A Hybrid Machine Learning Approach for Planning Safe Trajectories in Complex Traffic-Scenarios

Amit Chaulwar*, Michael Botsch*, Wolfgang Utschick †
*Faculty of Electrical Engineering
Ingolstadt University of Applied Sciences, 85049 Ingolstadt, Germany
Email: {firstname.lastname}@thi.de
†Department of Electrical Engineering, Technical University of Munich, 80333 Munich, Germany
Email: utschick@tum.de

Abstract*—Planning of safe trajectories with interventions in both lateral and longitudinal dynamics of vehicles has huge potential for increasing the road traffic safety. Main challenges for the development of such algorithms are the consideration of vehicle nonholonomic constraints and the efficiency in terms of implementation, so that algorithms run in real time in a vehicle. The recently introduced Augmented CL-RRT algorithm is an approach that uses analytical models for trajectory planning based on the brute force evaluation of many longitudinal acceleration profiles to find collision-free trajectories. The algorithm considers nonholonomic constraints of the vehicle in complex road traffic scenarios with multiple static and dynamic objects, but it requires a lot of computation time. This work proposes a hybrid machine learning approach for predicting suitable acceleration profiles in critical traffic scenarios, so that only few acceleration profiles are used with the Augmented CL-RRT to find a safe trajectory while reducing the computation time. This is realized using a convolutional neural network variant, introduced as 3D-ConvNet, which learns spatiotemporal features from a sequence of predicted occupancy grids generated from predictions of other road traffic participants. These learned features together with hand-designed features of the EGO vehicle are used to predict acceleration profiles. Simulations are performed to compare the brute force approach with the proposed approach in terms of efficiency and safety. The results show vast improvement in terms of efficiency without harming safety. Additionally, an extension to the Augmented CL-RRT algorithm is introduced for finding a trajectory with low severity of injury, if a collision is already unavoidable.

I. Introduction

The introduction of exteroceptive sensors like radar, camera, laser-scanner, etc. in vehicles enabled the development of new vehicle safety functions like Autonomous Emergency Braking. Inclusion of these safety functions in vehicles have steadily reduced deaths and injury rates [1]. But these functions represent initial steps for the long term goal of zero fatalities in road traffic. An important challenge to achieve this goal is the collision-free trajectory planning considering physical constraints of the vehicle. Even if a collision is unavoidable the trajectory can be modified, such that the predicted severity of injury is low. Such trajectory planning needs accurate modeling of road participants dynamics, appropriate estimates of the severity of injury, and efficiency in terms of implementation, so that algorithms run in real time and can be integrated in a vehicle.

Trajectory planning is an already widely researched area in the robotics community. Trajectory planning algorithms can be classified mainly into three types: potential field based methods, grid based methods, and sampling based methods. The potential field approach [2] cannot always find a trajectory despite using highly simplified dynamic models, even if one exists, because they can get trapped in a local minima. The deterministic path planning algorithms such as A* [3] and its dynamic extensions D* [4] and D*-lite [5] cannot find a path considering nonholonomic constraints of a vehicle.

Consequently, a probabilistic approach called *Rapidly-exploring Random Tree* (RRT) algorithm [6] has gained a lot of popularity because of its fast runtimes and ability for planning the trajectory with nonholonomic constraints of the vehicle. Many variants of RRT algorithm [7], [8], [9] to find a safe path with dynamic constraints of the vehicle have been proposed. But, these algorithms do not incorporate any longitudinal dynamics intervention. In [10] B-spline curves and the RRT algorithm are combined to find a collision free trajectory by taking into account both lateral and longitudinal dynamics but only among linearly moving obstacles. [11] claims to find collision-free trajectories in real time, but it needs to compute lots of paths before the start of a maneuver. Also, all the above algorithms use simple vehicle dynamic models which is not suitable for modeling harsh vehicle maneuvers.

Augmented CL-RRT [12] uses a two-track model for trajectory planning in a traffic scenario with multiple static and dynamic road participants. In order to have simultaneous longitudinal and lateral dynamics interventions, it uses many predefined longitudinal acceleration profiles sequentially to find multiple collision-free trajectories. A brief description of the Augmented CL-RRT can be found in Section III.

All the above mentioned algorithms either need powerful processors or need a lot of precomputation. Therefore, these algorithms are still not realizable in a vehicle. Statistical learning methods offer a possibility to describe complex relationships between multiple input-output signals with low computing resources. The use of statistical learning in subtasks of trajectory planning for the vehicle are mentioned in [13], [14], however these subtasks are not safety critical modules. In [15] 'Support Vector Machines' are used to find a curve separating two

978-1-5090-6167-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ICMLA.2016.17

objects which is taken as a collision-free path. This approach works with only two static objects, and 'learning' is not being done in a classical sense. Statistical learning methods are not used in safety critical applications like trajectory planning as they are purely data-based methods. They are seen as 'black-box' methods. Hybrid machine learning methods, a combination of statistical learning methods and physical models, open a new way to work around the disadvantages of data-based methods and simultaneously exploit their advantages.

The prediction of the severity of injuries before a collision occurs based on information from exteroceptive sensors is a new research area. While the potential of exteroceptive sensors for the estimation of the severity of injury is explained in many works, mostly summarized in [16], still a suitable method for estimating the severity of injury before a collision and its inclusion in a trajectory planning algorithm is missing.

This paper proposes a hybrid machine learning approach by using machine learning methods in combination with the Augmented CL-RRT algorithm. The Augmented CL-RRT algorithm offers a flexibility in finding a solution because of the randomness in lateral dynamics interventions and the use of many different longitudinal acceleration profiles. But, the computation time increases with the number of acceleration profiles. Therefore, an approach for predicting suitable longitudinal acceleration profiles is proposed in this paper.

The paper is organized as follows: Section II formulates the problem of planning safe trajectories and explains the proposed hybrid machine learning approach. Section III gives a brief description of the Augmented CL-RRT algorithm and also introduces a methodology for the prediction of the severity of injury with the Augmented CL-RRT. Section IV presents the convolutional neural networks which are used as a machine learning algorithm in this work. A data generation and the features learning procedure is explained in Section V followed by evaluation of the proposed approach and its comparison with the brute force analytical approach in Section VI. Section VII concludes with a short summary and future scopes.

Throughout this work, matrices are denoted by upper case bold letters, and vectors are denoted by lower case bold letters.

II. Problem Formulation and Proposed Approach

In this work the primary objective is the design of an efficient hybrid machine learning algorithm to find a collision-free trajectory in critical dynamic multi-object traffic-scenarios, with simultaneous interventions in the longitudinal and the lateral dynamics of the vehicle. If a collision is unavoidable, then the algorithm should find a trajectory with low severity of injury.

The Augmented CL-RRT algorithm uses N longitudinal acceleration profiles $\boldsymbol{a}_{\mathbf{x}}^{n}$, where $n = 1, \dots, N$, sequentially to find l safe trajectories. In this work the trajectories that are collision-free or with low predicted severity of injury are considered as safe trajectories. An increase in the number of acceleration profiles increases the chance of finding safe trajectories even in a complex multi-object dynamic traffic scenario, but it also increases the computation time. Therefore, a machine learning algorithm is proposed to predict only the best m out of N acceleration profiles for a particular traffic scenario, and use them with the Augmented CL-RRT algorithm to find a safe trajectory. The advantage of this approach is that the safe trajectories are still found by the model-based Augmented CL-RRT algorithm, and the runtime is reduced.

The construction of the feature vector is extremely important for statistical learning. For predicting longitudinal acceleration profiles, one can use hand-designed features such as road traffic participant's velocity, acceleration, position, etc. as features. The disadvantage of using such features is that a statistical learning algorithm trained for a scenario with a particular number of road traffic participants cannot be used for similar scenarios with a different number of road traffic participants as the dimension of the input vector will change.

In this paper, a traffic scenario is converted into a sequence of predicted occupancy grids $\{\mathcal{G}_{t_1}, \dots, \mathcal{G}_{t_P}\}$ for the prediction interval $[t_1, t_P]$. Such a sequence is denoted in this work as $\mathbf{M}$. An occupancy grid $\mathcal{G}_t$ represents predicted occupancies of road traffic participants other than the EGO vehicle, i. e., the vehicle in which the proposed algorithm runs. An example of an occupancy grid $\mathcal{G}_t$ of size 20×40 meters (each cell 1×1 meter) is shown in Fig. 1 with the EGO vehicle position at initial time t_0 and other road traffic participants predicted positions at time t. Dynamic models are used for finding each traffic participant's predicted positions.

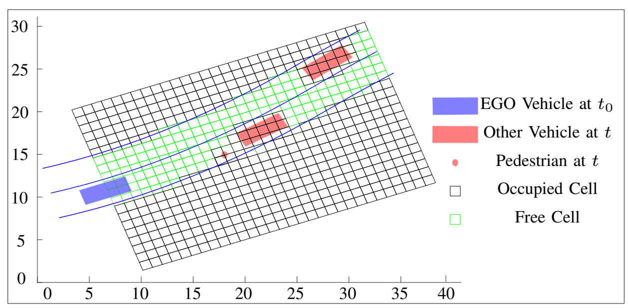

Figure 1. Occupncy grid $\mathcal{G}_t$

A scenario can be defined uniquely by a sequence of predicted occupancy grids $\mathbf{M}$ and the EGO vehicle physical parameters $\boldsymbol{\theta}$. A penalty function $\mathcal{P}(\mathbf{M}, \boldsymbol{\theta}, \boldsymbol{a}_{\boldsymbol{x}}^{n}) \in [0, 1]$ is used to find a penalty for each acceleration profile given a scenario described by $\{\mathbf{M}, \boldsymbol{\theta}\}$. Acceleration profiles that have found collision-free trajectories have low penalties while acceleration profiles that have found trajectories with a collision have high penalties close to 1. The penalty for collision-free trajectories also depends on acceleration values in the acceleration profiles with breaking having the lowest penalty followed by constant velocity and acceleration. In case of trajectories with collisions, the penalty increases with the severity of a collision. The penalty value 1 expresses that no safe trajectory could be found. The acceleration profile with the lowest penalty value is used as label for that scenario. This means when no collision-free trajectory is found, the acceleration profile

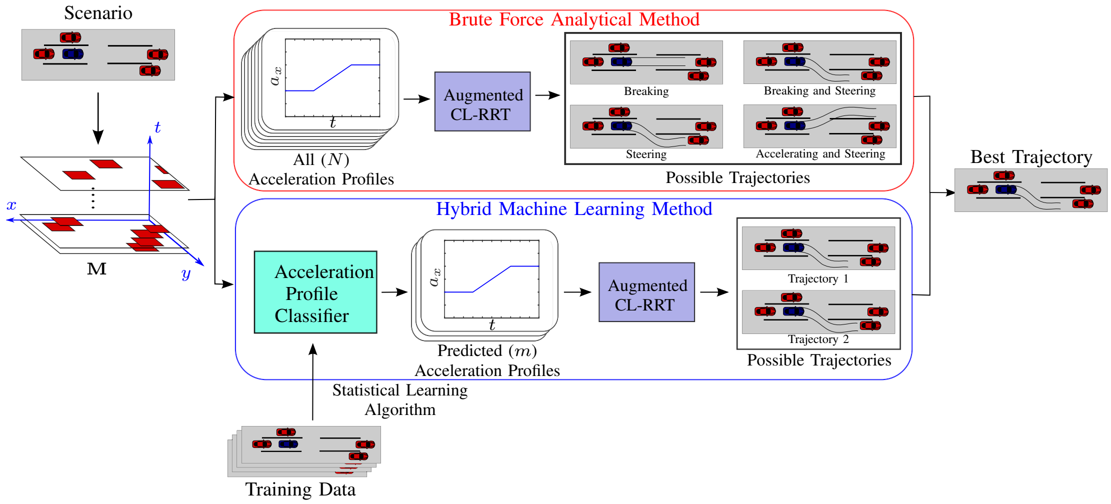

Figure 2. Brute force analytical and proposed hybrid machine learning algorithm

is selected only based on the severity of injury. The selected acceleration profile is represented by a one-hot vector $\mathbf{y}$ of dimension equal to $N+1$, i. e., the total number of classes.

The sequence of predicted occupancy grids $\mathbf{M}$ and the EGO vehicle physical parameters $\boldsymbol{\theta}$ are used as input for the machine learning algorithm. The goal of the machine learning algorithm is to find a function $\boldsymbol{f_\gamma}(\mathbf{M},\boldsymbol{\theta}) \mapsto \hat{\mathbf{y}}$ with parameters $\boldsymbol{\gamma}$ to minimize the risk $R(\boldsymbol{f_\gamma})$ which is defined as expectation

$$R(\boldsymbol{f_\gamma}) = E_{\mathbf{M},\boldsymbol{\theta},\mathbf{y}}\{\mathcal{L}(\mathbf{y},\boldsymbol{f_\gamma}(\mathbf{M},\boldsymbol{\theta}))\}, \tag{1}$$

where $\mathcal{L}(\mathbf{y},\hat{\mathbf{y}})) = -\sum_k y_k \log(\hat{y}_k)$ represents the cross-entropy loss function, with y_k and $\hat{y}_k$ being the true and estimated posterior probabilities for the k^{th} class. The risk $R(\boldsymbol{f_\gamma})$ is replaced by the empirical risk

$$R_{emp}(\boldsymbol{f_\gamma}) = \left(\frac{1}{L}\sum_{i=1}^{L}\left(\mathcal{L}(\mathbf{y}_i,\boldsymbol{f_\gamma}(\mathbf{M}_i,\boldsymbol{\theta}_i))\right)\right), \tag{2}$$

where L is total number of training data. Thus, the parameters $\boldsymbol{\gamma}$ of the classifier are found by minimizing $R_{emp}(\boldsymbol{f_\gamma})$.

Convolutional neural netowork (ConvNet) [18] is a specialized kind of neural network used for computing estimated probability distribution $\hat{\mathbf{y}}$ given an input that has grid-like topology such as $\mathbf{M}$. In order to learn spatiotemporal features from the input $\mathbf{M}$, a 3D ConvNet is used instead of 2D ConvNet. A network is designed which uses features extracted by convolution layers and hand-designed features of the EGO vehicle for the classification task.

The Augmented CL-RRT algorithm also uses predictions of other road traffic participants for the safe trajectory planning. Therefore, constructing occupancy grids from these predictions does not lead to much additional computation. The advantage of using such representation is that the dimension of the input remains the same, even if the number of road participants or the road infrastructure is changed. The procedure in the brute force analytical and in the hybrid machine learning method is visualized in Fig. 2. Initially, when a critical situation is detected, a scenario is converted into a sequence of predicted occupancy grids $\mathbf{M}$ using suitable models for road traffic participants. The brute force analytical method uses all N acceleration profiles while the hybrid machine learning method uses only predicted m acceleration profiles with the Augmented CL-RRT algorithm to find safe trajectories from which the best trajectory is selected.

III. Augmented CL-RRT and its Extension

The Augmented CL-RRT algorithm [12] considers vehicle nonlinear dynamics for trajectory planning in the form

$$\dot{\mathbf{s}}(t) = f(\mathbf{s}(t),\mathbf{u}(t)), \tag{3}$$

where $\mathbf{u}(t) \in \mathbb{R}^m$ is the control input and $\mathbf{s}(t) \in \mathbb{R}^2$ represents the area occupied by the vehicle at time t. In an iterative process, the algorithm samples a random point $\mathbf{s}_{rand}$ with some bias towards goal region S_{goal} and extends the tree by incremental motion towards $\mathbf{s}_{rand}$ from the closest $\mathbf{s}_k(t)$, that is stored previously in a tree, for the time interval Δt using differential constraints as in Eq. (3) to get the new state $\mathbf{s}_{k+1}(t+\Delta t)$. This state is added to the tree, if $\mathbf{s}_{k+1}(t+\Delta t) \in S_{free}(t+\Delta t)$, where $S_{free}(t+\Delta t)$ expresses the road area in $\mathbb{R}^2$ which is not occupied by other road traffic participants predictions at prediction-time $t+\Delta t$. The algorithm stores these positions along with yaw angle Ψ, the velocity v, the steering angle δ, and time $(t+\Delta t)$ at which the position will be reached. The Augmented CL-RRT uses a two-track model [17] of the vehicle, so the control input is

$$\mathbf{u}(t) = [\delta_{fl},\delta_{fr},\delta_{rl},\delta_{rr},s_{l,fl},s_{l,fr},s_{l,rl},s_{l,rr}], \tag{4}$$

where $\delta_{fl},\delta_{fr},\delta_{rl},\delta_{rr}$ are the four angles of the wheels with respect to longitudinal axis of the vehicle, and $s_{l,fl},s_{l,fr},s_{l,rl},s_{l,rr}$ are longitudinal slip values of the tires. The letters in subscript stand for "front-left", "front-right", "rear-left", and "rear-right". The longitudinal slip values are provided by the predefined longitudinal acceleration profiles considering a fixed friction coefficient between the tires and the road, and the Augmented CL-RRT algorithm finds steering wheel angles.

The algorithm divides a large prediction interval $[t_0, t_0+\tau]$ into two subintervals $[t_0, t_0+\tau_1]$ and $[t_0+\tau_1, t_0+\tau]$ with t_0 being the initial time. In the first subinterval $[t_0, t_0+\tau_1]$, it uses different acceleration profiles to compute multiple collision-free trajectories. The level of safety of these trajectories is measured by the maximum steering angle input δ_e, named as *steering effort*, required for the vehicle to follow the road safely in the second subinterval $[t_0+\tau_1, t_0+\tau]$ (from the end state of the collision-free trajectories found in first subinterval $[t_0, t_0+\tau_1]$). Trajectories with lower steering efforts correspond to a high level of safety and the one with the lowest steering effort is chosen by the vehicle to follow.

A. Prediction of Severity of Injury

The Augmented CL-RRT algorithm provides the possibility of just finding a collision-free trajectory. An extension is made to the Augmented CL-RRT algorithm to find a trajectory with a low severity of injury when a collision is unavoidable.

Instead of rejecting a state $\mathbf{s}(t)$, when $\mathbf{s}(t) \notin S_{free}(t)$, it is stored in a tree if its predicted severity of injury is low. A tree is not extended from such states as they are not safe states. Vehicle's delta-v [19], which is a change in velocity between pre-collision and post-collision trajectories of a vehicle, is considered as best single predictor of crash severity. But to apply the delta-v concept, a detailed model of a crash is necessary which is not a focus of this work. In this work a model proposed by [20] is used. It has developed the generalized relationship between impact speeds, impact angles, and the severity of injury probability for common crash types based on several simple assumptions. As the two-track model is used for estimating vehicle states, the estimated impact velocity and the estimated impact angle are available. Critical impact speeds for different crash types for which the probability of fatal or serious injury is less than 10% rounded to nearest 5 km/h is used in this work and shown in the Table I. If the impact speed is less than the critical impact speed, then the trajectory is considered as nonsevere. Nonsevere trajectories are further compared based on the impact velocity.

Table I
APPROAXIMATE CRITICAL IMPACT SPEEDS FOR COMMON CRASH TYPES

Crash-type	**Critical impact speed (km/h)**
Pedestrian-vehicle	20
Frontal crash	30
Side crash	30
Rear crash	55

When the algorithm fails to find a trajectory without any fatal or serious injury, the vehicle should break sharply.

IV. Convolutional Neural Networks

ConvNets are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. Mainly, three types of layers are stacked one after another to build ConvNets: convolution layer, pooling layer, and fully-connected layer.

Convolution layer uses two important ideas to improve a machine learning system: *sparse interaction* and *parameter sharing* [21]. In traditional neural network layers, the interaction between each output unit and input unit is described by a separate parameter. Convolution layer have *sparse interaction* by making the convolution kernels smaller than the input. This results into fewer number of parameters and operations to compute output decreasing memory and time requirements of the model. Further, every kernel is slided over the entire input dimension and convolved. This *parameter sharing* approach leads to further reduction in parameters of the model. The convolution layer is usually followed by a differential activation function such as sigmoid, rectified linear unit, etc.

The pooling layer is generally inserted between successive convolution layers. It converts the convolutional layer output into a number of slices and finds a single summary representation for each slice. Thus, it reduces the amount of the parameters and computation in the network. Commonly used pooling functions are max pooling, average pooling, etc.

After a number of convolution and pooling layers follows a fully-connected layer. The output layer has one neuron per class in the classification task. A softmax function is used, so that each neuron's output represents the posterior class probability.

A. 3D ConvNets

In order to extract spatiotemporal features from $\mathbf{M}$, a 3D convolution [22] is used. Then an additive bias b is applied and the result is passed through an activation function σ. As shown in Fig. 3, the value of unit $\mathbf{V}^{xyt}$ in a feature map at position (x, y) and time t is obtained by

$$\mathbf{V}^{xyt} = \sigma\left(b + \sum_{i=0}^{P-1}\sum_{j=0}^{Q-1}\sum_{n=0}^{R-1}\mathbf{W}_{ijn}\mathbf{U}_{ijn}^{xyt}\right), \tag{5}$$

where P, Q, and R are the dimensions of the kernel $\mathbf{W}$ and $\mathbf{U}_{ijn}^{xyt}$ is the input at position $(x+i\Delta x, y+j\Delta y, t+n\Delta t)$.

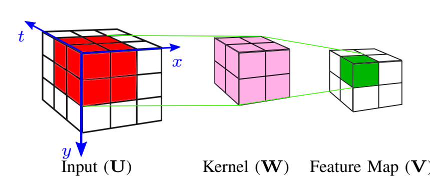

Figure 3. 3D Convolution

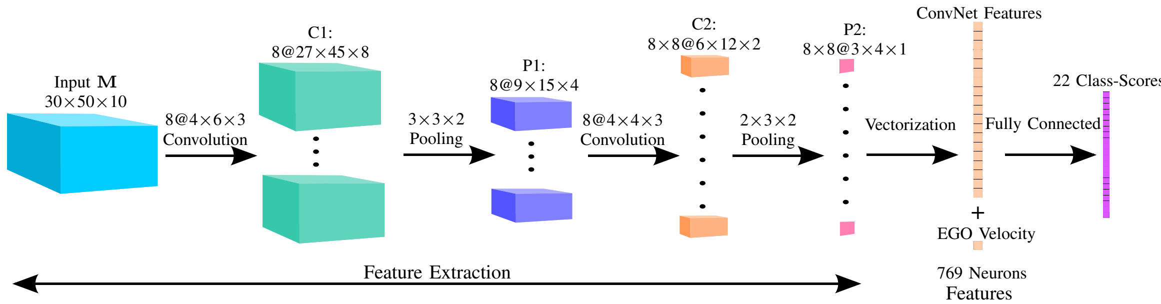

Figure 4. 3D-ConvNet architecture

V. Feature Learning and Training

A. Data Generation

A Matlab-based simulation environment is developed to design and simulate traffic scenarios and generate data. Two types of road designs are used to generate data: 1) curved road and 2) intersection. In each road design initial velocities, initial positions, and also number of road participants are changed to generate many traffic scenarios. For the curved road design, the curve radius is also changed. An example of a traffic scenario for both road designs is shown in Fig. 5 and 6.

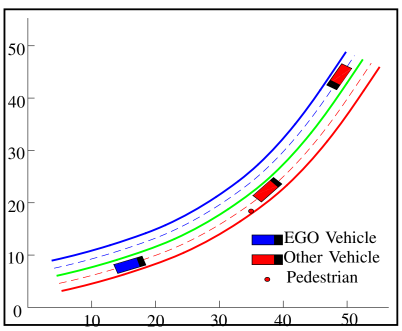

Figure 5. Curved road

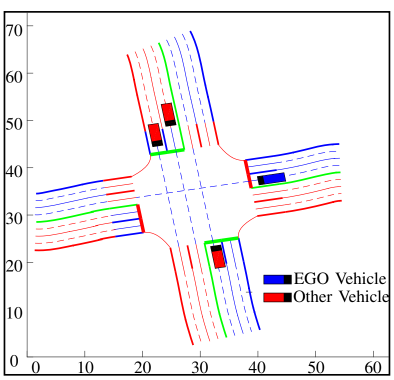

Figure 6. Intersection

The scenarios are simulated using suitable models for road traffic participants with assumption that vehicles tend to follow the road with constant velocity and pedestrian travel linearly with constant velocity. The scenarios in which a collision is detected with any other road traffic participant within 2 seconds in future which is not avoidable just by braking are taken as critical scenarios and considered for training. In total, 42191 scenarios for curved road designs and 13487 critical scenarios for intersection design are generated. In each of these scenarios safe trajectories are found by the Augmented CL-RRT algorithm. 21 different piecewise acceleration profiles designed by combination of braking (-8 and -4 m/s^2), acceleration ($+2$ and $+4$ m/s^2), and constant velocity are used with the Augmented CL-RRT for the time interval $[t_0, t_0+\tau_1]$ to find multiple safe trajectories.

As mentioned in Section II, for the labeling of the classification task, a one-hot vector $\mathbf{y}$, i. e., only one acceleration profile is needed to be assigned for each scenario. In this work the selection of the best acceleration profile is done based on penalty function $\mathcal{P}$ defined in Section II.

B. 3D-ConvNet Architecture

A variety of 3D ConvNet can be designed by changing the number of layers, the number of kernels, the size of kernels, etc. Three different architectures were tried in this work. The dimensions of kernels and the strides in the respective convolution layer are not changed. All convolution layers are followed by rectified linear unit activation function and average pooling in all architectures. The architecture providing good results with least number of parameters is finally selected. In this architecture, shown in Fig. 4, 10 predicted occupancy grids $\{\mathcal{G}_{t_0+\Delta t}, \ldots, \mathcal{G}_{t_0+\tau_1}\}$ of the size 30×50 meters (each cell 1×1 meter) over the prediction time interval $[t_0+\Delta t, t_0+\tau_1]$ are used as input. τ_1 and Δt are chosen to be 2 seconds and 0.2 seconds, respectively. 3D convolutions with 8 kernels of size $4\times 6\times 3$ (4×6 in the spatial dimension and 3 in temporal dimension) are applied to generate 8 sets of feature maps C1 of size $27\times 45\times 8$. The number of trainable parameters in this layer is 584. This is followed by $3\times 3\times 2$ pooling on each feature map of C1 layer to generate the same number of feature maps with reduced dimension of $9\times 15\times 4$. There are no trainable parameters for this layer. The layer C2 with dimensions of each feature map $6\times 12\times 2$ is obtained by applying convolutions with 8 kernels of size $4\times 4\times 3$. The number of trainable parameters is 392. The number of feature maps generated are 64 as 8 kernel have applied convolution on 8 feature maps from the P1 layer. A $2 \times 3 \times 2$ pooling is again performed on each of these feature maps to generate 64 feature maps of size $3 \times 4 \times 1$. Finally, all these feature maps are vectorized and the EGO vehicle initial velocity is added as hand-designed feature to form a vector of 769 numbers which is fully connected to the output layer. 22 class-scores, which correspond to 21 longitudinal acceleration profiles and one for no safe trajectory found, are calculated by matrix multiplication followed by the bias offset and a softmax classifier. The number of trainable parameters in this layer is 16940. The total number of learnable parameters in the whole network is 17916.

C. Training

The weights of the convolution layers are initialized randomly, and the weights of fully connected layer are initialized randomly with normalized initialization suggested in [23]. All biases are initialized with zero. The parameters are learned by the stochastic gradient descent with momentum method [24] using mini-batches of size 128. The momentum coefficient is kept fixed to 0.9 throughout the training. The learning rate is initialized at 0.1 and is halved after each epoch. The training is stopped after 15 epochs. The 3D-ConvNet is implemented in Matlab by modifying 2D-ConvNet implementation in [25].

D. Evaluation Metric for 3D-ConvNet

A *top*-i classification error, which tests if the reference class was within i hypotheses with highest probability, is used in this work. This metric is suitable as the goal of this work is to find m best acceleration profiles, i. e., top m classes out of N for using with the Augmented CL-RRT algorithm to find a safe trajectory. The softmax classifier used in the 3D-ConvNet architecture gives probabilities for each class which are used to get top m classes with highest probabilities. *top*-2 and *top*-3 classification error for evaluation of 3D-ConvNet are used.

VI. Results

A. Classification results of 3D-ConvNet

3D-ConvNets with different architectures (having different number of convolutional layers and kernels) are used to train acceleration profile classifiers for both curved road and intersection road designs shown in Fig. 5 and 6.

The results in Table II shows that networks with 2 convolution layers and 8 filters in each convolution layer, as shown in Fig. 4, have least *top*-2 and *top*-3 errors, and also have the least number of parameters which makes them most efficient in terms of memory. This architecture is further used to measure the efficiency of the 3D-ConvNet method with top 3 acceleration profiles.

Less accuracy in intersection design shows the importance of large data for 3D-ConvNet.

B. Comparison of Brute Force Analytical and 3D-ConvNet Method

The proposed method is compared with the brute force analytical method based on two criteria: *efficiency* and *safety*.

Table II
COMPARISON BETWEEN 3D-CONVNET ARCHITECTURES

Architecture	**1 Convolution**	**2 Convolution**	**2 Convolution**
Number of Kernels in Convolutional Layer	8	8	12
Number of Learnable Parameters	47934	17916	39524
Curved Road			
-*top*-3 (%) error	4.14	2.21	3.89
-*top*-2 (%) error	18.34	9.32	11.64
Intersection			
-*top*-3 (%) error	17.62	13.72	14.9
-*top*-2 (%) error	30.13	20.82	26.36

The efficiency is measured based on the time and the number of random samples required for the execution, and the safety is measured based on the success-rate, i. e., percentage of scenarios in which it was able to find a safe trajectory. 1890 random test scenarios for curved road design and 820 random test scenarios for intersection design are simulated in the Matlab simulation environment, and the brute force analytical and the 3D-ConvNet methods are used to find safe trajectories. Maximum of 100 samples are used with every acceleration profile to find a safe trajectory.

Table III
BRUTE FORCE ANALYTICAL AND 3D-CONVNET METHOD COMPARISON

	Curved Road		**Intersection**	
Criteria	**Brute Force Analytical**	**3D-ConvNet**	**Brute Force Analytical**	**3D-ConvNet**
Maximum Samples	1978	300	2100	300
Minimum Samples	678	21	810	61
Average Samples	1272	109	1868	237
Maximum Time (Sec.)	14.2441	1.4798	11.1528	2.4521
Minimum Time (Sec.)	2.1595	0.2801	4.0976	0.6281
Average Time (Sec.)	5.0518	0.6230	6.6989	1.3126
Collision-free Trajectory Found (%)	100	99.63	60.85	51.95
Nonsevere Trajectory Found (%)	0	0.37	39.15	42.37

Table III shows the results of the simulation by both methods. The average number of samples required by the brute force analytical method in curved road and intersection scenarios was 1272 and 1868, respectively. On the other hand, the 3D-ConvNet needed only 109 and 237 samples in both road designs on an average. Thus, 3D-ConvNet method shows 11.67 and 7.88 times improvement in terms of the number of samples required for the safe trajectory planning in curved road and intersection scenarios, respectively. Similarly, a comparison for the time required by both methods in curved road and intersection scenario shows 8.11 and 5.1 times improvement, respectively. The time required for constructing a sequence of predicted occupancy grids $\mathbf{M}$ is included in the time required for finding safe trajectories by the 3D-ConvNet method. As only 3 instead of 21 acceleration profiles are used, 7 times reduction in the number of samples and the time is expected. But, suitable acceleration profiles requires many a times less than 100 samples for finding a collision-free trajectory. Therefore, the proposed approach shows an improvement more than 7 times. Only intersection scenarios show less than 7 times improvement in terms of time required as many samples lead to collision states which further requires computation of the severity of injury.

Table III also shows the percentage of scenarios in which both algorithms were able to find safe trajectories. For the curved road design, the brute force analytical method was able to find 100% collision-free trajectories. Although, the 3D-ConvNet method was able to find a collision-free trajectory in 99.63% scenarios, in the rest of 0.37% scenarios it was able to find a nonsevere trajectory. Similarly, in intersection scenarios the brute force analytical method was able to again find a safe trajectory in 100% scenarios, but 3D-ConvNet was able to find a safe trajectory in 94.32% scenarios.

VII. Conclusion

This work is introducing a hybrid machine learning approach for planning a safe trajectory in complex road traffic-scenarios. A 3D-ConvNet, a variant of convolutional neural network, is used to learn spatiotemporal features from a sequence of predicted occupancy grids generated from predictions of other road traffic participants. These learnt features and hand-designed features of the EGO vehicle are used to estimate best acceleration profiles for the trajectory planning with the Augmented CL-RRT algorithm. A methodology for the prediction of the severity of injury with the Augmented CL-RRT algorithm is introduced, when a collision is unavoidable. Using simulations of a large number of scenarios for curved roads and intersections, a brute force analytical approach is compared with the proposed approach for the safe trajectory planning based on two criteria, i. e., efficiency and safety. The results show multiple times improvement in the efficiency without harming safety in most of the scenarios. It is planned to implement the proposed algorithm in a microcontroller followed by the testing and the validation in a vehicle on the test-track.

Acknowledgment

This research work is funded by the German Federal Ministry of Education and Research (BMBF) under grant number 03FH034PX4. The authors would also want to thank Automotive Safety Technologies GmbH for supporting this work.

References

[1] M. Kuehn, T. Hummel and J. Bende, “Benefit estimation of advanced driver assistance systems for cars derived from real-life scenarios", *21st International Technical Conference on the Enhanced Safety of vehicles*, 2009.

[2] S. Waydo and R. M. Murray, “Vehicle motion planning using stream functions", *International Conference on Robotics and Automation*, 2003.

[3] P. Hart, N. Nilsson and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths", *IEEE Transactions On System Science and Cybernetics*, 1968.

[4] A. Stentz, “Optimal and efficient path planning for partially-known environments", *IEEE International Conference on Robotics and Automation*, 1994.

[5] S. Koenig and M. Likhachev, “D* Lite", *National Conference on Artificial Intelligence*, 2002.

[6] J. Kuffner and S. LaValle, “Rrt-connect: An efficient approach to single-query path planning",*IEEE International Conference on Robotics and Automation*, 2000.

[7] A. Neto, D. Macharet and M. Campos, “Feasible RRT-based Path Planning Using Seventh Order Bezier Curves", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[8] E. Shan, B.Dai, J. Song and Z. Sun, “An Dynamic RRT Path Planning Algorithm Based On B-Spline”, Second International Symposeum on Computational Intelligence and Design, 2009.

[9] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, J. How, “Real-Time Motion Planning With Applications to Autonomous Urban Driving”, *IEEE Trans. on Control Systems Tech.*, 2009.

[10] K. Macek, M. Becked and R. Siegwart, “Motion planning for car-like vehicles in dynamic urban scenarios”, *IEEE/RSJ International conference on Intelligent Robots and Systems*, 2006.

[11] M. Otte and E. Frazzoli, “RRT-X: Real-Time Motion Planning/Replanning for Environments with Unpredictable Obstacles”, *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2014.

[12] A. Chaulwar, M. Botsch, T. Krueger , and T. Miehling, “Planning of safe trajectories in dynamic multi-object traffic-scenarios”, *Journal of Traffic and Logistics Engineering*, 2016.

[13] M. Buehler, K. Langnemma, and S. Singh, “The 2005 Darpa Grand Challenge", Springer Tracts in Advanced Robotics, 2007.

[14] M. Otte, A survey of machine learning approaches to robotic path-planning. [Online]. Available: https://www.cs.colorado.edu/~mozer/Teaching/Computational\%20Modeling\%20Prelim/Otte.pdf

[15] J. Miura, “Support vector path planning”,*International Conference on Intelligent Robots and Systems*, 2006.

[16] K. Kusano, “Methodology for Determining Crash and Injury Reduction from Emerging Crash Prevention Systems in the U.S.", *Virginia Polytechnic Institute and State Univeristy*, 2013.

[17] R. Jazar, “Vehicle Dynamics Theory and Application”, *2ed, Chapter 10, Vehicle Planar Dynamics, Springer*, 2014.

[18] Y. LeCunn, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to Documnet Recognition*, *IEEE*, 1998.

[19] H. C. Joksch, “Velocity change and fatality risk in a crash—a rule of thumb”, 1993.

[20] C. Jurewicza, A. Sobhania, J. Woolleyb, J. Dutschkeb, B. Corbenc, “Exploration of Vehicle Impact Speed – Injury Severity Relationships for Application in Safer Road Design”, *Transportation research procedia*, 2016.

[21] I. Goodfellow Y. Bengio, and A. Courville, *Deep Learning*. [Online]. Available: http://www.deeplearningbook.org/

[22] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, “3D convolutional neural networks for human action recognition,” *IEEE TPAMI*, 2013.

[23] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks”, *AISTATS*, 2010.

[24] B. Polyak, “Some methods of speeding up the convergence of iteration methods”, *USSR Computational Mathematics and Mathematical Physics*, 1964.

[25] Andrew Mass and Sameep Tondon, stanford_dl_ex_solution. Available: https://github.com/kloudkl/stanford_dl_ex-solutions.