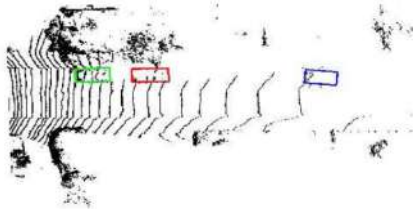
Deep-Prediction for Self-Driving Cars

Abhay Gupta, Nitin Singh
Advisor: Prof. Jeff Schneider

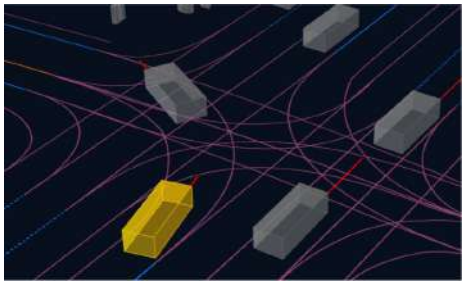
Motivation

Predicting behavior of traffic actors (vehicles/pedestrians/bicyclists) to prevent accidents and aid in better planning for Self-Driving Vehicles (SDVs)



Problem

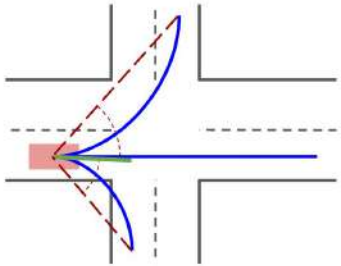
Simultaneously predict all possible trajectories of traffic actors given HD Maps of the surroundings of a SDV



Solution

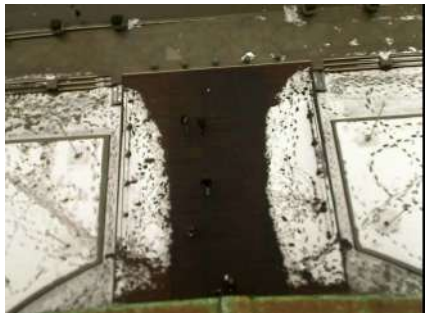
1. Traditional Methods:
    a. Constant Velocity Model
    b. Unscented/Extended Kalman Filter
2. Deep Learning Methods:
    a. Intermediate Representations
    b. Model interactions of traffic actors
    c. Model non-linear structure of motion

???



Spring 2019

Pedestrian Datasets



ETH

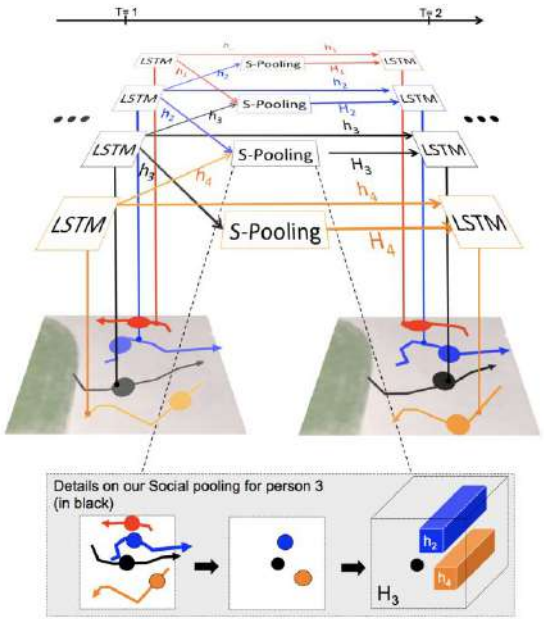
HOTEL

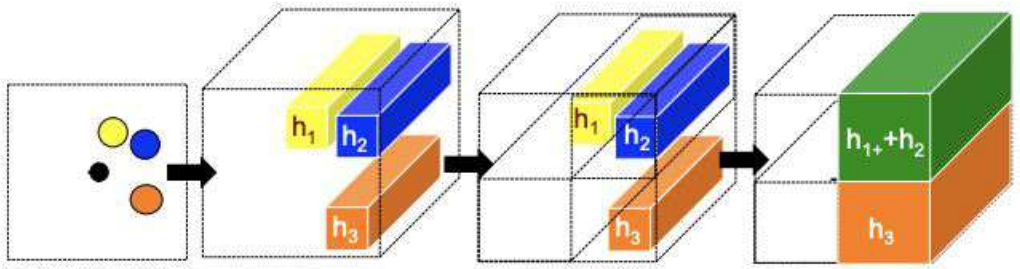
ZARA

UNIVERSITY

S. Pellegrini, A. Ess, and L. Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In Computer Vision–ECCV 2010, pages 452–465.Springer, 2010
L. Leal-Taix´e, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. InCVPR, pages 3542–3549. IEEE,2014

Social LSTM[1]



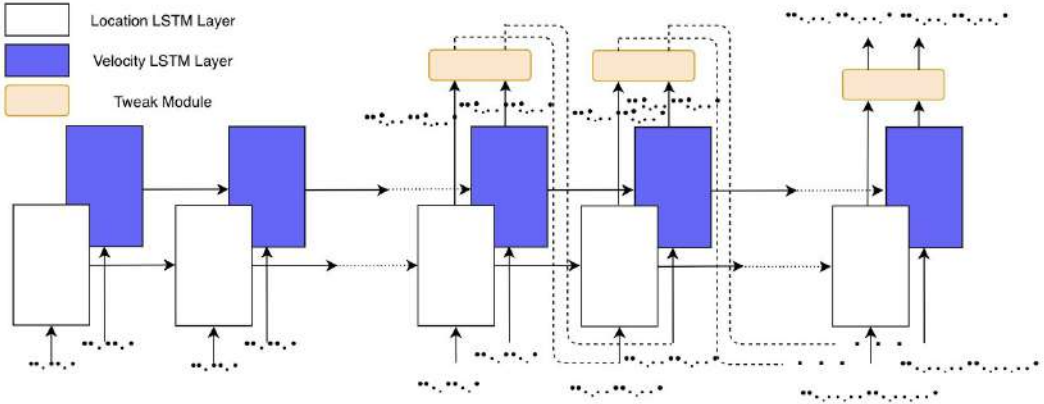


Combines social information in a local neighborhood and creates an aggregated representation

1 - Alahi, Alexandre, et al. "Social lstm: Human trajectory prediction in crowded spaces." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

2 - Bishop, Christopher M. *Mixture density networks*. Technical Report NCRG/4288, Aston University, Birmingham, UK, 1994.

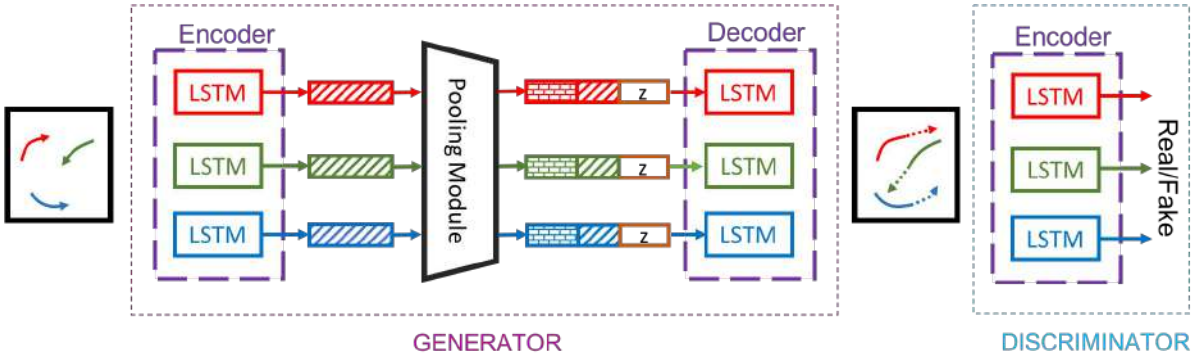
Location-Velocity-Attention LSTM



Attention is used to provide a weighted combination of location based prediction and velocity based prediction..

Xue, Hao, Du Huynh, and Mark Reynolds. "Location-Velocity Attention for Pedestrian Trajectory Prediction." *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019.

Social GAN



1. Scene-scale Pooling instead of neighborhood pooling
2. GANs - emulate more natural trajectories
3. Max-Pool -- helps to learn order invariant symmetric representations

Gupta, Agrim, et al. "Social gan: Socially acceptable trajectories with generative adversarial networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

Performance

1. **Average Displacement Error (ADE)** -The mean square error (MSE) over all estimated points of a trajectory and the true points
2. **Final Displacement Error (FDE)** - The mean square error (MSE) at the predicted final destination and the true final destination of the trajectory

1. Error reported in **meters**
2. Annotations are done at every 0.4 seconds
3. Predictions are done for 12 timesteps(4.8 secs)

Results

Prediction Length (4.8 sec) - ADE / FDE

	Constant Velocity	Vanilla LSTM	Social LSTM	Social GAN (k=20)	LVA LSTM
BIWI ETH	0.86 / 2.38	1.09 / 2.41	1.09 / 2.35	0.70 / 1.28	1.16/2.72
BIWI Hotel	0.37 / 0.81	0.86 / 1.91	0.79 / 1.76	0.48 / 1.02	2.15/5.18
UCY Zara1	0.41 / 0.98	0.41 / 0.88	0.47 / 1.00	0.34 / 0.69	0.48/1.14
UCY Zara2	0.36 / 0.82	0.52 / 1.11	0.56 / 1.17	0.31 / 0.65	0.39/0.99
UCY University	0.46 / 1.07	0.61 / 1.31	0.67 / 1.40	0.56 / 1.18	0.68/1.59

Social GAN performs best on ETH and ZARA dataset.
Constant Velocity performed well on HOTEL and University dataset.

Models

Model	Multi-Agent	Multi-Modal	Stochastic	Real-time Inference
Social-LSTM[1]	✓	X	X	✓
LVA-LSTM[2]	✓	X	X	X
Social-GAN[3]	✓	X	✓	✓

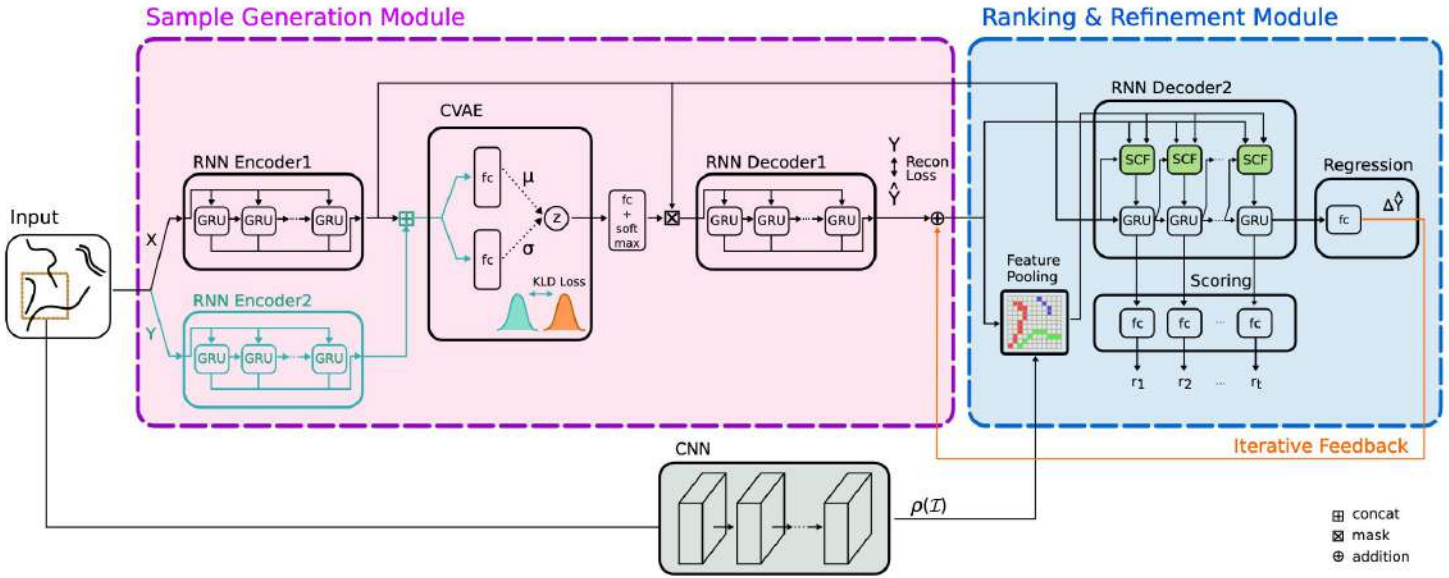
1 - Alahi, Alexandre, et al. "Social lstm: Human trajectory prediction in crowded spaces." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
2 - Xue, Hao, Du Huynh, and Mark Reynolds. "Location-Velocity Attention for Pedestrian Trajectory Prediction." *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019.
3 - Gupta, Agrim, et al. "Social gan: Socially acceptable trajectories with generative adversarial networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

Autonomous Vehicles Dataset

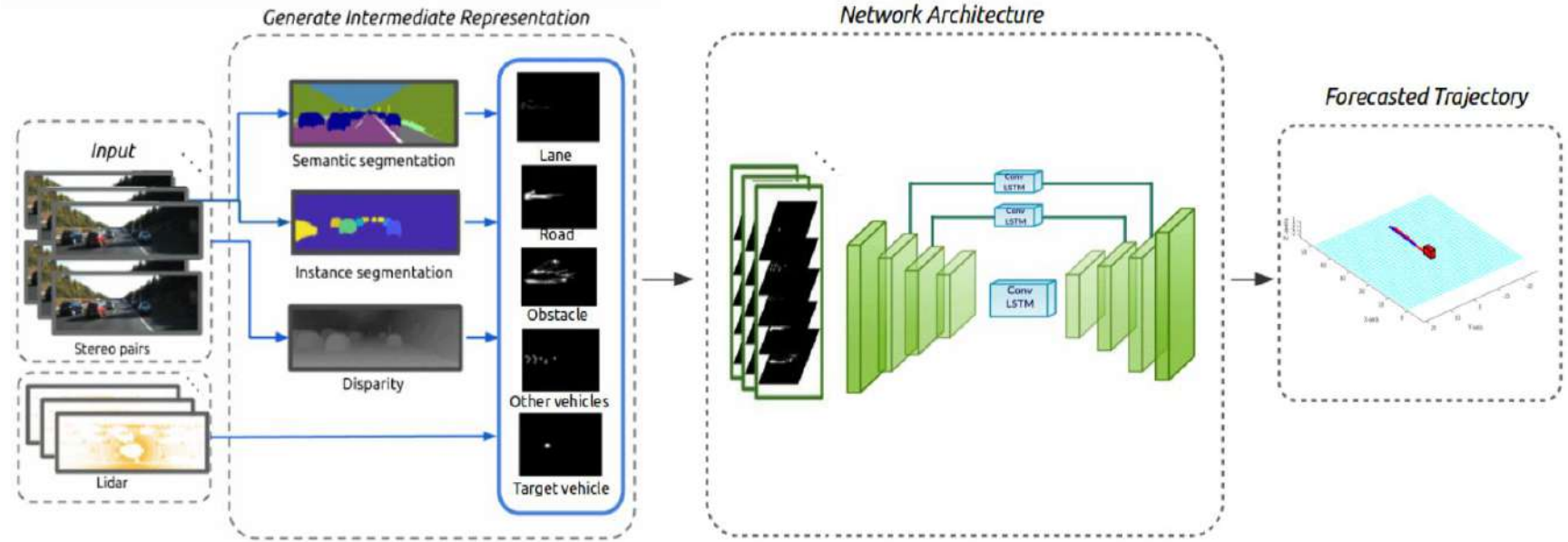
KITTI[1] Dataset

1 - Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012.

DESIRE


Sample Generation Module
Ranking & Refinement Module
Input
X
Y
RNN Encoder1
RNN Encoder2
GRU
CVAE
fc
μ
σ
z
KLD Loss
fc + soft max
RNN Decoder1
Y
Recon Loss
Ŷ
RNN Decoder2
SCF
Regression
Δ Ŷ
Feature Pooling
Scoring
r1
r2
rt
Iterative Feedback
CNN
ρ(I)
concat
mask
addition

INFER



Srikanth, Shashank, Junaid Ahmed Ansari, and Sarthak Sharma. "INFER: INtermediate representations for FuturE pRediction." *arXiv preprint arXiv:1903.10641* (2019).

Performance

1. **Average Displacement Error (ADE)** -The mean square error (MSE) over all estimated points of a trajectory and the true points

1. Error reported in **meters**
2. History is available for 2 seconds
3. Predictions are done for 4 seconds
4. To match metrics across papers, errors reported at each 1s interval

Results

Prediction Length – 4s (ADE)

	1s	2s	3s	4s
CVM	0.70	1.41	2.12	2.99
DESIRE-SI	0.31	0.70	1.39	2.12
INFER (ConvLSTM)	0.76	1.23	1.60	1.96
INFER (SkipLSTM)	0.53	0.89	1.22	1.56

All errors are reported in meters

Models

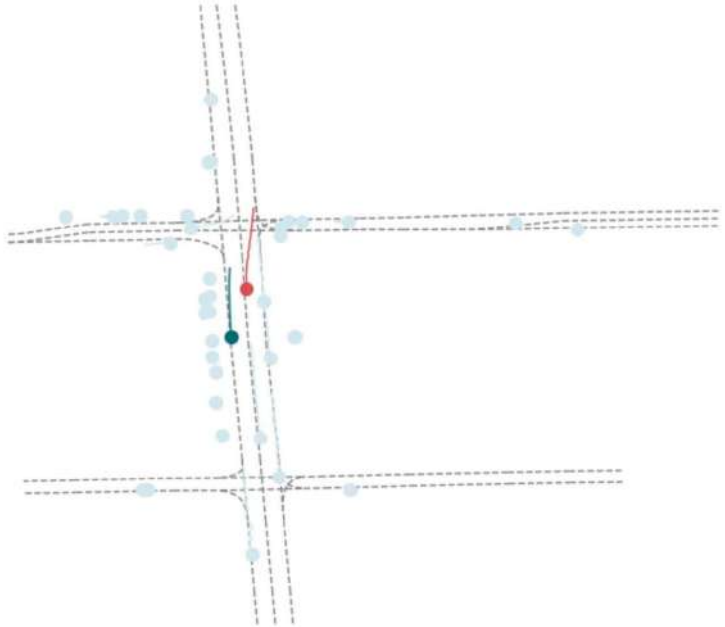
Model	Multi-Agent	Multi-Modal	Stochastic	Real-time Inference
Constant Velocity	✓	X	X	✓
DESIRE[1]	X	✓	✓	X
INFER[2]	✓	✓	X	X

1 - Lee, Namhoon, et al. "Desire: Distant future prediction in dynamic scenes with interacting agents." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
2 - Srikanth, Shashank, Junaid Ahmed Ansari, and Sarthak Sharma. "INFER: INtermediate representations for FuturE pRediction." *arXiv preprint arXiv:1903.10641* (2019).

Fall 2019

Argoverse[1] Motion Forecasting

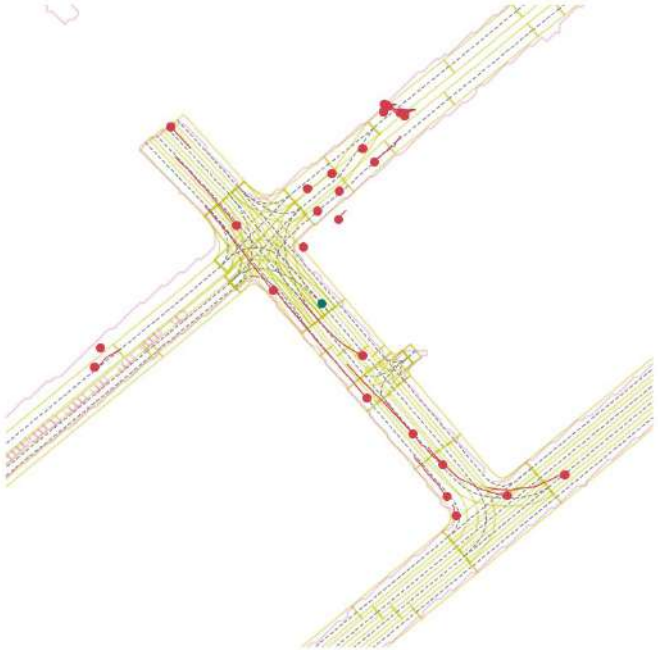
1. 3,23,557 sequences - each 5s long
   a. 2,05,942 train seq.
   b. 39,472 validation seq.
   c. 78,143 test sequences
2. Sampled at 10 Hz
3. 3s forecasting - modelling complex scenarios
   a. Traversing an intersection
   b. Slowing for merging vehicle
   c. Accelerating after a turn
   d. Slowing for pedestrian on road



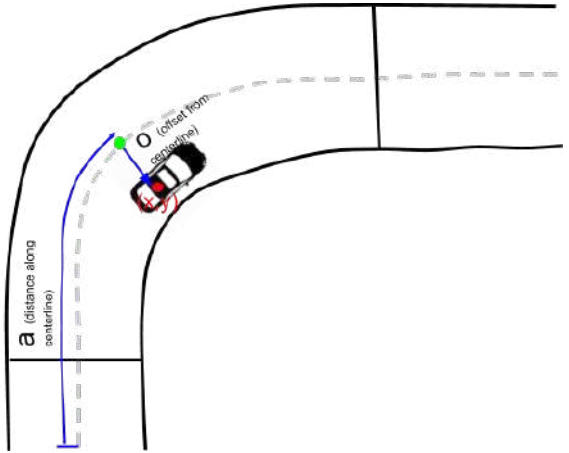
Green - AV; Red - Agent of Interest;
Light Blue - Other actors in scene

1 - Chang, Ming-Fang, et al. "Argoverse: 3D Tracking and Forecasting with Rich Maps." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

Data Formats

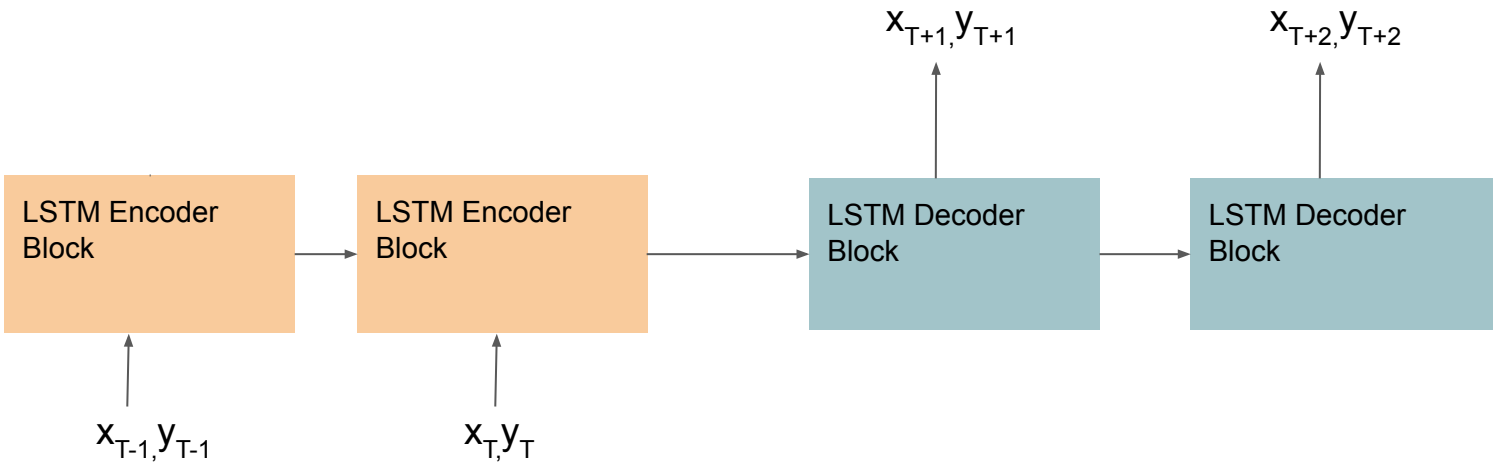


XY data

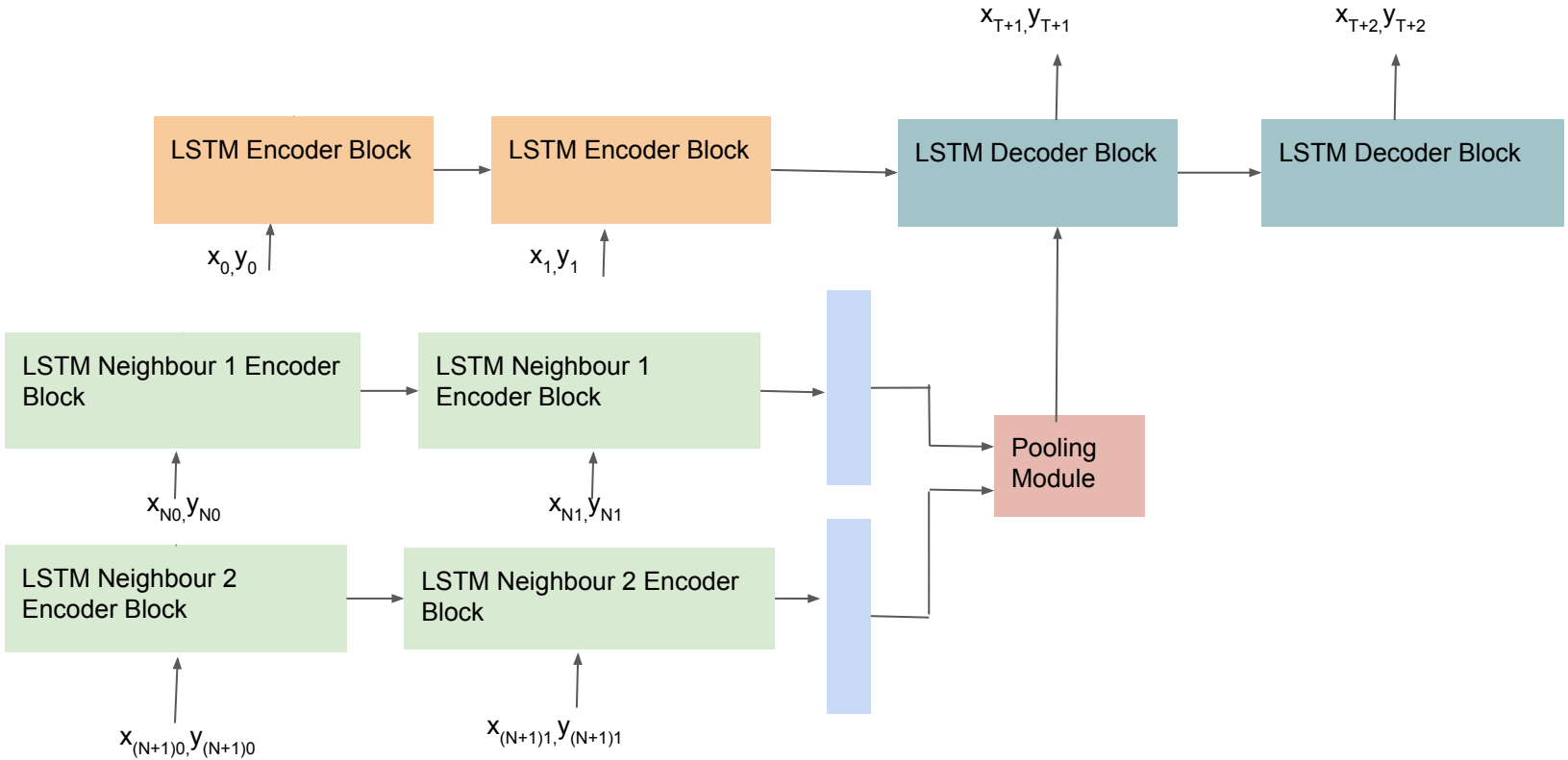


Centerline data (normal-tangential (nt))

LSTM Encoder-Decoder



Social LSTM Encoder-Decoder



Performance

1. **Average Displacement Error (ADE)** -The mean square error (MSE) over all estimated points of a trajectory and the true points
2. **Final Displacement Error (FDE)** - The mean square error (MSE) at the predicted final destination and the true final destination of the trajectory

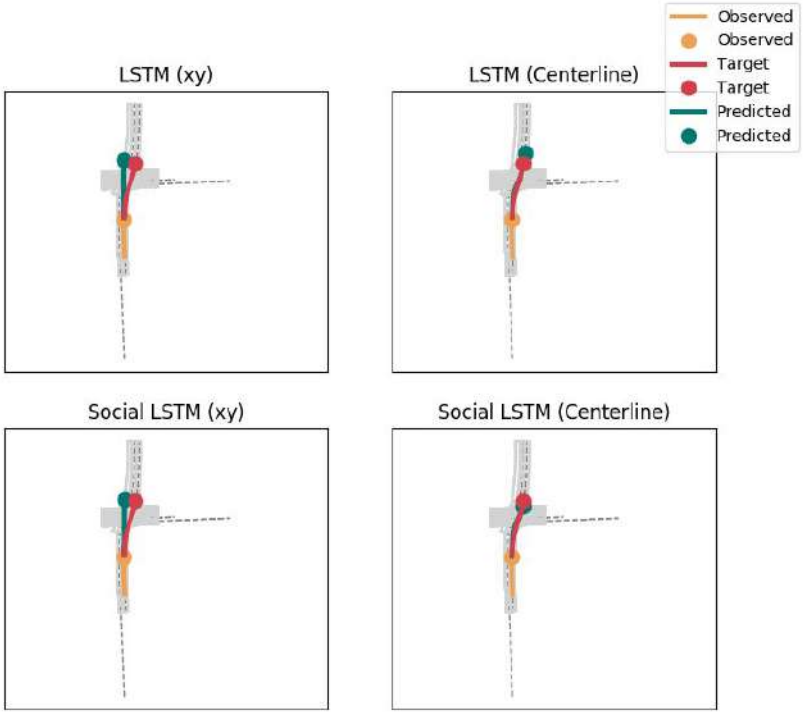
1. Error reported in **meters**
2. Annotations are done at every 10Hz
3. Predictions are done for 3 seconds (30 predictions)
4. Error reported for 1 and 3 second - ADE / FDE.

Results

	1 sec.		3 sec.	
Model	**ADE (m)**	**FDE (m)**	**ADE (m)**	**FDE (m)**
LSTM (xy)	0.68	1.02	1.88	4.19
Social LSTM (xy)	0.71	1.00	1.80	3.89
LSTM (nt)	0.73	1.04	1.79	3.69
Social LSTM (nt)	0.73	1.01	1.65	3.33
Constant Velocity	0.99	1.73	3.02	6.48

1. Results with the centerline (nt) data are better than xy data for 3 sec period.
2. Results with the Social models are better than the non-social counterparts for 3sec prediction.
3. Results for 1 sec prediction are quite similar for all the models.

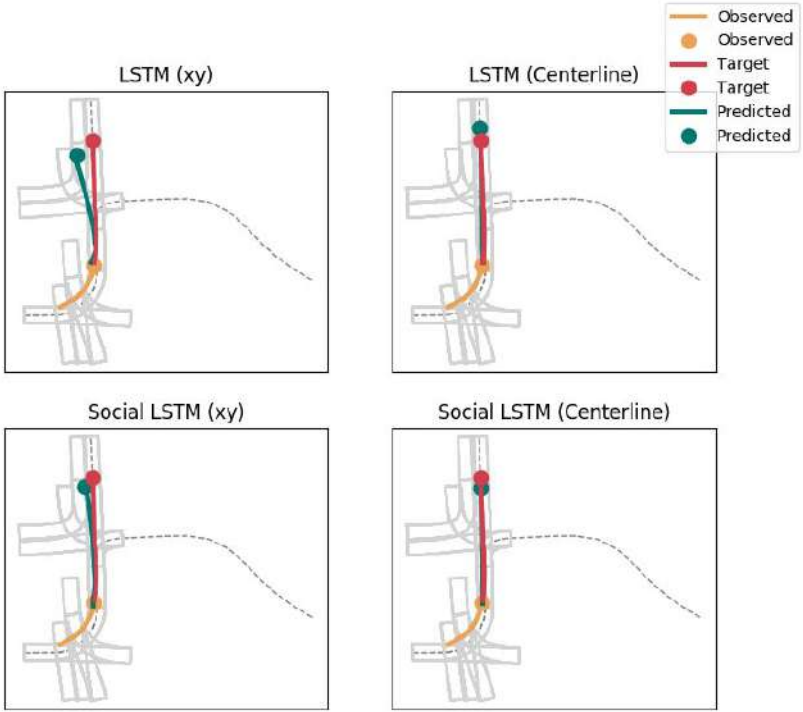
Results (Comparison)



1. The LSTM-models with xy data cannot model the lane curve ahead on the road while the LSTM model with centerline can.

2. The Social-LSTM model can accurately predict the speed in the trajectories whereas the non-social model face some errors with it.

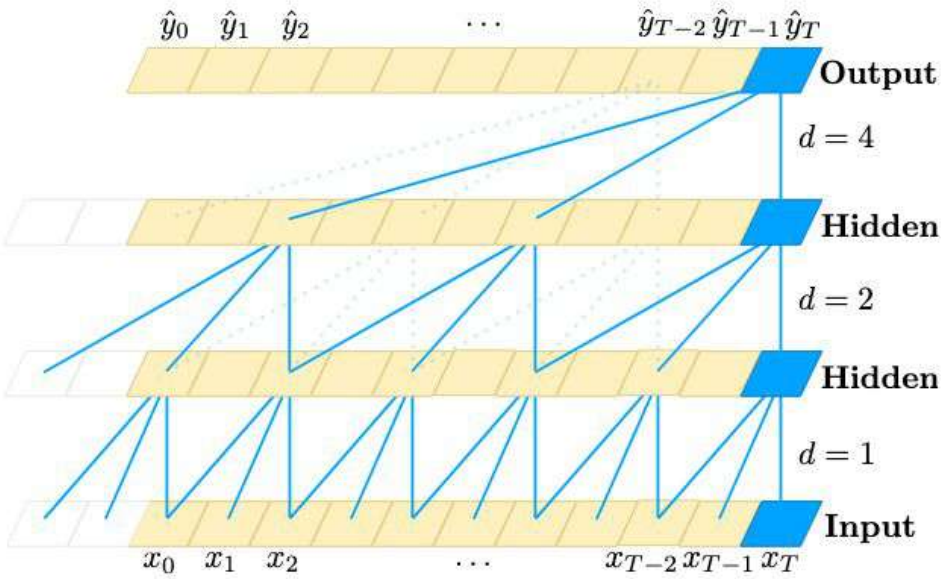
Results (Comparison)



1. The LSTM-models with xy data cannot model the lane curve ahead on the road while the LSTM model with centerline can.

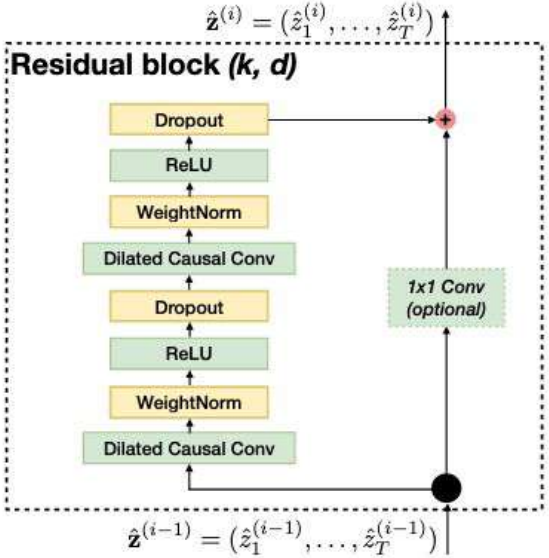
2. The Social-LSTM model can accurately predict the speed in the trajectories whereas the non-social model face some errors with it.

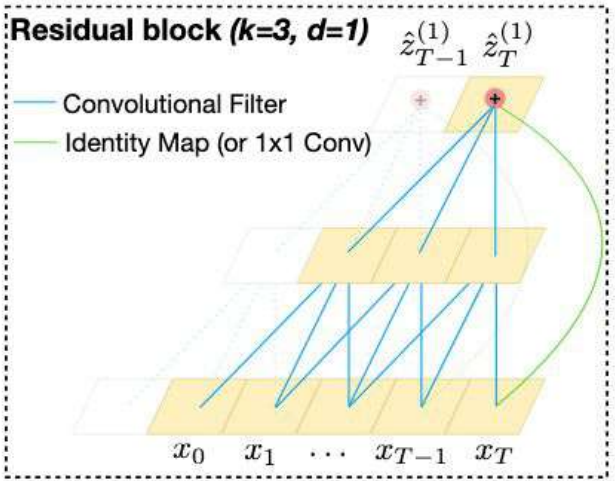
Temporal Convolutional Networks[1]



1 - Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." *arXiv preprint arXiv:1803.01271* (2018).

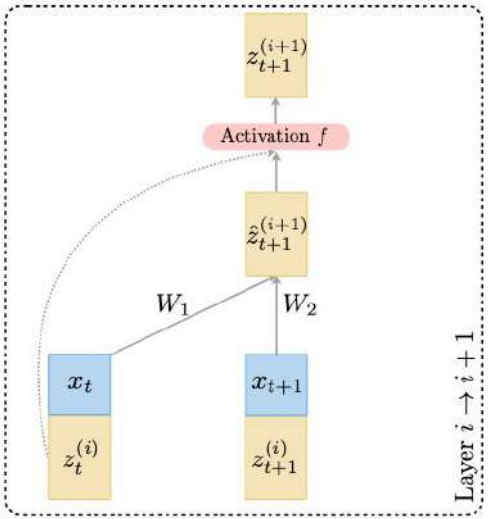
Temporal Convolutional Networks[1]



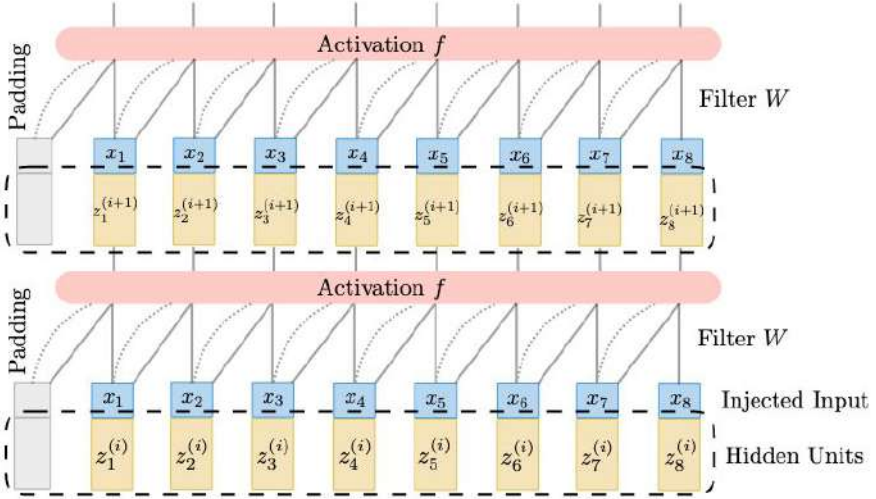


1 - Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." *arXiv preprint arXiv:1803.01271* (2018).

Trellis Networks[1]



(a) TrellisNet at an atomic level



(b) TrellisNet on a sequence of units

1 - Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "Trellis networks for sequence modeling." *International Conference on Learning Representations* 2019.

Equilibrium Points and the DEQ Model

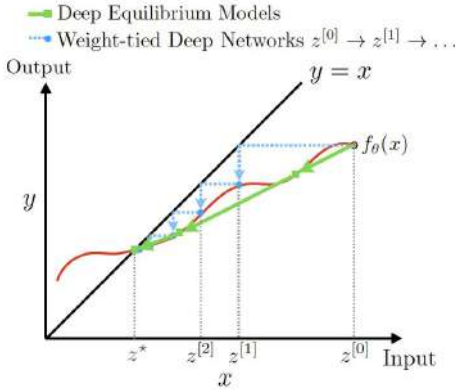
We now can think of a "deep" network as repeated applications of a function:

$$\mathbf{z}^{[i+1]} = f(\mathbf{z}^{[i]}; \mathbf{x})$$

In practice, after these types of models converge to an equilibrium point (i.e., an "infinite depth" network):

$$\lim_{i \to \infty} \mathbf{z}_{1:T}^{[i]} = \lim_{i \to \infty} f_\theta(\mathbf{z}_{1:T}^{[i]}; \mathbf{x}_{1:T}) \equiv f_\theta(\mathbf{z}_{1:T}^\star; \mathbf{x}_{1:T}) = \mathbf{z}_{1:T}^\star$$

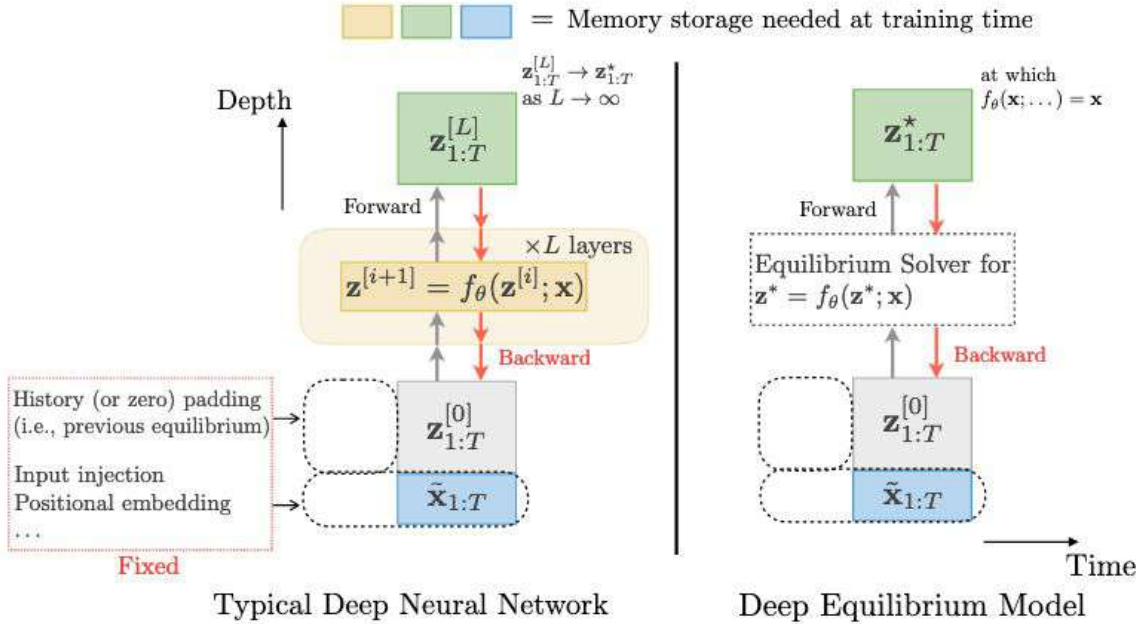
Deep Equilibrium (DEQ) Model: directly find this equilibrium/stable point via root-finding (eg, Broyden's method), rather than just iterating the forward model, and apply implicit differentiation for backpropagation.



1 - Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "Deep equilibrium models." *Advances in neural information processing systems* 2019.

Overview of DEQ Approach

To compare conventional deep networks with DEQ:



* slide courtesy of S. Bai (MLD-Phd @ CMU)

Performance

1. **Average Displacement Error (ADE)** -The mean square error (MSE) over all estimated points of a trajectory and the true points
2. **Final Displacement Error (FDE)** - The mean square error (MSE) at the predicted final destination and the true final destination of the trajectory

1. Error reported in **meters**
2. Annotations are done at every 10Hz
3. Predictions are done for 3 seconds (30 predictions)
4. Error reported for 1 and 3 second - ADE / FDE (Following Argoverse metric reporting)

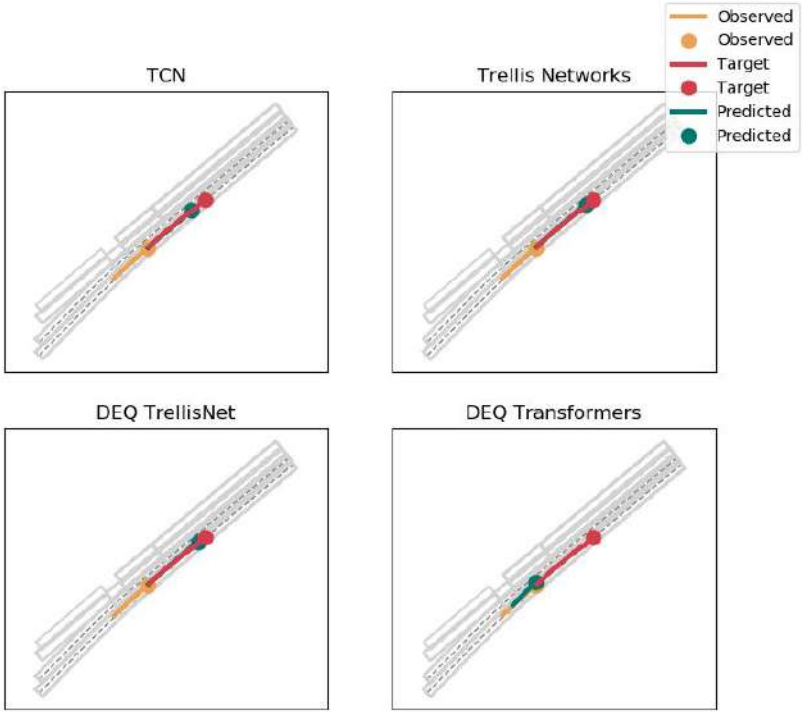
Results

	1 sec.		3 sec.	
Model	**ADE (m)**	**FDE (m)**	**ADE (m)**	**FDE (m)**
TCN	0.65	1.04	1.95	4.21
Trellis Networks	0.62	1.01	1.87	4.11
DEQ-TrellisNet	0.615	0.99	1.85	4.09
DEQ-Transformer[1]	3.5	4.07	4.76	6.60
Constant Velocity	0.99	1.73	3.02	6.48

1. We see that Trellisnets perform better than TCN models.
2. Also for the same model, DEQ framework performs slightly better.
3. Transformers do not seem to work well - in fact they perform worse than constant velocity

* Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.

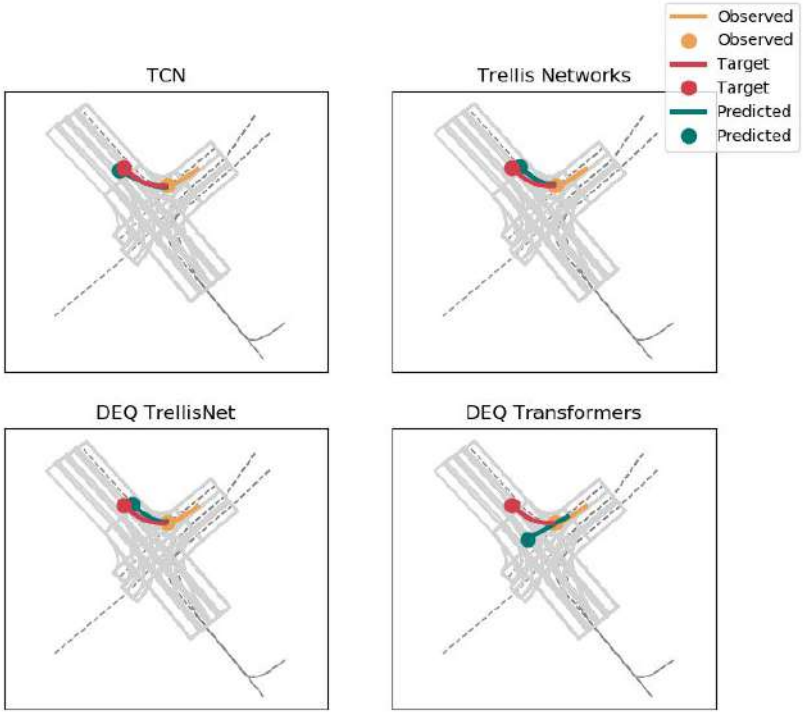
Results (Comparison)



1. TrellisNets perform better than TCN. DEQ Trellis have similar performance as TrellisNets.

2. Transformers learn very poorly because of shifted multi-head attention.

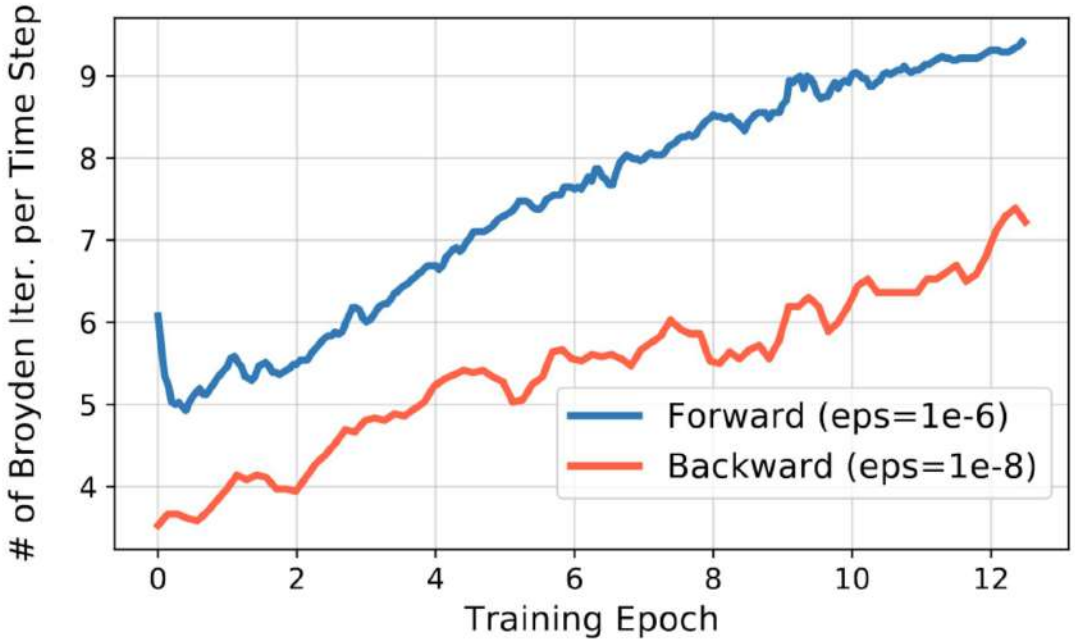
Results (Comparison)



1. TCN and TrellisNets are able to model curvature information to some extent

2. Transformers generally follow the curvature of the input trajectory

DEQ-Trellis Broyden Learning Graphs



*eps is stopping criteria hyper-parameter

Inference Analysis

Model	# Layers	Prediction Time - 30 steps (in ms)*	Memory on GPU*
TCN	20	1.2	2.5GB
Trellis Networks	40	3.5	16.9GB
DEQ-TrellisNet	40	5.8 (~1.65x slower)	2.39 (~7x smaller)
DEQ-Transformer	25	8.9	6.9GB

* All numbers reported to RTX 2080Ti GPU Cards on a batch of 8 samples

Conclusion

All Results

	1 sec.		3 sec.	
Model	**ADE (m)**	**FDE (m)**	**ADE (m)**	**FDE (m)**
LSTM (xy)	0.68	1.02	1.88	4.19
Social LSTM (xy)	0.71	1.00	1.80	3.89
LSTM (nt)	0.73	1.04	1.79	3.69
Social LSTM (nt)	0.73	1.01	1.65	3.33
TCN	0.65	1.04	1.95	4.21
Trellis Networks	0.62	1.01	1.87	4.11
DEQ-TrellisNet	0.615	0.99	1.85	4.09
DEQ-Transformer	3.5	4.07	4.76	6.60
Constant Velocity	0.99	1.73	3.02	6.48

Synopsis

Data:

1. Centerline data generally accounts for the curved error cases which comes from xy data
2. Providing embeddings of centerline curvature helps improve model inference.
3. The major error cases in all models is due to sudden acceleration/deceleration of agent - which means modeling velocity is an important aspect of this data
4. Use of Social information from the neighbouring trajectories improves prediction of agent.

Models:

1. Constant Velocity performs more poorly than the deep learning strategies - both recurrent and feedforward.
2. Modelling social interactions in the model is useful for prediction.
3. Feedforward are extremely fast in both training and inference
4. DEQ Models have much smaller memory consumption, while maintaining good inference speeds - proving to be a strong direction to consider for future improvements
5. Transformers seem to be a bad choice based on current evaluations for this type of data.

Thank you