Talk to the Vehicle: Language Conditioned Autonomous Navigation of Self Driving Cars

by

Sriram N N, Tirth Maniar, Jayaganesh Kalyanasundaram, Vineet Gandhi, Madhava Krishna

in

IEEE/RSJ International Conference on Intelligent Robots and Systems
(*IROS 2019*)

The Venetian Macao, Macau, China

Report No: IIIT/TR/2019/-1

Centre for Robotics
International Institute of Information Technology
Hyderabad - 500 032, INDIA
November 2019

Talk to the Vehicle: Language Conditioned Autonomous Navigation of Self Driving Cars

Sriram N. N.*[1], Tirth Maniar*[1], Jayaganesh Kalyanasundaram[1], Vineet Gandhi[1]
Brojeshwar Bhowmick[2], K Madhava Krishna[1]

***Abstract*— We propose a novel pipeline that blends encodings from natural language and 3D semantic maps obtained from visual imagery to generate local trajectories that are executed by a low-level controller. The pipeline precludes the need for a prior registered map through a local waypoint generator neural network. The waypoint generator network (WGN) maps semantics and natural language encodings (NLE) to local waypoints. A local planner then generates a trajectory from the ego location of the vehicle (an outdoor car in this case) to these locally generated waypoints while a low-level controller executes these plans faithfully. The efficacy of the pipeline is verified in the CARLA simulator environment as well as on local semantic maps built from real-world KITTI dataset. In both these environments (simulated and real-world) we show the ability of the WGN to generate waypoints accurately by mapping NLE of varying sequence lengths and levels of complexity. We compare with baseline approaches and show significant performance gain over them. And finally, we show real implementations on our electric car verifying that the pipeline lends itself to practical and tangible realizations in uncontrolled outdoor settings. In loop execution of the proposed pipeline that involves repetitive invocations of the network is critical for any such language-based navigation framework. This effort successfully accomplishes this thereby bypassing the need for prior metric maps or strategies for metric level localization during traversal.**

I. INTRODUCTION

Consider a complicated road and the driverless car is lost as it struggles to localize itself in the map provided. If it was a chauffeur-driven car instead, one could give simple instructions like *"take right from the traffic signal followed by second left"* or *"take first right at the crossroad"* and manage the situation. And if we could communicate with the car using natural language instructions, many such situations can be resolved with ease and it would be a giant step forward towards seamless integration of autonomous vehicles alongside humans.

It will also make autonomous navigation more efficient. For instance, traditional approaches to navigation have always relied on offline metric maps [23] generated from runs apriori. Map based approach is consistent across a variety of applications such as in Collaborative Indoor Service Robots or CoBots [4], MOBOTS used in Museums [18] and all the more so in outdoor autonomous driving scenarios, where companies have been spending extensively on getting detailed and up to date maps [7]. However, it can be argued that human navigation uses maps minimally and yet is able to reach locations without much ado. For instance, given the instruction *"keep following until you reach the end of the road and then take a right"* the vehicle can navigate without the need of precise localization on the metric map every instant. Hence, a natural language based instruction can augment the capability of the vehicle to work seamlessly even when the localization is erroneous (due to GPS lag or errors) or the precise metric maps are not available.

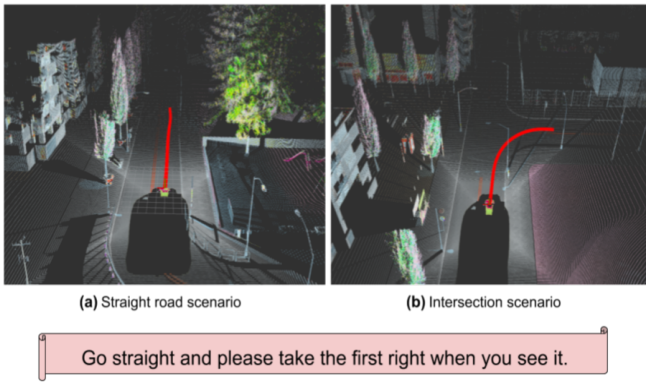

Fig. 1. **Language based Waypoint Generation**: The above instruction is given as input to the NLE network. (a) and (b) depict the output of WGN with a trajectory towards the predicted waypoint. (a) shows a straight road without turns where the network understands the scene and gives a straight waypoint. On the contrary, (b) shows an intersection where the network predicted a right waypoint for the same encoding from NLE network.

A fair amount of this stems from our ability to couple language with immediate semantic understanding to reach destinations accurately. For example humans can seamlessly interpret destination commands of the form, *"Take the second left from here and a right at the third traffic signal and your destination would be on the right"* and execute it by integrating language encodings with the local semantic structure. In this paper we propose a framework that captures this intuition of navigation that is driven/actuated by local semantics and language. While the semantics helps in decomposing the perceptual input into meaningful local representations, language helps in determining the intermediate and eventual goal over these semantic representations.

The framework consists of three modules: the first module is a natural language encoder (NLE), which takes as input the natural language instructions and translates them into high-level machine-readable codes/encodings. The second module is a Waypoint Generator Network (WGN), which combines the local semantic structure with the language encodings to predict the local waypoints (an example is

[1] International Institute of Information Technology, Hyderabad, India.
[2] TCS Research and Innovation Labs, Kolkata, India.
*Equal contribution from the first two authors.

illustrated in Fig. 1). The local semantic structure is defined by the occupancy map (obtained using depth maps) and the semantic segmentation maps (obtained using RGB image) of the current perspective of the car. The third module is a local planner that generates an obstacle avoiding trajectory to reach the locally generated waypoints and executes it by employing a low-level controller.

The entire framework is summarized in Fig. 2 The proposed framework is modular, scalable and works in real time. We believe that our approach works at the right granularity in contrast to the previous works which either avoid the navigation part altogether [26] or directly couple the local structure information with steering control [20], which would require exponentially more training data to reduce noise or to tackle corner cases [22]. The efficacy of the pipeline is verified thorough quantitative and qualitative results using the CARLA simulator environment as well as on local semantic maps built from real-world KITTI dataset. Formally, our work makes the following contributions:

- We propose a novel Language Conditioned Self Driving (LCSD) dataset, which consists of 140 ground truth trajectories from CARLA simulator with the corresponding natural language instructions for the end to end testing of language-based navigation task.
- A novel modular framework that provides accurate waypoints integrating Language Encodings with a Semantic Map. As we show in the experimental section by repeatedly invoking the WGN at the right granularity the agent (car) reaches its destination without the need for a prior map or accurate state estimation modules that depend on such maps. This stands in contrast with the prior art wherein the language command is processed only once and a global trajectory is planned to the eventual destination [12]. However, the subsequent execution of such a trajectory is contingent on access to accurate maps and state of the art localization modules.
- The efficacy of the pipeline is verified in realistic synthetic datasets such as CARLA [9] and real-world KITTI dataset [10]. We also show successful experimental runs on our Self Driving Electric Car.
- We compare with two versions of baseline architectures for WGN. The first version precludes semantics and the second includes semantics but integrates the NLE into a post-processing module. We show that the proposed framework outperforms both the baselines by a significant margin. Most importantly we show ablation studies that portray the robustness of WGN to control noise that translates to significant perturbations in viewing angle, which is now dominantly misaligned with the road direction.

II. Related Work

Previous works [16], [5], [26], [25], [17] have looked at the problem of taking language instructions and converting them into a sequence of machine-readable high-level codes/encodings, where each code defines a different set of actions. These approaches pose the problem analogous to Machine Translation. Earlier approaches [16], [5] employ statistical machine translation, while the recent ones frame it as an end-to-end encoder-decoder neural network architecture [17], [26], [25]. These approaches are limited to known metric maps [23] or topological maps [14] and also assume that robots can realize valid navigation based on the instructions. In our work, we employ a similar sequence to sequence machine translation strategy, however, we also couple it with the actual navigation. Additionally, our work does not rely on a known environment.

Another class of language-based navigation methods [20], [21] directly couple the steering control with the sensory input. The work by [20] builds upon the work by [26], by learning deep learning networks to imitate different navigation behaviors like entering the office, follow corridor etc. A separate convolutional neural network is trained for navigating in each behavior, which significantly increases the complexity of the problem. FollowNet [21] uses attention over language conditioned by the depth and visual inputs, to focus on the relevant part of the instruction at a given time. We take a modular approach and predict local waypoints instead of directly coupling the steering control with the sensory input, which may require an exponentially large amount of training data for precise measurements [22].

Several datasets [1], [8], [6] for language based navigation have been proposed in the recent past. The work by [1] present a Room-to-Room (R2R) dataset for visually-grounded natural language navigation in real buildings, using the Matterport3D Simulator. They pose navigation as a selection problem among several pre-defined discrete choices at each state, which dilutes the actual navigation component. Another recent work [6] proposes a dataset for language based navigation in google street maps, however, their experiments focus only on the visual grounding i.e. to identify the point in the image that is referred to by the description. The work by Chen et al. [8] works in the setting of a tourist guide communication and the focus is on locating the tourist and giving him the right instructions to move towards the target. Our work augments their work, as it focuses on actually navigating the vehicle given the instructions.

Another line of work [11], [12] has looked into identifying goals and constraints (admissible and inadmissible states in a known environment model) from natural language instructions. The work by Howard et al. [11] uses a Distributed Correspondence Graph (DCG) to infer the most likely set of planning constraints. More recent work by Hu et al. [12] uses LSTM networks to classify individual phrases into the goal, constraints, and uninformative phrases. A global plan is then created by grounding the goal in a known environment. Local cost maps are then derived considering the goal and the constraints, which is used to compute a local collision avoidance navigation of the robot. The dependency on accurate sentence/phrase parsing is a major weakness of these approaches.

Interestingly, most of the prior work has tackled the language based navigation problem only in indoor environments, which are either synthetically generated or/and

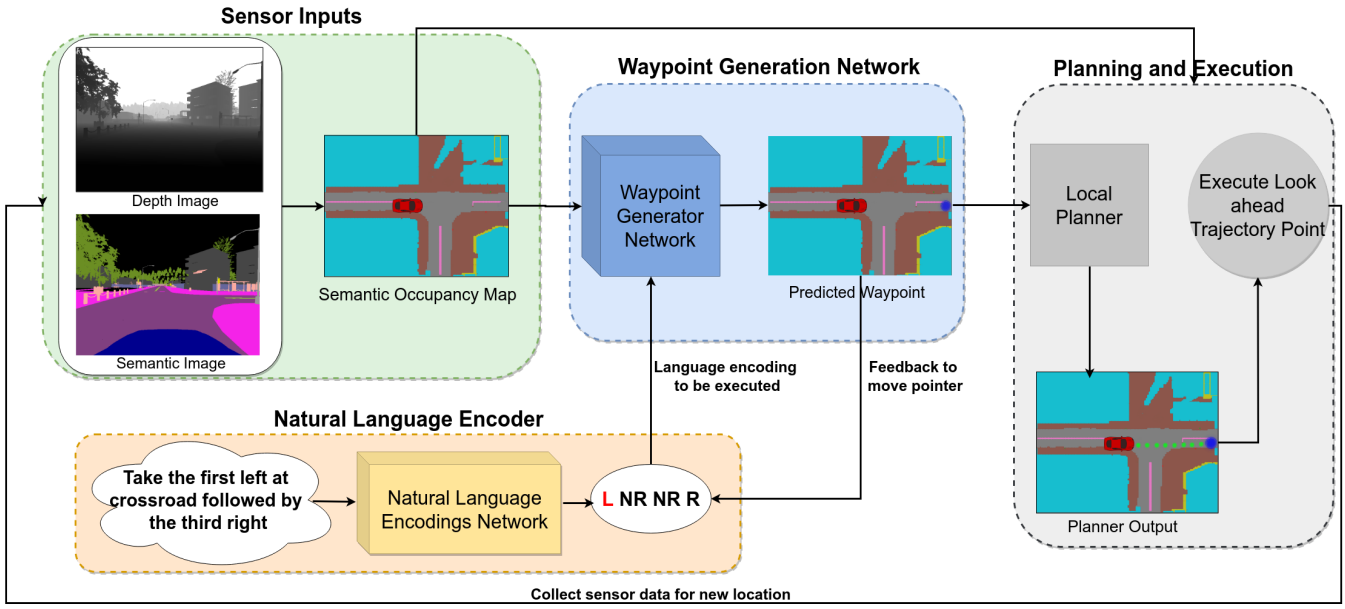

Fig. 2. The overall pipeline of the proposed approach. The figure illustrates a predicted local waypoint (blue dot) by WGN given sensory inputs conditioned by the language encoding. The language encoding currently points to 'L' (in red) i.e take next available left. The WGN predicts a straight waypoint. The planning and execution module, then predicts the trajectory to the local waypoint and executes it (the planned trajectory is shown with Green dots).

known apriori. We solve the problem on a self driving car without prior knowledge about the map/environment. Furthermore, we present results on a CARLA simulator (synthetic environment), real world KITTI dataset, and an actual implementation on our electric car to thoroughly illustrate the efficacy of our approach.

III. Method

The overall pipeline of our approach is illustrated in Fig. 2. The NLE module translates the natural language instructions into the sequence of machine-readable encodings as shown in Table I. For instance, an instruction *"Take the first left at the crossroad followed by the third right"* is translated into {L,NR,NR,R}, i.e. take left, skip a right (not right), skip a right (not right) and then take a right. The WGN module predicts the next local waypoint given the input occupancy map and the currently pointed output of the language encoding. For instance, the pointer will be at 'L' until a left turn is executed and the NLE pointer will move to 'NR' once the turn is complete. Fig. 2 shows WGN predicting a straight waypoint (blue dot) when input from NLE is 'L', as there is no left available. The planning and execution module takes as input the occupancy map and the generated waypoints to predict and execute an obstacle avoiding trajectory towards the waypoint. The trajectory is planned using a RRT* algorithm [13]. When a small arc-length of the trajectory is executed, a new waypoint is predicted and trajectories are replanned. We now provide a detailed description of the NLE and WGN modules.

A. Natural Language Encoder

The NLE module predicts a sequence of encodings given the natural language instructions. The sequence of encodings steers the high-level behavior of the autonomous vehicle. We pose this problem as a machine translation problem and train a LSTM network with attention mechanism [24], [15], [2] to solve it.

We employ an encoder-decoder architecture similar to [24] using a two-layer LSTM with local attention. The input embeddings are of size 500 and the number of hidden units in each layer is set to 1024. The target vocabulary size is 8, corresponding to the eight behaviors considered in our work (illustrated in Table I). We use the framework by Rico et al. [19] for training the NLE module.

We train our network using a dataset of 20,000 natural language instructions and their corresponding encodings (15000 sequences were used for training, 2500 for validation and 2500 for testing). The natural language instruction varies from the length of 2 words to 50 words and corresponding ground truth encoding sequences vary from the length of 1 to 15. To account for the high variability in the way people describe routes, our dataset contains a variety of explanations for the same sequence of behavioral encodings. For example, the instructions *"skip the first right and take the next one"*, *"you should take the second right, not the first one"* or *"keep moving straight on the crossroads and then take a right"* all correspond to sequence {NR, R}.

Given the current output vocabulary size, our NLE module achieves near perfect test accuracy, which suggests studying the NLE module with richer output vocabulary. Such studies focused on the Natural Language Processing component, have been explored in previous works [25] and indicates that the machine to machine translation frameworks are conveniently scalable to larger vocabularies in predicting indoor behavioral graphs. Some example predictions of our NLE module are illustrated in Table II.

TABLE I

EIGHT HIGH LEVEL BEHAVIOURS INCLUDED IN OUR WORK.

Encodings	Meaning
L	Take the first left
R	Take the first right
NL	Not left or go straight when left is available
NR	Not right or go straight when right is available
TL	Take the first left at the traffic signal
TR	Take the first right at the traffic signal
TNL	Not left at the traffic signal
TNR	Not right at the traffic signal

TABLE II

SOME EXAMPLES OF NATURAL LANGUAGE INSTRUCTIONS WITH THE PREDICTED ENCODINGS.

Instruction	Encoding
Follow the road until the junction and take right followed by a left in the next crossroad and keep going straight till you see the traffic signal and take a left at the signal.	R L TL
Please take the second left at the crossroad and then take right at the next intersection followed by a left.	NL L R L
Skip the first right and take the next one post that, take a right at the traffic signal.	NR R TR

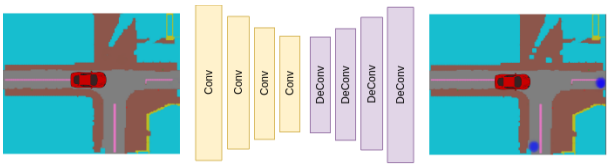

Fig. 3. **WGN Encoder-Decoder style**: Shows the network architecture of WGN E-D style where the network takes semantic occupancy map as input and predicts all possible waypoints as output.

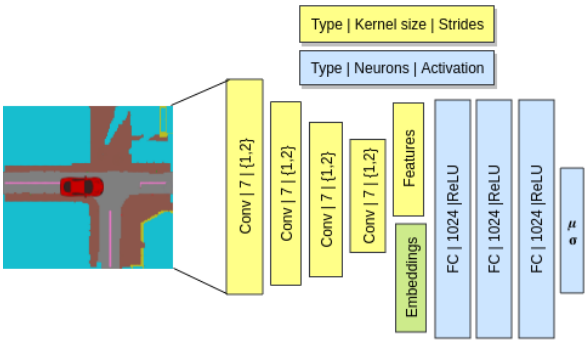

Fig. 4. **WGN CNN-Dense style**: The network takes a semantic occupancy map and a language encoding as input and predicts the parameters of a Gaussian distribution μ, σ, where μ gives the x, y location of the predicted waypoint.

B. Waypoint Generation Network

Waypoint Generation network is the second module in our framework which predicts a probable goal point that can be used by an autonomous vehicle to traverse towards its intended direction. We propose two variants of our Waypoint Generation Network, the first approach is based on an Encoder Decoder (E-D) architecture without language encoding as input where the network learns to forecast all plausible goal locations that the vehicle can take. We use this method as a baseline (Baseline 2) in our approach. We also train the same network without semantics and use it as a Baseline 1. The second variant is our proposed CNN+Dense (CNN-D) architecture where the prediction is conditioned on the language encoding and predicts only a single waypoint.

1) Inputs: The two variants of the Waypoint Generation Network have a semantically labeled occupancy map as input. The network takes in this semantic map $\mathcal{O}^9$ of 9 channels where, each $\mathcal{O}^i$ represents a binary activation map of a particular semantic label (road, buildings etc.). First, a 3D semantically labelled point cloud is generated and is projected on a 2D-grid in birds-eye view form to get a stack of mutually exclusive binary masks. Here we make use of 9 different semantics including unlabelled regions, that are common in an autonomous driving setting such as, Buildings, Road, Lanes, Vegetation, Vehicles, Traffic Lights, Poles and Pedestrians. Therefore, the input to the baseline WGN E-D style is given by $\mathcal{I}_{ED} = \{\mathcal{O}\}$.

The second variant of Waypoint Generation Network with CNN-Dense (WGN CNN-D) architecture takes an additional language prior $\mathcal{Q}$ as input that comes from the first module of our pipeline, which are the encodings given by the Language Network. Effectively, the input to the WGN CNN-D is $\mathcal{I}_{CD} = \{\mathcal{O}, \mathcal{Q}\}$. $\mathcal{I}$[1] has ego-centric vehicle information in both these cases.

2) Training Data: In order to obtain the ground truth in our framework, we query for waypoints in CARLA that are at a particular distance within the vicinity of our semantic map. These waypoints $\mathcal{W}$ are used as our ground truth locations for the network to learn. To this end, for a network input $\mathcal{I}_t \in \{\mathcal{I}_{train}\}$, we pick a waypoint $\mathcal{W}_t \in \{\mathcal{W}_i\}$ which represents various possible destination locations in our vicinity and i various from 1 to N based on number of points.

C. WGN Encoder-Decoder Style

We use a network architecture as shown in Fig. 3. The logits of the network have a sigmoidal activation. The network is trained in a supervised fashion by generating pairs of $< \mathcal{I}_t, \mathcal{W}_t >$ and by picking a ground-truth $\mathcal{W}_t$ from a pool of possible destination locations in our vicinity. The network outputs a 2 channel map where each one represents the corresponding pixel is classified as a goal point or not. The network is trained with a weighted cross entropy loss $\mathcal{L}_{CE}$ as given below.

$$\mathcal{L}_{CE} = \sum_{k=1}^{2} \alpha^k \mathcal{W}(-\log(z^k)) + (1-\mathcal{W})(-\log(1-z^k)) \quad (1)$$

where z is the output of sigmoidal activation from logits of the network. α represents the weight of each class.

[1]For simplicity we use $\mathcal{I}$ as the input to the network in both variants

D. WGN CNN-Dense Style

1) Architecture: The structure of the neural network is shown in the Fig 4. The network consists of two residual building blocks such as CNN and Fully connected layers. We use a CNN based architecture that encodes the given semantic map to a feature vector which is concatenated with the encodings given by the language network. This input is fed to the next block of our Waypoint Generation Network where it passes through 3 consecutive dense layers of 1024 neurons which finally gives out a mean μ and variance σ.

2) Training: In this work we utilize the method similar to Mixture Density Networks [3] to predict a Gaussian distribution with a mean μ and a variance σ. The Waypoint Generation Network maps the input $\mathcal{I}$ to a Gaussian distribution which provides the probability density of the output $\mathcal{W}$ that is conditioned on the input features and the network parameters Θ, $P(\mathcal{W}|\mathcal{I},\Theta)$ where, the probability density is given as,

$$P\big(\mathcal{W}|\mathcal{I},\Theta\big) = \frac{1}{(2\pi)^{c/2}\sigma(\mathcal{I})^c}\exp\Big\{-\frac{\|\mathcal{W}-\mu(\mathcal{I})\|^2}{2\sigma^2(\mathcal{I})}\Big\} \quad (2)$$

Here, c is the output dimension of the vector μ. In this case, c has a value of 2 with $\mu = \{x, y\}$ where x, y specifies the predicted waypoint location in the semantic map. The parameters μ and σ of the distribution can be obtained from the network as,

$$\mu(\mathcal{I}) = z^{(\mu)}(\mathcal{I}) \quad (3)$$

$$\sigma(\mathcal{I}) = ELU(z^{(\sigma)}(\mathcal{I})) \quad (4)$$

where $z^{(\mu)}$ and $z^{(\sigma)}$ are the output activations of the Waypoint Generation Network. In training, we minimize the negative log-likelihood of the WGN given the input $\mathcal{I}_{train}$.

$$\Theta = \arg\min_{\Theta}\Big(-\log P(\mathcal{W}|\mathcal{I}_{train},\Theta)\Big) \quad (5)$$

IV. Results

We evaluate the proposed method in both real world and simulated environments. Quantitative results are shown on CARLA and KITTI dataset followed by qualitative analysis on both along with the result of the test run on an electric car.

A. Experimental Setup

a) **CARLA:** We train our WGN networks by collecting data through CARLA simulator. For the task, we mount camera sensors that provides us RGB, depth and semantic segmentation of the current scene. The semantic maps are then derived using the depth and semantic information. We then capture the sensor data at 4000 random locations in Town1 to train our WGN network. In order to test our framework, we randomly pick start points and goal destinations and compute their corresponding language encodings for the path predicted by CARLA. We collect 140 such trajectories (70 from Town1 and 70 from Town2) with their ground truth encodings which are then annotated with natural language instructions. This dataset is then used for the end to end evaluation of our approach.

TABLE III

TABLE SHOWS THE NATURE OF THE DATA (CARLA) IN TERMS OF THE ENCODING AND DISTANCE TRAVELLED BY THE CAR.

No. of turns	Avg. encoding length	Avg. Distance (m)
1	1.49	485.13
2	3.24	850.49
3	4.48	1140.82

TABLE IV

DIFFERENT SEQUENCE LENGTH COMPARISON IN TOWN1 OF CARLA

No. of turns	Baseline1	Baseline2	Our Model	Total seq.
1	30	37	**40**	40
2	7	18	**20**	20
3	0	7	**8**	10

b) **KITTI:** The WGN module was trained on CARLA and was fine-tuned on KITTI using sequences number 05, 06 and 07 by manually annotating waypoints. The testing was done on sequences 00, 01, 02 and 03. In order to evaluate the results on KITTI with language instructions we first generate a database of registered points with semantics. At any given pose, we query for the scene from its perspective thus, getting the 3D semantically labeled point cloud which is further projected down in the form of a semantic occupancy map. We then label each of the KITTI sequences with the corresponding natural language instructions. The goal of our framework is now to predict waypoints conditioned on the language instructions and imitate the ground truth trajectories taken by car. This testing is extremely rigorous as the setting is not as structured as CARLA and contains real-world noise in estimating the occupancy maps. Furthermore, at any given pose we know only partial data, which was seen from the KITTI cars perspective in the original run.

c) **Electric Car:** We show the ability of the network to follow language instructions by deploying the network in outdoor scenarios on an electric car. The outdoor tests were conducted in constrained passageways and on roads of our campus. In order to test on our University roads, a Mahindra e2o was mounted with Velodyne-16 and Xsens MTi-30 IMU.

TABLE V

DIFFERENT SEQUENCE LENGTH COMPARISON IN TOWN2 OF CARLA

No. of turns	Baseline1	Baseline2	Our Model	Total seq.
1	27	35	**40**	40
2	5	17	**19**	20
3	0	**8**	**8**	10

TABLE VI

PERTURBATION ANALYSIS ON KITTI DATASET

Degree	0	10	20	30	40	50	60
Accuracy	100	95.8	93.3	88.3	85	79.1	79.1

B. Quantitative Results

We compare our method with Baseline1 and Baseline2 which has been explained in the methods section. Table III presents some details about the LCSD dataset. It shows the average encoding length corresponding to the number of turns taken by the car. For instance, for instruction with two turns *"take second right and then third left"* the encoding length is 5 {NR, R, NL, NL, L}. It also gives the statistics on the average distance the car had to cover.

Table IV and Table V shows the comparison between Baseline1, Baseline2 and our proposed model on LCSD dataset, on the basis of the number of turns (indicating in some form the increasing complexity of the task). The last column shows the total number of episodes we run for sequences of different complexities. By an episode we mean a run of complete trajectory once. We consider the instruction to be successfully completed if the ego car passes through all the ground truth waypoints globally while reaching its destination. Our approach outperforms both the baselines by a significant margin. The significant improvement in Baseline2 over Baseline1 suggests a clear benefit of augmenting semantic information in the waypoint prediction task. The direct conditioning of language into the waypoint prediction network brings further improvements, compared to a post-processing selection mechanism as used in Baseline2.

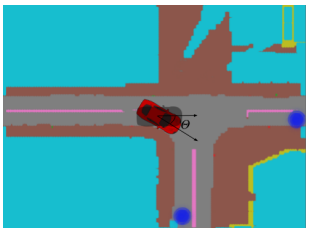

Fig. 5. **Perturbation:** the orientation of the car is pertubed by an angle θ and the occupancy grid so formed detects the same set of waypoints.

To further verify the robustness of WGN model we do a perturbation analysis on KITTI maps where we portray control noise as a perturbation in viewpoints of the car. For e.g. $30°$ perturbation implies that the car is offset from road direction by $30°$. Fig. 5 demonstrates perturbation and how the predicted waypoints are robust to the car's relative alignment to the road. The perturbation experiment on KITTI dataset also presents the ability of the WGN network to predict waypoints in partial data availability (as the sensor data is only known from KITTI's perspective in the original runs). The results in Table VI shows that the WGN is able to correctly predict correct waypoints with 80% accuracy, even with perturbations as large as $60°$.

C. Qualitative Results

a) CARLA: In Fig. 6 we show a metric map and overlay the waypoints executed by the car, corresponding to the instruction *"please take the second right followed by a left"* in CARLA simulator. The figure depicts that the network has an implicit understanding of the scene and predicts the right

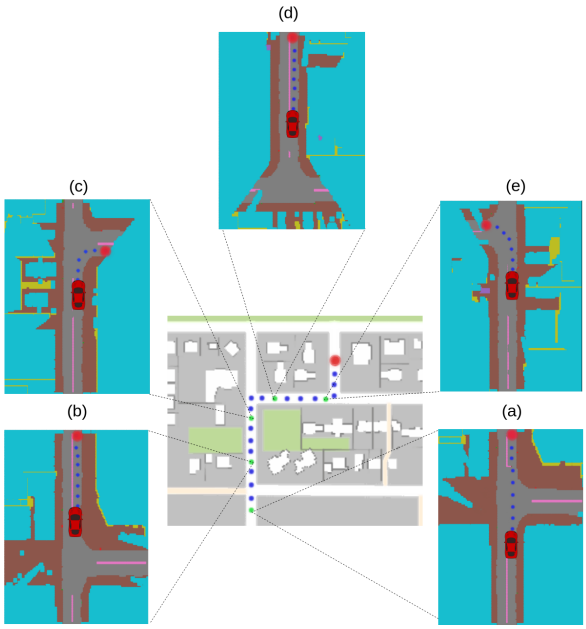

Fig. 6. Demonstrates the execution of language instruction in CARLA environment. The central picture shows a metric map with the waypoints that the car took (blue dot). Few sample locations where predictions were recomputed are shown with green dots along with the magnified view of the corresponding scene with its waypoint predicted by the network in red. The dotted blue lines in the semantic map shows the path towards the predicted goal. The corresponding NLE instruction executed for semantic maps a-e are $NR - R - R - L - L$ respectively.

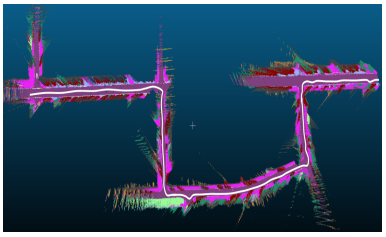

Fig. 7. White line shows the trajectory executed by updating the pose of ego-vehicle based on the path towards the predicted waypoint at every frame of KITTI in sequence 00

waypoints even when there are multiple dominant choices of destination locations that can be reached. For instance in Fig 6a the network was given encoding to not take a right (NR) and hence chose a straight waypoint. While in Fig 6b the encoding was to take a right (R) but the input scene did not have a right possible hence the network learned to predict a waypoint in straight. For the same encoding in 6c the network saw a possible right and gave a waypoint in that direction. The WGN network is re-run after executing a trajectory of fixed length. Such an in-loop at fine granularity is needed for successful navigation.

b) KITTI: In order to show results on KITTI dataset, as mentioned above we query for the point clouds from the current perspective from a database of registered points with semantics. The obtained semantic maps from these point clouds are given as input to the network along the direction of motion of these ego vehicles by annotating it

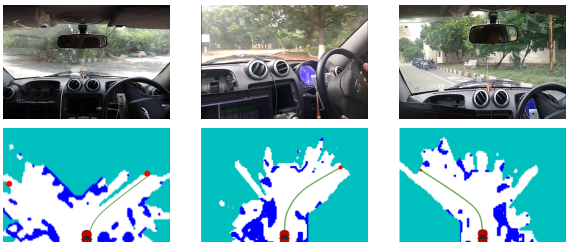

Fig. 8. **Top:** Shows the image of a real car at an intersection. **Bottom:** Red dot shows the waypoints of interest predicted by the network. The trajectory towards the predict goal points is shown using a green line.

with corresponding language sequence. For every iteration, a path to the waypoint predicted by the network is computed and then the current pose is updated based on the location and yaw of the predicted path as we move a distance of fixed length in the predicted trajectory. The result for one such run with an input instruction *"take the first right and then left at the next crossroad followed by another left at the next crossroad and then right at the next intersection"* is shown in Fig. 7 which demonstrates the successful completion of the instruction.

c) Electric Car: Fig. 8 show snapshots of the traversal of the e2o. We use Baseline 1 for the experiments since the scene semantics are yet to be annotated. Fig. 8 Top: shows the scene seen from the point of view of the car, while Fig. 8 Bottom Left: shows the output of the WGN with two waypoints. The waypoint on the right was chosen as the NLE instructed a right turn. Fig. 8 Bottom Center and Fig. 8 Bottom Right depict situations similar to Fig. 8 Bottom Left except that here the waypoint on the right and left was selected according to the NLE command. The overall language input to the car for the output shown was *"Take a right near the intersection and an immediate right after that and then take a left"*.

V. Conclusion

This paper proposes a novel pipeline that allows Natural language commands to be translated into suitable actuation on the car. The pipeline was tested on synthetic and real-world datasets with trajectories reaching their eventual destination more than 90% of the time even for NLE with longer sequence lengths. The paper established an end to end WGN that integrates both NLE and scene semantics as inputs which performs better than baselines that either does not integrate scene semantics or incorporate both in a decoupled fashion, which is not end to end. Successful real-world experiments on the Mahindra e2o further confirm the efficacy of the proposed pipeline. Future scope of the work is to include a more rich and diverse set of natural language commands that is replete with semantics and extensive validation of the method's ability to work without maps and state estimation modules on the electric vehicle.

References

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] C. M. Bishop. Mixture density networks. Technical report, Citeseer, 1994.

[4] J. Biswas and M. M. Veloso. Localization and navigation of the cobots over long-term deployments. *The International Journal of Robotics Research*, 32(14):1679–1694, 2013.

[5] D. L. Chen and R. J. Mooney. Learning to interpret natural language navigation instructions from observations. In *AAAI*, 2011.

[6] H. Chen, A. Shur, D. Misra, N. Snavely, and Y. Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. *arXiv preprint arXiv:1811.12354*, 2018.

[7] O. Dabeer, W. Ding, R. Gowaiker, S. K. Grzechnik, M. J. Lakshman, S. Lee, G. Reitmayr, A. Sharma, K. Somasundaram, R. T. Sukhavasi, et al. An end-to-end system for crowdsourced 3d maps for autonomous vehicles: The mapping component. In *IROS*, pages 634–641. IEEE, 2017.

[8] H. de Vries, K. Shuster, D. Batra, D. Parikh, J. Weston, and D. Kiela. Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367*, 2018.

[9] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[11] T. M. Howard, S. Tellex, and N. Roy. A natural language planner interface for mobile manipulators. In *ICRA*. IEEE, 2014.

[12] Z. Hu, J. Pan, T. Fan, R. Yang, and D. Manocha. Safe navigation with human instructions in complex scenes. *IEEE Robotics and Automation Letters*, 2019.

[13] J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. volume 2, pages 995–1001, 01 2000.

[14] B. Kuipers. Modeling spatial knowledge. *Cognitive science*, 2(2):129–153, 1978.

[15] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[16] C. Matuszek, D. Fox, and K. Koscher. Following directions using statistical machine translation. In *HRI*, pages 251–258. IEEE, 2010.

[17] H. Mei, M. Bansal, and M. R. Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *AAAI*, 2016.

[18] I. R. Nourbakhsh, C. Kunz, and T. Willeke. The mobot museum robot installations: A five year experiment. In *IROS*. IEEE, 2003.

[19] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Lubli, A. V. Miceli Barone, J. Mokry, and et al. Nematus: a toolkit for neural machine translation. *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

[20] G. Sepulveda, J. C. Niebles, and A. Soto. A deep learning based behavioral approach to indoor autonomous navigation. In *ICRA*, pages 4646–4653. IEEE, 2018.

[21] P. Shah, M. Fiser, A. Faust, J. C. Kew, and D. Hakkani-Tur. Follownet: Robot navigation by following natural language directions with deep reinforcement learning. *arXiv preprint arXiv:1805.06150*, 2018.

[22] S. Shalev-Shwartz and A. Shashua. On the sample complexity of end-to-end training vs. semantic abstraction training. *arXiv preprint arXiv:1604.06915*, 2016.

[23] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

[24] Z. Yang, Z. Hu, Y. Deng, C. Dyer, and A. Smola. Neural machine translation with recurrent attention modeling. *arXiv preprint arXiv:1607.05108*, 2016.

[25] X. Zang, A. Pokle, M. Vázquez, K. Chen, J. C. Niebles, A. Soto, and S. Savarese. Translating navigation instructions in natural language to a high-level plan for behavioral robot navigation. *EMNLP*, 2018.

[26] X. Zang, M. Vázquez, J. C. Niebles, A. Soto, and S. Savarese. Behavioral indoor navigation with natural language directions. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 283–284. ACM, 2018.