GENNAV: Polygon Mask Generation for Generalized Referring Navigable Regions

Kei Katsumata[1] **Yui Iioka**[1] **Naoki Hosomi**[2] **Teruhisa Misu**[3]
Kentaro Yamada[2] **Komei Sugiura**[1]
[1]Keio University, Japan, [2]Honda R&D Co., Ltd., Japan, [3]Honda Research Institute USA, USA,
{ke59ka77, kmngrd1805}@keio.jp, naoki_hosomi@jp.honda,
tmisu@honda-ri.com, kentaro_yamada@jp.honda, komei.sugiura@keio.jp

Abstract: We focus on the task of identifying the location of target regions from a natural language instruction and a front camera image captured by a mobility. This task is challenging because it requires both existence prediction and segmentation, particularly for stuff-type target regions with ambiguous boundaries. Existing methods often underperform in handling stuff-type target regions, in addition to absent or multiple targets. To overcome these limitations, we propose GENNAV, which predicts target existence and generates segmentation masks for multiple stuff-type target regions. To evaluate GENNAV, we constructed a novel benchmark called GRiN-Drive, which includes three distinct types of samples: no-target, single-target, and multi-target. GENNAV achieved superior performance over baseline methods on standard evaluation metrics. Furthermore, we conducted real-world experiments with four automobiles operated in five geographically distinct urban areas to validate its zero-shot transfer performance. In these experiments, GENNAV outperformed baseline methods and demonstrated its robustness across diverse real-world environments. The project page is available at https://gennav.vercel.app/.

Keywords: Autonomous Driving, Vision and Language, Semantic Understanding

1 Introduction

Shortages of professional labor pose critical challenges to providing equitable access to transportation, particularly where public transportation systems are fragile. To address this issue, numerous efforts have been made to develop autonomous personal mobility that assists people with mobility impairments [1, 2]. Such systems are expected to facilitate user-friendly interactions between users and autonomous systems. In particular, appropriately understanding navigation instructions is crucial for enabling such interactions. However, most existing systems exhibit limited performance in understanding navigation instructions based on the surrounding environments.

In this study, we focus on the task of identifying the location of target region(s) from both a natural language instruction and a front camera image captured by a mobility. In autonomous driving scenarios, such instructions may not always remain valid if the landmark no longer exists in the image due to the movement of the mobility itself. Accordingly, the model is expected to predict a "no-target" response when no region in the image corresponds to the instruction. Fig. 1 shows a typical use case for our task. In the example, when a user gives the instruction, "Please park to the left of the black car on the right," the model generates a mask that corresponds to the region on the left of the black car. By contrast, the model is expected to predict "no-target" when the user requests "Park left of the red car on the right" but no red car exists.

Our target task is challenging because it requires both predicting the existence of target regions and generating segmentation masks for them. This difficulty is particularly pronounced for stuff-type target regions (e.g., road) [3], which lack clear boundaries, in contrast to thing-type regions (e.g., person, sign or mobility) that correspond to countable objects with clear boundaries. Moreover, existing methods [1, 2] including MLLMs remain insufficient, particularly in handling cases where the target is absent or multiple targets exist. This is partially because these methods lack an explicit

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

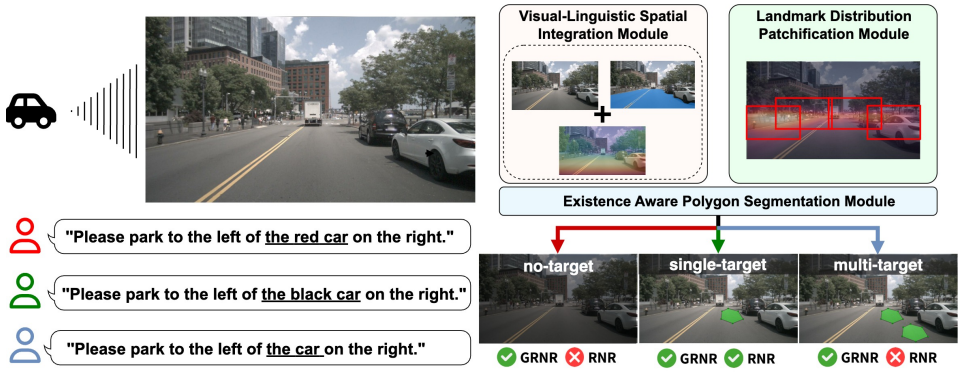


Figure 1: Overview of GENNAV. The model predicts target regions from a natural language instruction and a front camera image captured by a moving mobility.

mechanism for predicting no-target or multi-target responses, as we show in Section 4.2. By contrast, models that support multiple or no-target outputs [4, 5] are primarily designed for thing-type scenarios and do not account for the stuff-type target regions.

To address these limitations, we propose GENNAV which explicitly predicts whether target regions exist and generates segmentation masks for stuff-type target regions. GENNAV differs from existing approaches in the use of polygon based mask generation extended by existence prediction, which allows us to predict an arbitrary number of target regions in a computationally efficient manner. Classic pixel-based methods [1, 2] incur significant computational overhead by predicting every pixel, whereas polygon-based methods [6, 7] are more efficient but typically fail in no-target cases. In contrast, GENNAV extends existence prediction into polygon-based mask generation, effectively overcoming these limitations.

The main contributions of this study are as follows:

- We introduce Existence Aware Polygon Segmentation Module, which predicts the existence of target regions and generates polygon-based segmentation masks.
- We incorporate Landmark Distribution Patchification Module, which models fine-grained visual representations related to potential landmarks by patchifying images based on the spatial distribution of landmarks.
- We construct the GRiN-Drive benchmark, which includes three distinct types of samples: single-target, no-target, and multi-target.
- We validate the zero-shot transfer performance of GENNAV in real-world experiments involving four automobiles operating across five geographically distinct urban areas.

2 Related Work

Research on multimodal language processing for autonomous driving has been widely conducted [8, 9, 10, 11]. Zhou et al. [11] and Cui et al. [8] provide a comprehensive review of the use of LLMs and vision-language models in autonomous driving.

Visual Grounding. Research on visual grounding in driving scenarios has been widely conducted, encompassing various tasks such as referring expression comprehension (REC) tasks [12, 13, 14, 15, 16, 17, 18, 19], referring expression segmentation (RES) tasks [20, 7, 21, 6, 4, 5] and Referring Navigable Regions (RNR) tasks [1, 2]. REC aims to predict a rectangular bounding box that localizes the object described by a natural language phrase. Beyond single-object settings, multi-object and temporally dynamic variants such as Referring Multi-Object Tracking extend REC to sequences in which several landmarks are to be tracked simultaneously [15, 16, 12, 13].

Although REC has made significant progress, bounding box-level localization is often too coarse for the navigation task. To address this limitation, RES models aim to generate a pixel-level mask for the region referred to by the expression [20, 7, 21, 6, 22, 4, 5, 23, 24, 25, 26, 27]. Recent polygon-generation frameworks [20, 7, 21, 6] reformulate RES as sequential polygon prediction and achieve promising results. To accommodate diverse real-world scenarios, segmentation models that handle the existence of one or more targets have also been proposed [4]. Closely related to our study, the RNR task requires the segmentation of the destination region indicated by a navigation instruction [1, 2]. To solve the RNR and related tasks, TNRSM [2] integrates language, image,

and semantic-mask modalities in parallel to predict a destination region, whereas TRiP [28] models relative positional relationships between referred landmarks and target positions.

Vision and Language Navigation. While most VLN studies have focused on indoor environments [29, 30, 31, 32, 33], recent works have also explored its application to autonomous driving, where vehicles follow natural language instructions to reach specified destinations [34, 35, 36, 37, 38, 39]. Such tasks often require language-conditioned localization and path planning, where appropriately grounding the target location mentioned in the instruction is critical for navigation. Talk to the Vehicle [34] introduces a waypoint generation conditioned on natural language instructions. Furthermore, some approaches integrate foundation models to facilitate free-form and zero-shot navigation [35, 36]. For instance, Grounded then Navigate [37] uses CLIP [40] to identify navigable regions aligned with language instructions. While VLN focuses on the generation of a sequence of steps to reach a destination, RNR and GRNR require the generation of segmentation masks that directly identify the target region within an image, which serves as the designated goal.

Benchmarks. The survey by Liu et al. [9] reviews benchmark datasets for autonomous-driving scenes and highlights representative benchmarks such as the BDD100K [41], Cityscapes [42], KITTI [43], and nuScenes [44]. The KITTI vision benchmark suite is a comprehensive driving-scene dataset with ground truth for a variety of tasks, such as semantic segmentation and depth estimation. Refer-KITTI [15] and Refer-KITTI-V2 [16] were built on top of KITTI to establish the RMOT benchmark in which an arbitrary number of objects are tracked according to the given expression. The nuScenes dataset contains full-surround data across various cities and weather conditions, and is widely adopted for multi-view 3D object-detection benchmarks. Built on top of nuScenes, Talk2Car [14] introduces navigation commands that contain landmark-based referring expressions together with the locations of the referred landmarks and Talk2Car-RegSeg [1] further augments the dataset with pixel-wise segmentation masks of the intended navigable regions.

3 Method

3.1 Problem Statements

In this paper, we focus on the Generalized Referring Navigable Regions (GRNR) task, which predicts both the existence and location of target regions based on a natural language instruction and a front camera image taken from a moving mobility. In this task, the model is expected to determine whether any target region specified in the navigation instruction exists in the given image. If one or more target regions exist, the model is required to generate a segmentation mask for each target region.

Fig. 1 shows typical scenes from the GRNR task. Unlike the RNR task, the GRNR task involves instructions that specify any number of target regions, including multi-target and no-target instructions. In this task, we do not focus on trajectory prediction that corresponds to the given target region because existing lower-level functionalities can already perform such trajectory prediction sufficiently in practical applications. Further details of the task examples are available in the supplementary material.

The terminology used in this paper is defined as follows. An "navigation instruction" refers to an instruction in natural language to navigate the mobility to the destination. The "target region" refers to the region specified by a given navigation instruction. The relationship between the instruction and the target region follows a one-to-many mapping (if any). "Landmark" is an object used as a reference when specifying a target region.

3.2 GENNAV

We propose GENNAV which integrates polygon-based segmentation with spatially grounded multimodal representations and landmark-distribution-driven patchification. GENNAV models both the semantic and spatial characteristics with low computational overhead. It is inspired by polygon-based referring expression segmentation (RES) approaches [20, 7, 21, 6], and is widely applicable to various RES and RNR models. Fig. 2 shows the structure of GENNAV. It consists of three modules: Existence Aware Polygon Segmentation module (ExPo), Landmark Distribution Patchification Module (LDPM), and Visual-Linguistic Spatial Integration Module (VLSiM).

We define the input $\boldsymbol{x}$ to the model as follows: $\boldsymbol{x} = \{\boldsymbol{x}_{\text{img}}, \boldsymbol{x}_{\text{inst}}\}$, where $\boldsymbol{x}_{\text{img}} \in \mathbb{R}^{3\times W\times H}$ and $\boldsymbol{x}_{\text{inst}} \in \{0,1\}^{V\times L}$ denote a front-camera image and a navigation instruction, respectively, with the instruction represented as a sequence of one-hot vectors with the sequence length limited to a

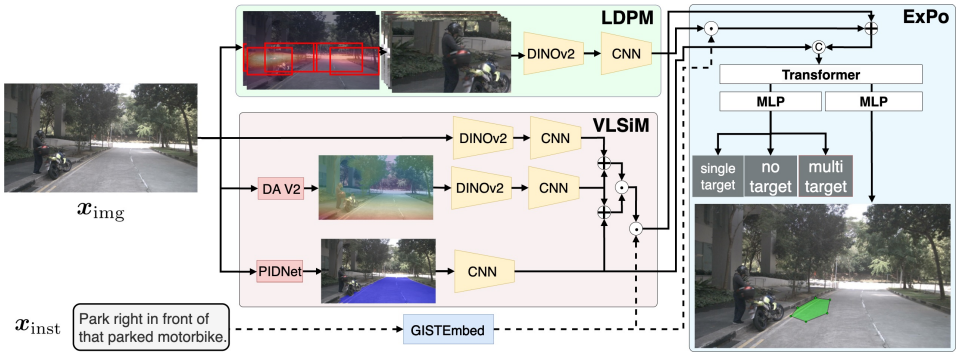


Figure 2: Overall architecture of GENNAV. DA represents Depth Anything [45]. The green, red, and blue regions in this figure represent the LDPM, VLSiM, and ExPo modules, respectively.

maximum of L over a vocabulary of size V. Here, W and H denote the image width and height. We preprocess $\boldsymbol{x}_{\text{inst}}$ with GISTEmbed [46] to obtain the linguistic feature $\boldsymbol{h}_{\text{inst}} \in \mathbb{R}^{C_{\text{inst}}}$, where C_{inst} denotes the dimensionality of the linguistic feature.

Landmark Distribution Patchification Module. This module models fine-grained visual representations related to potential landmarks by efficiently partitioning patches based on the spatial distribution of landmarks. Such representations are crucial because $\boldsymbol{x}_{\text{inst}}$ sometimes specify pedestrians/vehicles, however, most existing methods (e.g., [1, 2]) resize input images to a low resolution, resulting in an insufficient number of pixels being allocated to distant landmarks.

The input to this module is $\boldsymbol{x}_{\text{img}}$. We adopt a patchification strategy for utilizing high-resolution images. We introduce landmark-distribution-based patchification that covers regions where landmarks are likely to be densely distributed because classic uniform partitioning is inefficient. The left image within the LDPM (green) in Fig. 2 shows the patches (shown in red bounding boxes) and the distribution of landmarks in the training set of the Talk2Car dataset [14]. Each patch region is encoded by DINOv2 [47], followed by a CNN. The final feature $\boldsymbol{h}_{\text{ldp}}$ is obtained by integrating the resulting features across all patch regions.

Visual-Linguistic Spatial Integration Module. This module models the relationship between linguistic and spatial information by integrating $\boldsymbol{x}_{\text{img}}$, $\boldsymbol{h}_{\text{inst}}$, a pseudo-depth image, and road region. The inputs to VLSiM are $\boldsymbol{x}_{\text{img}}$ and $\boldsymbol{h}_{\text{inst}}$. First, we obtain visual features from $\boldsymbol{x}_{\text{img}}$ using a frozen DINOv2 [47], followed by a CNN. This procedure is denoted by $f_{\text{vis}}(\boldsymbol{x}_{\text{img}})$.

Next, we generate a pseudo-depth image $\boldsymbol{x}_{\text{depth}} \in \mathbb{R}^{3\times W\times H}$ from $\boldsymbol{x}_{\text{img}}$ using a monocular depth estimation model (e.g., Depth Anything V2 [47]) and we overlay the pseudo depth image with $\boldsymbol{x}_{\text{img}}$. Similar to $f_{\text{vis}}(\boldsymbol{x}_{\text{img}})$, we input the overlaid image to DINOv2 and an additional CNN. We introduce the CNN because DINOv2 is not specifically trained for overlaid images. The above procedure is denoted by $f_{\text{depth}}(\boldsymbol{x}_{\text{img}})$. Then, we integrate $\boldsymbol{h}_{\text{inst}}$ with a sum of $f_{\text{vis}}(\boldsymbol{x}_{\text{img}})$ and $f_{\text{depth}}(\boldsymbol{x}_{\text{img}})$ to model the relationship between linguistic and spatial features.

Moreover, to mitigate the generation of masks on inappropriate regions (e.g., the sky or areas occluded by objects), we integrate $f_{\text{depth}}(\boldsymbol{x}_{\text{img}})$ with a road representation. We obtain a mask of the road $\boldsymbol{x}_{\text{road}} \in \mathbb{R}^{3\times W\times H}$ using a semantic segmentation model (e.g., PIDNet [48]) given $\boldsymbol{x}_{\text{img}}$. Subsequently, we input the mask to a CNN. The procedure is denoted by $f_{\text{road}}(\boldsymbol{x}_{\text{img}})$. Finally, we obtain the following $\boldsymbol{h}_{\text{mm}}$ as the output of VLSiM:

$$\boldsymbol{h}_{\text{mm}} = \boldsymbol{h}_{\text{inst}} \odot \{f_{\text{vis}}(\boldsymbol{x}_{\text{img}}) + f_{\text{depth}}(\boldsymbol{x}_{\text{img}})\} \odot \{f_{\text{road}}(\boldsymbol{x}_{\text{img}}) + f_{\text{depth}}(\boldsymbol{x}_{\text{img}})\}, \tag{1}$$

where $\odot$ denotes the Hadamard product.

Existence Aware Polygon Segmentation Module. This module predicts the existence of target regions and generates polygon-based segmentation masks by integrating four types of information: $\boldsymbol{h}_{\text{inst}}$, $\boldsymbol{h}_{\text{mm}}$, $f_{\text{road}}(\boldsymbol{x}_{\text{img}})$ and $\boldsymbol{h}_{\text{ldp}}$. This module has two heads: a classification head that predicts the existence of target regions and a regression head that generates the segmentation masks of target regions. The regression head leverages polygon-based segmentation that enables more efficient inference than existing pixel-based segmentation approaches [1, 2], which is experimentally validated in Section 4.4.

The module takes $\boldsymbol{h}_{\text{inst}}$, $\boldsymbol{h}_{\text{mm}}$, $f_{\text{road}}(\boldsymbol{x}_{\text{img}})$ and $\boldsymbol{h}_{\text{ldp}}$ as inputs. It integrates $\boldsymbol{h}_{\text{inst}}$ and $\boldsymbol{h}_{\text{ldp}}$ to obtain a multimodal representation, $\boldsymbol{h}_{\text{cmm}} = (\boldsymbol{h}_{\text{ldp}} \odot \boldsymbol{h}_{\text{inst}}) + \boldsymbol{h}_{\text{mm}}$. Next, the regression head predicts the

polygon vertices of target regions $\hat{\boldsymbol{c}}_i = \text{MLP}\left([\boldsymbol{h}_{\text{cmm}}; f_{\text{road}}\left(\boldsymbol{x}_{\text{img}}\right)]\right)$, where $\text{MLP}(\cdot)$, $\hat{\boldsymbol{c}}_i \in \mathbb{R}^2$ $(i = 1, \ldots, n_{\text{pt}})$, and $[\cdot;\cdot]$ denote an MLP, the coordinates of each vertex of the predicted polygon ordering in a clockwise manner starting from the top-left, and concatenation, respectively. Similarly, the classification head predicts $p(\hat{\boldsymbol{y}}_{\text{ex}})$ over the categories $\{\text{no-target}, \text{single-target}, \text{multi-target}\}$ based on the same input as the regression head. Consequently, the output from the ExPo module is represented as $\hat{\boldsymbol{y}} = \{p(\hat{\boldsymbol{y}}_{\text{ex}}), \hat{\boldsymbol{c}}_i\}$.
We use the following loss function $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}\left(p(\hat{\boldsymbol{y}}_{\text{ex}}), \boldsymbol{y}_{\text{ex}}\right) + \lambda_{\text{pt}} \sum_{i=1}^{n_{\text{pt}}} \mathbb{I}\left[\boldsymbol{y}_{\text{ex}} \in \{\text{single-target}, \text{multi-target}\}\right] \cdot \ell_1\left(\hat{\mathbf{c}}_i, \mathbf{c}_i\right), \tag{2}$$

where λ_{pt}, $\mathcal{L}_{\text{CE}}(\cdot,\cdot)$, ℓ_1, and $\mathbb{I}[\cdot]$ denote the loss weight, cross-entropy loss, L1 loss and indicator function, respectively.

4 Experiments

4.1 Experimental Setup

Benchmark. To evaluate the performance of models on the three distinct types of cases in the GRNR task, we constructed the novel GRiN-Drive benchmark based on the Talk2Car-RegSeg [1] and Refer-KITTI-V2 [16] datasets. To the best of our knowledge, aside from Talk2Car-RegSeg, most datasets do not provide front camera images that are annotated with instruction texts and segmentation masks of target regions. In addition, the Talk2Car-RegSeg dataset alone is insufficient for the GRNR task because it lacks two critical components: (i) samples where no target region corresponds to the navigation instructions and (ii) samples that contain multiple landmarks in a single image. To ensure that datasets cover no-target, single-target, and multi-target cases, we newly collected no-target and multi-target samples and constructed the GRiN-Drive benchmark. Accordingly, the GRiN-Drive benchmark serves as a suitable benchmark for evaluating the performance of models in the three distinct types of cases in the GRNR task.
The GRiN-Drive benchmark consists of 17,114 samples. Each sample consists of an image, navigation instruction and zero or more segmentation masks for each target region. It has a vocabulary size, total number of words, and average sentence length of 2,699, 168,640, and 10.55 words, respectively. The training, validation, and test sets contained 14,973, 1,413, and 758 samples, respectively. We followed the original split [1, 15], as adopting a different dataset split would lead to an unfair comparison with baseline methods. The GRiN-Drive test set achieves a statistical power greater than 0.999. We used the training set for training models, validation set for adjusting their hyperparameters, and test set for evaluating the models. The details of the construction of GRiN-Drive benchmark are explained in the supplementary materials.

Baselines. We used two groups of baseline methods: pixel-based methods (a method by Rufus et al. [1], LAVT [22], TNRSM [2], and GSVA-Vicuna-7B [5]) and MLLMs (Gemini [49], GPT-4o [50], and Qwen2-VL [51]). Each method was used as a baseline method for the following reasons. We selected LAVT [22] and GSVA-Vicuna-7B [5] as a baseline method because of its successful application to pixel-wise referring image segmentation tasks. Similarly, we used the method by Rufus et al. and TNRSM [2] because of their effective application in the RNR task, which is closely related to the Generalized Referring Navigable Regions (GRNR) task. The details of the experimental settings of baseline methods are explained in the supplementary materials.

Evaluation Metrics. We used mean stuff Intersection over Union (msIoU), precision@K (P@K), accuracy (Acc.), and inference speed as evaluation metrics, with msIoU as the primary metric. We propose the new metric msIoU to evaluate single-target, multi-target, and no-target samples without biases. By contrast, most existing metrics yield biased evaluations: a correct no-target prediction yields an IoU of 1.0, whereas even accurate target segmentations yield scores lower than 1.0. Therefore, the metric favors trivial solutions focusing on target-existence classification, which is weighted more than mask generation. Indeed, in our task, even a trivial solution that always predicts "no target" can achieve a gIoU[4] score of 0.33, which is deceptively high, surpassing the human performance of msIoU 0.17 on target samples in the test set of the GRiN-Drive benchmark.
To address this, msIoU assigns a score of 1 to samples with IoU exceeding a threshold K, while normalizing IoU scores based on K for those below the threshold, as shown in Equation 3. By further averaging sIoU@k where (k =1, …, 1/K), msIoU achieves a balanced evaluation for both

Table 1: Quantitative comparison between GENNAV and the baseline methods on the test sets of the GRiN-Drive benchmark. The best score for each metric is in bold. The "Type" and "Inf. speed" columns specify the segmentation approach and inference speed of each method, respectively.

Method	Resolution	Type	msIoU [%]↑	P@0.1 [%]↑	P@0.2 [%]↑	Acc. [%]↑	Inf. speed [ms/sample]↓
Rufus et al. [1]	224×224	Pixel	35.44 ±3.12	21.53 ±4.42	17.52 ±2.41	51.61 ±9.86	39.87 ±1.10
LAVT [22]	224×224	Pixel	31.73 ±2.74	30.28 ±3.16	23.53 ±2.19	40.95 ±1.56	**9.29** ±0.24
TNRSM [2]	224×224	Pixel	37.90 ±0.82	44.05 ±1.33	34.09 ±0.58	67.23 ±2.72	502.69 ±0.22
Rufus et al. [1]	640×640	Pixel	37.84 ±1.95	11.48 ±4.42	7.68 ±3.64	52.19 ±4.34	65.00 ±2.11
LAVT [22]	640×640	Pixel	34.09 ±1.53	7.90 ±17.66	6.79 ±15.17	38.66 ±10.69	11.51 ±0.33
TNRSM [2]	640×640	Pixel	22.84 ±8.77	38.57 ±4.25	19.45 ±7.53	67.89 ±1.11	503.68 ±0.20
GSVA-Vicuna-7B [5]	1024×1024	Pixel	32.94 ±0.04	17.26 ±0.24	9.36 ±0.14	64.76 ±0.11	306.73 ±7.11
Gemini [49]	1600×900	Bbox	6.98 ±0.58	6.92 ±1.46	1.63 ±0.51	46.33 ±0.05	1793.68 ±37.96
GPT-4o [50]	1600×900	Bbox	23.41 ±5.72	5.04 ±1.37	0.04 ±0.09	62.44 ±9.35	3525.68 ±701.04
Qwen2-VL [51]	1600×900	Bbox	24.06 ±1.15	3.85 ±0.95	1.64 ±0.52	64.25 ±0.91	1768.99 ±0.41
Qwen2-VL [51]	1600×900	Polygon	12.16 ±4.86	0.08 ±0.18	1.88 ±0.74	44.66 ±5.97	1771.41 ±6.20
GENNAV (ours)	640×640	Polygon	**46.35** ±1.52	**49.60** ±1.12	**34.40** ±1.20	**75.41** ±0.67	31.31 ±0.05
Human	1600×900	Polygon	56.08	56.00	36.40	88.00	-

target and no-target samples. The details of the remaining evaluation metrics are explained in the supplementary materials. msIoU is defined as follows:

$$\text{msIoU} = \text{mean}\left(\frac{1}{N}\sum_{i=1}^{N}\text{sIoU@}k_i\right), \tag{3}$$

$$\text{sIoU@}k_i = \begin{cases} \min\left(k \cdot \text{IoU}(\hat{y}_i, y_i), 1\right) & \text{if TP,} \\ 1 & \text{if TN,} \\ 0 & \text{if FP or FN,} \end{cases} \tag{4}$$

where $\hat{y}_i$ and y_i denote the predicted and ground-truth masks for the i-th sample, respectively; N denotes the number of samples, and $\text{IoU}(\cdot)$ represents the IoU between them. Additionally, TP, TN, FP and FN denote the numbers of true positive, true negative, false positive, and false negative samples, respectively, which represents the prediction regarding the existence of target regions. The details of the remaining evaluation metrics are explained in the supplementary materials.

4.2 Quantitative Results

Table 1 shows the quantitative results of the baseline methods, GENNAV and human performance on the GRiN-Drive benchmark. The values in the table are the average and standard deviation over five trials. The "Type" and "Inf. speed" columns specify the segmentation approach and inference speed of each method, respectively. The detailed discussions of quantitative results and the subject experiment are provided in the supplementary material.

Table 1 indicates that GENNAV achieved highest msIoU of 46.35. These results show that GENNAV improved by 8.45 points over TNRSM (224×224), which achieved the highest msIoU among the baseline methods. Similarly, GENNAV achieved P@0.1 and accuracy of 49.60% and 75.41%, respectively, surpassing the baseline methods, including the MLLMs, with a 5.55 and 7.52 point improvement over the best-performing baseline, TNRSM (224×224). The inference speed per sample for GENNAV was 31.31 ms. Overall, GENNAV achieved faster inference speed than the baseline methods except for LAVT. Although GENNAV was slower than LAVT, it achieved substantially higher performance on other metrics. The improvement achieved by GENNAV in terms of msIoU, P@0.1, and accuracy was statistically significant ($p < 0.05$).

4.3 Qualitative Results

Fig. 3 shows the qualitative results of GENNAV and the baseline methods on the GRiN-Drive benchmark. Figs. 3 (i) shows single-target cases and Figs. 3 (ii) shows multi-target cases. Further qualitative results, including no-target cases, are provided in the supplementary materials. Figs. 3 (i) presents predictions given "Park my car next to the rack" as $\boldsymbol{x}_{\text{inst}}$. Masks should be generated for the region in front of the rack, because there is a bike rack located center-left in the image. GENNAV correctly generated the region whereas TNRSM predicted a "no-target" and LAVT and Qwen2-VL (bbox) generated inappropriate masks. In Figs. 3 (ii), $\boldsymbol{x}_{\text{inst}}$ was "Pull up right next to pedestrian

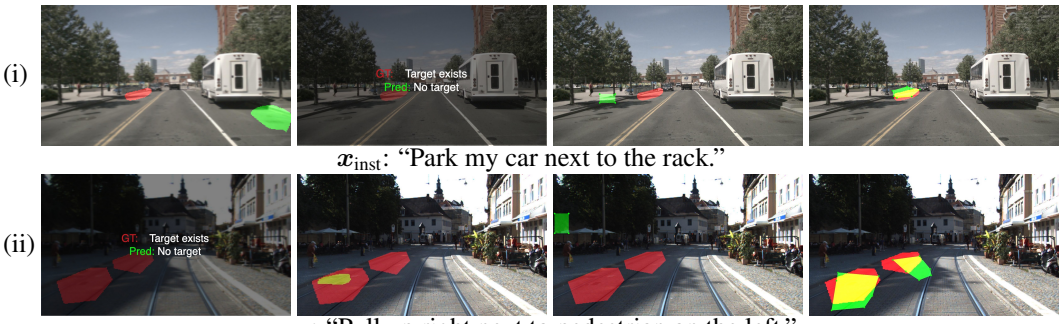


Figure 3: Qualitative results of GENNAV and baseline methods on the GRiN-Drive benchmark. Columns (a), (b), (c) and (d) show the prediction by LAVT, TNRSM, Qwen2-VL (bbox) and GENNAV, respectively. The green and red regions indicate the predicted and ground-truth regions, respectively; yellow indicates their overlap.

Table 2: Quantitative results of the ablation studies. The best values for each metric are highlighted in bold. The "Type" column specifies the segmentation approach of each method.

Model	Type	LDPM	VLSiM		msIoU [%]↑	P@0.1 [%]↑	P@0.2 [%]↑	Acc. [%]↑
			f_{depth}	f_{road}				
(i)	Polygon		✓	✓	43.05 $\pm$ 0.72	40.75 $\pm$ 1.77	24.48 $\pm$ 1.59	74.62 $\pm$ 0.86
(ii)	Polygon	✓		✓	44.13 $\pm$ 0.26	48.14 $\pm$ 0.44	31.63 $\pm$ 0.57	74.01 $\pm$ 0.79
(iii)	Polygon	✓	✓		44.53 $\pm$ 0.78	46.59 $\pm$ 2.56	30.95 $\pm$ 1.41	73.75 $\pm$ 0.49
(iv)	Pixel	✓	✓	✓	35.26 $\pm$ 1.21	27.18 $\pm$ 2.10	18.02 $\pm$ 1.67	62.93 $\pm$ 2.03
(v)	Polygon	✓	✓	✓	**46.35** $\pm$ 1.52	**49.60** $\pm$ 1.12	**34.40** $\pm$ 0.80	**75.41** $\pm$ 0.67

on the left." Therefore, the model is required to generate a mask for a region located next to each pedestrian positioned on the left side of the road. GENNAV successfully generated the mask on the appropriate region for each of the two pedestrians. However, none of the baseline methods were able to generate appropriate masks; for instance, TNRSM incorrectly predicted only a single region.

4.4 Ablation Study

We conducted three ablation studies to investigate the contribution of each module in GENNAV. Table 2 shows the results of the ablation studies. The table presents the average and standard deviation of the results of five trials. The highest score for each metric is in bold.

LDPM Ablation. We omitted LDPM to analyze its contributions. Table 2 shows that Model (i) achieved msIoU of 43.05, which was 3.3 points lower than that of Model (v), respectively. This indicates that LDPM enhanced the understanding of landmarks by partitioning patches based on landmark distributions.

VLSiM Ablation. To investigate the impact of the subnetworks $f_{\text{depth}}(\cdot)$ and $f_{\text{road}}(\cdot)$ on performance, we conducted ablation by removing each of them individually. First, we compare Model (v) (full) and Model (ii) (without $f_{\text{depth}}(\cdot)$). Model (ii) showed decreases of 2.22 points in msIoU. These results imply that $f_{\text{depth}}(\cdot)$ assisted the alignment between linguistic and spatial information by using generated pseudo-depth images. Similarly, we removed $f_{\text{road}}(\cdot)$ to assess its contribution. As shown in Table 2, msIoU of Model (iii) was 1.82 points lower, respectively, than that of Model (v). This indicates that $f_{\text{road}}(\cdot)$ facilitated the understanding of immovable regions and improved the model's mask generation capability.

Decoder Ablation. To analyze the impact of the decoder structure, we compare our polygon-based approach with a pixel-based method. Model (iv) underperformed Model (v) by 11.09 points in terms of msIoU. Notably, Model (iv), which employs pixel-based prediction, exhibits a decrease in inference speed to 47.92 ms/sample compared to polygon-based Model (v) 31.31 ms/sample. This demonstrates that our polygon-based approach successfully reduced the computational cost.

5 Real-World Experiments

5.1 Experimental Setup

We validated GENNAV in zero-shot, real-world experiments. In real-world experiments, we used four automobiles operated across five geographically distinct urban areas, which resulted in 20 video

recordings. Each vehicle was equipped with smartphone cameras (both iPhone and Android devices), which allowed us to evaluate GENNAV under varying hardware setups. Video data were collected under a wide range of traffic and environmental conditions (e.g., low-light settings, clear weather, and snow-covered roads), which covered diverse real-world scenarios. The model executed the GRNR task based on navigation instructions in environments that had not been observed during training. The experiments consisted of 120 episodes across single-target, no-target and multi-target scenarios, with 40 episodes per scenario. The experiment involved single-target, no-target and multi-target scenarios, each of which consisted of 40 scenarios. The details for the assignment of instructions and data selection are shown in the supplementary materials.

5.2 Quantitative Results

Table 3 shows the quantitative comparison between GENNAV and baseline methods in the real-world experiment. The values in the table are the average and standard deviation over five trials. We show the best baseline method for each resolution. The remaining results and detailed discussions are provided in the supplementary material. GENNAV reached an msIoU of 34.32, whereas a method by TNRSM (224×224), LAVT(640×640), and Qwen2-VL (bbox) resulted in scores of 23.11, 30.44, and 20.48, respectively. These results show that GENNAV achieved the highest msIoU among the baseline methods. Overall, GENNAV achieved higher P@0.1 (28.75%) and accuracy (67.50%) than the baseline methods in the real-world experiments conducted in a zero-shot manner.

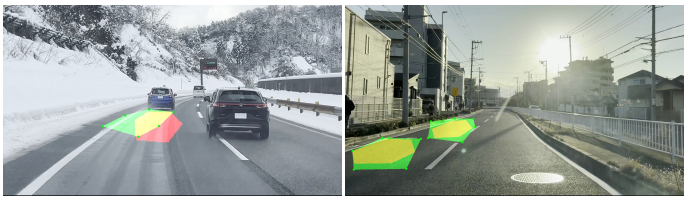
Table 3: Quantitative comparison between the proposed method and baseline methods in the real-world experiment. The best score for each metric is in bold.

Method	msIoU [%]↑	P@0.1 [%]↑	Acc. [%]↑
TNRSM [2] (224×224)	23.11 ± 5.88	24.25 ± 4.11	56.83 ± 5.08
LAVT [22] (640×640)	30.44 ± 1.63	11.73 ± 10.69	32.67 ± 19.74
Qwen2-VL [51] (bbox)	20.48 ± 2.33	5.75 ± 2.59	66.33 ± 3.36
GENNAV (ours)	**34.32** ± 2.97	**28.75** ± 5.08	**67.50** ± 2.95

5.3 Qualitative Results

Fig. 4 shows successful cases of GENNAV in the real-world experiments. Fig. 4 (i) shows a single-target case and Fig. 4 (ii) shows a multi-target case. Fig. 4 (i) presents an example where $\boldsymbol{x}_{\text{inst}}$ was "Please go near the blue car traveling in the same direction." GENNAV appropriately generated the region behind the blue car traveling in the same direction.

Fig. 4 (ii) illustrates an example of the $\boldsymbol{x}_{\text{inst}}$, "Stop to the right of the pedestrian on the left side." The models were required to generate a mask for a region to the right of each pedestrian on the left side. The proposed model successfully generated the mask appropriately for each of the two pedestrians.



(i) $\boldsymbol{x}_{\text{inst}}$: "Please go near the blue car traveling in the same direction." (ii) $\boldsymbol{x}_{\text{inst}}$: "Stop to the right of the pedestrian on the left side."

Figure 4: Qualitative results of GENNAV in the real world experiment. The color of regions are the same as in Fig. 2

6 Conclusion

In this paper, we focused on the GRNR task and proposed GENNAV, which predicts whether target regions exist and generates segmentation masks in cases that involve stuff-type target regions. Our contributions can be summarized as follows. We introduced GENNAV, which consists of three novel modules. (i) ExPo, which predicts the existence of target regions and generates polygon-based segmentation masks. (ii) LDPM, which models fine-grained visual representations related to potential landmarks by patchifying images based on the spatial distribution of landmarks. (iii) VLSiM, which models the relationship between language and spatial information. We constructed the GRiN-Drive benchmark, which includes three distinct types of samples: single-target, no-target, and multi-target scenarios. Our proposed method GENNAV outperformed the baseline methods on the GRiN-Drive benchmark. In real-world experiments, GENNAV outperformed the baseline methods, including the MLLMs, under a zero-shot transfer setting. These results indicate that GENNAV can be effectively integrated into real-world systems.

Limitations

Although we obtained promising results, the proposed method has several limitations. First, it exhibits limited semantic understanding of landmarks, which can lead to an inappropriate understanding of the scene. To address this, one potential solution is to overlay masks generated by more robust semantic segmentation models (e.g., Grounded SAM [52], Detic [53], Grounding DINO [54] and Open-Vocabulary SAM [55]) onto the input images before processing them with LDPM. Second, the current method lacks temporal modeling, which may lead to inconsistent predictions when applied to video-based GRES tasks. Incorporating information from adjacent frames could improve the coherence of predictions across time. In future work, we may explore temporal modeling strategies to maintain consistency across frames and improve overall performance.

References

[1] N. Rufus, K. Jain, K. Nair, V. Gandhi, and M. Krishna. Grounding Linguistic Commands to Navigable Regions. In *IROS*, pages 8593–8600, 2021.

[2] N. Hosomi, S. Hatanaka, Y. Iioka, W. Yang, K. Kuyo, T. Misu, K. Yamada, and K. Sugiura. Trimodal Navigable Region Segmentation Model: Grounding Navigation Instructions in Urban Areas. *IEEE RA-L*, 9(5):4162–4169, 2024.

[3] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic Segmentation. In *CVPR*, pages 9404–9413, 2019.

[4] C. Liu, H. Ding, and X. Jiang. GRES: Generalized Referring Expression Segmentation. In *CVPR*, pages 23592–23601, 2023.

[5] Z. Xia, D. Han, Y. Han, X. Pan, S. Song, and G. Huang. GSVA: Generalized Segmentation via Multimodal Large Language Models. In *CVPR*, pages 3858–3869, 2024.

[6] T. Nishimura, K. Kuyo, M. Kambara, and K. Sugiura. Object Segmentation from Open-Vocabulary Manipulation Instructions Based on Optimal Transport Polygon Matching with Multimodal Foundation Models. In *IROS*, pages 9549–9556, 2024.

[7] J. Liu, H. Ding, Z. Cai, Y. Zhang, R. Satzoda, V. Mahadevan, and R. Manmatha. PolyFormer: Referring Image Segmentation as Sequential Polygon Generation. In *CVPR*, pages 18653–18663, 2023.

[8] C. Cui, Y. Ma, X. Cao, W. Ye, Y. Zhou, K. Liang, J. Chen, J. Lu, Z. Yang, K.-D. Liao, et al. A Survey on Multimodal Large Language Models for Autonomous Driving. In *WACV*, pages 958–979, 2024.

[9] M. Liu, E. Yurtsever, J. Fossaert, X. Zhou, W. Zimmer, Y. Cui, B. Zagar, and A. Knoll. A Survey on Autonomous Driving Datasets: Statistics, Annotation Quality, and a Future Outlook. *IEEE T-IV*, pages 1–29, 2024.

[10] K. Huang, B. Shi, X. Li, X. Li, S. Huang, and Y. Li. Multi-modal Sensor Fusion for Auto Driving Perception: A Survey. *arXiv preprint arXiv:2202.02703*, 2022.

[11] X. Zhou, M. Liu, E. Yurtsever, B. Zagar, W. Zimmer, H. Cao, and A. Knoll. Vision Language Models in Autonomous Driving: A Survey and Outlook. *IEEE T-IV*, pages 1–20, 2024.

[12] Y. Du, C. Lei, Z. Zhao, and F. Su. iKUN: Speak to Trackers without Retraining. In *CVPR*, pages 19135–19144, 2024.

[13] P. Nguyen, K. Quach, K. Kitani, and K. Luu. Type-to-Track: Retrieve Any Object via Prompt-based Tracking. In *NeurIPS*, volume 36, pages 3205–3219, 2023.

[14] T. Deruyttere, S. Vandenhende, D. Grujicic, L. Gool, and M. Moens. Talk2Car: Taking Control of Your Self-Driving Car. In *EMNLP*, pages 2088–2098, 2019.

[15] D. Wu, W. Han, T. Wang, X. Dong, X. Zhang, and J. Shen. Referring Multi-Object Tracking. In *CVPR*, pages 14633–14642, 2023.

[16] Y. Zhang, D. Wu, W. Han, and X. Dong. Bootstrapping Referring Multi-Object Tracking. *arXiv preprint arXiv:2406.05039*, 2024.

[17] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion. MDETR-Modulated Detection for End-to-end Multi-modal Understanding. In *ICCV*, pages 1780–1790, 2021.

[18] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei. MOTR: End-to-End Multiple-Object Tracking with Transformer. In *ECCV*, pages 659–675, 2022.

[19] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu. FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking. *IJCV*, 129:3069–3087, 2021.

[20] C. Zhu, Y. Zhou, Y. Shen, G. Luo, X. Pan, M. Lin, C. Chen, L. Cao, X. Sun, and R. Ji. SeqTR: A Simple yet Universal Network for Visual Grounding. In *ECCV*, pages 598–615, 2022.

[21] Z. Cheng, K. Li, P. Jin, S. Li, X. Ji, L. Yuan, C. Liu, and J. Chen. Parallel Vertex Diffusion for Unified Visual Grounding. In *AAAI*, pages 1326–1334, 2024.

[22] Z. Yang, J. Wang, Y. Tang, K. Chen, H. Zhao, and P. H. Torr. LAVT: Language-Aware Vision Transformer for Referring Image Segmentation. In *CVPR*, pages 18155–18165, 2022.

[23] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia. LISA: Reasoning Segmentation via Large Language Model. In *CVPR*, pages 9579–9589, 2024.

[24] M. Cheng, S. Zheng, W. Lin, V. Vineet, P. Sturgess, N. Crook, N. Mitra, and P. Torr. ImageSpirit: Verbal Guided Image Parsing. *ACM Trans. Graph.*, 34(1), 2015.

[25] Y. Hu, Q. Wang, W. Shao, E. Xie, Z. Li, J. Han, and P. Luo. Beyond One-to-One: Rethinking the Referring Image Segmentation. In *ICCV*, pages 4067–4077, 2023.

[26] Z. Luo, Y. Wu, T. Cheng, Y. Liu, Y. Xiao, H. Wang, X. Zhang, and Y. Yang. CoHD: A Counting-Aware Hierarchical Decoding Framework for Generalized Referring Expression Segmentation. *arXiv preprint arXiv:2405.15658*, 2024.

[27] W. Li, Z. Zhao, H. Bai, and F. Su. Bring Adaptive Binding Prototypes to Generalized Referring Expression Segmentation. *arXiv preprint arXiv:2405.15169*, 2024.

[28] N. Hosomi, Y. Iioka, S. Hatanaka, T. Misu, K. Yamada, N. Tsukamoto, S. Kobayashi, and K. Sugiura. Multimodal Target Localization With Landmark-Aware Positioning for Urban Mobility. *IEEE RA-L*, 10(1):716–723, 2025.

[29] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Hengel. Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. In *CVPR*, pages 3674–3683, 2018.

[30] F. Zhu, X. Liang, Y. Zhu, Q. Yu, X. Chang, and X. Liang. SOON: Scenario Oriented Object Navigation with Graph-Based Exploration. In *CVPR*, pages 12689–12699, 2021.

[31] Y. Qi, Q. Wu, P. Anderson, X. Wang, Y. Wang, C. Shen, and A. Hengel. REVERIE: Remote Embodied Visual Referring Expression in Real Indoor Environments. In *CVPR*, pages 9982–9991, 2020.

[32] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang. NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation. In *RSS*, 2024.

[33] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee. Beyond the Nav-Graph: Vision-and-Language Navigation in Continuous Environments. In *ECCV*, pages 104–120, 2020.

[34] T. Deruyttere, D. Grujicic, M. B. Blaschko, and M.-F. Moens. Talk2car: Predicting Physical Trajectories for Natural Language Commands. *IEEE Access*, 10:123809–123834, 2022.

[35] M. Omama, P. Inani, P. Paul, and et al. ALT-Pilot: Autonomous navigation with Language augmented Topometric maps. *arXiv preprint arXiv:2310.02324*, 2023.

[36] D. Shah, B. Osiński, B. Ichter, and S. Levine. LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action. In *CoRL*, pages 492–504, 2022.

[37] K. Jain, V. Chhangani, A. Tiwari, K. M. Krishna, and V. Gandhi. Ground then Navigate: Language-guided Navigation in Dynamic Scenes. In *ICRA*, pages 4113–4120, 2023.

[38] J. Xiang, X. Wang, and Y. Wang. Learning to Stop: A Simple yet Effective Approach to Urban Vision-Language Navigation. In *EMNLP*, pages 699–707, 2020.

[39] R. Schumann and S. Riezler. Analyzing Generalization of Vision and Language Navigation to Unseen Outdoor Areas. In *ACL*, pages 7519–7532, 2022.

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, pages 12888–12900, 2021.

[41] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *CVPR*, pages 2636–2645, 2020.

[42] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *CVPR*, pages 3213–3223, 2016.

[43] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, pages 3354–3361, 2012.

[44] H. Caesar, V. Bankiti, A. Lang, S. Vora, V. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *CVPR*, pages 11621–11631, 2020.

[45] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth Anything V2. *arXiv:2406.09414*, 2024.

[46] A. Solatorio. GISTEmbed: Guided In-sample Selection of Training Negatives for Text Embedding Fine-tuning. *arXiv preprint arXiv:2402.16829*, 2024.

[47] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, et al. DINOv2: Learning Robust Visual Features without Supervision, 2023.

[48] J. Xu, Z. Xiong, and S. Bhattacharyya. PIDNet: A Real-Time Semantic Segmentation Network Inspired by PID Controllers. In *CVPR*, pages 19529–19539, 2023.

[49] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, et al. Gemini 1.5: Unlocking Multimodal Understanding Across Millions of Tokens of Context. *arXiv preprint arXiv:2403.05530*, 2024.

[50] J. Achiam, S. Adler, S. Agarwal, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

[51] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, et al. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[52] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, et al. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[53] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting Twenty-thousand Classes using Image-level Supervision. In *ECCV*, pages 350–368, 2022.

[54] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv preprint arXiv:2303.05499*, 2023.

[55] H. Yuan, X. Li, C. Zhou, Y. Li, K. Chen, and C. Loy. Open-Vocabulary SAM: Segment and Recognize Twenty-thousand Classes Interactively. In *ECCV*, pages 419–437, 2024.

[56] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and Comprehension of Unambiguous Object Descriptions. In *CVPR*, pages 11–20, 2016.

[57] N. Fiedler, M. Bestmann, and N. Hendrich. ImageTagger: An Open Source Online Platform for Collaborative Image Labeling. In *RoboCup*, pages 162–169, 2019.

[58] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *ICCV*, pages 10012–10022, 2021.

[59] A. Kumar, T. Kashiyama, H. Maeda, and Y. Sekimoto. Citywide Reconstruction of Cross-Sectional Traffic Flow from Moving Camera Videos. In *Big Data*, pages 1670–1678, 2021.

Appendix

A Task Details

Figure 5: Typical examples of the GRNR task. Left: single target. Center: multi-target. Right: no-target. The goal is to generate zero or more segmentation masks (shown in green). Unlike the RNR, the GRNR accommodates instructions that specify an arbitrary number of landmarks, including cases where multiple target regions exist or no target region exists.

Fig. 5 shows typical scenes from the Generalized Referring Navigable Regions (GRNR) task. The bounding boxes in the figure indicate the landmarks referenced in the instructions. In Fig. 5 (i-a), given the instruction "Pull up the left side of the right bicycle," the model is required to identify the target region and generate the mask depicted in green. Similarly, in Fig. 5 (i-b), because the instruction is "Park beside a walking pedestrian" and multiple pedestrians exist in the image, the model should generate a mask for each of the pedestrians. Finally, in Fig. 5 (i-c), the instruction "Stop behind the truck on the left" is provided but no truck exists, therefore the model should determine that there is no target region in the given image. Unlike the RNR task, the GRNR task involves instructions that specify any number of target regions, including multi-target and no-target instructions.

The input to this task is a front camera image and a navigation instruction, and the output is a binary indicator that represents whether at least one target region is present in the image, accompanied by a set of segmentation masks (with no masks generated if no target region is present).

B GRiN-Drive Details

Table 4: Comparison of datasets.

Dataset	Images	Region type	No	Single	Multi	Expression type
Refcocog [56]	26,711	Thing	–	✓	–	Phrase
gRefcoco [4]	19,992	Thing	✓	✓	✓	Phrase
Talk2carRegSeg [1]	10,016	Stuff	–	✓	–	Instruction
Reffer-Ksitti-V2 [15]	6,650	Thing	✓	✓	✓	Phrase
GRiN-Drive (ours)	17,114	Stuff	✓	✓	✓	Instruction

Most existing benchmarks lack coverage of no-target cases, which fundamentally limits their practicality. In contrast, as summarised in Table 4, the GRiN-Drive benchmark comprehensively covers single-, no-, and multi-target scenarios, serving as a more representative benchmark of real-world

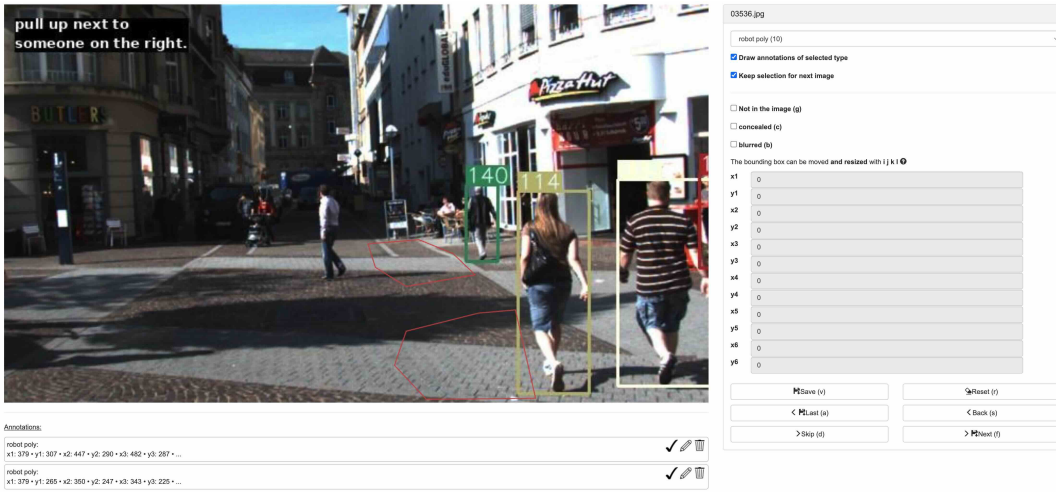


Figure 6: Annotation interface for multi-target samples in the GRiN-Drive benchmark. Annotators were instructed to provide polygons for an arbitrary number of target regions in the given image, corresponding to the navigation instruction.

scenarios. To evaluate the performance of models on the three distinct types of cases in the GRNR task, we constructed the novel GRiN-Drive benchmark based on the Talk2Car-RegSeg [1] and Refer-KITTI-V2 [16] datasets. The images from Talk2Car-RegSeg have a resolution of 1600×900, whereas those from Refer-KITTI-V2 were originally captured at 1280×384. To standardize the aspect ratio to 16:9, the Refer-KITTI-V2 images were cropped to 656×369, with the cropping centered on the lower part of the image. The segmentation masks of the multi-target samples in the GRiN-Drive benchmark were provided by 244 annotators, with an average of 29.07 samples per annotator. We constructed the GRiN-Drive benchmark following the procedure described below.

B.1 Single / No-target Samples

We constructed single-target and no-target samples based on the Talk2Car-RegSeg dataset. Given that it contains exclusively single-target samples, we created no-target samples using the following steps: (i) We randomly selected samples from the dataset and swapped their instructions. (ii) We used GPT-4o [50] to assess whether the landmark specified in each swapped instruction existed in the corresponding image. (iii) If GPT-4o indicated that the landmark was still present, we replaced the instruction again and repeated the assessment. (iv) Finally, we manually inspected the samples classified as containing no target region.

B.2 Multi-target Samples

To create multi-target samples, we leveraged the Refer-KITTI-V2 [16] dataset, which was designed for Referring Multi-Object Tracking tasks in urban driving scenes. We did not use the Talk2Car-RegSeg dataset to create multi-target samples because creating sample with a many-to-many relationship between landmarks and target regions using the Talk2Car-RegSeg dataset is challenging. This is mainly because the test set of the Talk2Car-RegSeg dataset includes samples that contain no landmarks in navigation instruction. Moreover, because these samples lack positional annotations for objects other than landmarks, the construction of a multi-landmark test set from the Talk2Car-RegSeg dataset would be impractical because of the high annotation costs. Therefore, we used the Refer-KITTI-V2 dataset as an alternative because it explicitly annotates multiple landmarks within a single image.

To create the multi-target samples for the GRiN-Drive benchmark, we first extracted relevant images from Refer-KITTI-V2. We selected frames from Refer-KITTI-V2 according to the following procedure: For each video, we identified frames containing multiple landmarks corresponding to noun phrases. Some landmarks may have been only partially visible, because we cropped wide-

format images from the original dataset for our analysis. To ensure the validity of landmarks, we only considered landmarks whose bounding boxes had more than half of their area within the image boundaries, thereby avoiding cases where objects were significantly truncated. To prevent redundancy and maintain diversity in the dataset, we excluded 20 subsequent frames adjacent to each selected frame for the same noun phrase instance.

Next, we generated navigation instructions for the extracted images based on Talk2Car-RegSeg instructions as follows: First, we leveraged MLLMs to create instruction templates from the original instructions in the Talk2Car-RegSeg dataset by replacing noun phrases that specify landmarks with <landmark> tags. For example, given a Talk2Car-RegSeg instruction such as "Stop next to the pedestrian on the right," the MLLM generated the following template: "Stop next to <landmark>pedestrian on the right</landmark>." Second, we converted the noun phrases from Refer-KITTI-V2 into their singular forms using MLLMs and verified the conversions manually. Third, we randomly replaced the <landmark> slots in the templates with singularized noun phrases to create the final navigation instructions.

Fig. 6 shows the annotation interface used in this study. Using the generated instruction-image pairs we asked annotators to identify the target regions for each sample. To assist annotators during the annotation process, the bounding boxes associated with noun phrases were displayed as visual references. For instructions referring to multiple landmarks, the annotators specified a target region that corresponds to each landmark separately using polygons with a minimum of six vertices. We used SoSci Survey[1] to manage the annotation procedure and Imagetagger [57] to collect the annotation of target regions. Finally, we manually removed inappropriate samples, such as cases with a significant overlap between target regions and landmarks or samples with unusually short response times, to enhance dataset quality.

C Implementation Details

C.1 GENNAV

Table 5 shows the experimental settings for GENNAV. In the Landmark Distribution Patch Module, we divided each region of 560×315 into patches and resized them to 224×224. For all other modules, we resized the original images to 224×224 to reduce the computational cost. To ensure a fair comparison, we evaluated the baseline method by Rufus et al., LAVT, and TNRSM at both 224×224 and an additional resolution of 640×640, which is equivalent to the resolution used in Landmark Distribution Patchification Module.

Table 5: Experimental settings of GENNAV

Epoch	100
Batch size	384
Learning rate	1×10^{-4}
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.98$)
Loss weights	$\lambda_{\text{pt}} = 3$

Our model had approximately 67.9M trainable parameters and 4.20T multiply-add operations. We trained our model on a GeForce RTX 4090 with 24 GB of GPU memory and an Intel Core i9-13900KF with 64 GB of RAM. The learning rate was warmed up during the first 5 epochs and then decayed by a factor of 0.1 after the 75th epoch. The training time for the proposed model was approximately 3 hours, and the inference time was approximately 31.3 milliseconds. We calculated the msIoU on the validation set every three epochs. To evaluate performance on the test set, we used the model that achieved the highest msIoU on the validation set.

C.2 Baselines

We used two groups of baseline methods: pixel-based methods (a method by Rufus et al. [1], LAVT [22], TNRSM [2], and GSVA-Vicuna-7B [5]) and MLLMs (Gemini [49], GPT-4o [50], and Qwen2-VL [51]). We selected Gemini, GPT-4o and Qwen2-VL because they are representative multimodal LLMs that have been pre-trained on large-scale datasets and have demonstrated outstanding performance on various vision-and-language tasks [49, 50, 51]. In our comparative experiments, we used eight baseline methods with the following experimental settings.

[1] https://www.soscisurvey.de/

Pixel-based methods (Rufus et al., LAVT, TNRSM, and GSVA-Vicuna-7B). We fine-tuned each model following the hyperparameter settings described in its respective paper. If no mask is generated for any pixel, the model is considered to have predicted a no-target.

MLLMs (Gemini, GPT-4o and Qwen2-VL). We conducted zero-shot evaluations under bounding box settings. The bounding box-based outputs are commonly recommended for object detection tasks for representative MLLMs, such as Gemini, GPT-4o, and Qwen2-VL. We conducted five experiments by varying the temperature parameter. To predict the existence of a target region and its corresponding bounding box, we used the following prompt: *"Given an image and a movement instruction, analyze if there is a target region for the movement. If it exists, identify the region by specifying a bounding box with two points in 2D coordinates. The top-left and bottom-right corners of the box that surrounds the target area. If there are multiple options, list them all. Return the response in the following JSON format: {"has_target": boolean, "bbox": [[[x1, y1], [x2, y2]], ... [[x1, y1], [x2, y2]]] or null}. The coordinates should be in pixels relative to the top-left corner of the image. If there is no target region, set 'bbox' to null"*. To select the above prompt, we conducted preliminary experiments using over ten prompts and selected the one yielding the best results. To compare performance under settings similar to GENNAV, we also conducted an experiment with polygon-based method using Qwen2-VL. We used the following prompt: *"Given an image and a movement instruction, analyze if there is a target region for the movement. If it exists, identify the region by specifying a 6 points polygon with 12 points in 2D coordinates. If there are multiple options, list them all. Return the response in the following JSON format: '{"has_target": boolean, "polygon": [[[x1, y1], [x2, y2], [x3, y3], [x4, y4], [x5, y5], [x6, y6]], ... [[x1, y1], [x2, y2], [x3, y3], [x4, y4], [x5, y5], [x6, y6]]] or null}'. The coordinates should be in pixels relative to the top-left corner of the image. If there is no target region, set 'polygon' to null."*

D Evaluation Metrics Details

We employed $\mathrm{P@}K$ because it is a standard metric for GRNR and RNR tasks. $\mathrm{P@}K$ counts the percentage of samples with IoU higher than the threshold K. $\mathrm{P@}K$ is defined as follows:

$$\begin{aligned} \mathrm{P@}K &= \frac{N_K}{N}, \\ N_K &= \sum_{i=1}^{N} \mathbb{1}\left[\mathrm{IoU}(\hat{y}_i, y_i) > K\right]. \end{aligned} \tag{5}$$

We set the threshold to 0.1 and 0.2 for $\mathrm{P@}K$. This is because setting the IoU threshold to 0.1 and 0.2 is considered appropriate in the GRNR task where human agreement substantially deviates from 1.0. As shown in Table 1, even at low thresholds such as $\mathrm{P@}0.1$ and $\mathrm{P@}0.2$, human performance was only 56.00% and 36.40%, respectively. This indicates that even humans struggle to predict consistent target regions. Moreover, we observe that even relatively simple models, when trained on data from a specific (in-domain) distribution, can sometimes surpass human performance. Such cases raise concerns about whether $\mathrm{P@}K$ remains a valid and informative metric, particularly when model performance exceeds that of humans. In fact, the baseline methods already outperform human performance at $\mathrm{P@}K$ with $K \geqq 0.3$. Given these observations, we consider that $\mathrm{P@}K$ at thresholds higher than 0.3 should be excluded from our evaluation because it no longer reflects meaningful or reliable distinctions in model capability.

We also evaluated the accuracy of predicting the existence of target regions. Accuracy is defined as follows:

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}. \tag{6}$$

E Additional Results

E.1 Additional Quantitative Results

Table 1 shows the quantitative results of the baseline methods, GENNAV and the human performance on the GRiN-Drive benchmark. The values in the table are the average and standard deviation over five trials. The "Type" and "Inf. Speed" columns specify the segmentation approach and inference speed of each method, respectively.

Table 6: Confusion matrix of target region existence for our method on the GRIN-Drive benchmark.

		Predicted target region existence	
		Positive	Negative
Ground truth target region existence	Positive	359	143
	Negative	81	175

The msIoU of GENNAV and the baseline methods are as follows, grouped by input resolution. In the 224×224 setting, the method by Rufus et al., LAVT, and TNRSM achieved msIoU scores of 35.44, 31.73, and 37.90, respectively. Under the 640×640 condition, GENNAV achieved an msIoU of 46.35, whereas the method by Rufus et al., LAVT, and TNRSM resulted in scores of 37.84, 34.09, and 22.84, respectively. In the high-resolution (1024×1024 and 1600×900) setting, GSVA and the MLLM baselines Gemini, GPT-4o, Qwen2-VL (bbox), and Qwen2-VL (polygon) yielded msIoU scores of 32.94, 6.98, 23.41, 24.06, and 12.16, respectively. These results show that GENNAV improved by 8.45 points over TNRSM (224×224), which achieved the highest msIoU among the methods.

Similarly, the $\mathrm{P@0.1}$ scores of the baseline methods by Rufus et al., LAVT, and TNRSM in the 224×224 setting were 21.53, 30.28, and 44.05, respectively. At the 640×640 resolution, the $\mathrm{P@0.1}$ scores of GENNAV and the baseline methods by Rufus et al., LAVT, and TNRSM were 11.48, 7.90, and 38.57, respectively. Moreover, at the 1024×1024 resolution, the $\mathrm{P@0.1}$ score of GSVA-Vicuna-7b was 17.26. Furthermore, the MLLM baselines, that is, Gemini, GPT-4o, Qwen2-VL (bbox), and Qwen2-VL (polygon), achieved $\mathrm{P@0.1}$ scores of 6.92, 5.04, 3.85, and 0.08, respectively. These results demonstrate that GENNAV outperformed the baseline methods in terms of $\mathrm{P@0.1}$, demonstrating a 5.55 point improvement over the best-performing baseline, TNRSM (224×224). Moreover, GENNAV achieved an accuracy of 75.41%, surpassing the baseline methods, including the MLLMs, with a 7.52 point improvement over the best-performing baseline, TNRSM (224×224).

Accordingly, the results can be summarized as follows: GENNAV outperformed the baseline methods in terms of msIoU, $\mathrm{P@0.1}$, $\mathrm{P@0.2}$ and accuracy. Moreover, the improvement achieved by GENNAV in terms of msIoU, $\mathrm{P@0.1}$, and accuracy was statistically significant ($p < 0.05$). Using high-resolution images with the baseline methods (the method by Rufus et al., LAVT, and TNRSM) did not lead to improved performance in terms of msIoU, $\mathrm{P@}K$, and accuracy. This limitation arose mainly because these methods used backbones that were fine-tuned on a resolution of 224×224, such as the Swin Transformer [58]. Consequently, they were not well-suited for processing high-resolution inputs, which hindered potential performance gains. Furthermore, the use of high-resolution images substantially increased the computational cost, which resulted in longer inference speeds across all methods compared to inputs with a resolution of 224×224.

Finally, we compared the inference speeds of all methods. The inference speed per sample for GENNAV was 31.31 ms. For comparison, the inference speeds of the baseline methods were as follows. In the 224×224 setting, the method by Rufus et al., LAVT and TNRSM took 39.87 ms, 9.29 ms and 502.69 ms, respectively. Under the 640×640 resolution setting, the method by Rufus et al., LAVT, and TNRSM took 65.00 ms, 11.51 ms, and 503.68 ms, respectively. By contrast, MLLM-based methods exhibited significantly slower inference speeds: Gemini, GPT-4o, Qwen2-VL (bbox), and Qwen2-VL (polygon) took 1793.68 ms, 3525.68 ms, 1768.99 ms, and 1771.41 ms, respectively. Overall, GENNAV achieved faster inference speed than the baseline methods except LAVT. Although GENNAV was slower than LAVT, it achieved substantially higher performance on other metrics.

We conducted a subject experiment to evaluate human performance on the GRiN-Drive test set. In total, we recruited 11 participants (aged 21-24) who all had technical backgrounds. First, we asked the participants to determine whether any target region specified by the navigation instruction was present in the image. If one or more target regions were identified by the annotators, they provided these regions by drawing polygons. The human performance in terms of msIoU, $\mathrm{P@0.1}$, $\mathrm{P@0.2}$, and accuracy were 56.08, 56.00, 36.40, and 88.00, respectively, as shown in Table 1.

(i)

$\boldsymbol{x}_{\text{inst}}$: "Pull over where that man is."

(ii)

$\boldsymbol{x}_{\text{inst}}$: "Slow down and let that gold car pass in front of us."

Figure 7: Qualitative results of the proposed method and baseline methods on the GRiN-Drive benchmark. Columns (a), (b), (c), and (d) show the predictions by LAVT, TNRSM, Qwen2-VL (bbox) and GENNAV, respectively. The green and red regions indicate the predicted and ground-truth regions, respectively, while the yellow region represents the overlap between the predicted and ground-truth regions.

(a) (b) (c) (d)

$\boldsymbol{x}_{\text{inst}}$: "Pull up next to the pedestrian on our right close to the tree."

Figure 8: Qualitative results of a failed case. Columns (a), (b), (c), and (d) show the predictions made by LAVT [22], TNRSM [2], Qwen2-VL [51] (bbox), and GENNAV, respectively. The green regions indicate the predicted segmentation.

Table 6 presents the confusion matrix for our method GENNAV using GRiN-Drive test set. There were 359, 175, 81, and 143 samples corresponding to TP, TN, FP, and FN, respectively.

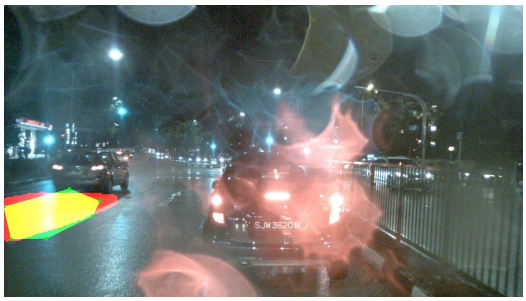
E.2 Additional Qualitative Results

Figs. 9 (i) illustrates an example, where the $\boldsymbol{x}_{\text{inst}}$ is, "Pull over where that man is." In this example, the model is expected to generate masks on the road in front of the man on the right side of the image. The baseline methods LAVT and TNRSM generated the inappropriate masks on regions around the truck and Qwen2-VL (bbox) inappropriately predicted the central region of the road. By contrast, our method was able to generate an appropriate mask on the region next to the man. Figs. 9 (ii) demonstrates a no-target case with $\boldsymbol{x}_{\text{inst}}$: "Slow down and let that gold car pass in front of us." The appropriate prediction should be a "no-target", because the gold car does not exist in the scene. GENNAV successfully predicted this sample as a "no-target." The baseline methods were unable to appropriately predict the absence of landmarks, which resulted in mask generation in inappropriate regions of the road

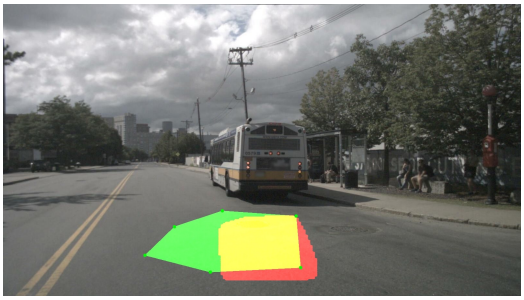
Fig. 8 shows the qualitative result of a failed case. In the figure, Columns (a), (b), (c), and (d) show the prediction made by LAVT, TNRSM, Qwen2-VL (bbox) and GENNAV, respectively. The green regions indicate the predicted segmentation. Fig. 8 presents a failed example in a no-target sample. In this example, the instruction was "Pull up next to the pedestrian on our right close to the tree." All methods including GENNAV predicted the region around the tree. However, because there are no pedestrians around the tree, predicting a "no-target" is appropriate. GENNAV was influenced by the word "tree," thereby leading to the incorrect prediction of the surrounding region.

Fig. 9 provide additional success examples of GENNAV on the GRiN-Drive benchmark. As shown in Fig. 9 (vii) and Fig. 9 (viii), GENNAV is capable of generating different predictions for the same image when provided with different instructions.

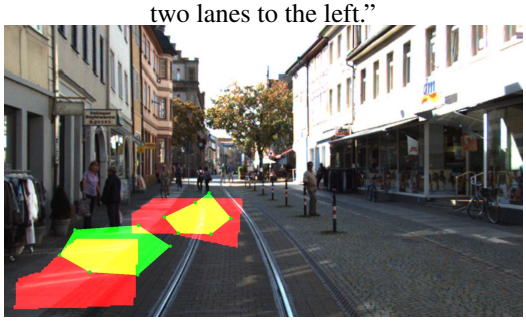
(i) $\boldsymbol{x}_{\text{inst}}$: "Slow down near that person."



(ii) $\boldsymbol{x}_{\text{inst}}$: "Switch lanes and follow that gray car two lanes to the left."

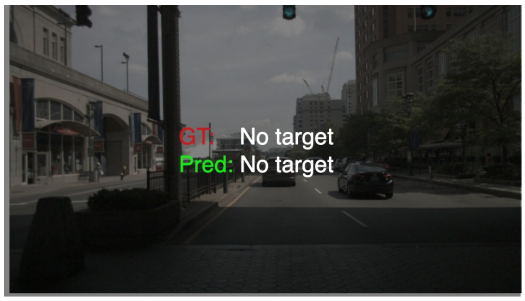


(iii) $\boldsymbol{x}_{\text{inst}}$: "Slow down a bit, that bus might pull out into traffic."

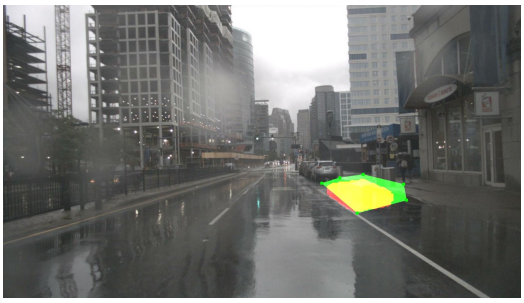


(iv) $\boldsymbol{x}_{\text{inst}}$: "Stop near those to the left."

(v) $\boldsymbol{x}_{\text{inst}}$: "Park behind that truck that is way up ahead."



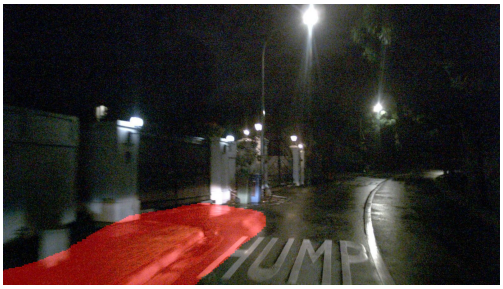
(vi) $\boldsymbol{x}_{\text{inst}}$: "Pull up in front of that gate left of the green trash bin."



(vii) $\boldsymbol{x}_{\text{inst}}$: "Pull up to that girl on the right closer to the road."

(viii) $\boldsymbol{x}_{\text{inst}}$: "Park behind that truck on the right side of the road."

Figure 9: Qualitative results of the proposed method in the GRiN-Drive benchmark. The green and red regions indicate the predicted and ground-truth regions, respectively, while the yellow region represents the overlap between the predicted and ground-truth regions.



(a)

$\boldsymbol{x}_{\text{inst}}$: "Park on the grass in front of the traffic cone."

(b)

$\boldsymbol{x}_{\text{inst}}$: "Stay behind that car in front of us."

Figure 10: Qualitative analysis of failure cases of the proposed method categorized as Reduced Visibility Conditions. Regarding limited or poor visibility conditions, such as those caused by inadequate lighting, reflections from rain, or cloudy weather, errors may arise.

F Error Analysis

To investigate the limitations of GENNAV, we analyzed cases where the method did not perform as expected. In this study, failures are defined in two Cases: (i) Existence Prediction: The prediction regarding the existence of a target region is incorrect (FP or FN). (ii) Polygon Generation: The generated mask does not overlap with the ground truth mask of target regions (i.e., IoU = 0). GENNAV failed in 187 samples for Case (i) and 44 samples for Case (ii) in the test set.

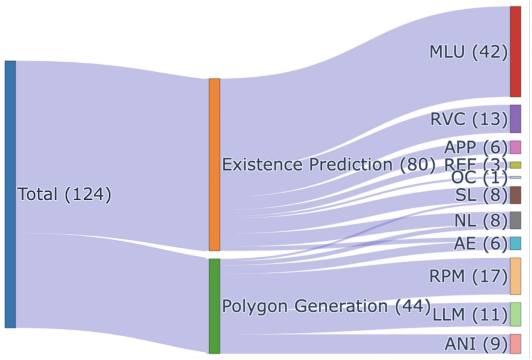


Figure 11: Categorization of failure modes. Error analysis results categorized into Cases (i) Existence Prediction and (ii) Polygon Generation.

Fig 11 shows the results of the error analysis for Cases (i) Existence Prediction and (ii) Polygon Generation. We randomly selected 40 samples each for false negative sample and true negative sample groups for Case (i). We classified them into the following eight categories for Case (i): Multimodal language understanding for landmarks (MLU): This refers to cases where the model incorrectly predicts the existence of landmarks that do not actually exist in the given image.

Appearance misunderstanding of landmarks (AML). This category includes cases where errors in existence prediction occurred because of the misrecognition of appearance (e.g., although the instruction specifies "next to the yellow car," the model incorrectly identified the target region as being next to the black car).

Reduced visibility conditions (RVC). This category encompasses errors that arise from limited or poor visibility conditions, such as those caused by inadequate lighting, reflections from rain, or cloudy weather. Fig. 10 shows a sample categorized as reduced visibility conditions. In each case, poor visual conditions are one of the factors that cause GENNAV to predict "no-target" despite the existence of a target region.

Referring-expression misinterpretation (REF). This category covers cases where visual information and navigation instruction sentences are interpreted incorrectly, particularly because of the misinterpretation of referring expressions.

Small landmark (SL). This involves cases where the landmark is too small or visually subtle, which causes the model to misunderstand the existence of target regions.

No landmark (NL). This category includes cases where errors arise because the instruction lacks any landmarks, which makes it difficult for the model to ground the action appropriately (e.g., vague instructions such as "turn left" or "stay here" without specifying a landmark).

Annotation error (AE). This category includes cases of discrepancies or mistakes in the annotation process itself, which can lead to misleading ground-truth data and subsequent model errors.

Occlusion (OC). This refers to cases where the target regions or landmarks are occluded, or located in visually accessible but physically unnavigable regions—such as across a guardrail or behind a transparent barrier.

Moreover, for Case (ii), we added three additional categories and classified the data accordingly.

Relative position misunderstanding (RPM). This category includes cases where the model fails to understand the positional relationship between the landmark(s) and the intended target region appropriately (e.g., misunderstanding "in front of" vs. "behind").

Landmark location misunderstanding (LLM). This refers to cases where the model misunderstands the location of one or more landmarks, leading to inaccurate polygon generation.

Ambiguous navigation instruction (ANI). This refers to cases where the navigation instructions are ambiguous, because they lack sufficient clarity or specificity to uniquely identify the target region.

Consequently, the model predicts a mask that does not overlap with the ground-truth target region, not because of incorrect reasoning, but because the ambiguity in the instruction leads to multiple plausible candidate regions.

As shown in Table 11, the main bottleneck in Case (i) was MLU, presumably because of an insufficient semantic understanding of landmarks. A possible solution to address this issue is to overlay masks generated by more robust semantic segmentation models (e.g., Grounded SAM [52], Detic [53], Grounding DINO [54] and Open-Vocabulary SAM [55]) onto the input images before handling them into LAPM. This approach aims to enhance the semantic understanding of landmarks and thereby mitigate the bottleneck in MLU. By contrast, the main bottleneck for Case (ii) was RPM, which was likely to have arisen from an inadequate understanding of the 3D structure of environments, including vehicles. A possible solution is to incorporate tasks such as vehicle orientation classification [59] into the pre-training process. These additions would strengthen the model's capability to understand 3D structural representations, thereby alleviating RPM-related limitations.

G Details of Real-World Experiments

G.1 Experimental Setup Details

We used GPT-4o to generate navigation instructions. Specifically, we used Grounding DINO [54] to detect candidate landmarks based on the categories defined in the Talk2Car dataset, and generated visual inputs by overlaying bounding boxes on the identified landmarks. We then provided these images as input to GPT-4o, along with prompts instructing it to refer to detected landmarks when generating navigation instructions. To evaluate whether the generated instructions incorporated the designated landmarks appropriately, we conducted manual inspections under both single-target and multi-target conditions. Additionally, for the no-target condition, we constructed samples by swapping instructions across different samples, following the approach used for the GRiN-Drive benchmark, to simulate scenarios in which no valid target landmarks were present. In these cases, we manually verified that the generated instructions did not reference any specific landmarks inappropriately. The instructions typically included referring expressions grounded in the visualized landmarks and described navigation tasks for directing a mobile agent to a designated destination.

G.2 Additional Quantitative Results

Table 7 shows the overall quantitative comparison between GENNAV and baseline methods in the real-world experiment. The values in the table are the average and standard deviation over five trials. The "Type" column specifies the segmentation approach of each method. The msIoU of GENNAV and the baseline methods were as follows, grouped by input resolution:

Table 7: Quantitative comparison between the proposed method and baseline methods in the real-world experiment. The best score for each metric is in bold. The "Type" column specify the segmentation approach of each method.

Method	Resolution	Type	msIoU [%]↑	P@0.1 [%]↑	P@0.2 [%]↑	Acc. [%]↑
Rufus et al. [1]	224×224	pixel	25.95 ±3.96	8.50 ±2.60	5.83 ±3.28	41.83 ±8.09
LAVT [22]	224×224	pixel	22.84 ±3.35	10.00 ±3.06	6.00 ±2.05	55.00 ±5.17
TNRSM [2]	224×224	pixel	23.11 ±5.88	24.25 ±4.11	13.25 ±4.56	56.83 ±5.08
Rufus et al. [1]	640×640	pixel	21.68 ±5.95	3.50 ±2.97	2.00 ±2.54	40.42 ±6.02
LAVT [22]	640×640	pixel	30.44 ±1.63	11.73 ±10.69	5.00 ±5.80	32.67 ±19.74
TNRSM [2]	640×640	pixel	28.94 ±4.98	7.25 ±12.10	4.50 ±8.69	47.43 ±18.60
Gemini [49]	1600×900	bbox	4.81 ±0.40	4.25 ±0.69	3.00 ±0.69	61.67 ±1.44
Qwen2-VL [51]	1600×900	bbox	20.48 ±2.33	5.75 ±2.59	1.50 ±1.05	66.33 ±3.36
Qwen2-VL [51]	1600×900	polygon	11.17 ±0.46	0.00 ±0.00	0.00 ±0.00	50.33 ±0.75
GENNAV (ours)	640×640	polygon	**34.32** ±2.97	**28.75** ±5.08	**16.75** ±2.27	**67.50** ±2.95

In the 224×224 setting, the method by Rufus et al. [1], LAVT, and TNRSM achieved msIoU scores of 25.95, 22.84, and 23.11, respectively. Under the 640×640 condition, GENNAV achieved an msIoU of 34.32, whereas the method by Rufus et al., LAVT, and TNRSM achieved scores of 21.68, 30.44, and 28.94, respectively. Notably, GENNAV outperformed the MLLMs that had high-resolution inputs. These results demonstrate that GENNAV achieved the highest msIoU among the baseline methods.

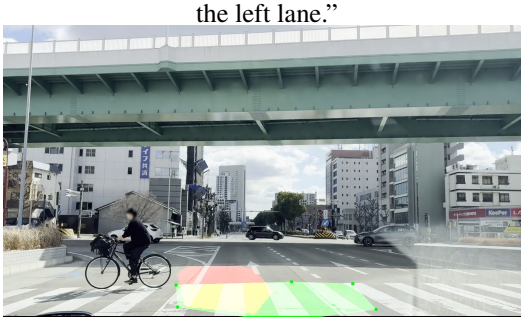
Similarly, the P@0.1 scores of the baseline methods by Rufus et al., LAVT, and TNRSM in the 224×224 setting were 8.50, 10.00, and 24.25, respectively. At the 640×640 resolution, the P@0.1 scores of GENNAV and the method by Rufus et al., LAVT, and TNRSM were 28.75, 3.50, 11.73, and 7.25, respectively. Furthermore, the MLLM baselines, that is, Gemini, Qwen2-VL (bbox), and Qwen2-VL (polygon), achieved P@0.1 scores of 4.25, 5.75, and 0.00, respectively. These results demonstrate that GENNAV outperformed the baseline methods in terms of P@0.1, demonstrating a 4.5 point improvement over the best-performing baseline, TNRSM (224×224). Overall, GENNAV achieved higher accuracy (67.50%) than the baseline methods in the real-world experiments conducted in a zero-shot manner.

G.3 Additional Qualitative Results

Fig. 12 provides additional success examples of GENNAV on the real-world experiment. These results indicate that GENNAV can be effectively integrated into real-world systems. Furthermore, as shown in Fig. 12 (vii) and Fig. 12 (viii), GENNAV is capable of generating different predictions for the same image when provided with different instructions.

(i) $\boldsymbol{x}_{\text{inst}}$: "Park next to the black car moving on the left lane."

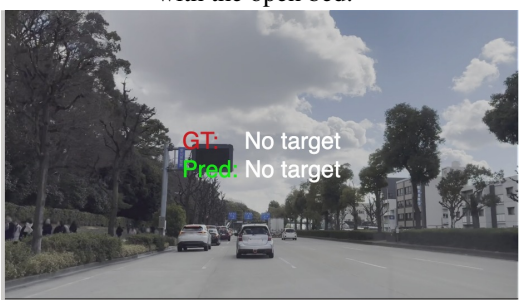
(ii) $\boldsymbol{x}_{\text{inst}}$: "Stop behind a moving vehicle in the traffic lane."



(iii) $\boldsymbol{x}_{\text{inst}}$: "Stop around the cyclist crossing the street."

(iv) $\boldsymbol{x}_{\text{inst}}$: "Stop to the right of the large truck with the open bed."

(v) $\boldsymbol{x}_{\text{inst}}$: "Park behind that truck that is way up ahead."



(vi) $\boldsymbol{x}_{\text{inst}}$: "Stop to the left of the silver vehicle in the left lane."

(vii) $\boldsymbol{x}_{\text{inst}}$: "Park near the person standing near the intersection."

(viii) $\boldsymbol{x}_{\text{inst}}$: "Please follow the vehicle in front."

Figure 12: Additional qualitative results of the proposed method in the real-world experiments. The green and red regions indicate the predicted and ground-truth regions, respectively, while the yellow region represents the overlap between the predicted and ground-truth regions.