

Ground then Navigate: Language-guided Navigation in Dynamic Scenes

Kanishk Jain*, Varun Chhangani*, Amogh Tiwari, K. Madhava Krishna and Vineet Gandhi

Abstract— We investigate the Vision-and-Language Navigation (VLN) problem in the context of autonomous driving in outdoor settings. We solve the problem by explicitly grounding the navigable regions corresponding to the textual command. At each timestamp, the model predicts a segmentation mask corresponding to the intermediate or the final navigable region. Our work contrasts with existing efforts in VLN, which pose this task as a node selection problem, given a discrete connected graph corresponding to the environment. We do not assume the availability of such a discretised map. Our work moves towards continuity in action space, provides interpretability through visual feedback and allows VLN on commands requiring finer manoeuvres like "park between the two cars". Furthermore, we propose a novel meta-dataset CARLA-NAV to allow efficient training and validation. The dataset comprises pre-recorded training sequences and a live environment for validation and testing. We provide extensive qualitative and quantitative empirical results to validate the efficacy of the proposed approach. Code is available at https://github.com/kanji95/carla_nav.

I. INTRODUCTION

Humans have exceptional navigational abilities, which, combined with their visual and linguistic prowess, allow them to perform navigation based on the linguistic description of the objects of interest in the environment. This is in direct consequence of the human capability to associate visual elements with their linguistic descriptions. Referring Expression Comprehension (REC) [1] and Referring Image Segmentation (RIS) [2] are two tasks for associating the visual objects based on their linguistic descriptions using bounding-box-based and pixel-based localizations, respectively. However, it is non-trivial to utilize these localizations directly for a navigation task [3]. For example, consider the linguistic command "take a right turn from the intersection," an object-based localization is not usable for navigation as it does not answer the question "which region" on the road to navigate to. To solve the aforementioned issue, the task of Referring Navigable Regions (RNR) was proposed in [4] to localize the navigable regions in a static front camera image on the road corresponding to the linguistic command.

Although these single image-based visual grounding methods [4], [2], [3] showcase excellent ability of neural models to correlate visual and linguistic data, they are limited in many ways. These methods are trained on carefully paired data, assuming that the region to be grounded is always visible in the frame. They give an erroneous output when the region to be grounded is not currently visible, is occluded, or goes out



Fig. 1. A major limitation of single image based grounding methods is that they fail if the language command is not immediately visible, which restricts these methods to be used for VLN. Here, we show an example result using the RNR model and on an image from their dataset [4]. The model accurately localizes the black car (middle), however, it completely confuses when asked to predict the left turn, which does not exist in the current view.

of frame (as the carrier moves). Such scenarios are part of everyday language-guided navigation; for instance, consider a command "Take a right once you see the traffic signal" the traffic signal here may not be immediately visible. Figure 1 illustrates an example, where the single image-based RNR method gives incorrect output, uncorrelated with the linguistic command. As a second major limitation, single image-based predictions [4] are devoid of any temporal context (short term or long term), which is crucial in successful navigation, especially in a dynamically changing environment. Finally, since single image methods are evaluated on frames from pre-recorded videos, they cannot be validated appropriately on their ability to complete the entire episode (from start to desired finish). Our work addresses these limitations and re-formulates the RNR approach to perform language-guided navigation in a dynamically changing environment by grounding intermediate navigable regions when the referred navigable region is not visible.

Our work also contrasts with the prior art for language-guided navigation in both indoor and outdoor environments. Most existing works on indoor navigation [5], [6], [7] assume that the navigational environment is fully known. This allows them to discretize the known map into a graphical representation, where the nodes are the set of navigable regions (landmarks) that the agent can navigate given the linguistic command. However, such an approach is not practical for outdoor settings (as studied in our work) where the environment is unknown. Moreover, even if the environment is known, discretization of the maps is not feasible when more refined localization and manoeuvres are required (e.g. "stop beside the person with a red cap").

Finer control remains a challenge in indoor and outdoor VLN methods, which model navigation as a selection from a set of discrete actions [8], [9], [10] or as a reinforcement learn-

*Equal Contribution

The authors are with Kohli Center on Intelligent Systems, International Institute of Information Technology, Hyderabad, India, 500032. kanishk.j@research.iiit.ac.in

The work was supported in part by a Qualcomm Innovation Fellowship

ing problem [5], [11]. For instance, one of the commonly used Touchdown dataset [12] consists of pre-recorded google street view images and allows navigation across street views by choosing from a set of four discrete actions, i.e. FORWARD, RIGHT, LEFT and STOP. Discretizing the action space (and the environment) limits the type of navigational manoeuvres that can be performed. For instance, these methods [8], [9], [10] cannot be used for commands like "park between the two cars on the right", requiring fine-grained control.

The aforementioned issues become apparent in a dynamically changing environment, where fine-grained control of the car's navigation and a fully navigable environment is required to adapt to the dynamic surroundings and perform navigational manoeuvres based on the linguistic command. In this paper, we present a novel meta-dataset in the CARLA environment [13] for outdoor navigation, which addresses the limitations associated with the existing navigation datasets. Additionally, the visual grounding-based approach combined with a planner allows us to have a fine-grained control over the vehicle as it enables navigation to any drivable region on the road.

Another concern with the previous methods [8], [9], [5] is that their predictions are not human interpretable. It is non-trivial to understand their predictions as there is no feedback. Instead, in the visual-grounding-based approach, there is visual feedback associated with each prediction in terms of the segmentation mask corresponding to the navigable region on the road. We take a step forward and also predict the short term intermediate trajectories, using a novel multi-task network. Moreover, we perform live inference on the proposed meta-dataset in a dynamic environment. To the best of our knowledge, this is the first attempt towards live language-based navigation in outdoor environments. Overall, our paper makes following contributions:

- We present a vision language navigation tool, CARLA-NAV, on the CARLA simulator which provides fine-tuned control of the vehicle to execute various language-based navigational manoeuvres.
- We propose a novel multi-task network for trajectory prediction and per-frame RNR tasks in dynamic outdoor environments. The prediction for each task is explainable and interpretable in the form of segmentation masks.
- We perform real-time navigation in the CARLA environment with a diverse set of linguistic commands.
- Finally, extensive qualitative and quantitative ablations are performed to validate the practicality of our approach.

II. RELATED WORK

A. Visual Grounding

Visual grounding aims to help associate the linguistic description of entities with their visual counterparts by localizing them visually. There are two prevalent approaches for visual grounding: (a) proposal based and (b) segmentation based. Proposal-based grounding is formally referred to as Referring Expression Comprehension (REC). Most methods in REC follow a propose-then-rank strategy, where the

ranking is done using similarity scores [1], [14], [15], [16] or through attention-based methods [17], [18], [19], [20]. The other approach is to localize the objects by their pixel-level segmentation mask, formally known as Referring Image Segmentation (RIS). In RIS, methods use different strategies to fuse the spatial information of the image with the word-level information of the language query [21], [22], [23], [24]. Recently, [4] proposed the Referring Navigable Region (RNR) task to directly localize the navigable regions on the road corresponding to the language commands. However, their work is limited to predictions on static images in pre-recorded video sequences [25]. We propose reformulating the RNR task for dynamic outdoor settings and performing real-time navigation based on language commands.

B. Language-based Navigation

Majority of efforts on Vision Language Navigation (VLN) have focused on the indoor scenario. Availability of interactive synthetic environments has played a key role in indoor navigation research. The environments are either designed manually by 3D artists [26], [27] or are constructed using RGB-D scans of actual buildings [28]. Existing methods have approached language guided navigation in variety of ways, including imitation learning [29], [30], behavior cloning [31], sequence-to-sequence translation [5] and cross-modal attention [32]. In these methods, the navigation is modelled as traversing an undirected graph, presuming known environment topologies. In recent work, [33] suggest that the performance in prior 'navigation-graph' settings may be inflated by strong implicit assumptions. Hu *et al.* [34] questions the role of visual grounding itself by highlighting that models which only use route structure outperform their visual counterparts in unseen new environments. Most indoor VLN methods are also hindered by limiting the output to a discretized action space [35].

For outdoor VLN, Sriram *et al.* [36] use the CARLA environment to perform navigation as waypoint selection problem, however, their work limits to only turning actions. The Talk2Car dataset is limited to localizing the referred object [3]. Another line of work focuses on interactive navigation environment of Google Street View [37]. The Touchdown dataset [12] proposes a task of following instructions to reach a goal (identifying a hidden teddy bear). Map2Seq dataset [38] learns to generate navigation instructions that contain visible and salient landmarks from human natural language instructions. The navigation on both datasets is modelled as node selection in a discrete connectivity graph. Most methods [8], [9], [10] using these datasets, solve outdoor VLN as sequence to sequence translation in a discrete action space. The role of vision modality remains illusive when tested in unseen areas [8]. In this work, we propose a paradigm shift towards utilizing RNR-based approaches for VLN. The explicit visual grounding forces the network to utilize visual information. Integrating with a local planner, the navigation is performed in a continuous space, without any reliance on the map information.

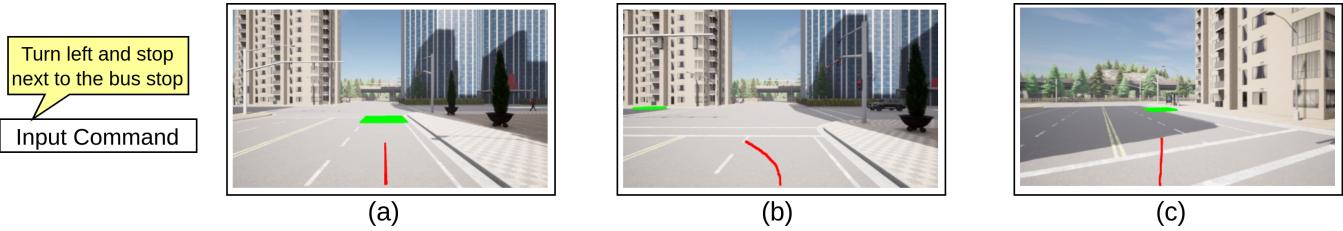


Fig. 2. Ground truth annotations of three sampled frames from an episode of the CARLA-NAV dataset. The textual command for the episode is shown on the top. The green rectangle illustrates the navigable region, and the red curve corresponds to the short future trajectory. (a) At the start, the left turn or the final navigable region is not visible, so a straight path is chosen as the intermediate mask; (b) the intermediate mask corresponding to the left turn; (c) the final navigable region (stop next to bus stop).

III. DATASET

The proposed CARLA-NAV dataset was curated using the open-source CARLA Simulator for autonomous driving research. It contains episodic level data, where each episode consists of a language command and the corresponding video from CARLA Simulator of navigation towards the final goal region described by the command. Example ground truth annotations from an episode from the CARLA-NAV dataset are shown in Figure 2. The ground truth segmentation mask for each frame either corresponds to the final or an intermediate navigable region. Each frame is additionally annotated with a plausible future trajectory of the vehicle in the next few frames.

The dataset includes video sequences captured in 8 different maps, 14 distinct weather conditions, and a diverse range of dynamically moving vehicles and passengers in the environment. The language commands in our dataset contain detailed visual descriptions of the environment and describe a wide range of manoeuvres. Commands can either have a single manoeuvre (e.g. park behind the black car on the left) or multiple manoeuvres (e.g. take a right turn and park near the bus stand). A maximum of three manoeuvres are included in the dataset. The dataset statistics for the CARLA-NAV dataset are illustrated in Table I. Since, each episode contains multiple frames, the overall dataset size is 83,297 frames for all the splits. During data collection, in each episode, the vehicle is spawned in a randomly selected map at a random position. During the training phase, we use the pre-recorded sequence for the network training. However, during the inference phase on validation and test splits, for each episode, we spawn the vehicle at the corresponding starting location, and the navigation is performed live based on network prediction and not on the pre-recorded sequences.

A. Dataset Creation

We created a data-collection toolkit on top of Carla's API and plan to open-source it upon acceptance. The data collection process involves an observer, a navigator and a verifier. The annotation happens in a three step manual process: (a) observer: providing a language command, (b) navigator: navigating the Carla environment through mouse clicks corresponding to navigable regions and (c) verifier: verifying the recorded episode for inclusion in dataset, if it correctly maneuvers corresponding to the given command.

split	# episodes	# frames	command length (words)	clicks
train	500	75,010	6.92	1.94
val	25	3,300	6.76	2.00
test	34	4,987	7.44	1.97

TABLE I
DATASET STATISTICS FOR THE CARLA-NAV DATASET. EXCEPT "# EPISODES" AND "# FRAMES", AVERAGE VALUES PER EPISODE ARE REPORTED FOR THE OTHER COLUMNS.

The navigator is provided with a "restart" option to restart the navigation in case of erroneous clicks.

The 2D point corresponding to the mouse click in the front view of the car is transformed into the 3D world coordinates using Inverse Projective transform; and this 3D position is passed as input to the local planner to navigate the CARLA environment. We use CARLA's default planner for our case; however owing to the modular design, in future it can be replaced by any state-of-the-art planner.

An episode comprise of multiple mouse clicks, until the final navigable region is not visible in the front view. These intermediate mouse clicks signify the intermediate navigable regions, and the last mouse click depicts the final goal region corresponding to the command. The mouse clicks are converted into segmentation masks by drawing a $3m \times 4m$ rectangle (approx size of a car) in the top view, centered at the mouse click. The rectangle is then projected in the front-camera. Overall, only the language commands and mouse clicks require manual effort, the rest of the annotation process is fully automated.

Furthermore, for allowing actionable intervention we predict and visualize the future short term trajectory of the vehicle. For the trajectory prediction task, we take the 3D position of the vehicle in the successive frames and project the 3D positions to the front-camera image using the projective transformation. We treat trajectory prediction as a dense prediction task. The task is meant for added human interpretability and is not quantitatively evaluated.

IV. PROBLEM STATEMENT

Given an input of video frames $V = \{v_{t-k}, v_{t-k+1}, \dots, v_t\}$, contextual historical trajectory P and a language command $L = \{l_1, l_2, \dots, l_N\}$, where t is the current timestamp, k is

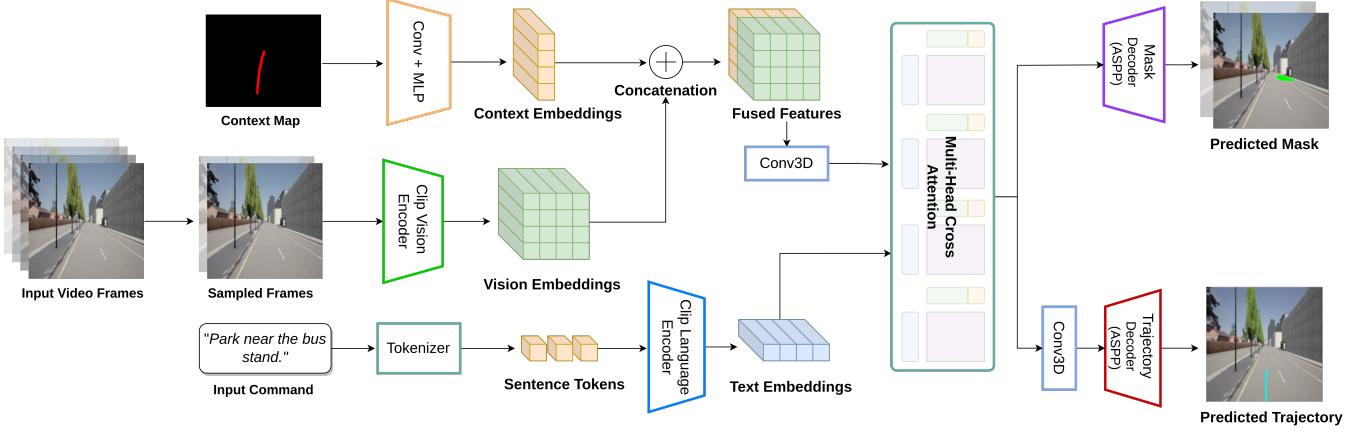


Fig. 3. Overall pipeline of the proposed approach. Given the visual frames, already executed past trajectory (context map) and the textual command, the network predicts a segmentation map corresponding to the navigable region and a plausible future trajectory.

the window size for historical frames and N is the maximum number of words in the linguistic expression, the goal is to predict the navigable mask y_t and the future trajectory mask z_t corresponding to the frame from current timestamp, i.e. v_t . The contextual trajectory P is utilized to ensure that the network gets the contextual information necessary to identify which part of the linguistic command has been executed. For example, if the linguistic command is "turn left and park near the blue dustbin", the contextual trajectory will provide information regarding the trajectory already taken by the vehicle, i.e. whether the "left turn" has been taken or not. The spatial location of the navigation mask should determine the trajectory path's direction; similarly, the orientation of the trajectory path should determine the location of the navigable regions. In the next section we describe the network architecture and the training process.

V. METHODOLOGY

We propose a novel multi-task network for navigation region prediction and future trajectory prediction tasks. Both tasks are treated as dense prediction tasks to make them interpretable for practical scenarios. We convert the dense pixel points to 3D world coordinates using inverse projective transformation during real-time inference. The architecture for our model is illustrated in Figure 3. In this section, we describe the feature extraction process and the architecture in detail.

We utilize CLIP [39] to extract both linguistic and visual features. For the linguistic expression $L = \{l_1, l_2, \dots, l_N\}$, where l_i is the i^{th} word of the expression, we tokenize the linguistic command using CLIP tokenizer and pass it through CLIP architecture to compute word-level feature representation $F^l = \{f_1^l, f_2^l, \dots, f_N^l\}$ of shape $\mathbb{R}^{N \times C_l}$. For visual frames $V = \{v_{t-k}, v_{t-k+1}, \dots, v_t\}$, the CLIP architecture encodes the video features as $F^v = \{f_{t-k}^v, f_{t-k+1}^v, \dots, f_t^v\}$. Finally, for the trajectory context P , we project the past trajectory on an image having same size as the input video frames v_t 's and pass it through convolution and a MLP layer to get feature map F^p with the same feature size as video features f_t^v 's.

The input to our network are the video frames V , historical trajectory context P in bird's eye view and the language command L . Specifically, for video V , we get visual features F^v of shape $\mathbb{R}^{C_v \times T \times HW}$, where H , W , T , and C represent the height, width, time, and channel dimensions, respectively. The trajectory context feature $F^p \in \mathbb{R}^{C_v \times 1 \times HW}$ contains information about the past trajectory taken by the vehicle. Following feature extraction, we concatenate the trajectory context feature F^p with video features F^v along the temporal dimension resulting in joint feature $F^{vp} \in \mathbb{R}^{C_v \times (T+1) \times HW}$ capturing the video and trajectory related contextual information. Finally, we apply multi-head self-attention over the joint contextual feature F^{vp} and linguistic feature F^l in the following manner,

$$\begin{aligned} F &= F^{vp} \odot F^l \\ A &= \text{Mhead}(F, F, F) \\ M &= \text{Conv3D}(A * F) \end{aligned} \quad (1)$$

Here, \odot represents the length-wise concatenation of the word-level linguistic features F^l and the joint feature F^{vp} , Mhead is the multi-head self-attention over the multi-modal features F and $*$ represents the matrix multiplication. Conv3D represents 3D convolution operation and is used to collapse the temporal dimension, M is the final multi-modal contextual feature with information from both visual and linguistic modalities.

Next, we describe the procedure for predicting the navigation and trajectory prediction masks. We want the future trajectory and the navigable region for the current time-step to be correlated with each other, i.e. the future trajectory should point in the direction of the predicted navigable region. Consequently, we utilize the multi-modal contextual feature M to predict the segmentation masks corresponding to the navigation and trajectory prediction tasks. For each task, we have a separate segmentation head, where each segmentation head comprises of sequence of convolution layers with upsampling operation. For training the segmentation masks, we utilize combo loss [40] which is a combination of binary

cross-entropy loss and dice loss:

$$\begin{aligned} L_{bce} &= -(y_t \log(\hat{y}_t) + (1 - y_t) \log(1 - \hat{y}_t)) \\ L_{dice} &= 2 * \frac{\hat{y}_t \cap y_t}{\Sigma \hat{y}_t + \Sigma y_t} \\ L_{combo} &= \lambda L_{bce} - (1 - \lambda) L_{dice} \end{aligned} \quad (2)$$

The proposed approach is end-to-end trainable and the predicted trajectory is highly correlated with the predicted navigation mask, as a result the predicted trajectory is interpretable in the sense that it suggests the future route to be taken by the autonomous vehicle.

VI. EXPERIMENTS

Implementation Details: We utilize CLIP backbone [39] for feature extraction. The frames are selected with a stride of 10 and are resized to 224×224 resolution. After feature extraction, we get per-frame visual features of spatial resolution $H = W = 7$ and channel dimension $C_v = 512$. For the historical contextual trajectory, we plot the trajectory from the starting location of episode to the current timestamp and resize it to 640×480 spatial resolution image, this is passed as input through convolution + MLP layers to obtain trajectory features with same resolution as per-frame visual features. For linguistic features, we use the CLIP tokenizer followed by the CLIP language encoder to compute the word-level features corresponding to the linguistic command. Maximum length of command is set to $N = 20$ and the channel dimension is $C_l = 512$. We use batch size of 32 and our network is trained using AdamW [41] optimizer, the initial learning rate is set to 10^{-4} and polynomial learning rate decay with power of 0.5 is used. For the combo loss, we set $\lambda = 0.3$. All the methods and baselines were trained from scratch using the CARLA-NAV dataset. For the single frame methods [4], individual image and language pairs were used for training (with and without context map).

Live-Navigation: In order to utilize the segmentation mask corresponding to the navigable region directly for navigation, we first need to sample a point from the predicted region. We take the largest connected component from the predicted mask and use its centroid as the target point for the local planner. As we move closer to the final navigable region, the distance between the current car location and the centroid target location consistently decreases. Simultaneously, the area of the predicted mask should increase as we move closer to the target region due to the perspective viewpoint of the front camera. Consequently, we use an area-based threshold to determine if the predicted navigation mask corresponds to the final navigable region or not. If the area of the predicted navigation mask is larger than the threshold for five consecutive times, we treat the predicted region as the final goal region corresponding to the linguistic command and stop the navigation. The threshold was found empirically.

Evaluation Metrics: Like previous approaches to VLN [8], [10], [12], we use the gold standard **Task Completion** metric to measure the success ratio for the navigation task. In addition, we use **Frechet Distance** [42] and **normalized Dynamic Time Warping (nDTW)** [43] metrics to compare

the predicted navigation path during live inference with the ground truth navigation path.

A. Experimental Results

We compare our proposed approach CLIP-MC (multiple frames with contextual trajectory) against the RNR-based approach proposed in [4]. We use their proposed approach with CLIP-based backbone, CLIP-S (single frame without contextual trajectory) as the baseline for our experimental results. The original RNR approach is limited to using a static scene with linguistic commands for navigation, which fails in a dynamically changing environment where the scene can change drastically when we start the navigation. Additionally, we motivate the benefits of contextual trajectory and multiple frames by presenting two variant baselines, (1) multiple frames without a contextual trajectory CLIP-M and (2) single frame with contextual trajectory CLIP-SC. Table II presents the results on the gold standard *Task Completion* metric and Table III presents the results on *Frechet Distance* and *nDTW* metrics.

Method	Task Completion	
	Val	Test
RNR-S	0.44	0.29
RNR-SC	0.52	0.32
CLIP-S	0.48	0.47
CLIP-SC	0.52	0.50
CLIP-M	0.56	0.55
CLIP-MC	0.72	0.68

TABLE II
RESULTS ON THE *Task Completion* METRIC. THE SUPERIOR PERFORMANCE OF PROPOSED APPROACH CLIP-MC, SHOWCASES THE EFFECTIVENESS OF HISTORICAL CONTEXT FOR THE NAVIGATION TASK.

We observe that our proposed approach CLIP-MC outperforms all the other variants. Introducing historical contextual trajectory consistently helps improve performance as it increases by 4% and 16% in cases of single-frame approaches (CLIP-SC, CLIP-S) and multi-frame approaches (CLIP-MC, CLIP-M), respectively on the validation split. Furthermore, the multi-frame approach CLIP-M gives an improvement of 8% on both the validation and test splits, respectively, over the single-frame approach CLIP-S. These results indicate that a combination of multiple frames and contextual trajectory are required to effectively tackle the VLN task.

In Table III, we present experimental results on the *Frechet Distance* and *nDTW* metrics. Our reformulated approach CLIP-MC outperforms all other variants by significant margins. However, we would like to stress that these metrics are not robust indicators of the average performance on the actual navigation task, as a single outlier can drastically affect the final score. For example, on a single instance if "a left turn" is taken instead of "a right turn," the predicted trajectory will diverge from the ground truth trajectory, and will lead to significant deviation in the average scores.

Effect of feature extraction backbone: Additionally, we compare our CLIP-based single frame approaches with the

original non-CLIP RNR approach proposed in [4], referred to as RNR-S and RNR-SC (single frame without and with context, respectively) in Table II. Both RNR-S and RNR-SC are trained from scratch on the proposed CARLA-NAV dataset. The results showcase the advantage of superior multi-modal features captured by the CLIP-based approaches over non-CLIP approaches, as the performance consistently increases on the challenging test split in case of both with context (RNR-SC, CLIP-SC) and without context (RNR-S, CLIP-S).

Method	Frechet Distance ↓		nDTW ↑	
	Val	Test	Val	Test
RNR-S	28.14	42.45	0.35	0.16
RNR-SC	21.64	44.65	0.45	0.33
CLIP-S	40.30	42.53	0.23	0.24
CLIP-SC	35.58	38.49	0.36	0.39
CLIP-M	32.92	53.10	0.39	0.26
CLIP-MC	13.54	15.06	0.54	0.59

TABLE III

EXPERIMENTAL RESULTS ON THE *Frechet Distance* AND *nDTW* METRICS.

↓ INDICATES LOWER VALUE IS BETTER AND ↑ INDICATES THAT THE HIGHER VALUE IS BETTER.

Effect of Number of Frames: In Table IV, we study the impact of the number of video frames on the multi-frame models for the Task Completion metric. As the number of video frames increases, the visual modality's contextual information also increases. We hypothesize that the network should utilize this additional contextual information and employ it effectively for the VLN task. The results in Table IV indeed corroborate our hypothesis, as we observe consistent performance gains as the number of video frames increases. The networks with $n = 1$ frame give the same performance as the corresponding single-frame variants. We obtain the best performance with $n = 8$ frames with CLIP-MC on both validation and test splits.

Method	Split	Task Completion				
		n=1	n=2	n=4	n=6	n=8
CLIP-MC	val	0.52	0.48	0.52	0.68	0.72
	test	0.50	0.53	0.56	0.62	0.68
CLIP-M	val	0.48	0.44	0.48	0.52	0.56
	test	0.47	0.47	0.50	0.53	0.55

TABLE IV

ABLATION ON THE NUMBER OF FRAMES FOR MULTI-FRAME MODELS FOR THE TASK COMPLETION METRIC.

B. Qualitative Results

In Figure 4, we qualitatively compare the proposed approach CLIP-MC with the RNR approach (RNR-S) proposed in [4]. We juxtapose the entire navigation path taken by each approach during live inference for a given linguistic command and overlay it on the aerial map of the CARLA environment. We showcase successful navigation scenarios of CLIP-MC in (a), (b) and (c). With additional contextual information from multiple frames and historical trajectory, CLIP-MC

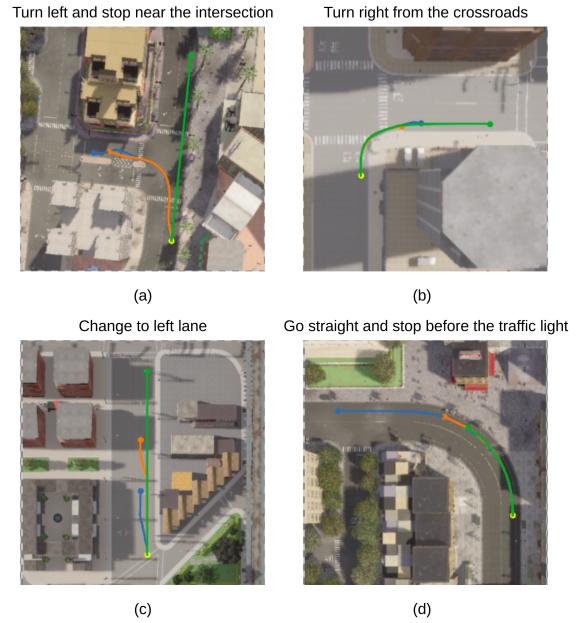


Fig. 4. Qualitative navigation results in the CARLA-NAV dataset. Yellow represents the starting point for the navigation. Orange is used to depict the navigational path taken by CLIP-MC network, green denotes RNR-S network's navigational path and blue represents the ground-truth path.

can successfully perform "turning" and "stopping" based navigational manoeuvres. While the RNR approach, without any contextual information and trained on static images, fails. For the command "change to the left lane", RNR-S fails to change the lane and continues in a straight line. While CLIP-MC manages to change the lane with a slight delay. For the example in the bottom-right corner, the road is curved in left direction and both the CLIP-MC and RNR-S stop much before the traffic light, as they mistake the curve with an intersection.

VII. CONCLUSION

This paper proposes a language-guided navigation approach in dynamically changing outdoor environments. We reformulate the RNR approach, designed for static scenes to make it amenable for dynamic scenes. Our approach explicitly utilizes visual grounding directly for the navigation task. Along the same lines, we propose a novel meta-dataset CARLA-NAV, containing realistic scenarios of language-based navigation in dynamic outdoor environments. Additionally, we propose a novel multi-task grounding network for the tasks of navigable region and future trajectory prediction. The predicted navigable regions are explicitly used for navigating the vehicle in the dynamic environment. The predicted future trajectories bring interpretability to our approach and correlate with the predicted navigable region, i.e., they indicate the vehicle's navigational route. Furthermore, the proposed approach allows us to perform live navigation in a dynamic CARLA environment. Finally, quantitative and qualitative results validate our approach's effectiveness and practicality. Future work should explore domain adaptation techniques like [44], [45] to ensure adaptability to real-world scenes and a learnable stopping criteria.

REFERENCES

- [1] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, “Grounding of textual phrases in images by reconstruction,” in *European Conference on Computer Vision*. Springer, 2016, pp. 817–834.
- [2] R. Hu, M. Rohrbach, and T. Darrell, “Segmentation from natural language expressions,” in *European Conference on Computer Vision*. Springer, 2016, pp. 108–124.
- [3] T. Deruyttere, S. Vandenhende, D. Grrijicic, L. Van Gool, and M. F. Moens, “Talk2car: Taking control of your self-driving car,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2088–2098.
- [4] N. Rufus, K. Jain, U. K. R. Nair, V. Gandhi, and K. M. Krishna, “Grounding linguistic commands to navigable regions,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2021.
- [5] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sunderhauf, I. Reid, S. Gould, and A. Van Den Hengel, “Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments,” in *CVPR*, 2018.
- [6] A. Shrestha, K. Pugdeethosapol, H. Fang, and Q. Qiu, “High-level plan for behavioral robot navigation with natural language directions and r-net,” 2020. [Online]. Available: <https://arxiv.org/abs/2001.02330>
- [7] 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,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2657–2666. [Online]. Available: <https://aclanthology.org/D18-1286>
- [8] R. Schumann and S. Riezler, “Analyzing generalization of vision and language navigation to unseen outdoor areas,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 7519–7532. [Online]. Available: <https://aclanthology.org/2022.acl-long.518>
- [9] W. Zhu, X. Wang, T.-J. Fu, A. Yan, P. Narayana, K. Sone, S. Basu, and W. Y. Wang, “Multimodal text style transfer for outdoor vision-and-language navigation,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 1207–1221. [Online]. Available: <https://aclanthology.org/2021.eacl-main.103>
- [10] J. Xiang, X. Wang, and W. Y. Wang, “Learning to stop: A simple yet effective approach to urban vision-language navigation,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 699–707. [Online]. Available: <https://aclanthology.org/2020.findings-emnlp.62>
- [11] J. Fu, A. Korattikara, S. Levine, and S. Guadarrama, “From language to goals: Inverse reinforcement learning for vision-based instruction following,” *arXiv preprint arXiv:1902.07742*, 2019.
- [12] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, “Touchdown: Natural language navigation and spatial reasoning in visual street environments,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12538–12547.
- [13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Conference on robot learning*. PMLR, 2017, pp. 1–16.
- [14] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell, “Natural language object retrieval,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4555–4564.
- [15] B. A. Plummer, P. Kordas, M. H. Kiapour, S. Zheng, R. Piramuthu, and S. Lazebnik, “Conditional image-text embedding networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 249–264.
- [16] N. Rufus, U. K. R. Nair, K. M. Krishna, and V. Gandhi, “Cosine meets softmax: A tough-to-beat baseline for visual grounding,” in *European Conference on Computer Vision*. Springer, 2020, pp. 39–50.
- [17] C. Deng, Q. Wu, Q. Wu, F. Hu, F. Lyu, and M. Tan, “Visual grounding via accumulated attention,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7746–7755.
- [18] H. Zhang, Y. Niu, and S.-F. Chang, “Grounding referring expressions in images by variational context,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4158–4166.
- [19] S. Yang, G. Li, and Y. Yu, “Dynamic graph attention for referring expression comprehension,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4644–4653.
- [20] H. Qiu, H. Li, Q. Wu, F. Meng, H. Shi, T. Zhao, and K. N. Ngan, “Language-aware fine-grained object representation for referring expression comprehension,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 4171–4180.
- [21] H. Shi, H. Li, F. Meng, and Q. Wu, “Key-word-aware network for referring expression image segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 38–54.
- [22] L. Ye, M. Rochan, Z. Liu, and Y. Wang, “Cross-modal self-attention network for referring image segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10502–10511.
- [23] S. Huang, T. Hui, S. Liu, G. Li, Y. Wei, J. Han, L. Liu, and B. Li, “Referring image segmentation via cross-modal progressive comprehension,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10488–10497.
- [24] K. Jain and V. Gandhi, “Comprehensive multi-modal interactions for referring image segmentation,” *Findings of the Association for Computational Linguistics: ACL*, 2022.
- [25] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11621–11631.
- [26] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, “Ai2-thor: An interactive 3d environment for visual ai,” *arXiv preprint arXiv:1712.05474*, 2017.
- [27] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian, “Building generalizable agents with a realistic and rich 3d environment,” *arXiv preprint arXiv:1801.02209*, 2018.
- [28] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3D: Learning from RGB-D data in indoor environments,” *International Conference on 3D Vision (3DV)*, 2017.
- [29] K. Nguyen, D. Dey, C. Brockett, and B. Dolan, “Vision-based navigation with language-based assistance via imitation learning with indirect intervention,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12527–12537.
- [30] K. Nguyen and H. Daumé III, “Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 684–695. [Online]. Available: <https://aclanthology.org/D19-1063>
- [31] A. Das, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, “Neural modular control for embodied question answering,” in *Conference on Robot Learning*. PMLR, 2018, pp. 53–62.
- [32] F. Landi, L. Baraldi, M. Cornia, M. Corsini, and R. Cucchiara, “Perceive, transform, and act: Multi-modal attention networks for vision-and-language navigation,” *ArXiv*, vol. abs/1911.12377, 2019.
- [33] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, “Beyond the nav-graph: Vision-and-language navigation in continuous environments,” in *European Conference on Computer Vision*. Springer, 2020, pp. 104–120.
- [34] R. Hu, D. Fried, A. Rohrbach, D. Klein, T. Darrell, and K. Saenko, “Are you looking? Grounding to multiple modalities in vision-and-language navigation,” in *ACL*, 2019.
- [35] M. Z. Irshad, C.-Y. Ma, and Z. Kira, “Hierarchical cross-modal agent for robotics vision-and-language navigation,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13238–13246.
- [36] N. Sriram, T. Maniar, J. Kalyanasundaram, V. Gandhi, B. Bhownick, and K. M. Krishna, “Talk to the vehicle: Language conditioned autonomous navigation of self driving cars,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5284–5290.
- [37] P. Mirowski, M. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, A. Zisserman, R. Hadsell *et al.*, “Learning to navigate in cities without a map,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.

- [38] R. Schumann and S. Riezler, “Generating landmark navigation instructions from maps as a graph-to-text problem,” *ACL/IJCNLP*, 2020.
- [39] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *ICML*, 2021.
- [40] S. A. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh, “Combo loss: Handling input and output imbalance in multi-organ segmentation,” *Computerized Medical Imaging and Graphics*, vol. 75, pp. 24–33, 2019.
- [41] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [42] T. Eiter and H. Mannila, “Computing discrete fréchet distance,” 1994.
- [43] G. Ilharco, V. Jain, A. Ku, E. Ie, and J. Baldridge, “General evaluation for instruction conditioned navigation using dynamic time warping,” *arXiv preprint arXiv:1907.05446*, 2019.
- [44] J. N. Kundu, A. Kulkarni, A. Singh, V. Jampani, and R. V. Babu, “Generalize then adapt: Source-free domain adaptive semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7046–7056.
- [45] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, and A. Hauptmann, “Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 3569–3580, 2020.