

Article

Deep Learning-Based Multimodal Trajectory Prediction with Traffic Light

Seoyoung Lee , Hyogyeeong Park , Yeonhwi You , Sungjung Yong  and Il-Young Moon * 

Department of Computer Science and Engineering, Korea University of Technology and Education,
Cheonan 31253, Republic of Korea; cengyi@koreatech.ac.kr (S.L.); sjshk@koreatech.ac.kr (H.P.);
wlwhs3258@koreatech.ac.kr (Y.Y.); zeros952@koreatech.ac.kr (S.Y.)

* Correspondence: iymoon@koreatech.ac.kr; Tel.: +82-41-560-1493

Abstract: Trajectory prediction is essential for the safe driving of autonomous vehicles. With the advancement of advanced sensors and deep learning technologies, attempts have been made to reflect complex interactions. In this study, we propose a deep learning-based Multimodal Trajectory Prediction method that reflects traffic light conditions in complex urban intersection situations. Based on existing state-of-the-art research, the multi-path of multi-agents was predicted using a generative model, and the actor's trajectory information, state, social interaction, and traffic light state, and scene context were reflected. Performance evaluation was conducted using metrics commonly used to evaluate the performance of stochastic trajectory prediction models. This study is meaningful in that trajectory prediction was performed by reflecting realistic elements of traffic lights in a complex urban environment. Future research will need to be conducted on efficient ways to reduce time and computational performance while reflecting different real-world environments.

Keywords: trajectory prediction; multimodal; traffic light; deep learning; GAN; LSTM; autonomous vehicle



Citation: Lee, S.; Park, H.; You, Y.; Yong, S.; Moon, I.-Y. Deep Learning-Based Multimodal Trajectory Prediction with Traffic Light. *Appl. Sci.* **2023**, *13*, 12339. <https://doi.org/10.3390/app132212339>

Academic Editor: Juan A. Gómez-Pulido

Received: 11 September 2023
Revised: 6 November 2023
Accepted: 12 November 2023
Published: 15 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Trajectory prediction for autonomous vehicles is essential for efficiently and safely navigating complex traffic conditions in urban centers. Previously, simple physics-based motion models were presented that represented vehicles as dynamic objects governed by the laws of physics, predicting their future behavior based primarily on conditions, such as the speed and acceleration of the target vehicle and the coefficient of friction of the road surface. In a slightly more advanced manner, a maneuver-based model based on adjustments to the driver's intentions was also presented [1,2].

More recently, research has attempted to reflect interactions between agents and the influence of surrounding objects on agents. Reflecting social interactions for a large number of agents is a very challenging task, but a way to consider its impact was proposed in [3], involving a social pooling technique via LSTM, and various variations and advances have also been proposed. In addition, with the emergence of high-performance deep learning models for image recognition, such as Transformer, ResNet, and VGG, the autonomous vehicle field is also attempting to perform situational awareness through vision processing, and there is active research on combining sensor information for situational awareness to understand the surrounding environment and perform trajectory prediction tasks [4,5]. Cameras are still the most commonly utilized sensors, but with the development of advanced sensors such as LiDAR and Radar, attempts to incorporate them into the perception phase of autonomous driving are also a current research trend, and datasets for autonomous vehicles containing advanced sensor information are emerging [6,7]. It has been, and continues to be, a catalyst for autonomous driving research and development at various research institutions. The approach to trajectory prediction path generation is changing from the deterministic regressors of the past to generative, multi-modal methods for safer and more efficient path prediction. Attempts are also being made to reflect the influence of various

objects in the driving environment. Therefore, advanced algorithmic research is being conducted to re-reflect various real-world information, such as route prediction considering the interaction of various multi-agents using deep learning technology and route prediction, reflecting the surrounding environment, such as lane shape, etc. Refs. [8–12].

However, to date, trajectory prediction models for autonomous driving that incorporate signal information have not been well studied, and datasets with traffic light information are still very rare. Existing studies have attempted to predict trajectories by presenting the results of inputting traffic light information as additional contextual information, the data-driven modeling of driver reactions to traffic lights, or modeling with discrete dependencies; however, they seem to have difficulty with generalization [13–15]. For efficient route prediction in urban areas, it is necessary to reflect the interaction of not only the target vehicle, but also nearby agents and objects, and to experience various trial and error and test scenarios.

In this paper, we propose a trajectory prediction model that reflects scene context information, interactions between agents, and the past state information of agents, as well as traffic light information, which is the closest traffic information, for route prediction reflecting the urban environment. We evaluated the performance of our model on the waterloo dataset [16], which provides multi-agent information and Bird's Eye View scene information for urban intersection situations.

The main approaches and contributions of our model are as follows:

- Among the surrounding environment elements, we incorporated traffic light information. We considered various signal information environment that occur in urban intersection situations.
- Previously, sequence prediction with social interaction mainly used the trajectory information of the agent as input. Our model used not only trajectory information but also agent state information such as speed and acceleration as input.
- To reflect the scene context, we used the ResNet18 model, which shows high performance in image recognition, to extract image features and reflect them in the model.
- Predict multiple paths based on generative models.

2. Related Works

2.1. LSTM for Sequence Prediction

A recurrent neural network is a type of artificial neural network in which, unlike a forward neural network, hidden nodes are connected by directed edges to form a circular structure. It is a deep learning model that is often used to process time-dependent or sequential data because the previous state can affect the next state. Recently, it has been frequently applied in the fields of natural language processing and time-series prediction. Representative examples include recurrent neural networks (RNN), gated recurrent units (GRU), and long short-term memory (LSTM). In particular, the emergence of LSTM models has enabled the storage of long-term temporal data, and the performance and utilization of recurrent neural network (RNN) models have been widely demonstrated. The trajectories of pedestrians and dynamic agents also provide spatial location information over time, and many recent attempts have been made to model and predict them using recurrent neural networks [17–19].

2.2. Agent-Agent Models

For unmanned vehicles to drive without collisions, it is essential for them to not only consider nearby static objects and terrain characteristics but also to change their speed, direction, etc., through interactions with other vehicles. Therefore, learning models that reflect the interaction between agents to perform trajectory prediction have also emerged. Following the success of RNN-based models in modeling the dynamic behavior of agents, there is a sense in which interdependencies can be considered. Representative state-of-the-art models in this field include Social LSTM, Social Attention, Convolutional Social Pooling, and Social GAN [20–22].

2.3. Multi Modal and Generative Modeling

Given the number of real-world factors that must be considered for driverless driving and the complexity of variables and priorities, it can be argued that providing multiple paths to a goal is an essential feature for real-world driving. The GAN model comprises two networks: generators and discriminators. The generator generates multiple possible paths, and the discriminator identifies the feasibility of the generated paths. GAN models are widely used in trajectory prediction owing to their high performance and ability to generate multiple paths [23,24]. In addition, models that represent multiple probability distributions for future trajectories, such as Multimodal Trajectory Prediction (MTP) and Multi-Path, have also been widely studied [15,25].

2.4. Scene Context-Aware Prediction

For trajectory prediction, there are many attempts to recognize the surrounding environment from 2D images or map information based on vision sensors. The inputs for scene context are not only RGB images from traditional camera sensors, but also LiDAR point clouds, HD maps, semantic segmentation maps, etc. In addition, the shape of the image can be categorized into Ego-Camera View and Bird's Eye View (BEV) depending on the perspective, and the BEV shape has the advantage of easily reflecting more information through a wider field of view, so it is more commonly used in deep learning-based trajectory prediction models to date [26]. Based on OpenCV for collecting traffic flow parameters and traffic information, research is underway to establish an adaptive model for a stationary background and to detect, track, and monitor vehicles through image processing [27].

In a related study, SoPhie used attention techniques for scene information to focus on salient regions [20]. Social-BiGAT applied graph attention network (GAT) and self-attention to images to consider social and physical features of the scene [28]. Trajectron++ shows the results of the impact of heterogeneous data for trajectory prediction through map encoding [29].

3. Materials and Methods

Trajectory prediction is commonly thought of as the regression of a sequence of output trajectories on a sequence of input trajectories. In this study, we define the problem as predicting the future position of all agents in a scene based on their past position inputs.

3.1. Problem Definition

In this study, the input trajectory information, status information, and traffic light information for all agents in the scene at that time were obtained. Past trajectory information for input is denoted by X as follows. Based on this information, the future trajectory information for all agents in the scene to be predicted is written as Y , as follows. In addition, the agent's state value is indicated as S , and v is the agent's speed (km/h), $acc1$ represents the tangential acceleration (m/s^2), $acc2$ is the lateral acceleration (m/s^2), and ang is the vehicle angle at each location.

$$\begin{aligned}
 X &= \left[(x_i^{t-h}, y_i^{t-h}), \dots, (x_{i+N}^{t-h}, y_{i+N}^{t-h}), \dots, (x_{k+M}^t, y_{k+M}^t) \right], \\
 S &= \left[(v_i^{t-h}), \dots, (v_{i+N}^{t-h}), \dots, (v_{k+M}^t) \right], \\
 SS &= \left[(v_i^{t-h}, acc1_i^{t-h}, acc2_i^{t-h}), \dots, (v_{i+N}^{t-h}, acc1_{i+N}^{t-h}, acc2_{i+N}^{t-h}), \dots, (v_{k+M}^t, acc1_{k+M}^{t-h}, acc2_{k+M}^{t-h}) \right], \\
 SA &= \left[(v_i^{t-h}, acc1_i^{t-h}, acc2_i^{t-h}, ang_i^{t-h}), \dots, (v_{i+N}^{t-h}, acc1_{i+N}^{t-h}, acc2_{i+N}^{t-h}, ang_{i+N}^{t-h}), \dots, (v_{k+M}^t, acc1_{k+M}^t, acc2_{k+M}^t, ang_{k+M}^t) \right], \\
 Y &= \left[(x_i^{t+1}, y_i^{t+1}), \dots, (x_{i+N}^{t+1}, y_{i+N}^{t+1}), \dots, (x_{k+M}^{t+pred}, y_{k+M}^{t+pred}) \right] \quad (1)
 \end{aligned}$$

The trajectory information is given as position information in a two-dimensional plane of x and y . (x_i^{t-h}, y_i^{t-h}) represents the position before h s of the first agent in frame i , and $(x_{k+M}^{t+pred}, y_{k+M}^{t+pred})$ represents the position after $pred$ s of the M -th agent in frame k . The

time interval t is 0.4 s, and the trajectory information predicted through the model will be written as $\hat{Y}$. h denotes the past time seconds of the agent to be observed, and $pred$ denotes the future time seconds of the agent to be predicted. The time interval was set in increments of 0.4 s. The predicted data are denoted by $\hat{Y}$. We set the problem to simultaneously predict all trajectories of all agents in the same frame at a certain time.

3.2. Model Overview

The structure of the proposed model is as follows:

The overall structure of the proposed model is shown in Figure 1. Scene images are passed to a CNN-based scene context encoder to extract features. Traffic light information and agent state values corresponding to each agent at each time from the dataset are input through an LSTM-based encoder. Trajectory information was entered into a pooling model, which was then merged to enter into a decoder for the predicted trajectories. The trajectory generated by the decoder generates a feasible trajectory using the discriminator.

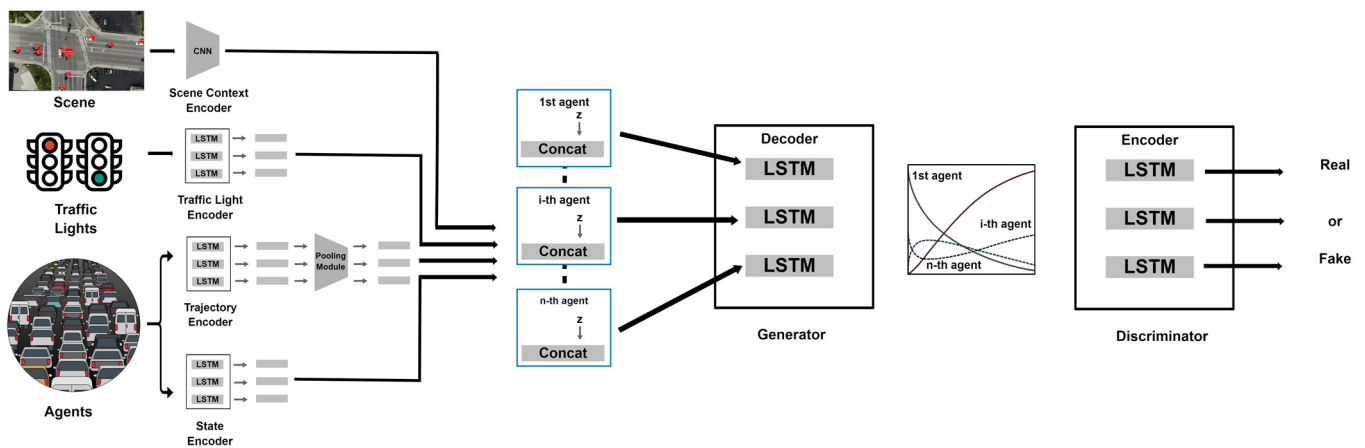


Figure 1. The proposed model architecture.

3.3. Generating Traffic Signal Information

Traffic light information in all areas inside the frame at a specific time included in the dataset was combined and classified into 26 states. Subsequently, at each time point, the classified traffic light state values (RGB) were assigned to all agents belonging to each frame. The 2-digit code of the generated signal information is shown in Table A1.

3.4. Model Details

3.4.1. Scene Context Encoder

The scene in the type of BEV (Bird's Eye View) was passed to the CNN-based Scene Context Encoder to extract features. To reduce the size of the operation, the image was resized to 640×360 and put into the encoder. The CNN model used ResNet18, which shows high performance in terms of performance and the speed of image recognition, and the tensor after the last convolution before the fully connected layer as the feature vector. I is the scene image, W_{cnn} is the weight of the CNN model, and R_c is the feature value extracted from the image.

$$R_c = \text{CNN}(I, W_{\text{cnn}}) \quad (2)$$

3.4.2. LSTM-Based Encoder

There are three encoders, and each encoder receives trajectory, state, and traffic light information. An encoder was constructed using an LSTM model to model the time-series characteristics. In this study, the structure and internal dimensions of the encoders used were SGAN (Social GAN), a state-of-the-art model in this field, which is described in detail in [30].

Trajectory information refers to the agent's two-dimensional (x, y) position information at each time and is represented by the UTM coordinate value in the zone of the dataset. The state of an agent includes four values: velocity, tangent acceleration (m/s^2), lateral acceleration (m/s^2), and the vehicle angle. The traffic light information converted the listing of each traffic light value (RGB) at 10 points shown in the frame into a code, and the code was assigned to all agents at every time step. The values at that point in time for each agent were embedded through a single multilayer perceptron (MLP) and converted to a fixed-length vector, which was passed as input to the LSTM. This can be expressed as an equation as follows. In addition, e_{1i}^t , e_{2i}^t , and e_{3i}^t represent a fixed-length vector, $\varphi(\cdot)$ is embedding function, W_X , W_S , and W_T are embedding matrices with ReLU non-linearity and W_{en1} , W_{en2} , and W_{en3} represent the LSTM weight of each LSTM encoder model.

$$\begin{aligned} e_{1i}^t &= \varphi(X_i^t, W_X), h_{ei}^t = \text{LSTM}\left(h_{ei}^{t-1}, e_{1i}^t, W_{en1}\right) \\ e_{2i}^t &= \varphi(S_i^t, W_S), h_{si}^t = \text{LSTM}\left(h_{si}^{t-1}, e_{2i}^t, W_{en2}\right) \\ e_{3i}^t &= \varphi(T_i^t, W_T), h_{ti}^t = \text{LSTM}\left(h_{ti}^{t-1}, e_{3i}^t, W_{en3}\right) \end{aligned} \quad (3)$$

Here, h_{ei}^t , h_{si}^t , and h_{ti}^t denote the hidden state of the LSTM encoder at time t of the i -th agent. In addition, e_{1i}^t , e_{2i}^t , and e_{3i}^t represent a fixed length vector, and W_{en1} , W_{en2} , and W_{en3} represent an LSTM weight.

3.4.3. Social Interaction

To reflect the influence of social context between agents, we utilized trajectory information. To combine the past trajectory information of all agents extracted from the trajectory encoder, we utilized the pooling module proposed by SGAN, which demonstrates the performance of state-of-the-art in this field. In this study, a single MLP was used to compute the relative positions between the target agent and all agents in the frame, which were then combined using the max-pooling technique. Consequently, for each agent, a vector is extracted that combines the distance context between all agents.

$$\begin{aligned} e_{1i}^t &= \varphi(X_i^{t-1}, W_d) \\ P_i &= \text{pool}\left(h_{ei}^{t-1}, \dots, h_{en}^t\right) \end{aligned} \quad (4)$$

3.4.4. Feature Fusion

The feature vectors reflecting the agent's trajectory, state, and signal information extracted from the three encoders described in the previous step are combined with the feature vectors reflecting the social interaction passed through the pooling module. Each feature value is extracted for each agent, and after combining, it is passed as an input to the GAN-based LSTM decoder.

$$\begin{aligned} f_{1i}^t &= \text{MLP}(P_i, e_{1i}^t; W_{f1}) \\ f_{2i}^t &= \text{integrate}\left(R_c^t, f_{1i}^t, h_{si}^t, h_{ti}^t; W_{f2}\right) \end{aligned} \quad (5)$$

3.4.5. GAN-Based Decoder

The combined values from the previous steps are passed through a GAN-based decoder. The generator was trained to return the various expected trajectory distributions. The discriminator is also trained to take the original and predicted trajectories as inputs and classify them as real/fake based on the predicted classification values. This module also returns a sequence of trajectories for each agent and is constructed using an LSTM. The objective function of GAN is given by Equation (7). G wants to decrease the function

V, and D wants to increase. Similar to the conditional GAN, it uses a noise vector z as the input and generates samples [31].

$$h_{dec_i}^t = [f_{2i}^t, z]$$

$$D_{LSTM}(MLP(P_i, h_{dec_i}^{t-1}), e_{1i}^t, W_{decoder}) \quad (6)$$

We also used L2 loss, which measures how far the predicted trajectory is from the actual trajectory, and among the multiple paths, the one with the least distance error was selected. This is expressed in (8). G represents the generator, which takes the latent variable z as the input, where D denotes the discriminator.

$$\min_{G, D} V(G, D) = E_{X \sim P_{data}(x)} [\log D(x)] + E_{Z \sim P(z)} [\log(1 - D(G(z)))] \quad (7)$$

$$L_{variety} = \min_k \|Y_i - \hat{Y}_i^{(k)}\|_2 \quad (8)$$

4. Experiments

4.1. Dataset

To learn and train the model, we used the Waterloo multi-agent traffic intersection dataset [14]. Waterloo is an approximately one-hour video dataset of aerial views of a crowded urban intersection in Waterloo, Canada. The dataset provides 14 data files containing trajectory and context information as well as a video with an agent tag for each frame. We used this dataset to reflect the impact of traffic light information on the trajectory prediction of agents.

The videos in the dataset were framed in 2.5 Hz (0.4 s) increments, and the dataset was preprocessed to provide trajectory information, agent state, signal state, and time for all agents in each frame. The data were split into frames for training, validation, and testing. Partitioning was performed in the following ratio: 5505 frames for training, 2609 frames for validation, and 1454 frames for testing out of a total of 9568 frames. Figure 2. shows a combined image of the 4 scene information for each frame in the dataset.

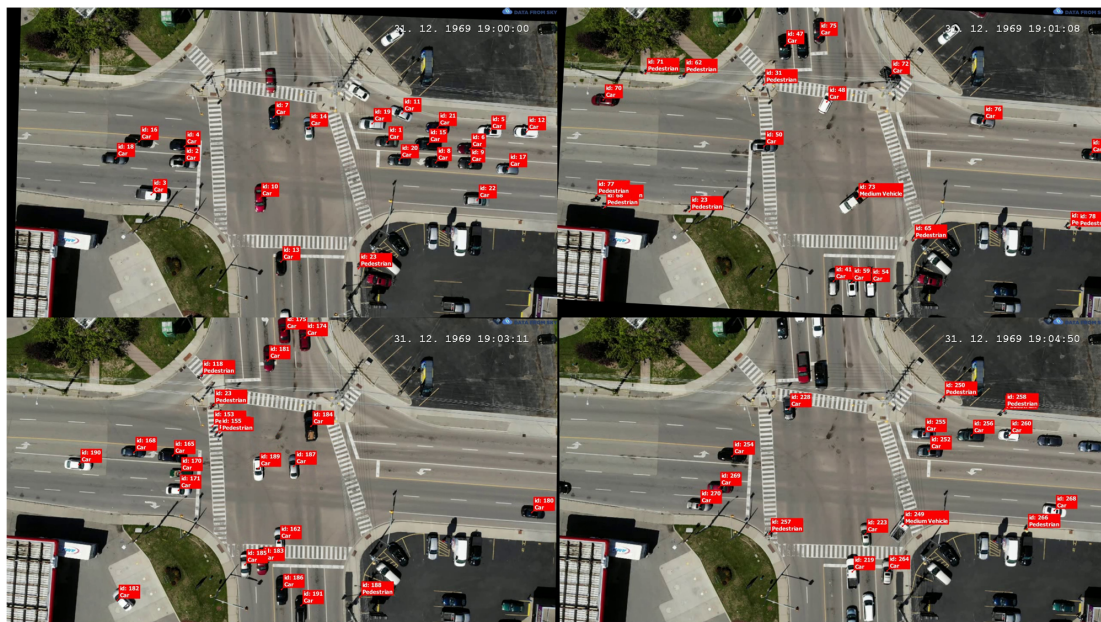


Figure 2. Image of the dataset. Each frame, divided into 0.4 s increments, contains between approximately 10 and 30 agents. Each agent is labeled with its tag and id number information.

4.2. Preprocessing

The preprocessing of the dataset resulted in the following sequence shape: $Seq_i^t = (F_i, A_i, x, y, v, acc1, acc2, ang, tl, time)$. F_i is the frame number; A_i is the index of the agent in the frame; and $x, y, v, acc1, acc2$; and ang are the position, velocity, acceleration, and angle of the agent, respectively. In addition, tl is the code for the combination of traffic light states in that frame, and time is the time in 0.4 s increments.

4.3. Experimental Environments

The software and hardware environments with which we experimented are listed in Table 1.

Table 1. Software and hardware environment used for experiment.

Operating System	GPU Card	Library
Ubuntu 20.04	NVIDIA RTX A5000 24GB × 2	PyTorch v1.13.0 PyTorch CUDA 11.6

4.4. Metrics

As evaluation metrics for our experiments, we used the metrics that have been widely used in previous studies for generative trajectory prediction.

4.4.1. ADE (Average Displacement Error)

The average L2 distance in meter between the predicted and actual values over all prediction times is expressed in (9), where i is the specific agent in the dataset, and T_{pred} is the predicted time in the future.

$$ADE(i) = \frac{\sum_{t=T_t+1}^{t=T_t+pred} \left[\sqrt{(\hat{x}_i^t - x_i^t)^2 + (\hat{y}_i^t - y_i^t)^2} \right]}{T_{pred}}, \quad (9)$$

$$ADE = \frac{\sum_{i=1}^{i=n} ADE(i)}{n}$$

4.4.2. FDE (Final Displacement Error)

This is the average L2 distance between the predicted final position and the final actual position after the T_{pred} time and is as shown in Equation (6), where n is the total number of agents included in the test set.

$$FDE = \frac{\sum_{i=1}^{i=n} \sqrt{\left(\hat{x}_i^{T_t+pred} - x_i^{T_t+pred} \right)^2 + \left(\hat{y}_i^{T_t+pred} - y_i^{T_t+pred} \right)^2}}{n} \quad (10)$$

4.5. Implementation Detail

The Adam optimizer was used to train the generator and discriminator models. We set the batch size to 8, epoch to 50, and learning rate to 0.0005 (trained for approximately 3 h on the waterloo dataset). The input and output dimensions for the trajectory, state value, and traffic light encoders were set to 64, the encoder hidden dimension for the generator was set to 64, and the decoder hidden dimension was set to 128. We also set the encoder hidden dimensions of the discriminator to 64. The experiment was performed by predicting the trajectory for the next 8 s based on the agent's past 8 s. The L2 loss was calculated for the trajectory with the highest probability. In addition, during training, we observed eight steps (3.2 s) and attempted to predict the next 12 steps (4.8 s). Additional experiments that altered the observed time and prediction time during training are presented in Tables A2 and A3.

4.6. Results

We ran two separate experiments, one with and one without scene information. The results of the first experiment, which did not include any scene information from the image, showed that the quantitative results of these experiments are presented in Table 2. We set the prediction time to 4 (1.6 s), 8 (3.2 s), and 12 time steps (4.8 s), and the results were compared with the existing model, SGAN, Ours only speed, and Ours. Ours only speed means that our model's state encoder only places the speed value into the state. We can see that the lower the prediction time, the higher the results are for both ADE and FDE. In addition, if a traffic light was not provided as an input signal, the overall performance was reduced.

Table 2. Results from the first experiment without scene information. Shows the change in the metric value over the prediction time. Comparison of the results with SGAN, our model (state only with speed and acceleration), our model state with only speed, and acceleration (without angle state). Lower result value is better performance.

Prediction Time (s)/Time Steps	Metrics	SGAN	Ours (State Only with Speed, Acc)	Ours (State with Speed, Acc, Angle)
1.6 s/4	ADE	3.26	2.60	3.10
	AVG	4.34	3.36	4.22
	FDE	5.41	4.12	5.34
3.2 s/8	ADE	5.53	3.94	5.46
	AVG	8.03	5.63	8.03
	FDE	10.52	7.32	10.59
4.8 s/12	ADE	7.55	4.90	7.50
	AVG	11.30	7.31	11.31
	FDE	15.04	9.71	15.11

When the prediction time is short, that is, the prediction is performed for four times steps, our proposed model performs the best; for eight time steps, our model performs better in terms of ADE; and the SGAN model performs better in terms of FDE. The prediction results for eight time steps, for which the prediction time is longer, show the best performance of our model in terms of the ADE, and SGAN models in terms of the FDE. In terms of the average of the two metrics, the SGAN and our model show similar results.

Overall, we found that the state encoder performed best when the agent's speed and acceleration values were input, which we believe is due to the fact that the agent's speed and acceleration also have a time-series nature, and we trained the entire deep learning model by constructing an encoder and decoder for the value based on the LSTM model. It showed a higher performance than SGAN, which only modeled the simple trajectories. When mixed acceleration and angle values were provided in the state encoder, no significant performance improvement was observed. A lower result value indicates better performance.

Figure 3 shows a visualization of the results of the predicted trajectory. You can see how the trajectory distributions for SGAN, Ours (state only with speed, acc), and Ours (state with speed, acc, angle) are for the trained models, as shown in Table 3.

The second experiment was performed by combining scene information. The raw image features (512 channels) extracted from the ResNet18 model were combined with traffic light information, agent state information (speed, acceleration, and angle), and agent trajectory information to the decoder, and the resulting values are shown in Table 3.

The model was trained to predict the trajectory of the next eight time steps from the trajectory of the past eight time steps. The evaluation of the model is also performed by comparing the results of trajectory prediction for eight time steps. In terms of the accuracy of performance, the model without encoding scene context performed better in both ADE and FDE metrics. The time (s) were measured in the experimental environment, as shown

in Table 1, with the batch size all set to 1. Including scene context was measured to take about 11.5% more time than not including scene context.

Figure 4 shows the results for trajectories with and without scene context.

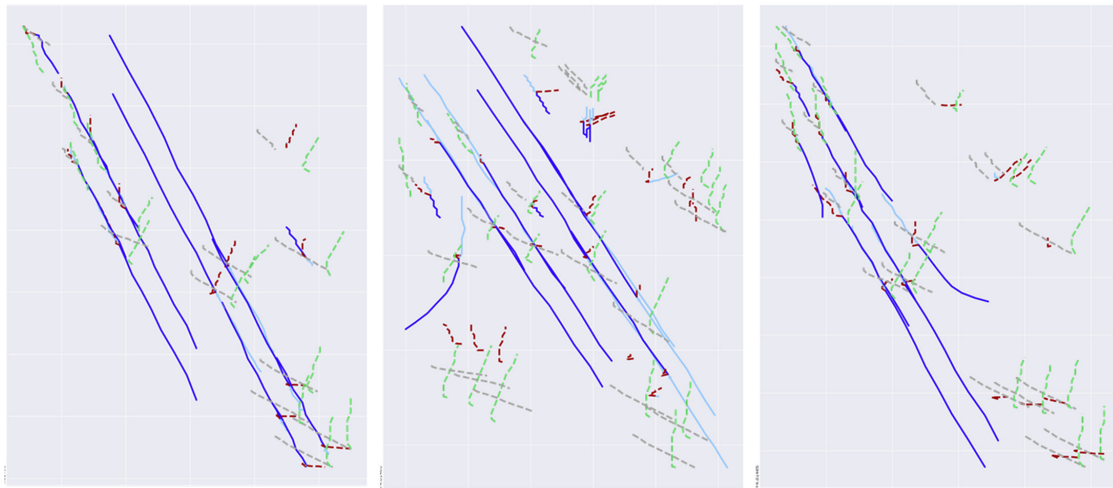


Figure 3. Visualization of trajectory predictions, each representing three separate scenes. The light blue color represents the actual observed historical trajectory, while the dark blue color represents the actual ground truth trajectory. Light green shows the predictions from the SGAN model, red shows our model (state only with speed, acc), and gray shows the results from our model.

Table 3. Results from the second experiment with scene information.

Prediction Time (s)/Time Steps	Metrics	Ours (with Scene Context)	Ours (without Scene Context)
3.2 s/8	ADE	7.35	5.46
	AVG	10.59	8.03
	FDE	13.83	10.59
	Time (s)	211	184

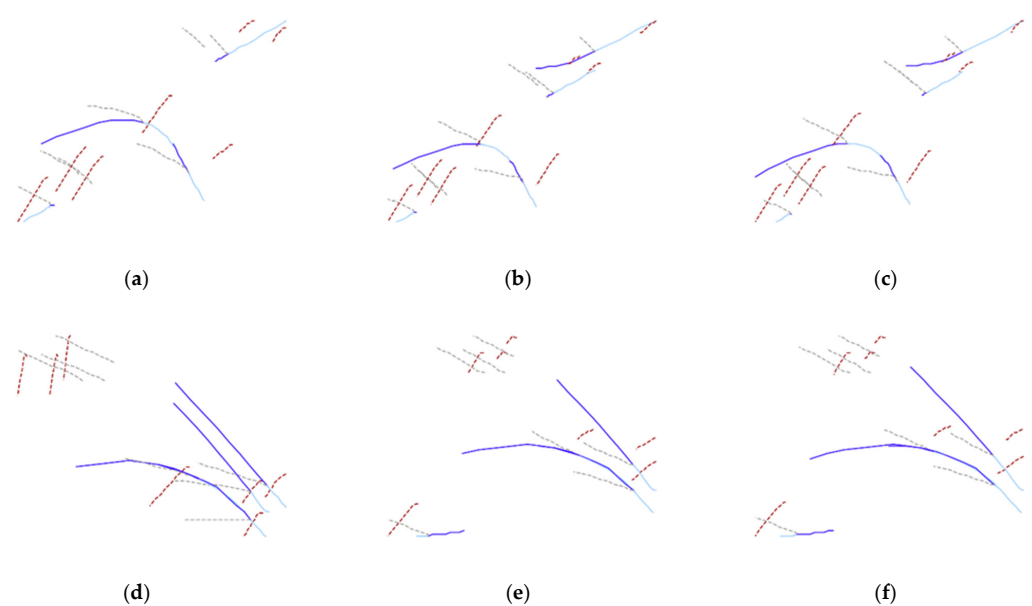


Figure 4. Trajectory visualization results with and without scene information. The red line shows the results of the model with scene context encoded, the gray line shows the results of the model without

scene context applied. Also the light blue color represents the actual observed historical trajectory, while the dark blue color represents the actual ground truth trajectory. (a–f) are three consecutive frames, respectively.

5. Conclusions

Trajectory prediction is an essential component of safe driving by autonomous agents. Attempts to model the complex interactions of multiple factors in the real world using deep learning techniques are ongoing, and researchers are striving to go beyond simply utilizing the location information of a single agent to reflect interactions with dynamic objects or to efficiently reflect the effects of road topography. In this paper, we study the prediction of multiple trajectories of multiple agents in an urban intersection context through a generative model. Building on existing research that models the temporal dependence and interdependence between agents on each agent's trajectory, we reflect the influence of dynamic agent states such as speed, direction, traffic light status, and scene context information. We utilized ADE and FDE, metrics used in deep learning models for multiple trajectory generation, to compare the results over time for each prediction, and compared the results with and without agent state values. We believe that this paper is meaningful in that it is a model that performs trajectory prediction in urban areas by considering interactions in various real-world situations. However, predicting multiple trajectories of multiple agents is considered a difficult task, and the results obtained in this study leave much to be desired in terms of accuracy. Research is needed to improve this accuracy, consider the temporal impact of trajectory generation, and better reflect traffic light conditions. Furthermore, future research could consider models that reflect factors such as lanes and road markings. Considering the characteristics of autonomous vehicles that require cognition and planning in real time, it is necessary to continue research and development to efficiently model complex environmental factors. It is also believed that changing the encoder, for example, with a Transformer or GCN, which is a modern model for time series data, could lead to an improvement in performance and time complexity [32]. Obtaining contextual information from sensors is also an area where autonomous communication technologies need to be integrated, so technologies such as edge caching may be able to improve the accuracy of trajectory prediction by reducing the latency involved [33]. Therefore, a combination of approaches from different perspectives is required to improve accuracy and speed.

Author Contributions: Written with joint contributions from the authors. Conceptualization, S.L. and H.P.; methodology, S.L. and H.P.; software, S.L. and H.P.; validation, S.Y. and Y.Y.; formal analysis, S.L.; investigation, S.L. and H.P.; resources, S.L. and H.P.; writing—original draft preparation, S.L.; writing—review and editing, H.P., S.Y. and Y.Y.; visualization, H.P. and S.L.; supervision, I.-Y.M.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Research Program through the National Research Foundation of Korea (NRF) and funded by the Ministry of Education (No. 2021R111A3057800), and the results were supported by “Regional Innovation Strategy (RIS)” through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-004).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Publicly available datasets, Waterloo Multi-Agent Traffic Dataset were analyzed in this study. This data can be found here: <https://wiselab.uwaterloo.ca/waterloo-multi-agent-traffic-dataset/intersection-dataset/> (accessed on 14 November 2023) or here: <https://uwaterloo.ca/waterloo-intelligent-systems-engineering-lab/datasets/waterloo-multi-agent-traffic-dataset-intersection>.

Acknowledgments: The authors would like to thank the editors and reviewers for constructive suggestions and comments that helped improve the quality of the article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Two-digit code generated by categorizing the state of a traffic light for an experiment.

Two-Digit Code	State (RGB Values of Traffic Light)
00	G G G R R G R R G R
01	G R R R R G R R R R
02	R R G R R G R R G R
03	R R R G G R G G R G
04	R R R R G R R G R G
05	R R R R R R R G G R R
06	R R R R R R R R R R R
07	R R R Y Y R R R R Y
08	R R R Y Y R Y Y R Y
09	Y Y Y R R Y R R Y R
10	G G R R R R R R R R
11	R R R G R R G R R R
12	R R R R R R Y Y R R
13	Y R R R R Y R R R R
14	G G G G G R R R R R
15	Y Y Y Y Y R R R R R
16	Y Y R R R R R R R R
17	R R R R R G G R R R
18	R R R R R G G G G G
19	R R R R R Y Y Y Y Y
20	R R R R R Y Y R R R
21	R G R G G R R R R R
22	R R R R R R G R G R
23	G R G R R R R R R R
24	R R R R R G R G R G
25	G R G R G R R R R R

Appendix B

It shows the change in performance metrics as observed trajectory time and predicted trajectory time change.

Table A2. Results from trained model with observed eight times steps (3.2 s) and attempt to predict the next four times steps (1.6 s).

Prediction Time (s)	Metrics	SGAN	Ours	Ours without Traffic Light
1.6	ADE	5.36	2.87	2.76
	AVG	7.15	3.65	3.64
	FDE	8.94	4.42	4.52

Table A3. Results from trained model with observed eight times steps (3.2 s) and attempt to predict the next twelve times steps (4.8 s).

Prediction Time (s)	Metrics	SGAN	Ours	Ours without Traffic Light
4.8	ADE	6.84	3.96	12.65
	AVG	10.18	5.91	19.04
	FDE	13.52	7.86	25.43

Appendix C

We performed additional experiments for 4, 8, and 12 time steps in the past observed eight time steps in the future with and without traffic light signal.

Table A4. Results from trained model with observed eight times steps (3.2 s) and attempt to predict the next four times steps (1.6 s) without traffic light signal.

(a) Excluding Traffic Light Signal		
Prediction Time (s)	Metrics	Ours without Traffic Light
1.6	ADE	3.20
	AVG	4.34
	FDE	5.47
3.2	ADE	5.62
	AVG	8.22
	FDE	10.82
4.8	ADE	7.77
	AVG	11.65
	FDE	15.52

Table A5. Ablation study. Results from trained model with observed eight times steps (3.2 s) and attempt to predict the next four (1.6 s), eight (3.2 s), twelve (4.8 s) time steps including traffic light. s is the velocity, Acc is the acceleration, and Ang is the angle of the agent.

(b) Including Traffic Light Signal						
S	Ablation Study		Metrics	1.6 s	3.2 s	4.8 s
✓			ADE	2.86	4.58	7.37
			AVG	3.83	6.68	10.85
			FDE	4.79	8.77	14.32
✓	✓		ADE	2.60	3.94	4.90
			AVG	4.72	5.63	7.31
			FDE	4.12	7.32	9.71
✓		✓	ADE	3.65	6.17	8.30
			AVG	4.86	8.82	12.11
			FDE	6.07	11.47	15.93
✓	✓	✓	ADE	3.10	5.46	7.50
			AVG	4.22	8.03	11.31
			FDE	5.34	10.59	15.11

References

1. Lefèvre, S.; Vasquez, D.; Laugier, C. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH J.* **2014**, *1*, 1. [[CrossRef](#)]
2. Ammoun, S.; Nashashibi, F. Real time trajectory prediction for collision risk estimation between vehicles. In Proceedings of the 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, 27–29 August 2009. [[CrossRef](#)]

3. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Li, F.F.; Savarese, S. Social LSTM: Human trajectory prediction in crowded spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. [\[CrossRef\]](#)
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. [\[CrossRef\]](#)
5. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 03762. [\[CrossRef\]](#)
6. Ettinger, S.; Cheng, S.; Caine, B.; Liu, C.; Zhao, H.; Pradhan, S.; Chai, Y.; Sapp, B.; Qi, C.; Zhou, Y.; et al. Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021. [\[CrossRef\]](#)
7. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. [\[CrossRef\]](#)
8. Nachiket, D.; Eric, M.W.; Oscar, B. Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals. In Proceedings of the 5th Conference on Robot Learning (CoRL 2021), London, UK, 8–11 November 2021. [\[CrossRef\]](#)
9. Luo, C.; Sun, L.; Dabiri, D.; Yuille, A. Probabilistic Multi-modal Trajectory Prediction with Lane Attention for Autonomous Vehicles. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021. [\[CrossRef\]](#)
10. Deo, N.; Trivedi, M.M. Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018. [\[CrossRef\]](#)
11. Zhang, Z. ResNet-Based Model for Autonomous Vehicles Trajectory Prediction. In Proceedings of the 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 15–17 January 2021. [\[CrossRef\]](#)
12. Sheng, Z.; Xu, Y.; Xue, S.; Li, D. Graph-Based Spatial-Temporal Convolutional Network for Vehicle Trajectory Prediction in Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 17654–17665. [\[CrossRef\]](#)
13. Zhang, Y.; Wang, W.; Guo, W.; Lv, P.; Xu, M.; Chen, W.; Manocha, D. D2-TPred: Discontinuous Dependency for Trajectory Prediction Under Traffic Lights. In Proceedings of the ECCV 2022: 17th European Conference (ECCV), Tel Aviv, Israel, 23–27 October 2022. [\[CrossRef\]](#)
14. Oh, G.; Peng, H. Impact of traffic lights on trajectory forecasting of human-driven vehicles near signalized intersections. *arXiv* **2020**, arXiv:1906.00486v4.
15. Yuning, C.; Benjamin, S.; Mayank, B.; Dragomir, A. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In Proceedings of the 2019 Conference on Robot Learning (CoRL 2019), Osaka, Japan, 30 October–1 November 2019. [\[CrossRef\]](#)
16. Multi Agent Traffic Dataset. Available online: <https://wiselab.uwaterloo.ca/waterloo-multi-agent-traffic-dataset/intersection-dataset> (accessed on 1 July 2023).
17. Kong, X.; Xing, W.; Wei, X.; Bao, P.; Zhang, J.; Lu, W. STGAT: Spatial-temporal graph attention networks for traffic flow forecasting. *IEEE Access* **2020**, *8*, 134363–134372. [\[CrossRef\]](#)
18. Kim, B.; Park, S.; Lee, S.; Khoshimjonov, E.; Kum, D.; Kim, J.; Kim, J.; Choi, J. Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021. [\[CrossRef\]](#)
19. Gupta, A.; Johnson, J.; Li, F.F.; Savarese, S.; Alahi, A. Social GAN: Socially acceptable trajectories with generative adversarial networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [\[CrossRef\]](#)
20. Sadeghian, A.; Kosaraju, V.; Sadeghian, A.; Hirose, N.; Rezaatofighi, H.; Savarese, S. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
21. Deo, N.; Trivedi, M.M. Convolutional Social Pooling for Vehicle Trajectory Prediction. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018. [\[CrossRef\]](#)
22. Lee, N.; Choi, W.; Vernaza, P.; Choy, C.B.; Torr, P.H.S.; Chandraker, M. DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [\[CrossRef\]](#)
23. Zhao, T.; Xu, Y.; Monfort, M.; Choi, W.; Baker, C.; Zhao, Y.; Wang, Y.; Wu, Y.N. Multi-Agent Tensor Fusion for Contextual Trajectory Prediction. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
24. Cui, H.; Radosavljevic, V.; Chou, F.C.; Lin, T.H.; Nguyen, T.; Huang, T.K.; Schneider, J.; Djuric, N. Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019. [\[CrossRef\]](#)

25. Phan-Minh, T.; Grigore, E.C.; Boulton, F.A.; Beijbom, O.; Wolff, E.M. CoverNet: Multimodal Behavior Prediction Using Trajectory Sets. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. [\[CrossRef\]](#)
26. Teeti, H.; Khan, S.; Shahbaz, A.; Bradley, A.; Cuzzolin, F. Vision-based Intention and Trajectory Prediction in Autonomous Vehicles: A Survey. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22), Vienna, Austria, 23–29 July 2022; pp. 5630–5637. [\[CrossRef\]](#)
27. Hou, W.; Wu, Z.; Jung, H. Video Road Vehicle Detection and Tracking based on OpenCV. *J. Inf. Commun. Conver. Eng.* **2022**, *20*, 226–233. [\[CrossRef\]](#)
28. Kosaraju, V.; Sadeghian, A.; Martin-Martin, R.; Reid, I.; Rezaiofighi, H.; Savarese, S. Social-BIGAT: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*. [\[CrossRef\]](#)
29. Salzmann, T.; Ivnovic, B.; Chakravarty, P.; Pavone, M. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In Proceedings of the Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Springer International Publishing: Cham, Switzerland, 2020. Part XVIII 16. pp. 683–700. [\[CrossRef\]](#)
30. Agrimgupta92, Sgan, GitHub Repository. Available online: <https://github.com/agrimgupta92/sgan> (accessed on 8 September 2023).
31. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784. [\[CrossRef\]](#)
32. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *arXiv* **2018**, arXiv:1707.01926v3.
33. Wu, Q.; Zhao, Y.; Fan, Q.; Fan, P.; Wang, J.; Zhang, C. Mobility-Aware Cooperative Caching in Vehicular Edge Computing Based on Asynchronous Federated and Deep Reinforcement Learning. *IEEE J. Sel. Top. Signal Process.* **2023**, *17*, 66–81. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.