Incorporating Multi-Context Into the Traversability Map for Urban Autonomous Driving Using Deep Inverse Reinforcement Learning

Chanyoung Jung and David Hyunchul Shim

Abstract—**Autonomous driving in an urban environment with surrounding agents remains challenging. One of the key challenges is to accurately predict the traversability map that probabilistically represents future trajectories considering multiple contexts: inertial, environmental, and social. To address this, various approaches have been proposed; however, they mainly focus on considering the individual context. In addition, most studies utilize expensive prior information (such as HD maps) of the driving environment, which is not a scalable approach. In this study, we extend a deep inverse reinforcement learning-based approach that can predict the traversability map while incorporating multiple contexts for autonomous driving in a dynamic environment. Instead of using expensive prior information of the driving scene, we propose a novel deep neural network to extract contextual cues from sensing data and effectively incorporate them in the output, i.e., the reward map. Based on the reward map, our method predicts the ego-centric traversability map that represents the probability distribution of the plausible and socially acceptable future trajectories. The proposed method is qualitatively and quantitatively evaluated in real-world traffic scenarios with various baselines. The experimental results show that our method improves the prediction accuracy compared to other baseline methods and can predict future trajectories similar to those followed by a human driver.**

Index Terms—**AI-based methods, autonomous vehicle navigation, intelligent transportation systems, learning from demonstration.**

I. Introduction

DESPITE significant efforts by academia and industry over the last decade, autonomous driving in urban environments with other road users (social navigation) remains challenging. One of the key objectives in autonomous driving is to generate a traversability map that considers multiple contexts of a driving scene for ego-motion planning [1], [2]. To ensure safety and avoid negatively impacting the traffic flow, three types of contextual cues should be considered: 1) Inertial context: The trajectory should be driven by a user's own intention under kinematic and dynamic constraints; 2) Environmental context: The trajectory should be collision-free with respect to stationary obstacles; 3) Social context: The trajectory should be collision-free, not only considering the current observed states (position and velocity) of the neighboring agents but also the near-future states and intentions.

Recently, various efforts have been made for ego-motion planning in a dynamic environment; this problem can be considered as a probabilistic future trajectory (or behavior) prediction and multi-context embedded traversability map prediction problem. Conventional filter and model-based methods [3], [4] are well-known approaches for predicting an agent's future trajectories. In these methods, the historical and current observation states (such as position, velocity, and yaw) are set as inputs to estimate the future behaviors based on a model. Therefore, they are suitable only in simple scenarios but have limitations in that they cannot reflect interactions between real-world agents that have their own intentions. With the success of deep learning, data-driven approaches have been applied to predict future behaviors or trajectories [5]–[8], consider social interactions [9], [10], and model the human driving policy [11]–[15].

However, there are several major limitations of previous studies. First, many recent studies utilized expensive prior knowledge such as a high-definition (HD) map, which provides rich environmental contextual cues about the driving scene, as an input to the network; however, an important practical limitation of this approach is the lack of scalability. Second, in the case of studies utilizing detection results as an input to the network, the error at the detection stage is inherently propagated to the prediction result. Finally, the majority of previous studies focused on predicting the future trajectory of an agent in a fixed third-person view. However, this approach is not applicable for an onboard autonomy system where the environmental context changes as it moves.

In this study, motivated by the limitations of existing studies, we extend the deep inverse reinforcement learning (DIRL) framework to predict an ego-centric traversability map that incorporates the essential contextual cues for autonomous driving in a dynamic environment, as shown in Fig. 1. We propose a convolutional long short-term memory (ConvLSTM)-based deep neural network (DNN) as a reward function approximator. Rather than utilizing expensive prior information of the driving scene such as an HD map, our network extracts multiple contextual cues from LiDAR and trajectory/route data represented in a

Manuscript received October 15, 2020; accepted February 4, 2021. Date of publication February 16, 2021; date of current version March 2, 2021. This letter was recommended for publication by Associate Editor T. Kulvicius and Editor T. Asfour upon evaluation of the reviewers' comments. This letter was the result of a research project supported by SK hynix Inc. *(Corresponding author: Chanyoung Jung.)*

The authors are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon KS015, South Korea (e-mail: hy910915@kaist.ac.kr; hcshim@kaist.ac.kr).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3059628, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3059628

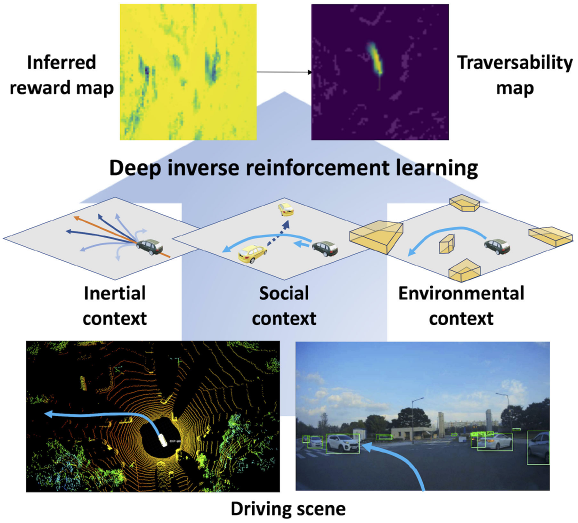

Fig. 1. Scheme for predicting traversability map that incorporates inertial, environmental, and social contexts using the deep inverse reinforcement learning (DIRL) framework.

local grid map and effectively incorporates them into the output reward map. The probability distribution of future trajectories can be computed based on the inferred reward map, and we set this as a future traversability map. Finally, we qualitatively and quantitatively evaluated our approach on real-world traffic scenarios and compared the results with multiple baseline methods. The experimental evaluation shows that our method outperforms existing methods. Furthermore, we performed thorough ablation study to validate our design choices. The main contributions of this work can be summarized as follows:

- We extend the DIRL framework to predict a traversability map that incorporates multi-context (inertial, environmental, and social) for urban autonomous driving.
- As a reward function approximator, a ConvLSTM-based DNN is proposed as a hidden reward function approximator that is able to effectively extract contextual cues from the driving scene and recover the reward function behind the human driving demonstration.
- Extensive evaluation and comparison of the existing methods are conducted based on real-world traffic scenarios.

The remainder of this latter is organized as follows. Section II introduces the existing studies on trajectory prediction and social navigation. The technical details of our method are given in Section III, and the evaluation results are provided with metrics and baselines in Section IV. Section V presents the conclusions and scope for future work.

II. Related Work

A traversability map can be expressed as a probability distribution of an ego-vehicle's future trajectories. In a static environment, it is sufficient to consider the inertial and environmental contexts when predicting a traversability map; however, in a dynamic environment where surrounding agents exist, the social context should also be considered for safety and traffic efficiency. Conventionally, filter and model-based methods have been widely used for trajectory prediction problems considering social interactions [16], [17]. However, the prediction results are valid only in the short term and fail to model the interactions with other agents.

To avoid loss of generality and to accurately model the interactions with multiple neighboring agents, recurrent neural networks (RNNs) and long short-term memory (LSTM) networks have been adopted for trajectory prediction [5]–[7], [9]. In [18], [19], generative adversarial networks (GANs) were adopted for pedestrian trajectory prediction. They modeled each observed pedestrian with an LSTM-GAN, which is an LSTM-based encoder-decoder framework that outputs the predicted trajectories. In [8], the authors presented an approach that can predict the future trajectories of heterogeneous traffic agents (e.g., other vehicles and pedestrians) in an ego-centric view. They designed a graph-based LSTM network that inputs the perception result from the detection module and outputs the future trajectory in the image coordinate. However, in contrast to our work, the perception error is propagated to the prediction performance because their networks take the detection results as an input.

Recently, methods using an HD map as a network input for a prediction task for autonomous driving have received increased interest, particularly by industry players (such as Uber and Lyft). The authors of [20]–[22] introduced a deep learning-based approach that takes into account the current world state and infers the future trajectories, utilizing raster images of each actor's vicinity on an HD map. In a similar fashion, the authors of [23] focused on learning driving policy from demonstration using pre-planned routes and top-views of an HD map. To encourage road rule compliance and penalize undesirable events such as collision, the model was trained using a number of domain-specific losses. In [24] and [25], end-to-end trainable network models that use LiDAR sweeps with an HD map as the network inputs for joint perception and prediction tasks for autonomous driving were introduced. Although an HD map provides rich environmental contextual cues of the driving scene to the network, this approach is not a scalable approach from a self-driving perspective.

In addition to the deep learning approaches, reinforcement learning (RL) and inverse reinforcement learning (IRL) have been applied to autonomous driving [11]–[15], [26], [27]. In [26], a deep RL approach for socially aware motion planning was proposed. They successfully performed autonomous navigation in a pedestrian-rich environment by inducing a simple social norm into the model. In [13], the authors proposed a maximum entropy DIRL (MEDIRL)-based approach that outputs the probabilistic distribution of future trajectories. Unlike the RL mechanism, which is trained based on a manually designed reward function, the DIRL approach, which aims to learn a nonlinear reward function from a driver's demonstration using a DNN, can model human intuition. In the extended work of [13], which also motivated our research, the authors of [14] proposed a method to incorporate ego-vehicle kinematics as well as environmental contexts into the DIRL framework and evaluated it in the off-road environment.

III. Proposed Methodology

In this section, the details of the proposed methodology are discussed. First, we briefly introduce the DIRL and Markov decision process (MDP). Subsequently, in the DIRL framework, a DNN architecture as a reward function approximator is presented.

A. Problem Statement

We model the autonomous vehicle as an agent whose process of future trajectory prediction follows a MDP. The MDP is a widely used framework for probabilistic decision-making processes and is a fundamental concept in RL and IRL. A finite-state MDP is defined as $M = \{S, A, P_{sa}, \gamma, R\}$, where $S = \{s_0, \ldots, s_N\}$ is the set of states, $A = \{a_0, \ldots, a_k\}$ is the set of possible actions, P_{sa} denotes the state transition probability functions upon taking action $a \in A$ in state $s \in S$, $\gamma \in [0, 1)$ is a discount factor and R is a reward function.

Given the demonstration sample $D_i = \{\nu_i, \zeta_i\}$, where ν_i is the driving scene representation and ζ_i is a sequence of states $(s_0, \ldots, s_M)$, the reward value of the trajectory ζ_i is calculated as follows:

$$R(\zeta_i) = \sum_{t=0}^{M} \gamma^t R(s_t) \tag{1}$$

Assuming that the demonstrations from a human are near-optimal behaviors in terms of maximizing the reward based on the hidden reward function, our objective is to recover the reward function that can explain demonstrations. Using the approximated reward function, we can compute the traversability map representing the probability of visiting each state.

B. Medirl

RL aims to learn an optimal policy $(\pi : S \rightarrow A)$ that maximizes the expected reward for each state with a given reward function in an MDP. However, it is difficult to accurately represent the attributes that should be exhibited by the reward function. Moreover, it is ambiguous to define what is the “desirable” behavior in an RL problem. Therefore, the DIRL approach was firstly proposed in [13], which aims to recover an expert’s hidden reward function using a DNN based on given demonstrations D. Here, we set human driving trajectories as the demonstrations and DNN as a reward function approximator $r(\phi(s)) = f(\phi(s); \theta)$, where $\phi(s)$ are features at state s, and θ denotes the network model parameters. Following the maximum entropy principle [28], the network is trained while maximizing the following joint log-likelihood of the demonstrations and the model parameters, θ, under the predicted reward functions, R^* :

$$L(\theta) = P(D, \theta | R^*) = L_D + L_\theta, \tag{2}$$

where:

$$\begin{aligned} L_D &= \log P(D|R^*) \\ L_\theta &= \log P(\theta). \end{aligned} \tag{3}$$

Allowing gradient descent methods to this problem with respect to the network parameters θ, the gradient can be expressed as follows:

$$\begin{aligned} \frac{\partial L}{\partial \theta} &= \frac{\partial L_D}{\partial \theta} + \frac{\partial L_\theta}{\partial \theta} \\ &= (\mu_D - \mathbb{E}[\mu]) \cdot \frac{\partial f}{\partial \theta} + \lambda \theta, \end{aligned} \tag{4}$$

where μ_D is the state visitation frequency (SVF) exhibited by the demonstration, $\mathbb{E}[\mu]$ is the expected visitation frequency from the predicted reward function, and $\frac{\partial L_\theta}{\partial \theta}$ represents the regularization term(here, we set L_2) which aims to prevent from the over-fitting. Thus, because the SVF distribution represents a set of probable trajectories, the network model parameters are trained to infer future trajectories similar to those that result from the expert’s demonstrations. Algorithms 1 and 2 show the overall training procedure and value iteration, respectively.

Algorithm 1: MEDIRL Training Procedure.

```
Input: $D, S, A, P_{sa}, \gamma, r$
Output: Trained network model parameters $\theta$
Initialize parameters $\theta$;
for iteration $i$ = 1 to $N$ do
    $r^i(\phi(s)) = f(\theta^i, \phi(s)) \forall s \in S$;
    $\pi^i = ValueIteration(r^i, S, A, P_{sa}, \gamma)$;
    $\mathbb{E}[\mu^i] = ComputeSVF(\pi^i, S, A, P_{sa})$;
    $\frac{\partial L_D^i}{\partial r^i} = \mu_D - \mathbb{E}[\mu^i]$;
    $\theta^{i+1} = BackPropagate(\theta^i, \frac{\partial L_D^i}{\partial r^i})$;
end
return $\theta$
```

Algorithm 2: Value Iteration.

```
Input: $r, S, A, P_{sa}, \gamma$
Output: Policy $\pi$
Initialize values $V(s) = -\infty$
repeat
    $V_t(s) = V(s)$;
    $Q(s,a) = r(s,a) + \gamma \cdot \sum_{s' \in S} P_{s'a} V(s')$;
    $V(s) = max_a(Q_i(s,a))$;
until $max_s(V(s) - V_t(s)) < \varepsilon$;
return $\pi(a|s) = \exp(Q(s,a) - V(s))$
```

C. Network Architecture

For autonomous driving in a dynamic environment in a safe and socially acceptable manner, it is necessary to predict a traversability map considering the inertial, environmental, and social contexts. These are implicitly considered in the case of human driving; for example, when we drive at a high speed, sudden steering is undesirable or may not even be possible. In addition, we interact with the surrounding vehicles as well as avoid collisions with obstacles while driving to the destination. From the perspective of how humans drive, we designed a ConvLSTM-based encoder-decoder architecture that can capture multiple contextual cues from the driving scene and accurately incorporate these into the traverse reward map, as shown in Fig. 2.

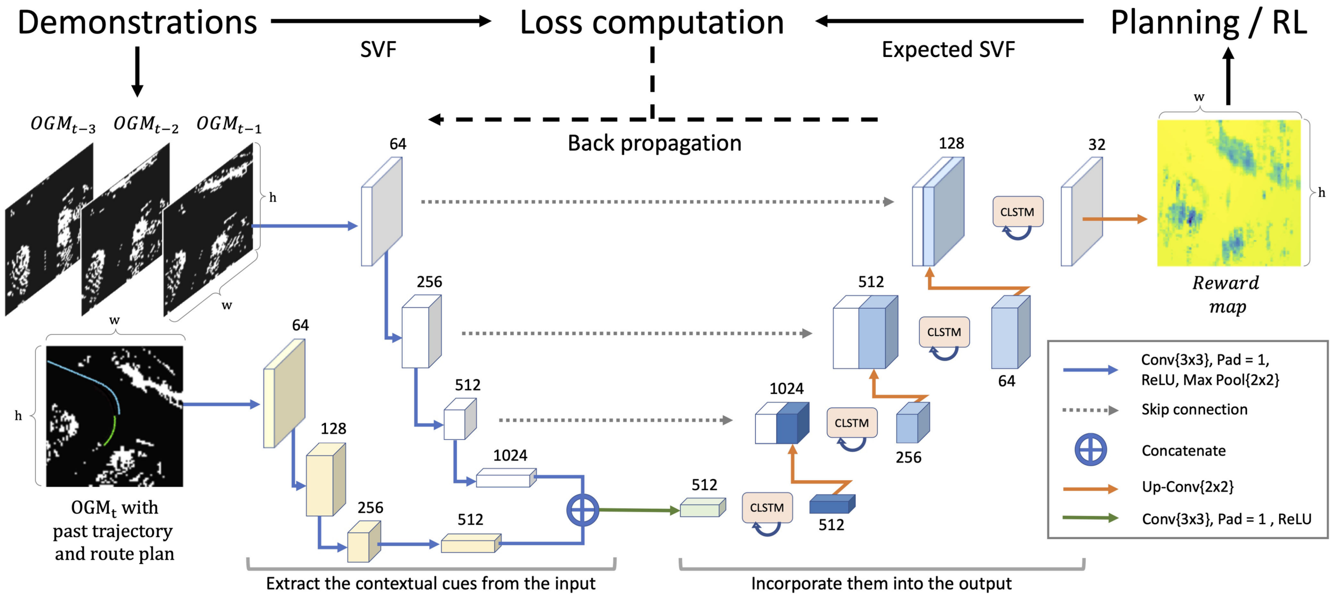
Demonstrations
SVF
Loss computation
Expected SVF
Planning / RL
Back propagation
OGM_{t-3} OGM_{t-2} OGM_{t-1}
h
w
64
256
512
1024
64
128
256
512
$\mathrm{OGM_t}$ with past trajectory and route plan
512
CLSTM
512
1024
256
512
64
128
32
w
h
Reward map
Conv{3x3}, Pad = 1, ReLU, Max Pool{2x2}
Skip connection
Concatenate
Up-Conv{2x2}
Conv{3x3}, Pad = 1 , ReLU
Extract the contextual cues from the input
Incorporate them into the output

Fig. 2. Illustration of the proposed network architecture as a reward function approximator, and training procedures. The encoder module with two branches extracts the contextual cues from the input data. The convolutional long short-term memory (ConvLSTM)-based decoder module is added to incorporate them into the output reward map. With the inferred reward map, the difference between the demonstration and the expected state visitation frequency (SVF) is used as a training signal.

The proposed network architecture is mainly composed of two branches of the encoder and decoder. The first branch of the encoder takes three channel-wise stacked occupied grid maps (OGMs) $x \in \mathbb{R}^{h \times w \times 3}$ over time, which contains the spatiotemporal dependencies. Each grid cell of the OGM is said to be occupied if the maximum height difference of the LiDAR points in the corresponding cell is more than a certain level. The frame interval is experimentally set to 1 s to maintain the temporal dependencies between the frames. The OGM of the current scene overlaid with the past trajectory and the lane-level route plan toward the destination $x \in \mathbb{R}^{h \times w}$ are set as the inputs to the second branch of the encoder. The authors of [14] showed that the simultaneous use of inertial and environmental contexts can improve the performance of trajectory prediction. They applied a middle fusion strategy, where the vehicle speed and curvature calculated from the past trajectory are inputted to the middle part of the network through an additional process. Unlike the way in which they merge the inertial context with the environmental features, we overlaid the past trajectory on the OGM and set it as the network input so that the inertial and environmental contexts can be combined at the kernel level. In addition, by inputting the lane-level route plan, we implicitly provide information that helps to predict future trajectories, such as intersections.

Given the inputs, the encoder module extracts features using convolutional layers. By gradually reducing (downsampled by two) the resolution of the feature maps, our network is capable to extract multi-scale spatiotemporal features. The outputs of each branch of the encoder are concatenated and then sent to the decoder module. In addition, the feature maps of every level in the first branch of the encoder are sent to the decoder via a skip connection mechanism, which aims to enhance the low-level features. To incorporate the spatiotemporal features into a reward map that can reason about human demonstrations, we adopted the 1×1 ConvLSTM unit [29] as the basic block of the decoder module. The feature map at different abstraction levels obtained from the encoder through the skip connection is subsequently merged with the upsampled output of the previous ConvLSTM layer. Finally, the network outputs a reward map, which has the same resolution as the input via the 1×1 convolution filter with the ReLU activation.

IV. Experiments

In this section, we introduce the experimental design and data collection process for training and validation, and then present the implementation details of our approach. Finally, we present the evaluation results and compare them with those obtained using existing methods.

A. Data Collection and Experimental Design

Fig. 3(a) shows our experimental platform with a sensory pack used for data acquisition installed on a full-scale modified vehicle. A 360° multi-layer 3D LiDAR (64 ch) was used to generate the OGM, and the real-time kinematic global positioning system (RTK GPS) and IMU data were fused using an extended Kalman filter (EKF) for a precise localization of the vehicle.

Using the experimental platform, we collected the dataset by recording 16 h of manual driving by more than five drivers; an example of the data sample is shown in Fig. 3(b). We acquired the data in various environments to prevent the network model from overfitting on specific road geometry and situation. Approximately 12 000 demonstrations were collected, of which 18% was set as a validation set. To clearly evaluate the performance, we classified the validation set into three cases, as listed in Table I.

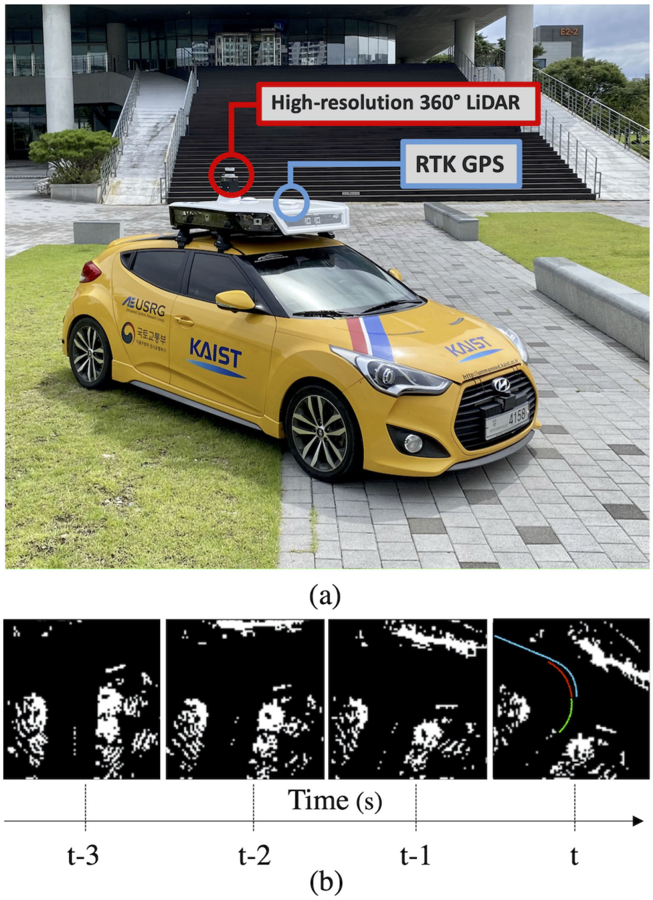

Fig. 3. (a) Full-scale vehicle platform. A high-resolution 360° LiDAR sensor and a real-time kinematic global positioning system (RTK GPS) antenna are installed on the roof sensor pack. (b) Visualization of the training sample. The past trajectory, route plan, and expert demonstration are depicted in yellow, cyan, and white, respectively, in the occupied grid map (OGM) at time t.

TABLE I
EXPERIMENTAL DESIGN

Case #	Description
1	Following the lane while minimally interacting with other agents
2	Traveling in an intersection without other agents
3	Traveling in an intersection with other agents

B. Implementation Details

Input representation: To create the OGM, the point cloud was cropped to reside within a region of size 40 m × 40 m, corresponding to the X and Y ranges, respectively. Subsequently, we removed the ground plane using the RANSAC filter. Each grid cell size of the OGM was set to 0.5 m, and it was considered occupied only when the height difference of the LiDAR points in the corresponding grid exceeded a certain threshold (here, we set 0.3 m). The past trajectory and human driving demonstration were converted to a grid map coordinate based on the vehicle's global position. The current vehicle position and heading were always set to the center and north of the grid map coordinates, respectively. The past trajectory and demonstration consisted of vehicle positions for the past 3 s and 3 s from the present time, respectively. A route plan that provides information about the destination was also generated in the same way and was only partially provided within the local grid map.

Environment modeling: We used the finite-state MDP with the same size as the network input OGM (80 × 80) to model the environment. Each cell in the grid map represents the state, and the output of the network (reward map) corresponds to the reward value at each state. The agent can perform a total of six actions (stay, east, west, north, northwest, and northeast) without considering the backward direction. Here, we set the discount factor to 0.9.

Network training and inference: We utilized a Pytorch deep learning framework for training and inference. We trained the network at the NVIDIA DGX station until convergence using the ADAM optimizer with a learning rate of 0.001 and a batch size of 1024. Inference was running on a 2080Ti GPU with i9 CPU and it took about 1.43 sec to predict the final output.

C. Evaluation Metrics and Baselines

For a quantitative evaluation of the predictive performance, we randomly sampled 300 trajectories from the trained policy and computed the average value over the following metrics:

- The negative log-likelihood (NLL) metric represents the probability of the demonstration under the learned policy. The lower the value, the higher the probability.
- The Hausdorff distance (HD) [30] metric represents the geometric similarity between demonstrations and trajectories sampled with the learned policy. The lower the value, the closer the distance between the paths.

Four baselines were reproduced to compare the predictive performance with our approaches:

- EKF: We assumed that the speed and the steering angle do not change during prediction.
- CSP-LSTM [10]: A variant of a social pooling LSTM layer adding convolutions to the network to learn interdependencies in vehicle motion and predict the predictive distribution over future trajectories.
- IEC-DIRL [14]: A two-stage fully convolutional network for predicting a reward map that aims to explain about inertial and environmental contexts using DIRL.
- MTP-DCN [20]: A deep convolutional network with LSTM takes bird's eye view(BEV) raster images that encode neighboring vehicle states and HD map that aims to account for and capture the inherent uncertainty of the prediction task.

As both CSP-LSTM and MTP-DCN assume that the states of the surrounding road users(position, heading) are provided by a separate perception system, we provided the ground truth states as defined by each model. Additionally, the HD map of KAIST campus was utilized to generate the rasterized BEV image as an input of MTP-DCN.

D. Evaluation Results

Comparison with baselines: Fig. 4 shows a visualization of the key experimental results for comparison with the baseline

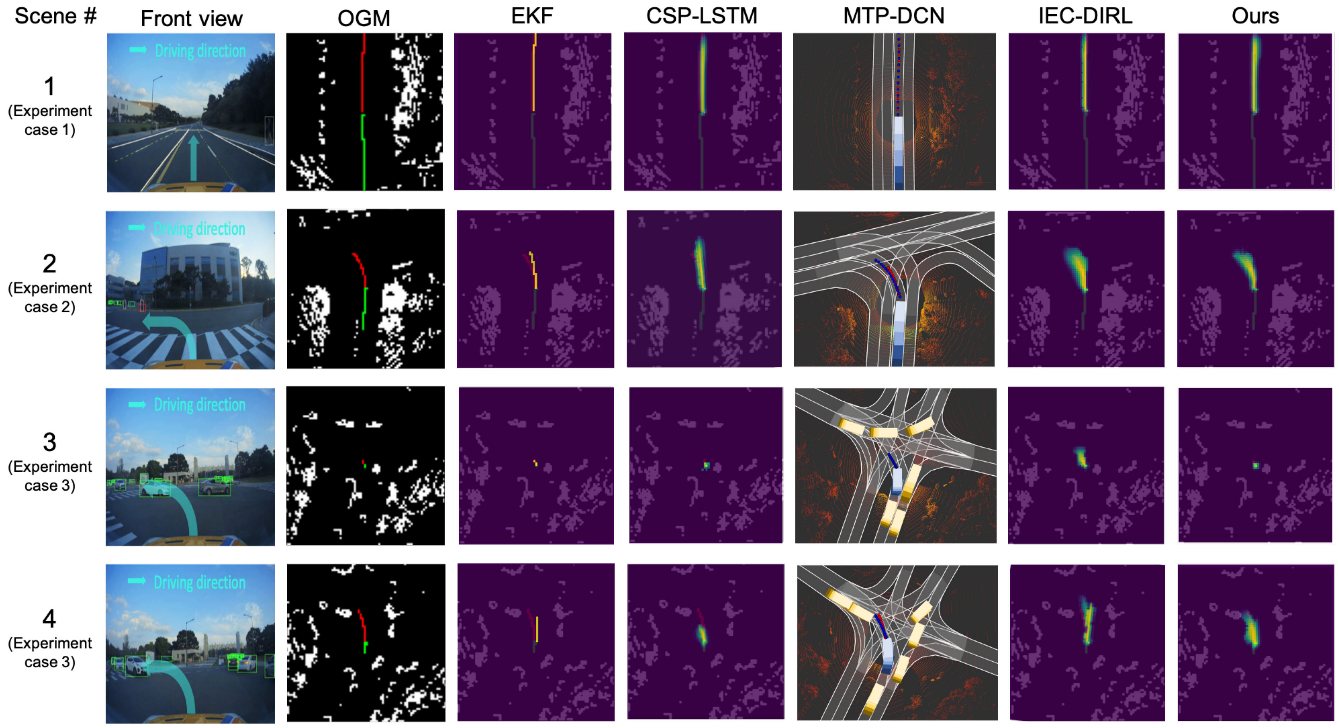

Fig. 4. Comparison of prediction results with baselines in various scenes. The first column shows the front-view image (not inputted to the network). In the second column, the past trajectory (in green) and demonstration (in red) are projected into the occupied grid map (OGM) at the current timestamp. Except for MTP-DCN, the prediction results of each method are overlaid over the OGM. In the fifth column, the rasterized input images for MTP-DCN with the surrounding vehicles (in yellow box), ego vehicle (in blue box), prediction result (dotted blue), and demonstration (dotted red) on the HD map are shown.

methods. The first row of Fig. 4 shows a situation where the ego-vehicle drives along a nearly straight lane in an environment with no nearby vehicles. In this situation, the social context is not very important for predicting the traversability map, and all the methods show reasonable results. The second row shows the result of predicting the traversability map at a non-signalized intersection with no surrounding vehicles.

Both EKF and CSP-LSTM, which do not consider the environmental context, failed to predict the future trajectories, because they did not grasping the topography of the intersection. On the other hand, MTP-DCN, IEC-DIRL, and our method, which provide environmental as well as inertial features to the network model using an HD map or stacked OGM with a route plan, predicted similar results to the human demonstration. The third and fourth rows correspond to experimental case 3 and show prediction results at an intersection where neighboring vehicles exist. Because the FIFO rule is not strictly adhered to, it is necessary to predict the future trajectory considering the social contexts. Specifically, scenes 3 and 4 represent situations where vehicles do not give way and do give way to the destination, respectively. Therefore, in these scenes, the future trajectory (expert demonstration) is a path that reflects the social context including interactions with surrounding vehicles as well as environmental and inertial contexts. In scene 3, the MTP-DCN and IEC-DIRL predicted that the vehicle would travel on a collision-free area, which are different from the human driving demonstration. These results indicate that the interdependencies between neighboring road users (social context) are not sufficiently considered in their outputs. On the other hand, our method predicted a very narrow area so as not to conflict with the flow of traffic in the surrounding area and our result gradually expanded the traversability map area by grasping the intentions of nearby vehicles that give way. Fig. 5 shows our predicted traversability map over time in more detail.

A quantitative evaluation was also conducted, and the results are listed in Table II. The EKF and CSP-LSTM model showed a good performance in case 1, where it was sufficient to predict the future trajectories by considering only the inertial context. However, because they do not take into account the environmental context, it can be observed that the performance decreased significantly in other cases. MTC-DCN and IEC-DIRL showed reasonable performance in case 1 and case 2 where there were no surrounding vehicles. However, in case 3, it can be observed that the performance was degraded in terms of both the evaluation metrics. Our method showed slightly higher HD evaluation result than the MTP-DCN in case 1, but outperformed other baselines in all other cases without any use of expensive prior knowledge about the driving scene. In particular, our model showed the largest performance difference compared with other baselines in case 3, where environmental, inertial, and social contexts should be considered concurrently.

Ablation studies: To further analyze our proposed method, we first conducted ablation studies on input representation. Maintaining the sensing range, we evaluated the prediction performance according to the grid resolution of the input OGM to check whether the proposed method works at more dense

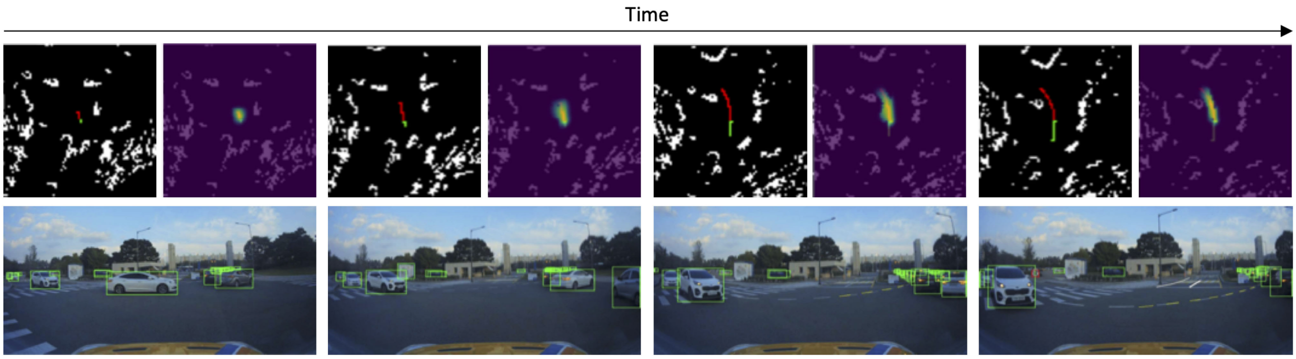

Fig. 5. Visualization of traversability map prediction result over time. The first row shows the occupied grid map (OGM) and the traversability map overlaid on the demonstration (in red) in order. The second row shows the front view image with the neighboring vehicles marked in green bounding boxes.

TABLE II
COMPARISON RESULTS BETWEEN BASELINES FOR PREDICTING A FUTURE TRAJECTORY/TRAVERSABILITY MAP. *M* COLUMN STATES WHETHER THE MODEL USES THE MAP OR NOT. *D* COLUMN STATES WHETHER THE MODEL USES THE DETECTION RESULT AS AN INPUT OR NOT

Method	M	D	Case 1		Case 2		Case 3		Overall	
			NLL ↓	HD ↓	NLL ↓	HD ↓	NLL ↓	HD ↓	NLL ↓	HD ↓
EKF			N.A.	8.52	N.A.	14.22	N.A.	16.78	N.A.	10.651
CSP-LSTM		✓	0.70	8.54	1.03	12.21	1.21	15.62	0.827	10.123
MTP-DCN	✓	✓	N.A.	**8.13**	N.A.	10.21	N.A.	12.45	N.A.	9.064
IEC-DIRL			0.71	8.20	0.94	11.14	1.10	13.62	0.802	9.438
Ours			**0.65**	8.15	**0.85**	**9.64**	**0.89**	**10.35**	**0.718**	**8.712**

TABLE III
ABLATION STUDY RESULTS ON INPUT REPRESENTATIONS AND NETWORK ARCHITECTURE

OGM size (grid resolution)	Stack length	ConvLSTM module	Case 1		Case 2		Case 3		Overall	
			NLL ↓	HD ↓	NLL ↓	HD ↓	NLL ↓	HD ↓	NLL ↓	HD ↓
80 x 80 (0.5 m)	1	✓	0.74	8.24	1.04	14.05	1.40	15.04	0.879	10.218
	3	✗	0.71	8.18	0.93	10.97	1.11	11.52	0.802	9.138
		✓	**0.65**	**8.15**	0.85	9.64	**0.89**	**10.35**	**0.718**	**8.712**
160 x 160 (0.25 m)	1	✓	0.73	8.43	1.01	13.81	1.31	14.94	0.855	10.287
	3	✗	0.73	8.30	1.02	10.58	1.08	11.57	0.830	9.148
		✓	0.67	8.18	**0.84**	**9.61**	0.96	10.85	0.738	8.786

resolutions. As shown in Table III, under the same conditions of stacked length of OGMs and network architecture, our approach showed similar performance in all cases regardless of grid map resolution. We further compared the performance depending on the length of the stack input. In the case of the model in which only the current scene of OGM (stack length, 1) is the input, performance degradations are occurred in all cases compared with the proposed model (stack length, 3). We found that the spatiotemporal feature between stacked OGMs is used as useful information for predicting the ego future traversability map considering the intention of the surrounding vehicles.

Next, we carried out the performance evaluation according to the ConvLSTM module of the proposed network. To achieve this goal, we replaced the ConvLSTM layer with a pooling layer for ablation models. As a result, we can see that adding the ConvLSTM module at the decoder part further improved the prediction performance. This indicates that the multiple contextual features extracted from the network are efficiently incorporated into the reward map using RNN architecture.

V. CONCLUSION AND FUTURE WORK

To ensure safety and traffic efficiency as one of the road users, an autonomous vehicle should be able to predict the future traversability map considering multiple contexts. In this study, we proposed a DIRL approach to address this key aspect of autonomous driving. We designed a novel ConvLSTM-based DNN that can model hidden reward function learning from human demonstrations. Instead of using expensive prior information of the driving scene, our network takes LiDAR-based OGMs, past trajectory, and the lane-level route plan as inputs to extract the contextual cues and effectively incorporate them into the reward map. Evaluation results show that our approach outperforms the various approach of baselines based on real-world traffic scenarios.

We believe that our method has several technical advantages: First, it considers the multiple contexts concurrently for social navigation of autonomous vehicle. Second, it does not rely on the expensive prior knowledge about the environment. Last but not least, since it uses OGMs generated by LiDAR raw data as network input, performance of our method is independent of perception noise.

There is room for improvement and will become part of our future works. Currently, our model did not consider traffic rules such as traffic lights and stop signs. Our future work would focus on incorporating these traffic rules into the model by utilizing additional input images or perceived results. Moreover, we intend to conduct field experiments in diverse and complex environments where various dynamic objects such as pedestrians exist in addition to the surrounding vehicles.

References

[1] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, “Human-aware robot navigation: A survey,” *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1726–1743, 2013.

[2] D. Ferguson and M. Likhachev, “Efficiently using cost maps for planning complex maneuvers,” *Lab Papers*, p. 20, 2008.

[3] C. G. Prevost, A. Desbiens, and E. Gagnon, “Extended kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle,” in *Proc. Amer. Control Conf.*, 2007, pp. 1805–1810.

[4] J. Wiest, M. Höffken, U. Krebßel, and K. Dietmayer, “Probabilistic trajectory prediction with gaussian mixture models,” in *Proc. IEEE Intell. Veh. Symp.*, 2012, pp. 141–146.

[5] N. Deo and M. M. Trivedi, “Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms,” in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1179–1184.

[6] A. Khosroshahi, E. Ohn-Bar, and M. M. Trivedi, “Surround vehicles trajectory analysis with recurrent neural networks,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 2267–2272.

[7] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, “Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network,” in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 399–404.

[8] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, “Trafficpredict: Trajectory prediction for heterogeneous traffic-agents,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6120–6127.

[9] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human trajectory prediction in crowded spaces,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 961–971.

[10] N. Deo and M. M. Trivedi, “Convolutional social pooling for vehicle trajectory prediction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1468–1476 .

[11] A. Kendall *et al.*, “Learning to drive in a day,” in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 8248–8254.

[12] X. Pan, Y. You, Z. Wang, and C. Lu, “Virtual to real reinforcement learning for autonomous driving,” 2017, *arXiv:1704.03952*.

[13] M. Wulfmeier, D. Rao, D. Z. Wang, P. Ondruska, and I. Posner, “Large-scale cost function learning for path planning using deep inverse reinforcement learning,” *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1073–1087, 2017.

[14] Y. Zhang, W. Wang, R. Bonatti, D. Maturana, and S. Scherer, “Integrating kinematics and environment context into deep inverse reinforcement learning for predicting off-road vehicle trajectories,” 2018, *arXiv:1810.07225*.

[15] Z. Zhu, N. Li, R. Sun, H. Zhao, and D. Xu, “Off-road autonomous vehicles traversability analysis and trajectory planning based on deep inverse reinforcement learning,” 2019, *arXiv:1909.06953*.

[16] M. Luber, J. A.Stork, G. D. Tipaldi, and K. O. Arras, “People tracking with human motion predictions from social forces,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 464–469.

[17] S. Ammoun and F. Nashashibi, “Real time trajectory prediction for collision risk estimation between vehicles,” in *Proc. IEEE 5th Int. Conf. Intell. Comput. Commun. Process.*, 2009, pp. 417–422.

[18] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social GAN: Socially acceptable trajectories with generative adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.

[19] S. Eiffert, K. Li, M. Shan, S. Worrall, S. Sukkarieh, and E. Nebot, “Probabilistic crowd GAN: Multimodal pedestrian trajectory prediction using a graph vehicle-pedestrian attention network,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5026–5033, Oct. 2020.

[20] N. Djuric *et al.*, “Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2095–2104.

[21] F.-C. Chou *et al.*, “Predicting motion of vulnerable road users using high-definition maps and efficient convnets,” 2019, *arXiv:1906.08469*.

[22] S. Fadadu *et al.*, “Multi-view fusion of sensor data for improved perception and prediction in autonomous driving,” 2020, *arXiv:2008.11901*.

[23] M. Bansal, A. Krizhevsky, and A. Ogale, “Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst,” 2018, *arXiv:1812.03079*.

[24] M. Liang *et al.*, “PnpNet: End-to-end perception and prediction with tracking in the loop,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11553–11562.

[25] W. Zeng *et al.*, “End-to-end interpretable neural motion planner,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8660–8669.

[26] Y. F. Chen, M.M. EverettLiu, and J. P. How, “Socially aware motion planning with deep reinforcement learning,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1343–1350.

[27] M. Wulfmeier, P. Ondruska, and I. Posner, “Maximum entropy deep inverse reinforcement learning,” 2015, *arXiv:1507.04888*.

[28] B.D.Ziebart, A. L. Maas, J.A.Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning,” in *Assoc. Adv. Artif. Intell.*,Chicago, IL, USA, vol. 8, pp. 1433–1438, 2008.

[29] S.H.I. Xingjian, Z.Chen, H. Wang, D.Y. Yeung, W.-K. Wong, and W.-c.Woo, “Convolutional lstm network: A. machine learning approach for precipitation nowcasting,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[30] K.M. Kitani, B.D. Ziebart, J. AndrewBagnell, and M.Hebert, “Activity forecasting,” in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2012, pp. 201–214.