Vehicle Interaction Learning

Xiaosong Jia

About double-car model

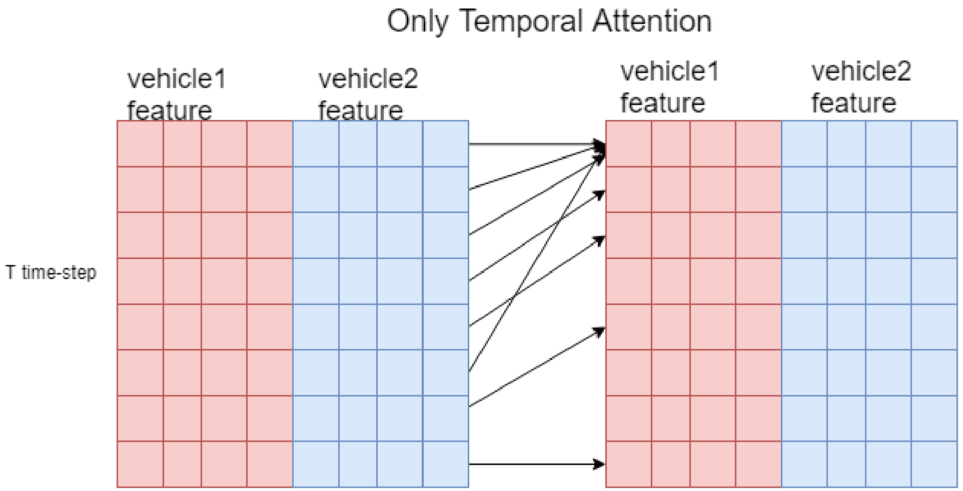
- Separate features for each car

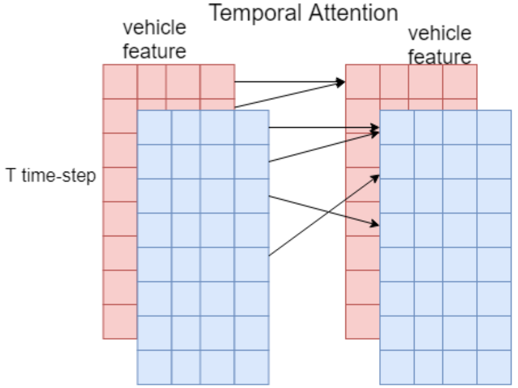
- Spatial-temporal encoder to fuse the information of other time-steps and the other car.

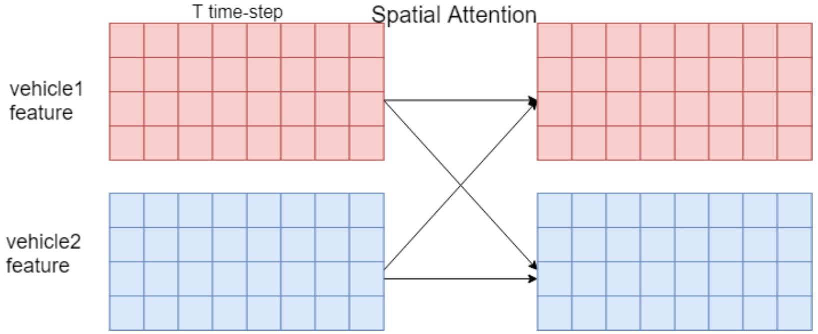
- Improve the generalization ability

Train on FT and validate on SR

IoU6 Acc: 20.4%(origin best) -> 32.4%


Only Temporal Attention


Temporal Attention


Spatial Attention

About Rotation and Scale

- Improve the generalization ability as well:

Train on FT and validate on SR

IoU6 Acc: 32.4% -> 43.7%

- Scale data with rotation to mean=0 and std=1 and keep the magnitude difference of x and y

Train on FT and validate on SR

IoU6 Acc: 43.7% -> 49.2%

Train on FT and finetune on SR:

IoU6 Acc: 45.5% (Train only on SR) -> 68.2%

Semi-supervised Dataset

- Label for entire trajectory:

stop-sign: 0-1s->negative, 1-3s->unknown, >3s->positive

TTC: 0-3s->positive, 3-8->unknown, >8s negative

- Label for each time-step:

Only label negative time-step, others set as unknown

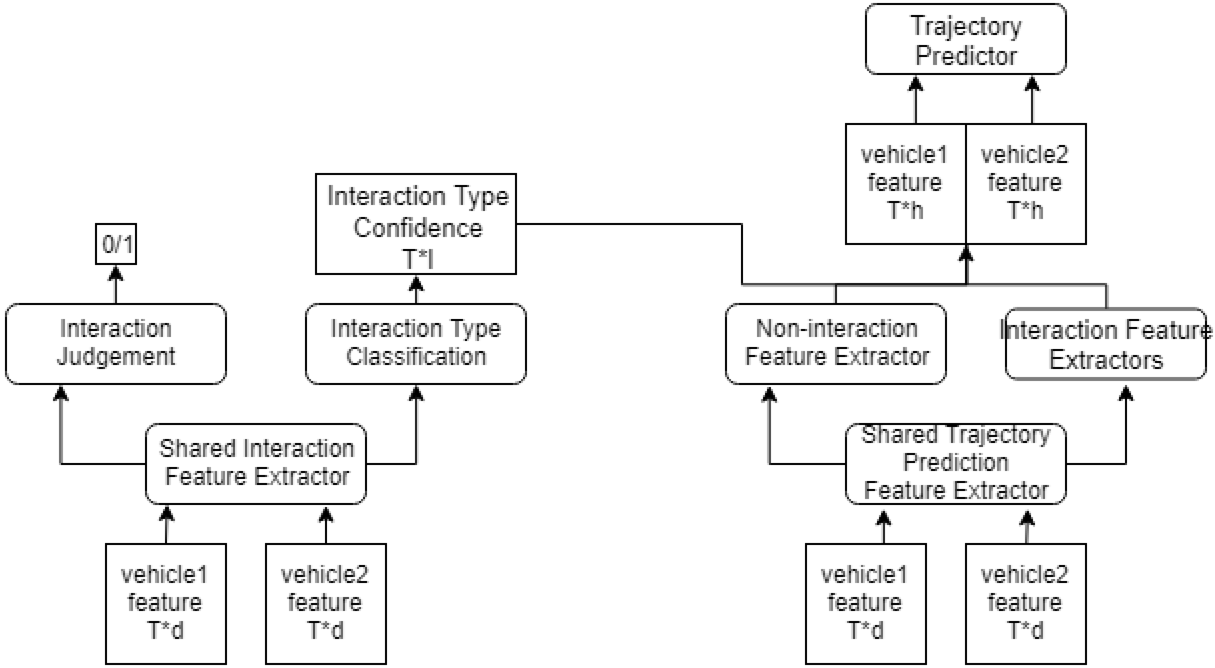
- All scenarios except NGSIM, HighD, CHN_merging, EUR_VA

Semi-supervised Model

- Interaction Judgement Part uses ST-transformer as feature extractor

- In Interaction Judgement Part, each time-step could see all the other time-step.

- To alleviate unbalanced problem, only samples with trajectory label 1/unknown could have other kinds of loss.



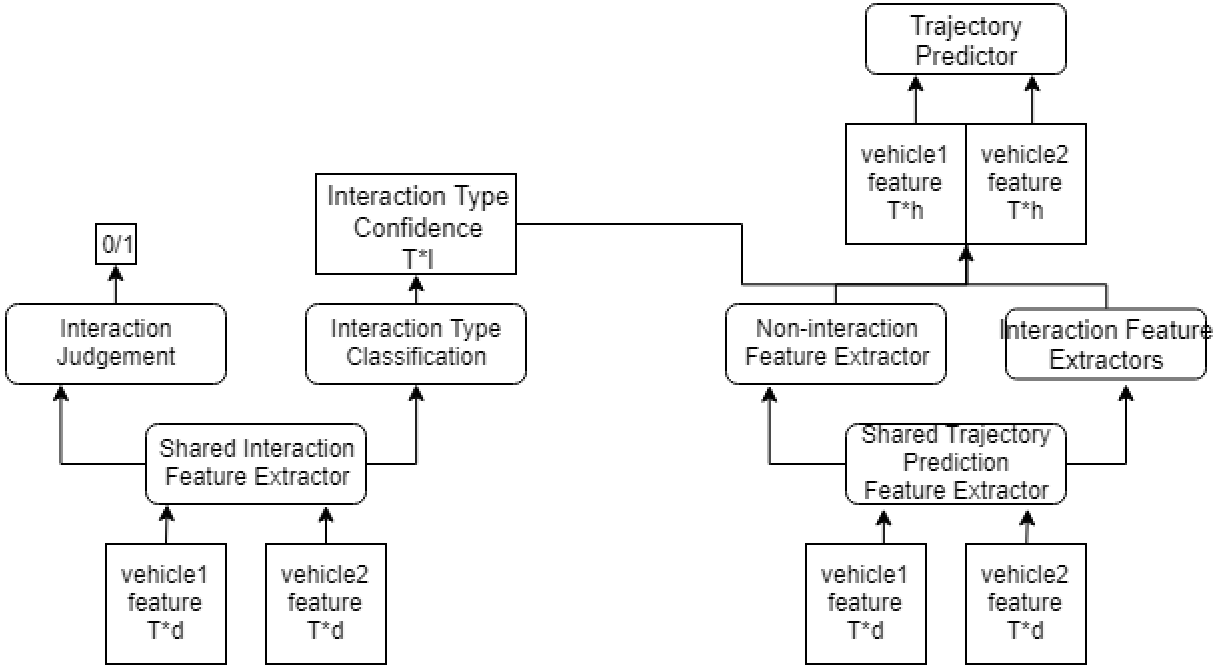
Trajectory Predictor
vehicle1 feature T*h
vehicle2 feature T*h
Interaction Type Confidence T*l
0/1
Interaction Judgement
Interaction Type Classification
Non-interaction Feature Extractor
Interaction Feature Extractors
Shared Interaction Feature Extractor
Shared Trajectory Prediction Feature Extractor
vehicle1 feature T*d
vehicle2 feature T*d
vehicle1 feature T*d
vehicle2 feature T*d

Semi-supervised Model

- In Trajectory Prediction Part, each time-step could only see the time-step before it.

- Non-interaction Feature Extractor only fuse the information of each vehicles itself.

- Each Interaction Feature Extractor uses ST-transformer



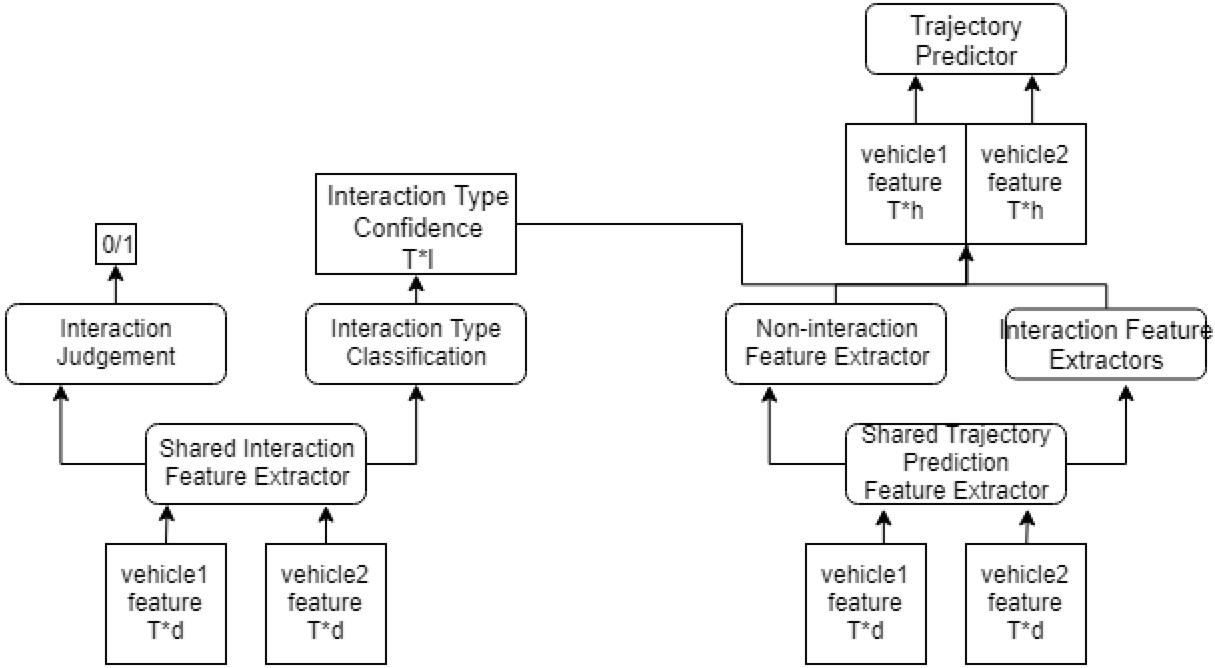
Trajectory Predictor
vehicle1 feature T*h
vehicle2 feature T*h
Interaction Type Confidence T*l
0/1
Interaction Judgement
Interaction Type Classification
Non-interaction Feature Extractor
Interaction Feature Extractors
Shared Interaction Feature Extractor
Shared Trajectory Prediction Feature Extractor
vehicle1 feature T*d
vehicle2 feature T*d
vehicle1 feature T*d
vehicle2 feature T*d

Semi-supervised Model

- At each time-step, the features of each vehicle are determined by their interaction types.

- Use LSTM with interaction features to do prediction

- Trick: to make the edge types of different time-steps smoother, use average pooling for Interaction Type Judgement Features



Trajectory Predictor
vehicle1 feature T*h
vehicle2 feature T*h
Interaction Type Confidence T*l
0/1
Interaction Judgement
Interaction Type Classification
Non-interaction Feature Extractor
Interaction Feature Extractors
Shared Interaction Feature Extractor
Shared Trajectory Prediction Feature Extractor
vehicle1 feature T*d
vehicle2 feature T*d
vehicle1 feature T*d
vehicle2 feature T*d

Loss Function

- Trajectory Classification Loss: predict whether a pair of vehicles interact (Cross Entropy Loss): 0/1/-100

$$L_{traj\_cls} = -\sum_{k} y_k log p'_k$$

- Trajectory Prediction Loss: predict next t step's relative movement

$$L_{traj\_pred} = \left|\left|pred - gt\right|\right|^2$$

- Prior Distribution (KL Loss)

Label Information: [0.5874, uniform]

$$L_{prior} = \sum_{k} p_k log \frac{p_k}{p'_k}$$

Loss Function

- Unsupervised Edge Entropy Loss: make model surer about edge types

$$L_{ent} = -\sum_{k} p'_k log p'_k$$

- Supervised Edge Loss (Cross Entropy Loss): 0/-100

$$L_{sup} = -\sum_{k} y_k log p'_k$$