Reproducibility report for It Is Not the Journey but the Destination

Anonymous Author(s)
Affiliation
Address
`email`

Reproducibility Summary

The following paper is a reproducibility report for Its not the journey but the destination [3]. The basic code was made available by the author via the GitHub repository <link>. To reproduce the rest of the ablation studies mentioned in the paper, we had to modify the model structure accordingly. The well-commented version of the code containing all ablation studies performed derived from the original code is available at <link> with proper instructions to execute experiments in ReadMe.

Scope of Reproducibility

We have verified all claims made by the paper and results from different experiments mentioned inside the paper to support the claims. The central claim of PECNet was to improve state-of-the-art performance on the Stanford Drone trajectory prediction benchmark by 20.9 percent and on the ETH/UCY benchmark by 40.8 percent, which has been verified to be true.

Methodology

The PECNet model was trained on the drone dataset with social pooling at different conditioned points and on the ETH/UCY datasets without social pooling. Results that were obtained matched with those claimed in the paper. Furthermore, the trained model was evaluated on the drone dataset (with social pooling) at different values of evaluated samples (referenced as 'k' in the paper). For the latter, GitHub was used as a reference with author-given code.

Results

Overall, we were able to reproduce all the results mentioned in the paper within 5% error compared to what was mentioned in the paper.

What was easy

Verification of the claims against the ETH/UCY benchmarks and Stanford drone benchmark trajectory prediction with the PECNet models was an easy task.

What was difficult

For the datasets of ZARA1 and ZARA2, there were gaps in the sequence of frames, and thus interpolation was done to ensure the continuity of way-points. This caused the ADE and FDE errors to increase. Also, to maintain common frequency for all the datasets, they were down-sampled accordingly. For the conditioned way-point positioning experiment (with and without ORACLE) experiment, ADE had to be calculated from 11 predicted positions to not alter the structure of the model, and FDE was also calculated from the 11th point. However, due to it, some ADE fluctuations

Submitted to 4th ML Reproducibility Challenge 2020. Do not distribute.

after the sixth way-point (and later) were larger than the claimed results. Similar fluctuations were observed for FDE as well, but the relative trends support the paper's claim.

Communication with original authors

We have not contacted any of the original authors as all the results were reproduced satisfactorily.

1 Introduction

The paper reproduced in this report aims to tackle multiple pedestrian trajectory predictions using rich multi-modal predictions for the use of autonomous vehicles, social robots, etc. Earlier approaches to this problem have been auto-regressive in nature, i.e., using n points (or analogically, data from the last t seconds) from the dataset to produce the immediately next point, and then this process is recurred.

In this paper, the end-point distribution conditioned on the past trajectory and the past trajectory features are modeled separately for each pedestrian. The future trajectory points are predicted based on the past and features from other pedestrians via social pooling. An assumption in this model is the absence of passive pedestrians or the fact that each pedestrian has an actual preconceived end-point or destination and is motivated to reach it.

To formulate this report, we have experimented on the author's code by adding/removing social pooling layers, using truncation tricks, visualisation tools, and changing between CVAE and VAE architectures to verify all the claims made by the author described in detail below. We also performed some experiments such as shifting origin to the current point, using different architecture for encoder and decoder networks with the hope of improving the results, which are also described in detail at the end.

2 Scope of reproducibility

The paper revolves around the claim that an important component of predicting the trajectory is the destination in multi trajectory forecasting. If the destination for the pedestrian is clear, then the trajectory can be easily resolved using a separate network that takes the past trajectory and the destination as input taking into account social interactions among fellow pedestrians. Hence the central idea and claim of the paper is to use Conditional Variational Auto Encoder (CVAE) to get the latent variable encoding conditioned on the destination from the ground truth, use the latent variable to infer the predicted destination, and use it for predicting the rest of the future trajectory. We take k samples of the latent variable for testing purposes to predict k different admissible trajectories as output for different destinations derived from the latent encoding. The overall reduction in the value of best ADE and FDE values for the Stanford Drone, ETH/UCY datasets by using the CVAE network is the central claim of the paper.

To support the argument that indeed given the destination, the rest of the predicted trajectory contributes much less error than the previous state of the art methods such as SGAN, which directly predict the future trajectory, the paper performs an ablation study where they give the ground truth of a way-point which they call as oracle instead of the best one from taking k samples of the latent variable to get the decoupled error of predicting the trajectory. The results strongly support the argument.

Further, they also experimented with different values of k to show that FDE tends to 0 as k increases and ADE tends to a certain value, which also shows the decoupled error in predicting the rest of the trajectory.

This paper also introduces a non-local social pooling layer and a "truncation-trick," which improves diversity and multi-modal trajectory prediction performance.

Hence the claims can be summarized as follows:-

1. Conditioning the destination on the past trajectory using CVAE helps in explicit decoupling of the destination prediction and path prediction errors. It hence helps reduce the destination prediction error and the subsequent path prediction error.

2. Using the social pooling layer helps reduce the error in predicting the path given the history and the destination.

3. Using truncation trick i.e., truncating the distribution for fewer values of k from which samples are taken helps reduce the destination prediction error. Also, taking a higher sigma value for larger values of k reduces the error.

3 Methodology

We used the GitHub repository provided by the author as the base. However, it only contained the base model for results on the drone dataset. In order to reproduce the rest of the experiments, we had to make changes accordingly.

3.1 Model descriptions

The model used in the paper consists of 2 parts:

First, the CVAE or Conditional Variational AutoEncoder to get the representation of the latent variable conditioned on destination and given the past trajectory.

Second, the predictor network consists of social pooling layers and an MLP network to get the future trajectory.

A representative diagram of the network is given in figure 1 and the architecture parameters are shown in table 1.

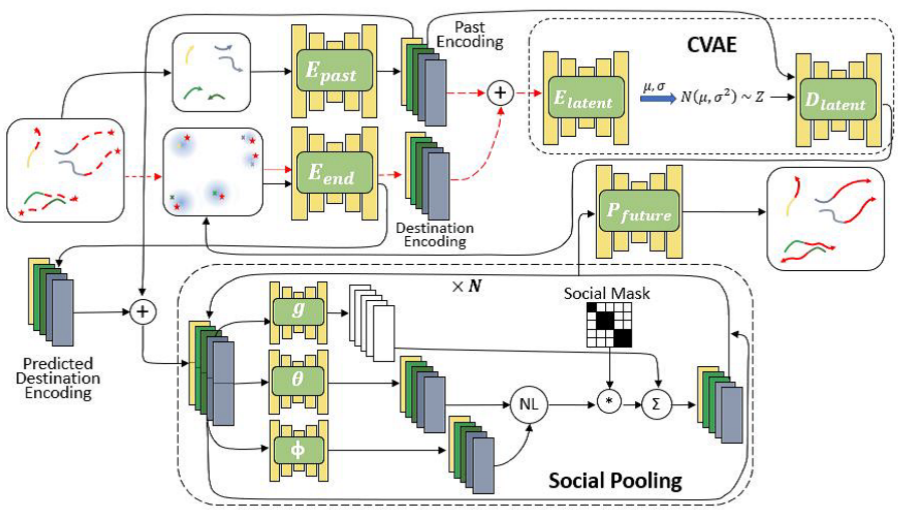

Figure 1: Model architecture

$$ADE = \frac{\sum_{j=t_i+1}^{t_p+t_f+1} \|\hat{\mathbf{u}}_j - \mathbf{u}_j\|_2}{t_f} \tag{1}$$

$$FDE = \left\|\hat{\mathbf{u}}_{t_p+t_f+1} - \mathbf{u}_{t_p+t_f+1}\right\|_2 \tag{2}$$

$$\mathcal{L} = \lambda_1 \underbrace{D_{KL}(\mathcal{N}(\mu,\sigma)\|\mathcal{X}(0,\mathbf{I}))}_{KL \text{ Div in latent space}} + \lambda_2 \underbrace{\left\|\hat{\mathcal{G}}_c - \mathcal{G}_c\right\|_2^2}_{AEL} + \underbrace{\left\|\hat{\mathcal{T}}_f - \mathcal{T}_f\right\|^2}_{ATL} \tag{3}$$

3.2 Datasets

We used Stanford Drone [5] and ETH [4] / UCY [2] datasets. The Stanford drone dataset was given in the author's code, but ETH/UCY was not given, so we took the dataset from opensource source.

	Network Architecture
E_{way}	2 -> 8 -> 16 -> 16
E_{past}	16 -> 512 -> 256 -> 16
E_{latent}	32 -> 8 -> 50 -> 32
D_{latent}	32 -> 1024 -> 512 -> 1024 -> 2
θ, Φ	32 -> 512 -> 64 -> 128
g	32 -> 512 -> 64 -> 32
$P_{predict}$	32 -> 1024 -> 512 -> 256 -> 22

Table 1: Model Architecture

3.3 Hyperparameters

We used Hyperparameters given in the paper. We occasionally changed them accordingly to perform the ablation studies described below.

3.4 Experimental setup

We ran code in google colab with GPU (NVIDIA-SMI 450.36.06 Driver Version: 418.67 CUDA Version: 10.1).

3.5 Computational requirements

Typically, it took less than an hour to train the model both for the drone and ETH/UCY datasets.

4 Results

The following experiments/ablation studies support the claims made earlier. A detailed description of the experiments and their results to support the claim are listed below:-

4.1 Experiment on drone dataset (with and without social pooling, truncation trick)

Stanford drone dataset: We did it with social pooling and got results within 95% accuracy from claim results. The preprocessed dataset for train and test were given on GitHub (by author). We used them to verify the results. We did two experiments with n-samples 5 and another with n-samples 20 as required for reproducing the results in the first table of the paper.

	O-S-TT	O-TT	Ours	PECNet-Ours
K	20	20	5	20
ADE	10.56 / 10.47	10.23 / 10.19	12.79 /14.16	9.96/10.04
FDE	16.72 / 16.43	16.29 / 15.9	25.88 / 26.73	15.96/16.20

Table 2: Comparisons of our results against those of the authors' and previous state-of-the-art methods. -S' '-TT' represents ablations of our method without social pooling truncation trick. We report results for in pixels for both K = 5 20 and for several other values of K. The format for each cell is <claimed result> / <reproduced result>

4.2 Experiment on ETH/UCY datasets (with and without social pooling, truncation trick

ETH-UCY: ETH/UCY dataset consists of 5 scenes eth, hotel, univ, zara1, zara2 extracted from another source <link> because, in the paper, the source was not mentioned. We Followed the conventional leave-one-out approach, i.e., trained on 4 sets and tested on the last set to get the results. We verified results within 98% accuracy from claimed results. The dataset was further downsampled by 6 to get a 0.4 second gap between consecutive frames as demanded by the paper. The result is shown below in the table. With these 2 experiments, the reduction in error with respect to the previous results by using CVAE and subsequent reduction by using social pooling layer and truncation trick can be demonstrated.

	O-S-TT		PECNet-Ours	
Datasets	ADE	FDE	ADE	FDE
ETH	0.58/.57	0.96/.98	0.54/.53	0.87/.87
HOTEL	0.19/.20	0.34/.35	0.18/0.18	0.24/0.23
UNIV	0.39/0.32	0.67/0.53	0.35/0.32	0.60/0.49
ZARA1	0.23/0.23	0.39/0.37	0.22/0.23	0.39/0.35
ZARA2	0.24/0.20	0.35/0.33	0.17/0.20	0.30/0.32

Table 3: Quantitative results obtained versus those of the authors' (in the form of ours/authors'). 'Our-S-TT' represents ablation of our method without social pooling truncation trick. The format for each cell is <claimed result> / <reproduced result>

4.3 Change in the structure of CVAE

In this experiment during training, the ground truth Eend (G_k) was used to predict the future T_f instead of the one obtained from the latent variable. We did it on the Stanford drone dataset with social pooling and got results within 95% accuracy from the claim results.

ADE : 10.87 / 10.945

FDE : 17.03 / 16.277

4.4 Effect of Number of samples (K)

We did this experiment on the Stanford drone dataset with social pooling. We trained the PECNet model with default sigma values and test on different k-sample value with and without truncation. Without truncation for k-sample<=3 we used σ with variance 1 and for k-sample > 3 we used σ with variance 1.3. With truncation for k-sample > 3 we used σ with variance 1 and for k-sample<=3 we used σ with variance c * $\sqrt{k-1}$. In this experiment we got results within 95 accuracy from the claim results.

	1	2	3	5	10	50	100	1000	10000
ADE	24.29	18.457	16.25	14.16	12.04	8.99	8.208	6.81	6.27
FDE	51.84	37.65	32.15	26.73	21.10	12.27	9.73	4.66	2.46
Truncated-ADE	17.62	16.67	15.71	14.788	12.10	8.54	7.70	6.39	6.02
Truncated-FDE	35.02	32.67	30.34	28.57	21.49	11.27	8.54	3.54	1.66

Table 4: Effect of no of samples (K) on ADE, FDE, Truncated-ADE, Truncated-FDE

4.5 Conditioned Way-point positions Oracles

In this experiment, we conditioned on future trajectory points other than the last observed point, which we refer to as way-points. This was not clear in the paper about how to calculate FDE error because we can not predict last observed point in the model so we calculated FDE from the 11th point of the predicted trajectory. It was done in two parts.

1. **With oracle**: During prediction of future trajectory (at time of testing and validation), we gave ground-truth value of conditioned point instead of the best guessed one from sampling to predict trajectory from the model. The Stanford drone data set with social pooling and truncation trick was used to match with the results on paper.

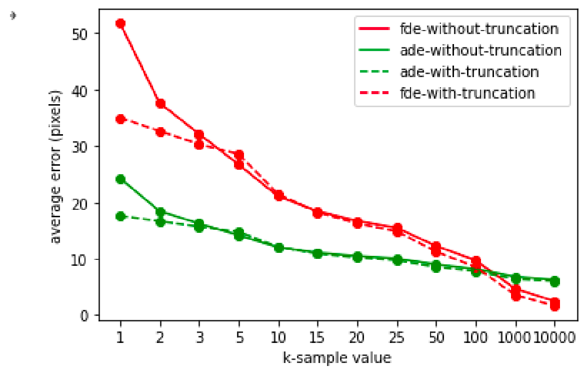

Figure 2: Graph of errors

2. **Without oracle**: The same thing was done here except during prediction of the future trajectory the best guess for the conditioned point(predicted by model) was taken (at time of testing and validation). Way-point Prediction Error was calculated as difference between ground truth of conditioned point and the one predicted by the model.

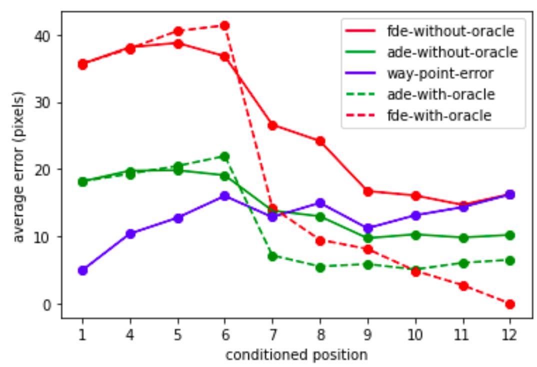

Figure 3: Graph of errors

	1	4	5	6	7	8	9	10	11	12
ADE	18.16	19.76	19.83	19.08	13.82	12.98	9.73	10.29	9.83	10.218
FDE	35.64	38.125	38.77	36.79	26.61	24.18	16.73	16.08	14.69	16.27
Way-point error	4.93	10.38	12.75	16.01	12.86	14.98	11.207	13.12	14.336	16.23
Oracle ADE	18.17	19.30	20.46	21.94	7.17	5.52	5.87	5.074	6.0552	6.51
Oracle FDE	35.68	37.93	40.54	41.38	14.30	9.48	8.13	4.892	2.745	0.0

Table 5: Conditioned Way-point positions and Oracles

5 Discussion

From each of the experiments, the claims made by the paper as described above can be strongly supported and empirically proved. In order to further study the choice of structure of the network, we performed the following experiments:-

5.1 Reference shift <link>[1]

We took the reference of the trajectory for each pedestrian as the current point instead of the first point of the past trajectory. This helped the CVAE network to get a better representation of the destination point as all input past trajectories have a common last point, which makes it easier for the encoder and decoder network to function; also, the predictor and social pooling network gets more easily trained. This showed about 8% further decrease in ADE and FDE metrics for drone dataset as follows:-

ADE : 8.64

FDE : 14.64

5.2 Using encoder and decoder LSTM network instead of MLP <link>[1]

We used encoder LSTM instead of MLP to form the encoding of the past trajectory to accommodate variable length of past trajectory and form a better representation as to the input temporal data. Also, we used the decoder LSTM network to predict the rest of the trajectory given the destination. However, the FDE error reduced by about 5 %, but the ADE is surprisingly more, demonstrating that decoder LSTM does not perform well given the destination point.

ADE : 26.9

FDE : 14.3

References

[1] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: `https://www.wandb.com/%5C%7D`.

[2] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. "Crowds by Example". In: *Computer Graphics Forum* 26.3 (2007), pp. 655–664. DOI: `https://doi.org/10.1111/j.1467-8659.2007.01089.x`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2007.01089.x`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2007.01089.x`.

[3] Karttikeya Mangalam et al. "It is Not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Aug. 2020.

[4] S. Pellegrini et al. "You'll never walk alone: Modeling social behavior for multi-target tracking". In: *2009 IEEE 12th International Conference on Computer Vision*. 2009, pp. 261–268. DOI: `10.1109/ICCV.2009.5459260`.

[5] A. Robicquet et al. "Stanford Drone Dataset". In: (). URL: `http://cvgl.stanford.edu/projects/uav_data/`.