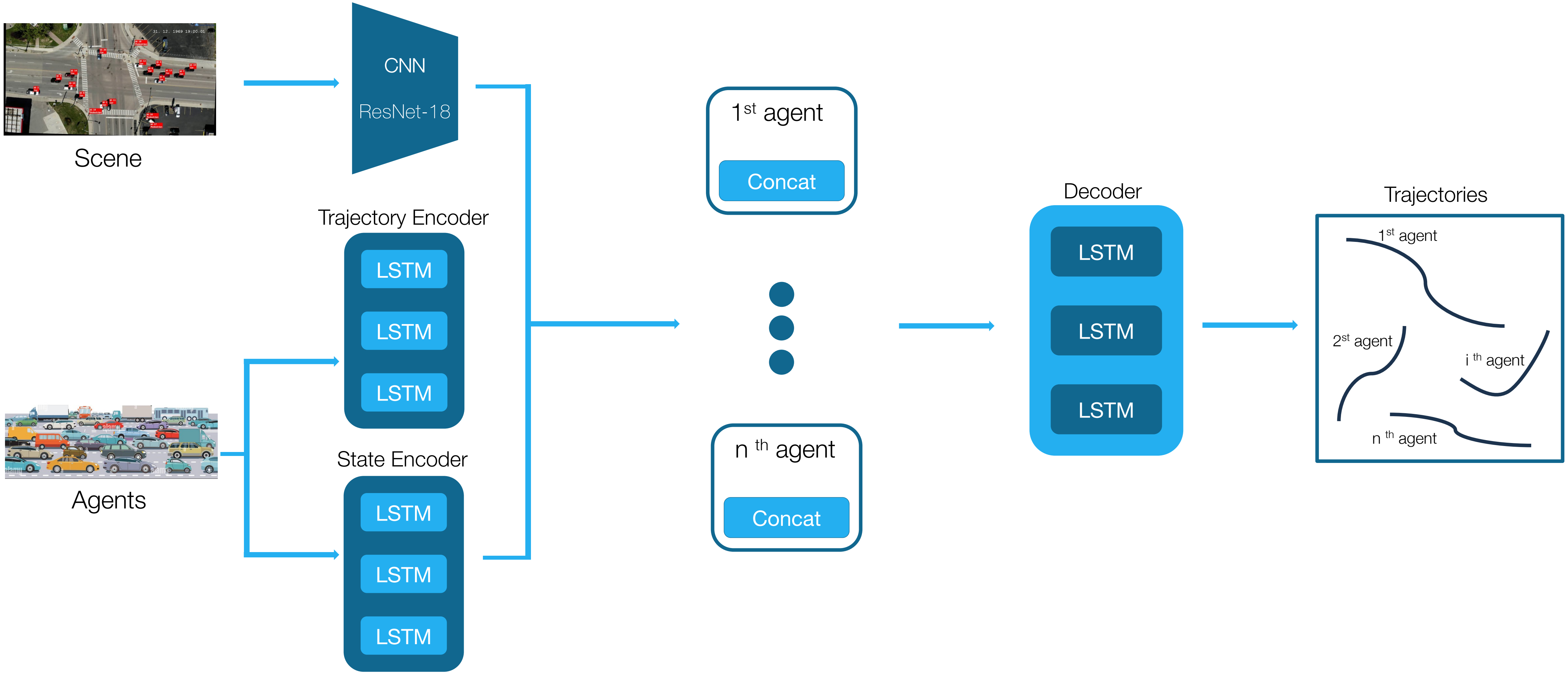
- **Efficient** and **safe** navigation
- Agent **state**
  - Speed
  - Tangential acceleration
  - Lateral acceleration
  - Angle
  - Time
- **Scene** context

**Physical Behavior**

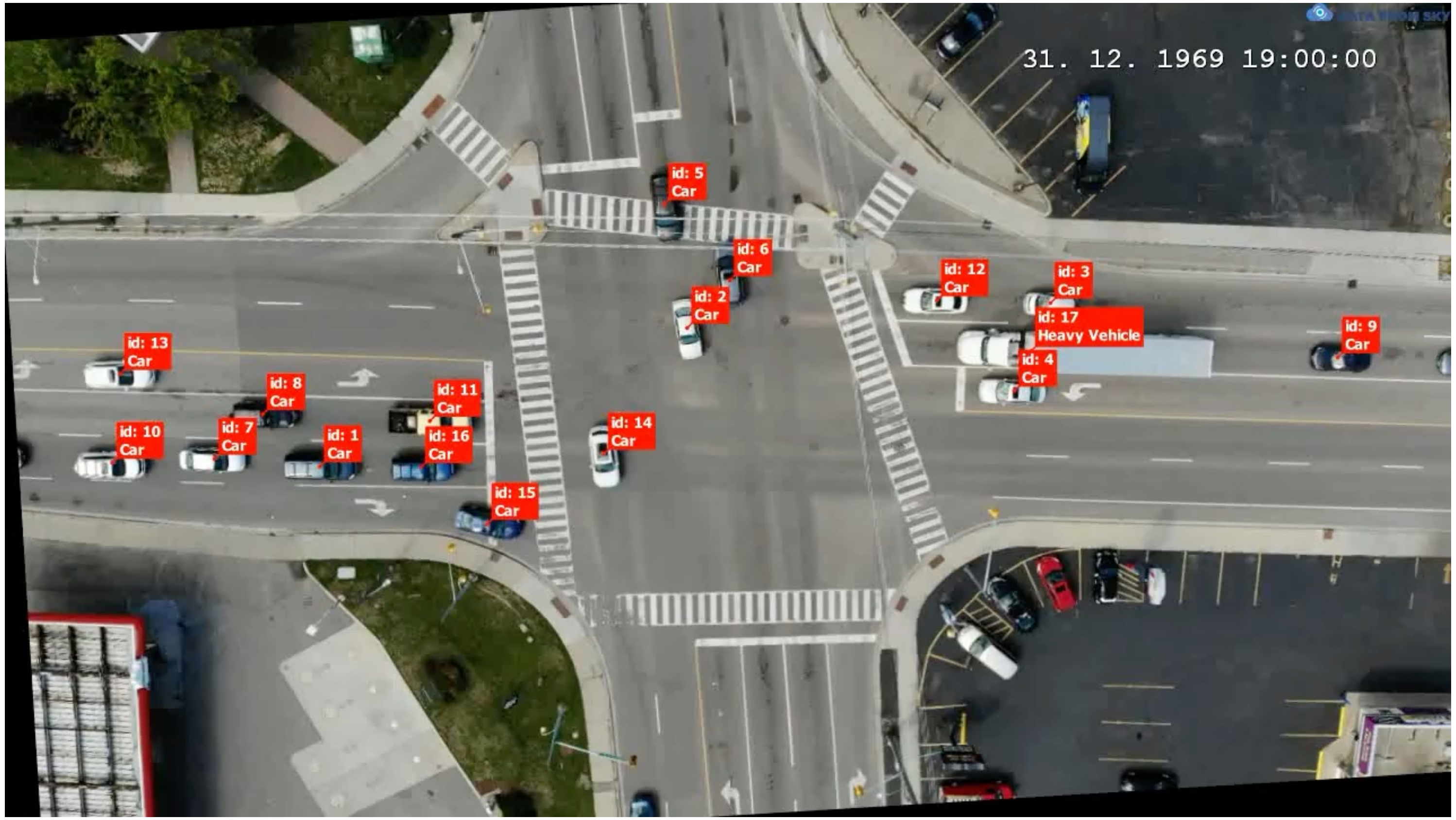
**+**

**Scene Context Information**

# Proposed method: PBSC



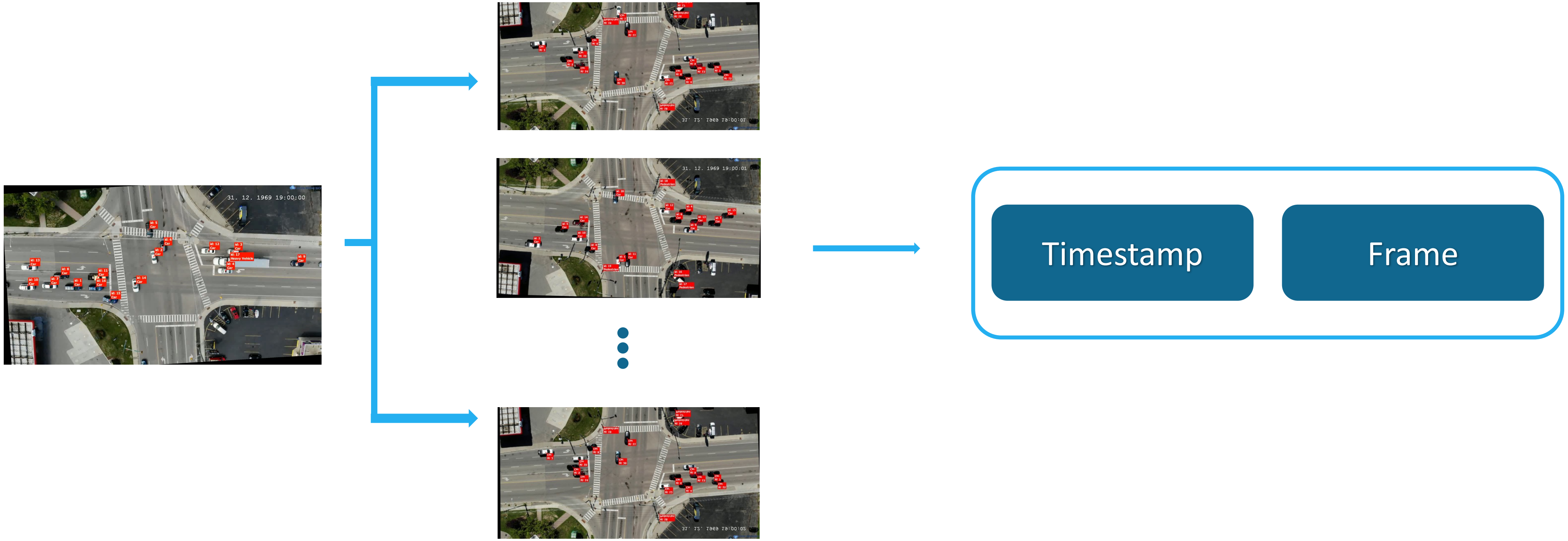
# Waterloo Intersection Dataset




- Multi-agent
- Conflict points
- Traffic Regions
- Traffic Lights
- Trajectories
  - Position (X,Y)
  - Speed
  - Tangential acceleration
  - Lateral acceleration
  - Angle
  - Time

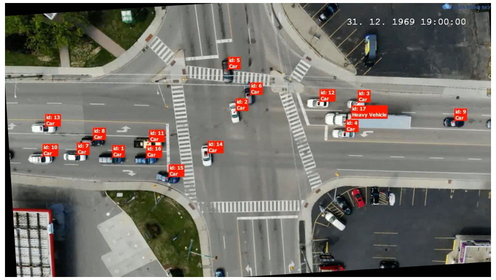
# Data Preparation -1

Conversion of videos into frame with 3Hz of sampling frequency and saving the corresponding timestamp



# Data Preparation -2

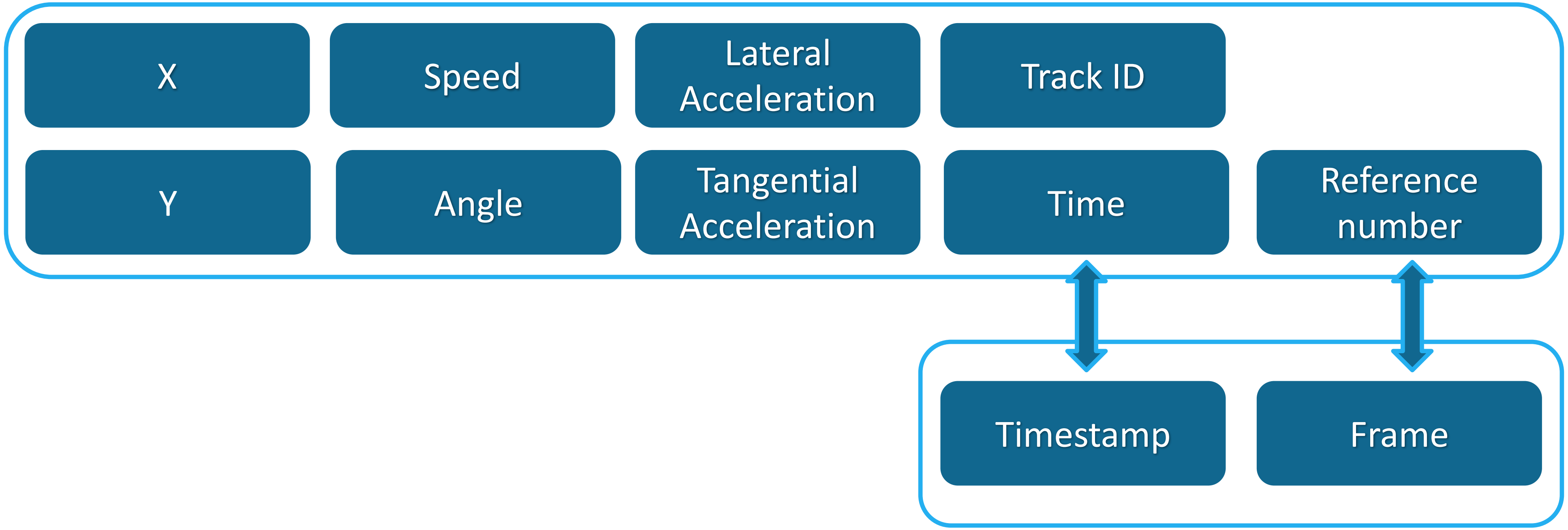
Add in the trajectories information the identifier of the related video




- X
- Y
- Speed
- Angle
- Lateral Acceleration
- Tangential Acceleration
- Track ID
- Time
- Reference number

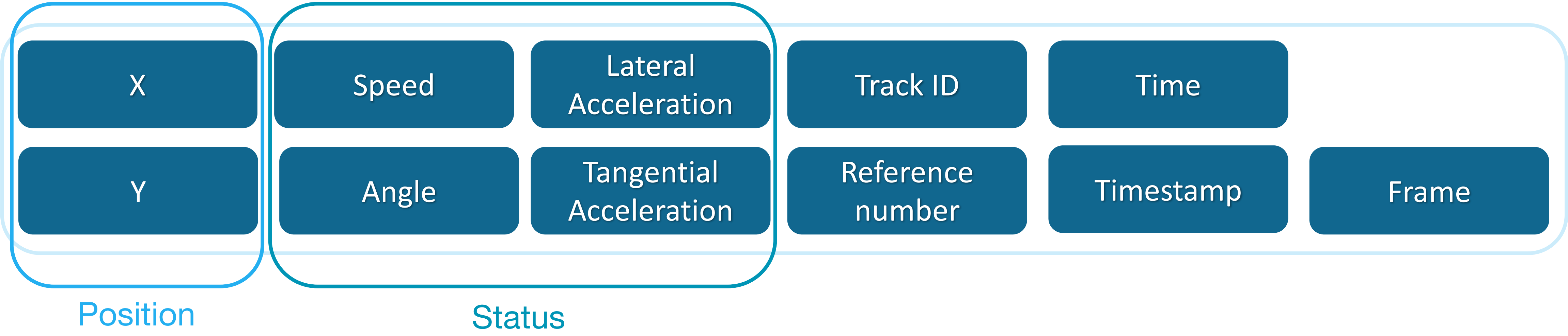
# Data Preparation -3

Merging the trajectories information with the corresponding frame, thanks to the timestamp and the identifier of the video



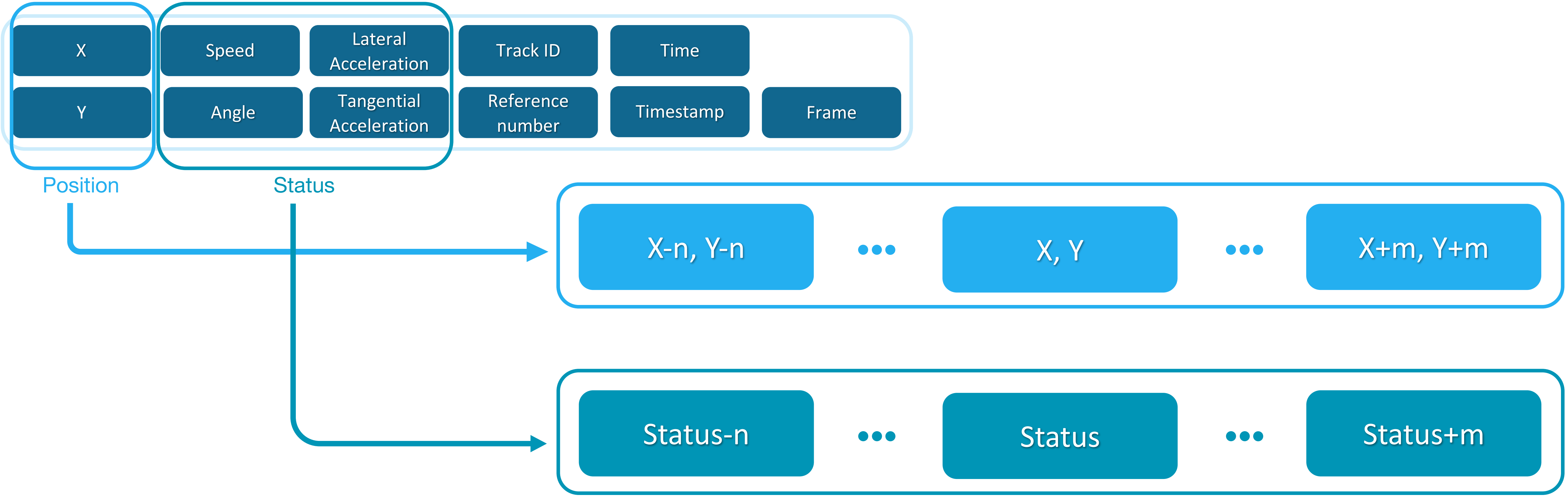
# Data Preparation -4

Each car in each frame is characterized by a position (X,Y) and his status (Speed, Angle, Lateral Acceleration, Tangential Acceleration).



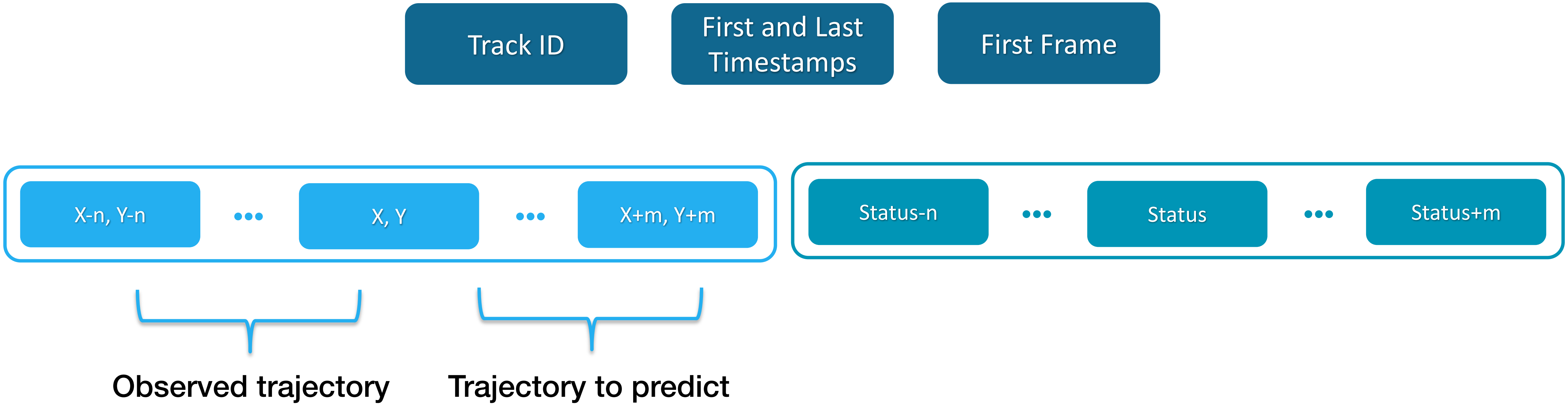
# Data Preparation -5

Reorganize the data in order to have temporalized information, obtaining all the positions and status from the moment in which the agent appears in the video (n) to the moment in which it completes his trajectory (m).



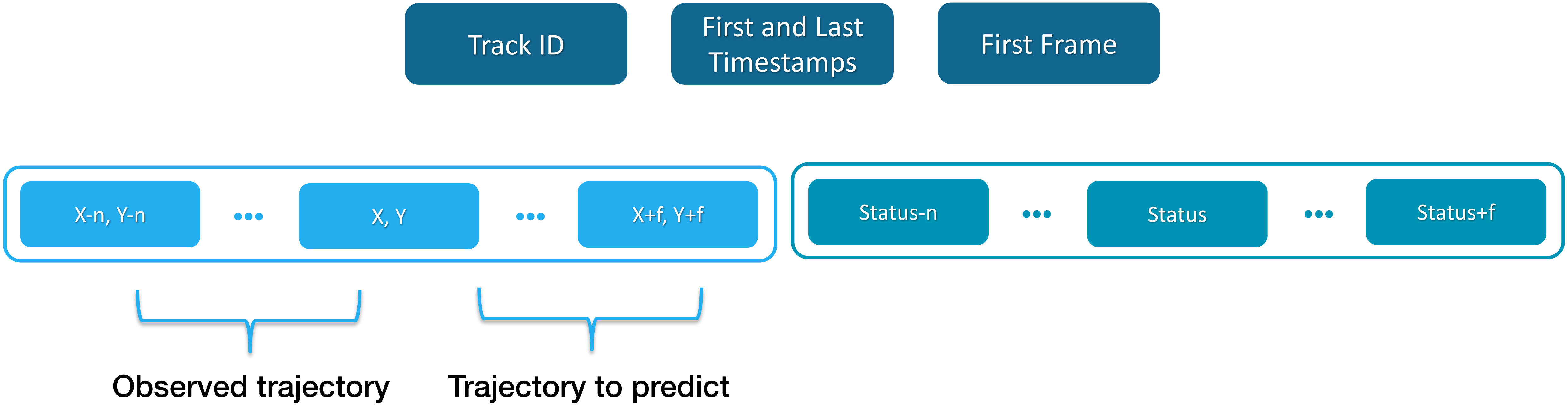
# Data Preparation -6

For each agent are available all the position and status information with the corresponding time interval and the frame relative to his first position



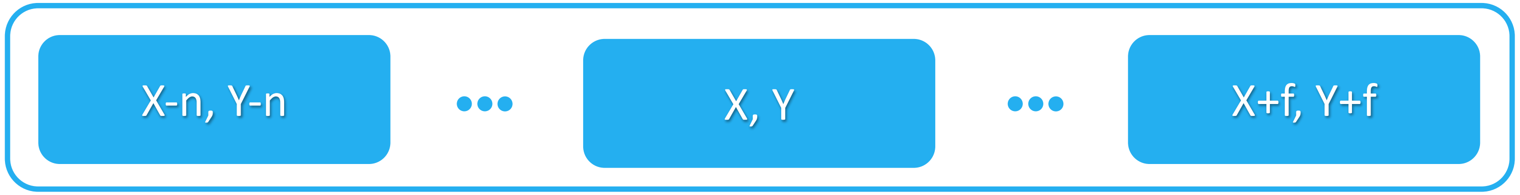
# Data Preparation -7

Since agents have different trajectories size, the data is filtered to have only target trajectories with the same fixed dimension.



# Data Preparation -8

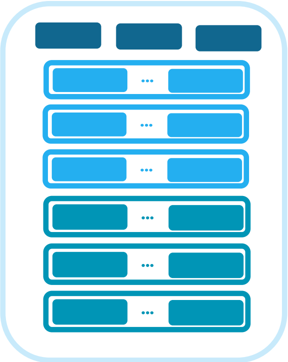
The position values are much higher than the status values and the image, which are normalized to deal with the ResNet-18 parameters.
The position data is normalized following the min-max normalization



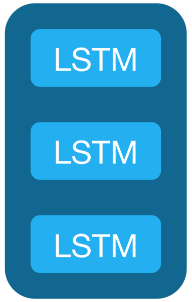

$$normalized\_x = \frac{x - minimum\_x}{maximum\_x - minimum\_x} \qquad normalized\_y = \frac{y - minimum\_y}{maximum\_y - minimum\_y}$$

# Implementation details -1

- Batch size = 8
- Sequence length = 12
- Target length = 5

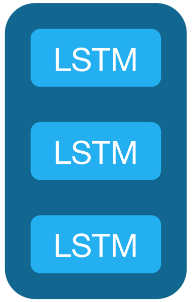


Trajectory Encoder




- Input size = 2
- Hidden size = 128
- Layers = 2

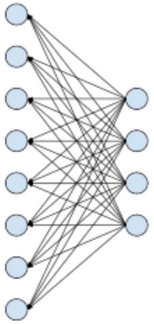
State Encoder




- Input size = 4
- Hidden size = 128
- Layers = 2

Decoder

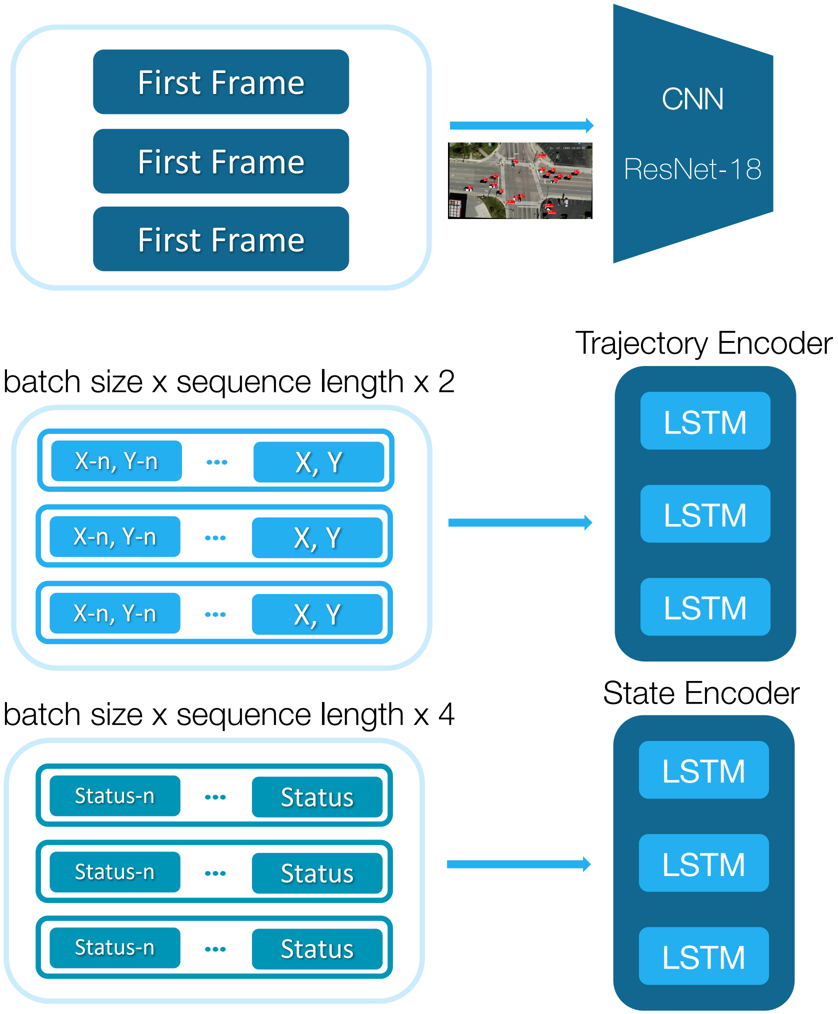

- Input size = 384
- Hidden size = 256
- Layers = 2



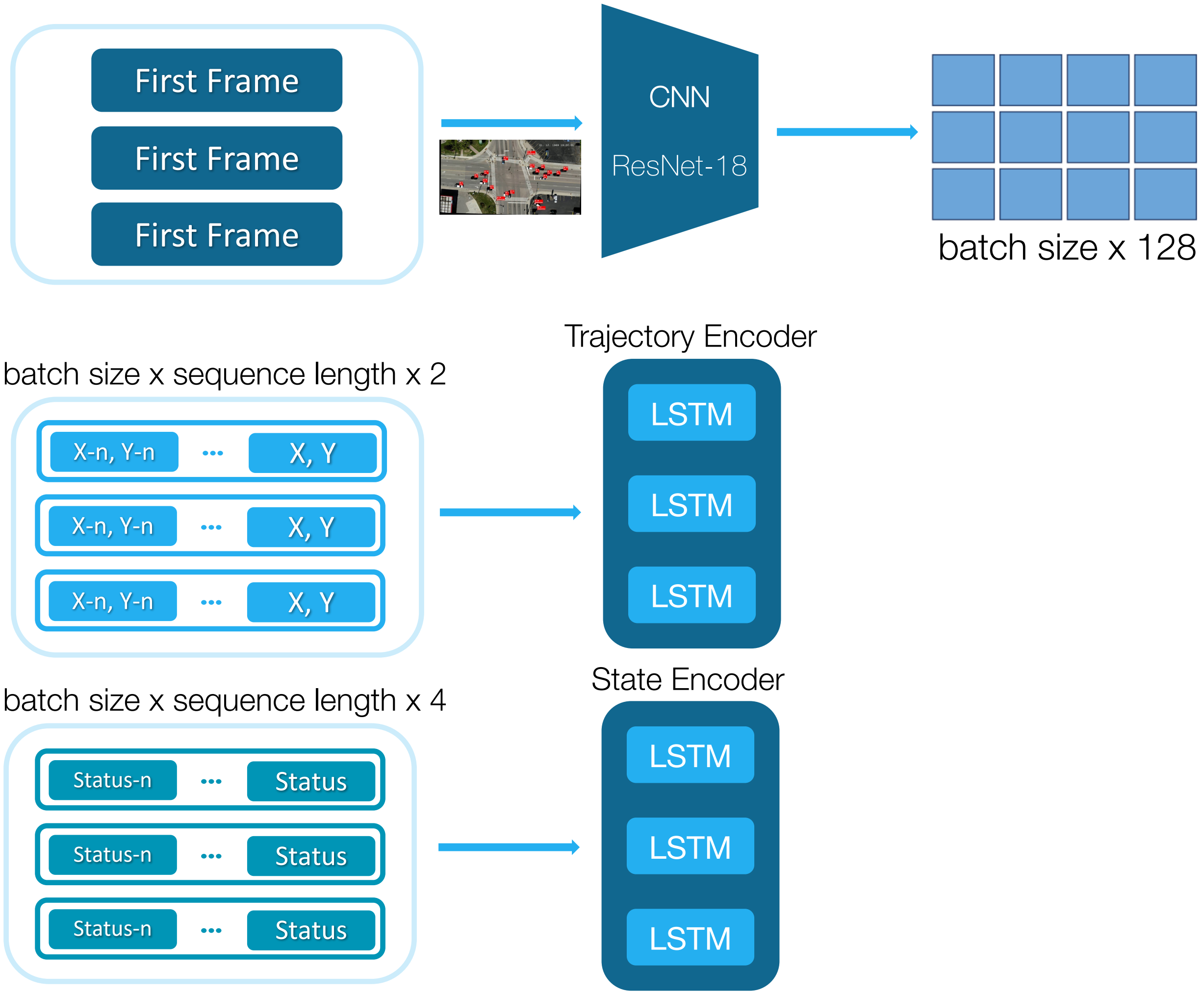
- Input size = 256
- Output size = 2

The total number of trainable parameters is 1584642

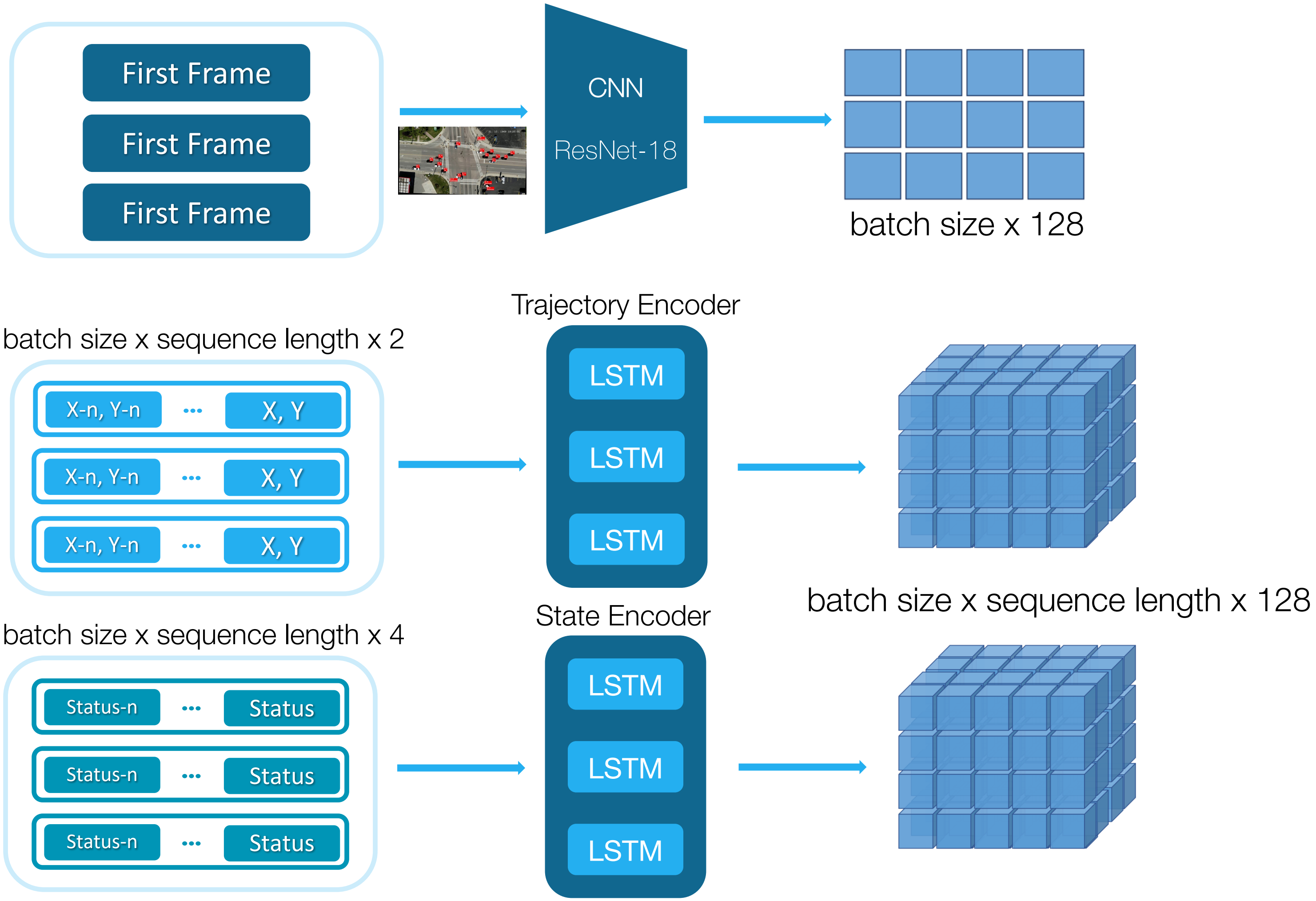
# Implementation details -2



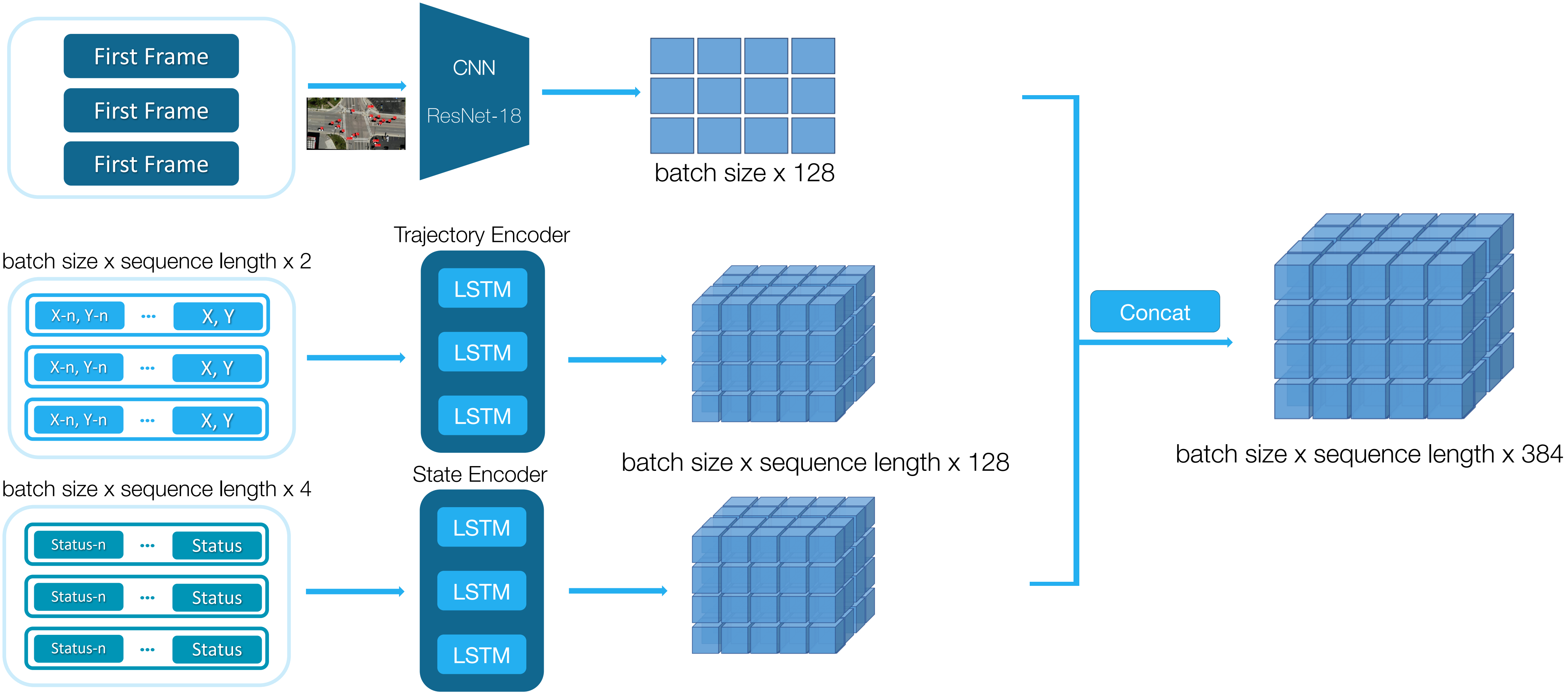
# Implementation details -2



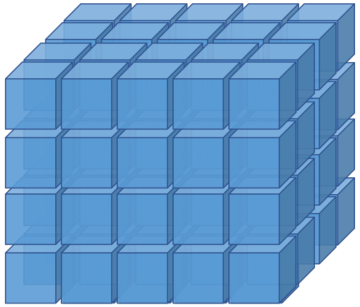
# Implementation details -2



# Implementation details -2

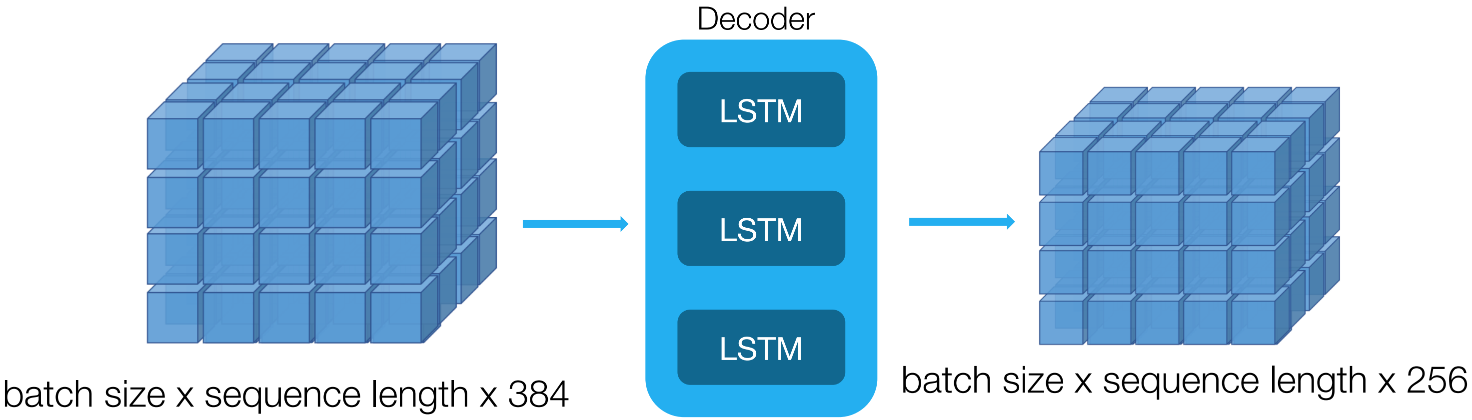


# Implementation details -3

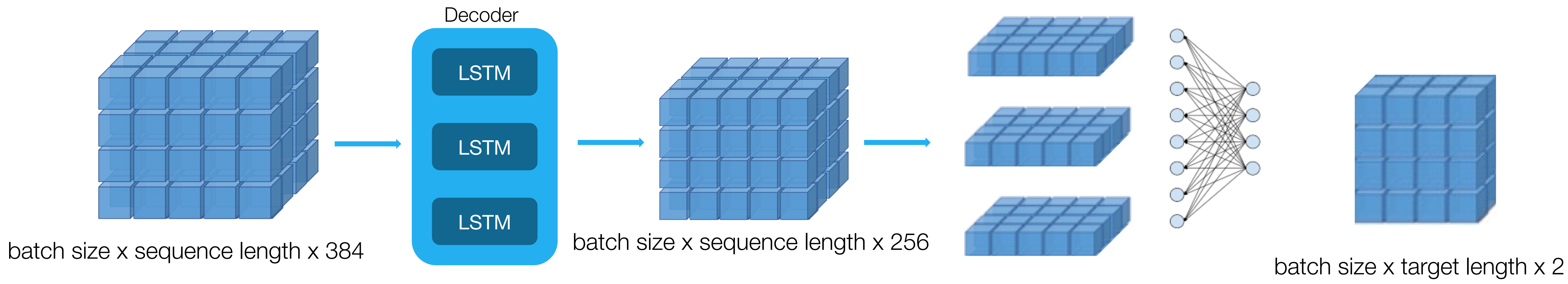


batch size x sequence length x 384

# Implementation details -3



# Implementation details -3



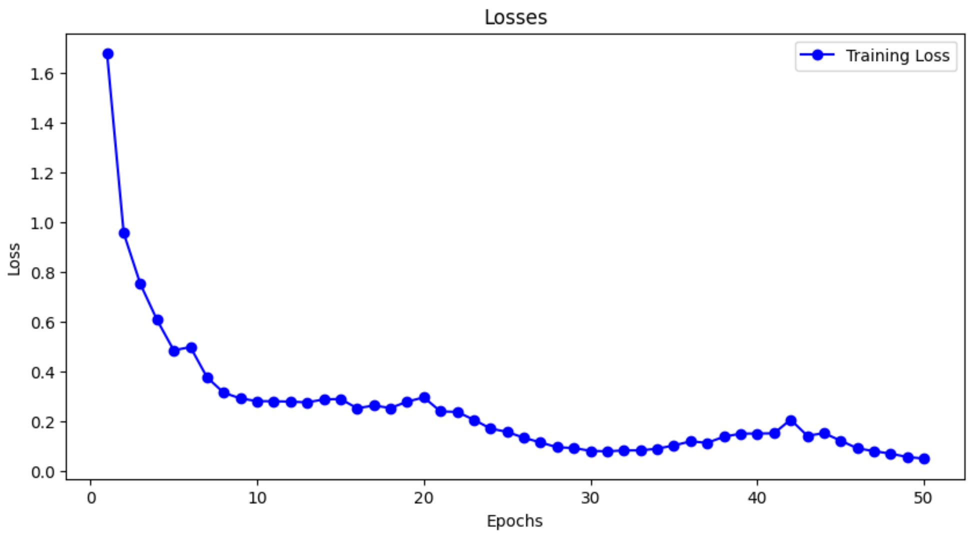
# Evaluation metric

The used metric is ADE: Average Displacement Error between the predicted and the actual positions.

$$ADE = \frac{\sum_{i=1}^{n} \sqrt{(x_{pred} - x_{target})^2 + (y_{pred} - y_{target})^2}}{n}$$

# Experimental Results -1

Training parameters:
- Adam Optimizer with learning rate 0.0005
- Cosine Annealing Learning Rate scheduler
- 50 epochs

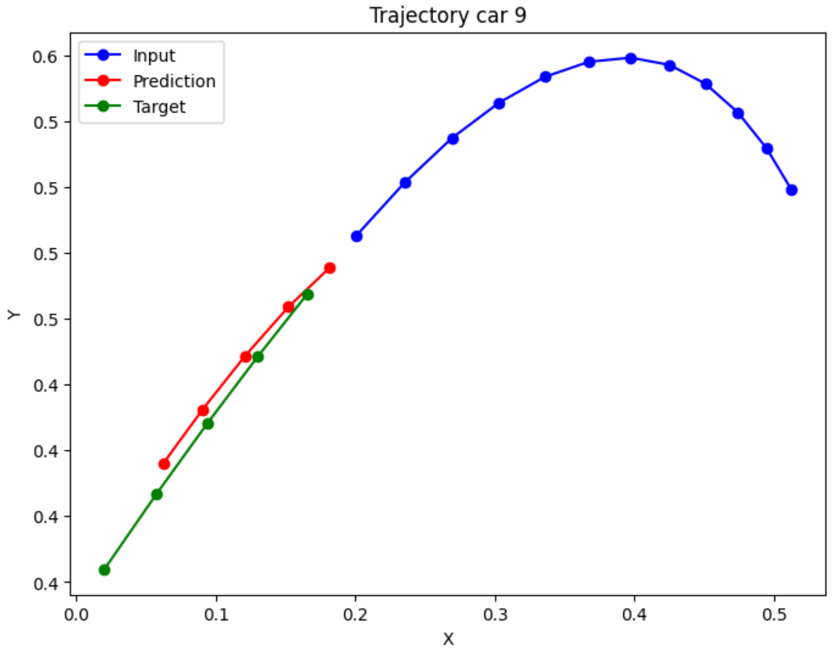


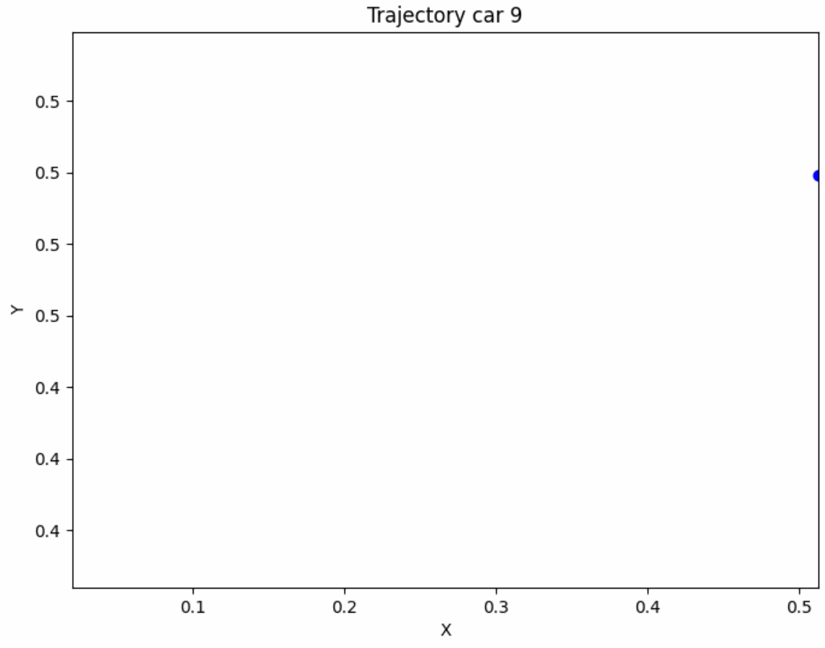

| | ADE |
|---|---|
| Train | **0.0510** |
| Test | **0.1605** |

Note that data is normalized

# Experimental Results -2

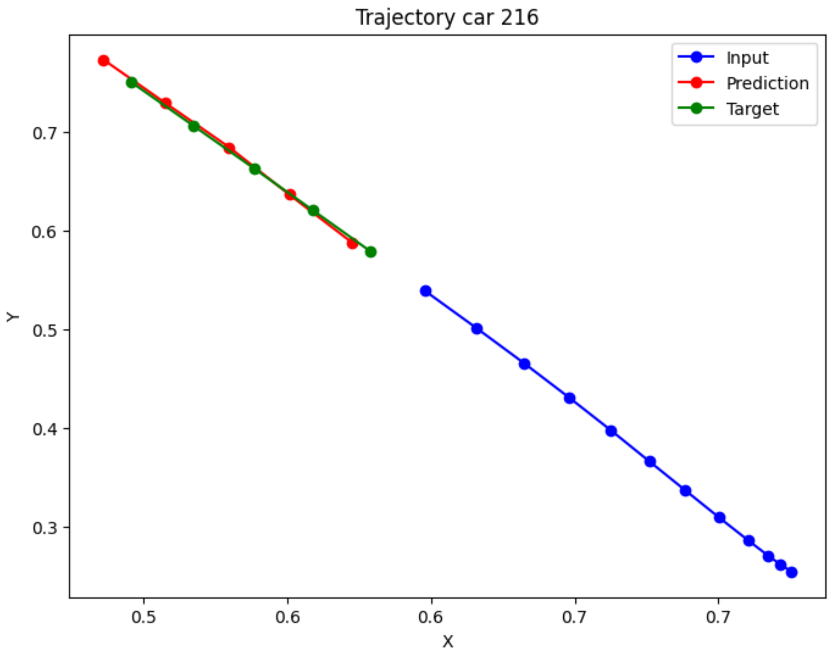
Observed trajectory   Target Trajectory   Predicted Trajectory



# Experimental Results -3

Observed trajectory Target Trajectory Predicted Trajectory




# Conclusion and future works

The integration of physical behavior and scene context enhances the ability of the proposed LSTM-based model to accurately predict agents' trajectories.

**Future works:**

- Use trajectories with variable lengths
- Include a Social Pooling mechanism
- Use additional information provided by the Waterloo multi-agent traffic dataset, like the traffic lights data.

# References

**Deep Learning-Based Multimodal Trajectory Prediction with Traffic Light** [1]
Seoyoung Lee ,Hyogyeong Park , Yeonhwi You , Sungjung Yong and Il-Young Moon

**Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks** [2]
Agrim Gupta, Justin Johnson, Fei-Fei Li, Silvio Savarese, Alexandre Alahi

**Theory to Practice: LSTM and Transformers | PyTorch** [3]
VSINGHSAN

# Authors


**Fabrizio Italia** 2143104

italia.2143104@studenti.uniroma1.it

**Stefano D'Urso** 2143081

durso.2143081@studenti.uniroma1.it

**Vincenzo Crisà** 2143080

crisa.2143080@studenti.uniroma1.it


SAPIENZA
UNIVERSITÀ DI ROMA