International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Enhancing the Accuracy and Autonomy of Autonomous Vehicles using DRL

T. Bal Nagender Singh[1]*, K. Bala Naga Raju*[2]*, S. Sai Mithun Reddy*[3]*, A.B.V.S Sayeesh*[4]*, S. Balaji Divya Dhatri*[5]*, Mr. Sabyasachi Chakraborty*[6]

[1,2,3,4,5] UG Student, B. Tech, School of Engineering, Hyderabad, India

[6] Guide, Assistant Professor, Department of CSE(AI-ML)-Malla Reddy University, Hyderabad, Telengana. & PhD Scholar, Department of CSE, C.V Raman Global University, Bhubaneshwar, Odisha

2111CS020073@mallareddyuniversity.ac.in[1], 2111CS020074@mallareddyuniversity.ac.in[2], 2111CS020076@mallareddyuniversity.ac.in[3], 2111CS020072@mallareddyuniversity.ac.in[4], 2111CS020075@mallareddyuniversity.ac.in[5],

ABSTRACT—

Self Driving Car is one of the biggest path breaking invention challenge in the current world of Automation Technology. Deep Learning Techniques in multilayer Neural Network are the determinants of perfection of executing driver activities, actions, reactions and responses to the captured images of path and it's various traffic signals and obstacles. Multilayer Convolutional Neural Networks train Deep Learning Techniques on the basis of images captured. In this paper, the Driver Imitation Algorithm is analyzed and attempted to mix with Reinforcement Learning Technique to achieve optimal accuracy. The Driver imitation algorithm, Behavioral cloning, obstacle and collision detection and avoidance are the key factors to be enhanced. The input images and signals are captured through the Camera, LIDAR and RADAR and used to train the Deep Reinforcement Learning process to apply on the various algorithms and techniques. The simulator is used to examine the training and testing of algorithms in a riskless environment and the results are analytically interpreted.

Keywords— ***Deep learning, neural network, Convolutional neural network, stimulator,Deep Reinforcement Learning, Behavioral Cloning, Imitation Learning. Introduction (*Heading 1*)***

I. INTRODUCTION

It has been observed that maximum number of road accidents are` caused by the Driver's errors. The humane imperfection and mistakes are the major responsible factors of maximum number of collisions and clashes. Introduction to Deep Learning and Artificial CNN has shown the ways of introducing self controlled and self driven vehicles where the human errors are attempted to be minimized with the help of enforcing full intelligent automation where the DL program have full control over all the necessary parts of the vehicle and direct and manage them with accurate responses after being trained with large number of input images.

II. Study of Existing Researches

Convolutional Neural Network (CNN) in accordance with various classifiers facilitates and enhances the processes of pattern recognition.In this process the major aspects are the Feature extraction and generation of knowledge . On a captured input image, CNN performs the image classification and feature extractions CNN in combination with Reward-Penalty based optimization of Reinforcement Learning brings the accuracy of training . 1) Huge data sets like "Image Net Large Scale Visual Recognition Challenge" (ILSVRC) have become more accessible and can be validated as well.

2) i) Overview of RL algorithm and it's application background for the automotive community.

ii) Detailed literature review of using RL for different autonomous driving tasks. • Discussion of the key challenges and iii) opportunities for RL applied to real world autonomous driving

The paper discussed the components of Self Driving Car e.g. LIDAR, RADAR etc , overview of Reinforcement Learning, application of Deep Reinforcement Learning and it's modified forms in Self Driving Car and finally the challenges in deployment in the real life situation.

3) The paper examines a Reinforcement leaning technique "Asynchronous Actor-Critic Agent Algorithm" in combination with deep CNN to achieve the optimal controlling and tackling methods of driverless vehicle.

4) The paper "Discrete Control in Real –World driving environments using Deep Reinforcement Learning " introduces discrete control in real-world environments by introducing perception, planning, and control in real-world driving environments. The RL agent learns with minimal video data, and minimal training using the proposed multi-agent setting and training procedures on the basis of the assumption of unidirectional road and driving direction. The research based on the action space with two components : speed (f) and direction angle(θ) and uses R-CNN and Masked R-CNN to extend the analysis.

5) Here, the Behavioral Cloning Algorithm is used in accordance with CNN to train an end-to end model for controlling lateral motion of autonomous vehicles. In this paper , the CNN's ability to drive in similar but unknown situation by imitating driver actions is analyzed. . In a simulated environment, with the help of a small amount of training data, the resultant network was able to successfully drive the car .

6) Here, Behavioral cloning is implemented using CNN in which ELU activation function is used, which uses mean square of error loss , applied for validating data. Here the training is done on a bigger dataset and the trained model has achieved very good results in test driving.

7) In this paper, the research is based on the learning of CNN with multiple parallel GPUs based on DARPA autonomous vehicles where training input is extracted through the videos captured through the cameras as well as the reading of steering angles . The research examines the NVIDIA's new effort on DARPA Autonomous Vehicle (DAVE) to minimize the need of recognizing the human designated features like Lane Markings, Guard Rails or the movements of other cars. It takes the input images extracted from the videos captured through the cameras on timely basis and simultaneously the steering angles in the respective points of time which is obtained by tapping the vehicle's Controller Area Network(CAN) bus.

Here the steering command is represented by 1/r where r is the turning radius of the car. 1/r <0 for left turns and 1/r >0 for right turns and for straight drive , 1/r =0 because r is infinite and thus there is no movement of steering

Research Results: i. Simulation Result: Simulated test estimates the percentage of time the network could drive the car independently which is referred as "Autonomy" .

In the simulated environment, the human intervention is required if the simulated vehicle deviates by over a meter from the center line . The human intervention is assumed to take 6 seconds on an average to re-center the vehicle and then after the self starting mode could be restarted.

The Autonomy(the percentage estimation) is estimated by the derived formula :

[1- {(no. of interventions x 6Sec)/Elapsed Time (in seconds)}] x 100

On the basis of this formula, if we have 10 human interventions in 600 seconds , then ,

Autonomy= {1-(10 x6)/600} x 100 = (1-1/10) x 100=90.

Therefore , in that case the car achieves 90 percent autonomy.

The simulated test achieved a result of 99 percent autonomy as it required the human intervention of only 2-3 times in 600 seconds driving in test run.

ii.On Road Test Result: On Road Test Result achieved 98 percent autonomy . It didn't require any human intervention in a 10 miles test drives at a stretch.

8) The Mathematical model of Behavioral Cloning from observations is studied . In this paper it is concluded that BCO, an algorithm for performing imitation learning requires neither access to demonstrator actions nor post-demonstration environment interaction. In the experimental results in this paper show that it performs favorably with other imitation learning techniques that use demonstrator's action. However it uses post demonstration environment slight bit. This BCO algorithm tested to reduce the execution time.

[9] The combination of Reward based Reinforcement Learning and Imitation Learning from Observations is made through Reinforced Inverse Dynamics Modeling (RIDM) is made. Here the Reinforcement Learning is typically aimed at maximizing the number of reward points achieved upon moving from one state to the next as a result of a given action taken by the agent.

In contrast to RL, the Imitation Learning is concerned with the imitation of Expert's behavior by the learner rather than maximizing the external reward.

In this study, RIDM is introduced to find an action which is responsible to make a transition from current state to the next desired state and by doing so, the the sequence of generated actions is aimed at maximizing the cumulative reward from the environment. The RIDM is based on the idea of no dependence on demonstrator action information as the action is assumed to be derived from the current state and the desired next state by the Inverse Dynamics Model function . It is investigated whether or by which extent, the independence of expert demonstration is viable by evaluating RIDM . Finally the result depicts the success of RIDM and the success is based on a single State-only demonstration which is the initial state and absolutely without any expert action demonstration.

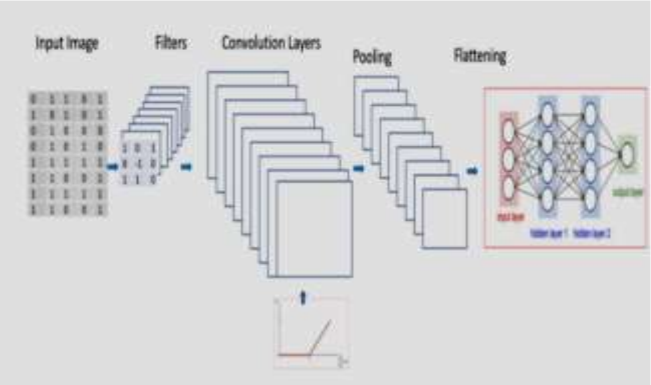
The next section contains the detailed study on it.

II. I Key Concepts and Applications extracted from the study

A) CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN is a special type of Neural Network that contains hidden layers used to perform the processing image like 2D matrix data. There are the filters **or** feature detectors also referred as Kernels that represent the color channels. CNN applies the filters to the input image **to** generate the feature maps or the activation maps. Relu activation function is used for this to omit out the negative values.. Feature detectors or filters help identify different features present in an image like edges, vertical lines, horizontal lines, bends, etc

CNN has the following layers:



Layers of CNN

If any of the layers fail to perform their task then the process will never be executed.

In the convolution layer we first line up the feature along with picture and then multiply the pixel value with the corresponding value of filter and then adding them up and driving them with the absolute value of pixels.

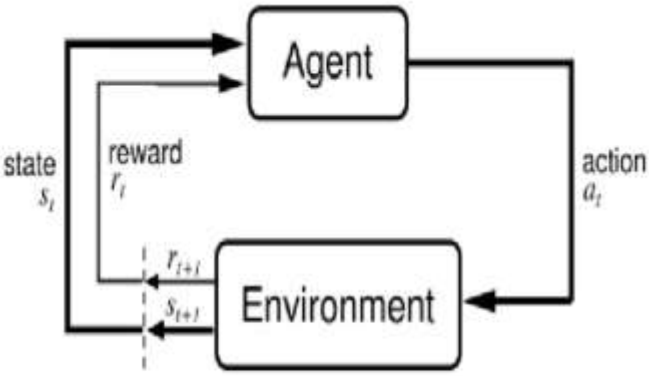
Relu layer represents rectified layer unit crafted by this layer is to expel all the negative qualities from the filter picture and afterward change it with zero. The node is activated if the image is above a particular quality and it is linearly dependent on the variable considering the input is above threshold value. Image received from Relu is shrunk in the pooling layer.

The actual classification is done in a fully connected layer or “Flattering Layer”. We take the shrieked image and up that in a single list. And then we compare that image with our previously-stored list and judge the image. The last is the output layer which gives the output of the classified image.

pictures caught in a recreated situation (street, markers, scene) are a decent estimate of pictures that could be caught in reality.

The stimulator additionally gives security and convenience. Information assortment isn't troublesome we can gather information effectively, and if the model fails then there is no risk to life. To study and improve different model structures a stimulator is the best stage. We can implement our model afterward in a real car with real cameras, but for that, we should have to make it successful run efficiently in our stimulator.

B) Reinforcement Learning Process



Basic Framework of RL Process

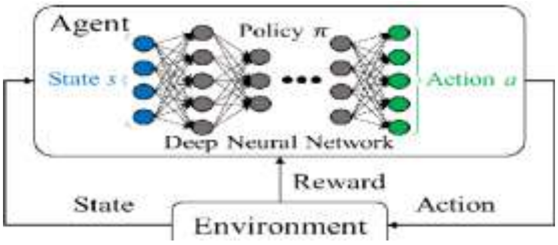
Reinforcement learning is a special type of machine learning technique in which a computer learns to perform a task through repeated interactions with a dynamic environment.

In the training process, unlike supervised learning, the input data are not labeled and the program(a software module referred as Agent) learns from a series of repeated interactions with the environment and for a good result, a reward is achieved and for a failure or bad result, a penalty is assigned.

The process will go in repeating till the penalty is minimized and/or the reward is maximized. The goal of RL is to find a strategy that would optimize the outcome.

Here, this trial-and-error learning approach enables the computer to make a series of decisions without human intervention and explicitly written programs to perform the task.

C) Deep Reinforcement Learning



Deep RL Framework

Next step of study is the combination of Deep Neural Network and Reinforcement Learning. Here the Deep Neural Network is used for getting input from the environment along with Reward or Penalty points and determining the agent actions. Here in applying for Self Driving Cars, Deep Neural Network is actually a Deep CNN where the environment inputs and reward points will pass through a number of layers to produce optimal actions in each state and by doing so it may minimize the total no of steps of RL. In self driving car, R-CNN and Mask-CNN s are used.

D) BEHAVIORAL CLONING

In “Behavioral cloning” method, human sub cognitive skills can be captured and reproduced in a computer program. Whenever a human being performs a task, his action (which includes both physical action as well as sub psychological aptitudes like perceiving and understanding an object while simultaneously performing an action) along with the space are recorded. These records are used as inputs of a learning program. In one word, human sub cognitive , sub psychological and physical skills, aptitudes and actions along with the situation or the action space responsible for that action are imitated to a computer program. The learning program outputs a set of rules that reproduce the skilled behavior. In order to automate a control system, classical control theory appears to be inadequate, but behavioral cloning can be best choice to be used for that.

From [5]: The training phase contains the following function :

$f(x)$ = max(0, x) -------------------------------(i)

This represents Rectified Linear Unit (RELU) activation function where output will be 0 if the input is negative , otherwise the output carries forward the input itself. RELU is acted as the activation function when a 5 x5 stride is used during convolution.

In the Evaluation and testing phase, the following parameters are tested to find their quantitative values in percentage representing the degree of success.

i) Degree of Autonomy in percentage

$\alpha = (1 - N\times 2 / T) \times 100$ -----------------------------------(ii)

Here α indicates the degree of autonomy (in percentage) of the lateral vehicle control system in a multilane track

N is the number of lane changes and T is the total time elapsed in seconds.

ii) Degree of Staying ability in the Lane in percentage

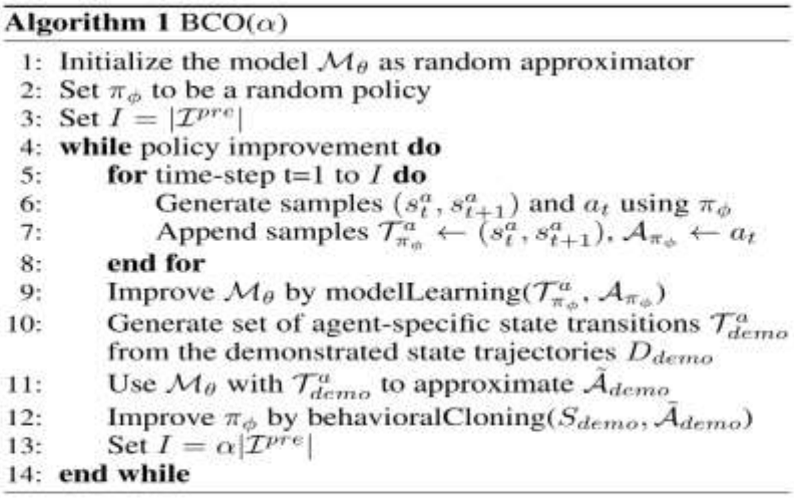
$\kappa = (1- \tau / T) \times 100$ ------------------------------(iii)

Here k represents the percentage of time , for which the vehicle stayed within the lane in a single lane track. Here τ is the time in seconds for which the vehicle left the lane and T is the total elapsed time.

In the simulated test, 96.62% autonomy was achieved here . For each lane change, a penalty of 2 seconds was added because simulated environment was very fast to execute an action. In the simulated Test Run, there was no human intervention. In the tracks with a single lane, the model achieved an autonomous lane keep ratio (percentage of time, for which the vehicle stayed within the lane in a single lane track) of 89.02%. The time for which the car deviated from the track is recorded and compared with the total time spent on track in seconds.

CNN learning enhanced with behavioral cloning from visual images and steering angle examined in [7] and discussed in the previous section in detail. The training process achieves it’s optimum by repeated back propagation and weight adjustments to dynamically update CNN results which in turn progresses towards “Autonmy”.

The Existing BCO algorithm is as follows:



```
Algorithm 1 BCO($\alpha$)
Initialize the model $\mathcal{M}_\theta$ as random approximator
Set $\pi_\phi$ to be a random policy
Set $I = |\mathcal{I}^{pre}|$
while policy improvement do
    for time-step t=1 to $I$ do
        Generate samples $(s_t^a, s_{t+1}^a)$ and $a_t$ using $\pi_\phi$
        Append samples $\mathcal{T}^a_{\pi_\phi} \leftarrow (s_t^a, s_{t+1}^a)$, $\mathcal{A}_{\pi_\phi} \leftarrow a_t$
    end for
    Improve $\mathcal{M}_\theta$ by modelLearning($\mathcal{T}^a_{\pi_\phi}$, $\mathcal{A}_{\pi_\phi}$)
    Generate set of agent-specific state transitions $\mathcal{T}^a_{demo}$
    from the demonstrated state trajectories $D_{demo}$
    Use $\mathcal{M}_\theta$ with $\mathcal{T}^a_{demo}$ to approximate $\tilde{\mathcal{A}}_{demo}$
    Improve $\pi_\phi$ by behavioralCloning($S_{demo}$, $\tilde{\mathcal{A}}_{demo}$)
    Set $I = \alpha|\mathcal{I}^{pre}|$
end while
```

IV. The Proposed Research :

Combination of Reinforced Learning and Behavioral Cloning or Imitation Learning from Observation

The studies [8],[9] are made on behavioral cloning from observations, imitation learning , inverse dynamic model , reinforced inverse dynamic model and the combination of the Reinforcement Learning and Imitation Learning .

Behavioral Cloning from Observation (BCO) framework is proposed in [8]. Here BCO is represented by α .Here The agent is initialized with a (random) policy which interacts with the environment and collects data to learn its own agent-specific inverse dynamics model. Then, given state-only demonstration information, the agent uses this learned model to infer the expert's missing action information. Once these actions have been inferred, the agent performs imitation learning. The updated policy is then used to collect data and this process repeats

The study is made to combine the Reinforced Learning and Imitation Learning from observation[9] is carried out with a new concept "Reinforced Inverse Dynamic Modeling" (RIDM) as outlined in the review section.

The study is based on the preliminary principle of RL with Markov decision Process (MDP) which is defined by the tuple:

M={S,A,T,R}

Where S is set of state spaces, A is the set of actions taken by the agent in a state to make a transition to another state. T is the environment transition function gives the probability that an agent can have a transition from one state to the next as a result of a given action taken by the agent .

$T : S_i$ x A x S_{i+1} -→ (0,1)

and R is the reward function that represents the reward points achieved by the agent upon moving from one state to another as a result of a given action taken.

The RL optimizes the agent's behavior in order to find a control policy

$\prod^*$: S→A

By which the agent can have successful transitions and maximize the total cumulative reward .

The Imitation Learning on the other hand , is concerned with the imitation of expert demonstration by the Learner rather than maximizing the eternal reward.

The expert demonstration is defined as :

D^e ={S_t^e , a_t^e }

Where S_t^e is the state of the expert at a time instance t . a_t^e denotes the action taken by the expert at that time .

The goal of IL is to learn a control policy that the learning agent can use to produce behavior similar to the expert. Now if it is assumed that the action information is absent , we have, D^e ={S_t^e} , which is called imitation from observation. Here the expert action {a_t^e } must be inferred to get {a_t^e }* for each state {S_t^e }.

Reinforced Inverse Dynamic Modeling (RIDM) is a new method of integrating Imitation from Observation and Reinforced Learning . In the RIDM learning strategy, the agent can select action { a_t } that will allow it to achieve a higher level of task performance when there is a single state only expert demonstration $D^e = \{S_t^e\}$. RIDM performs this by learning and using a task specific inverse dynamic model M_θ that infers which action to be taken at any given time instant , based on agent's current state and the desired next state .In other word, the RIDM function is used to select the action by which the agent can have a transition from the current state to the next state . Thus the action is the M_θ function of S_t^e and $S_{t+1.}$

$a_t = M_\theta (S_t , S_{t+1}^e)$

where S_t is the current state of the agent and S_{t+1}^e is the state of expert at t+1 th time .

The goal of RIDM function is to find out the optimal θ so that the sequence of actions generated will maximize the cumulative number of Reward points from the environment.

Here we have to maximize the reward function of M_θ

Maximize: $R(M_\theta)$

In this paper , the given studies of literatures .are extended to find a direction to combine Deep CNN, Reinforcement Learning, Behavioral cloning algorithm or Driver imitation algorithm to achieve a higher degree of accuracy in learning and execution. Firstly, behavioral cloning process is applied in Deep CNN and then the Deep –CNN to be incorporated into the learning process of Reinforcement Learning. This paper is attempts to apply behavioral cloning or imitation algorithm in Deep Reinforcement Learning to achieve an optimum accuracy.

The existing BCO algorithm is proposed to be Modified and altered by incorporating Deep CNN aspects for image capturing and preprocessing . In short, the state images of BCO are enhanced by Deep CNN theoretically and algorithmically .After incorporating Deep CNN in input image preprocessing and Processing, it is proposed to Combine Reinforcement Learning to incorporate Reward Function on BCO to optimize the cloning process from observation, or maximize the reward function $R(M_\theta)$ as mentioned before.

The Suggestive Process:

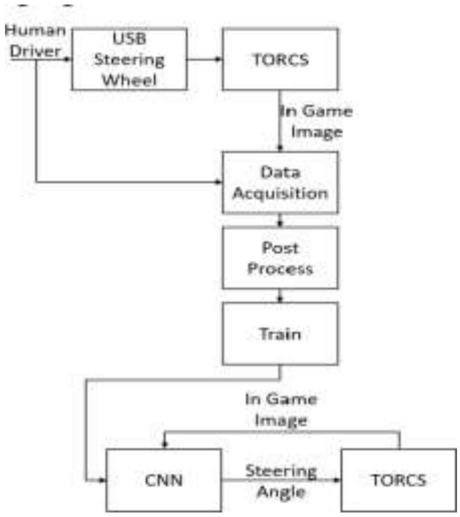
Step-1: Designing and Building the Driving Environment. and customizing it to make the RL agent able to train on it.

Step-2: Build and Train the Deep Convolutional Neural Network (DCNN) . In this step, the different layers of DCNN defined, built and trained with Reinforcement Learning process and incorporated with behavioral cloning or imitation algorithm.

Step-3: Test and Drive in simulated environment.

NETWORK ARCHITECTURE

Basic Block Diagram of Deep CNN with Behavioral Cloning/Imitation Learning



Behavioral Cloning with Deep CNN for Lateral Motion Control.

*__TORCS__ (**The Open Racing Car Simulator**) is a driving simulator to be used here. It is capable of simulating the essential elements of vehicular dynamics such as mass, rotational inertia, collision, mechanics of suspensions, links and differentials, friction and aerodynamics.

V. CONCLUSION

This paper proposes the combination of Deep Reinforcement learning and the behavioral cloning /imitation learning algorithm to make the .learning process more accurate, optimized and with less flaws in learning . It plans to implement the proposed model in simulated environment and test the outcomes. In our next work , we could train and test the model and analyze the outcome .In out next paper , it is planned to materialize the proposed system.

References :

"Autonomous Decision Making for a Driver-less Car", IEEE 2017. By Nicolas Gallardo; Nicholas Gamez; Paul Rad; Mo Jamshidi

"Deep Reinforcement Learning for Autonomous Driving:" A Survey Article in IEEE Transactions on Intelligent Transportation Systems,IEEE · February 2021

"Tackling Real-World Autonomous Driving using Deep Reinforcement Learning" by Paolo Maramotti1 , Alessandro Paolo Capasso2 , Giulio Bacchiani2 and Alberto Broggi2, IEEE Explore. rXiv:2207.02162v1 [cs.RO] 5 Jul 2022

"Discrete Control in Real World driving environments using Deep Reinforcement Learning "-Cornel University Jounal, Cited as -arXiv:2211.15920v2 [cs.AI] 30 Nov 2022 By Avinash Amballa, Bosch, Advaith P-IIT-Hyderabad, Pradip Sasmal –IIT Jodhpur, Sumahana Channapayya , IIT-Hyderabad

"Behavioral Cloning for Lateral Motion Control of Autonomous Vehicles Using Deep Learning" Conference Paper · May 2018 . By

Girma Tewolde Kettering University, Jaerock Kwon University of Michigan-Dearborn, IEEE Explore.

Self Driving Car using Deep Learning Technique By Chirag Sharma Vellore Institute of Technology, Chennai Tamil Nadu, India S. Bharathiraja Vellore Institute of Technology, Chennai Tamil Nadu, India G. Anusooya Vellore Institute of Technology, Chennai Tamil Nadu, India , International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 ,Published by : www.ijert.org Vol. 9 Issue 06, June-2020

End to End Learning for Self-Driving Cars by Mariusz Bojarski NVIDIA Corporation Holmdel, NJ 07735 Davide Del Testa NVIDIA Corporation Holmdel, NJ 07735 Daniel Dworakowski NVIDIA Corporation Holmdel, NJ 07735 Bernhard Firner NVIDIA Corporation Holmdel, NJ 07735 etc. NVIDIA ; Cited as : arXiv:1604.07316v1 [cs.CV] 25 Apr 2016

Behavioral Cloning from Observation By Faraz Torabi1 , Garrett Warnell2 , Peter Stone1 1 The University of Texas at Austin 2 U.S. Army Research Laboratory .{faraztrb,pstone}@cs.utexas.edu, garrett.a.warnell.civ@mail.mil ; Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

"Reinforced inverse Dynamics Modeling for Learning from a single observed demonstration." - By Brahma S. Pavse†,1 , Faraz Torabi†,1 , Josiah Hanna2 , Garrett Warnell3 , and Peter Stone4 In IEEE Robotics and Automation Letter, Presented at IROS 2020 July 2020