

Appendix

A Task Details



Figure 5: Typical examples of the GRNR. 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

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 [A1] and Refer-KITTI-V2 [A2] 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

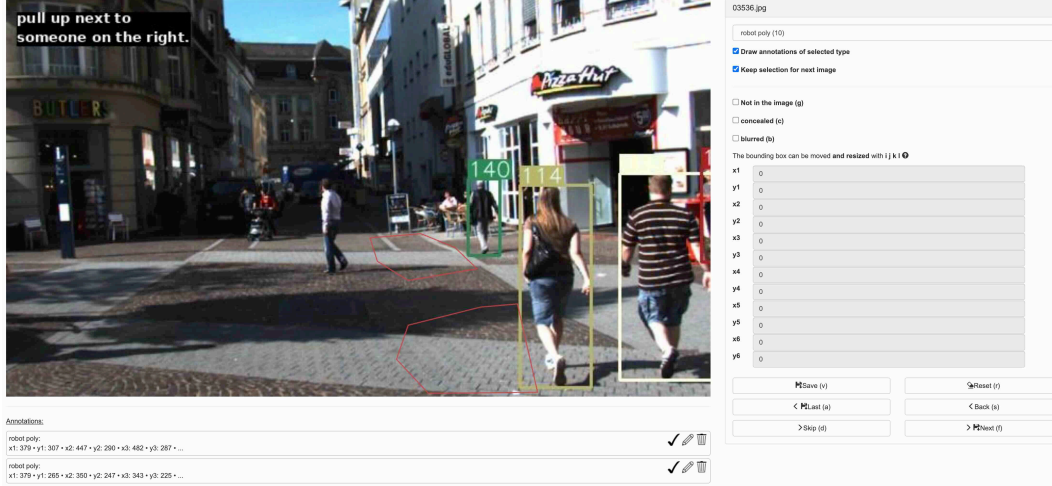


Figure 6: Annotation interface for multi-target samples in the GRiN-Drive benchmark. Annotators were instructed to provide polygons for an arbitrary number of target regions in the given image, corresponding to the navigation instruction.

used GPT-4o [A3] 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 [A2] 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 Survey¹ to manage the annotation procedure and Imagetagger [A4] 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 4 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 4: 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	$\lambda_{pt} = 3$

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. [A1], LAVT [A5], and TNRSM [A6]) and MLLMs (Gemini [A7], GPT-4o [A3], and Qwen2-VL [A8]). 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 [A7, 3, 8]. In our comparative experiments, we used seven baseline methods with the following experimental settings.

Pixel-based methods (Rufus et al., LAVT and TNRSM). 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

¹<https://www.soscisurvey.de/>

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.”

137 D Evaluation Metrics Details

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

$$P@K = \frac{N_K}{N}, \quad (5)$$

$$N_K = \sum_{i=1}^N \mathbb{1}[\text{IoU}(\hat{y}_i, y_i) > K].$$

140 We set the threshold to 0.1 and 0.2 for $P@K$. This is because setting the IoU threshold to 0.1 and
141 0.2 is considered appropriate in the GRNR task where human agreement substantially deviates from
142 1.0. As shown in Table 1, even at low thresholds such as $P@0.1$ and $P@0.2$, human performance
143 was only 56.00% and 36.40%, respectively. This indicates that even humans struggle to predict
144 consistent target regions. Moreover, we observe that even relatively simple models, when trained
145 on data from a specific (in-domain) distribution, can sometimes surpass human performance. Such
146 cases raise concerns about whether $P@K$ remains a valid and informative metric, particularly when
147 model performance exceeds that of humans. In fact, the baseline methods already outperform human
148 performance at $P@K$ with $K \geq 0.3$. Given these observations, we consider that $P@K$ at thresholds
149 higher than 0.3 should be excluded from our evaluation because it no longer reflects meaningful or
150 reliable distinctions in model capability.

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

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (6)$$

153 E Additional Results

154 E.1 Additional Quantitative Results

155 Table 1 shows the quantitative results of the baseline methods, GENNAV and the human perfor-
156 mance on the GRiN-Drive benchmark. The values in the table are the average and standard devia-
157 tion over five trials. The “Type” and “Inf. Speed” columns specify the segmentation approach and
158 inference speed of each method, respectively.

159 The msIoU of GENNAV and the baseline methods are as follows, grouped by input resolution. In the
160 224×224 setting, the method by Rufus et al., LAVT, and TNRSN achieved msIoU scores of 35.44,
161 31.73, and 37.90, respectively. Under the 640×640 condition, GENNAV achieved an msIoU of
162 46.35, whereas the method by Rufus et al., LAVT, and TNRSN resulted in scores of 37.84, 34.09,
163 and 22.84, respectively. In the high-resolution (1600×900) setting, the MLLM baselines Gemini,
164 GPT-4o, Qwen2-VL (bbox), and Qwen2-VL (polygon) yielded msIoU scores of 6.98, 23.41, 24.06,
165 and 12.16, respectively. These results show that GENNAV improved by 8.45 points over TNRSN
166 (224×224), which achieved the highest msIoU among the methods.

167 Similarly, the $P@0.1$ scores of the baseline methods by Rufus et al., LAVT, and TNRSN in the
168 224×224 setting were 21.53, 30.28, and 44.05, respectively. At the 640×640 resolution, the $P@0.1$

Table 5: Confusion matrix regarding the existence of the target regions for our method using the GRiN-Drive benchmark.

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

scores of GENNAV and the baseline methods by Rufus et al., LAVT, and TNRSM were 11.48, 7.90, and 38.57, respectively. Furthermore, the MLLM baselines, that is, Gemini, GPT-4o, Qwen2-VL (bbox), and Qwen2-VL (polygon), achieved 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 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, P@0.1, P@0.2 and accuracy. Moreover, the improvement achieved by GENNAV in terms of msIoU, 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, 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 [A9]. 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, P@0.1, P@0.2, and accuracy were 56.08, 56.00, 36.40, and 88.00, respectively, as shown in Table 1.

Table 5 presents the confusion matrix for our method GENNAV using GRiN-Drive test set. There were 359, 175, 81, and 143 for TP, TN, FP and FN samples, respectively.

E.2 Additional Qualitative Results

Figs. 9 (i) illustrates an example, where the x_{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 x_{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

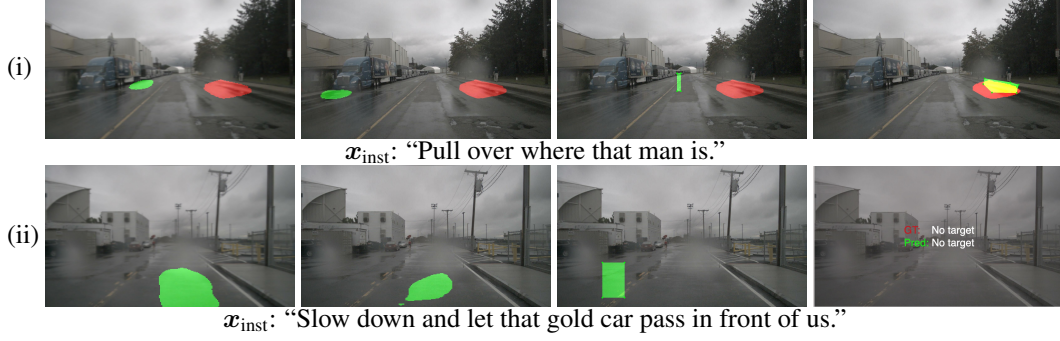


Figure 7: Qualitative results of the proposed method and baseline methods on the GRiN-Drive benchmark. Columns (a), (b), (c) and (d) show the prediction by LAVT, TNRS, Qwen2-VL (bbox) and GENNAV, respectively. The green and red regions indicate the predicted and ground-truth regions, respectively, while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



Figure 8: Qualitative results of a failed case. Columns (a), (b), (c), and (d) show the prediction made by LAVT [A5], TNRS [A6], Qwen2-VL [A8] (bbox), and GENNAV, respectively. The green regions indicate the predicted segmentation.

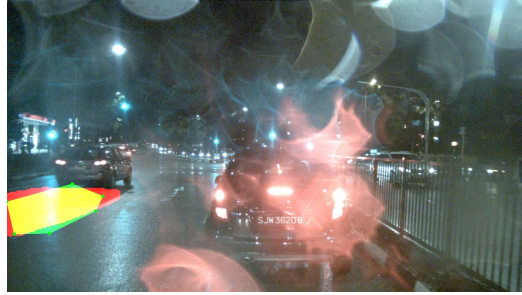
213 to appropriately predict the absence of landmarks, which resulted in an inappropriate generation of
 214 masks on the region on the road.

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

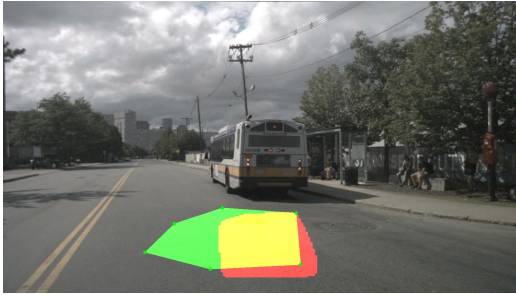
222 Fig. 9 provide additional success examples of GENNAV on the GRiN-Drive benchmark. As shown
 223 in Fig. 9 (vii) and Fig. 9 (viii), GENNAV is capable of generating different predictions for the same
 224 image when provided with different instructions.



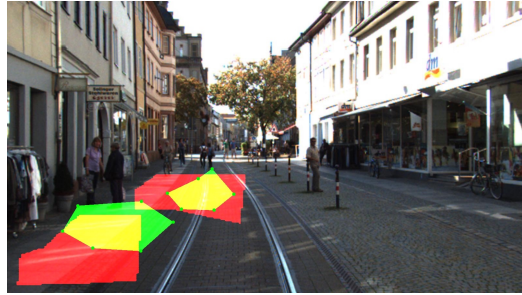
(i) x_{inst} : “Slow down near that person.”



(ii) x_{inst} : “Switch lanes and follow that gray car two lanes to the left.”



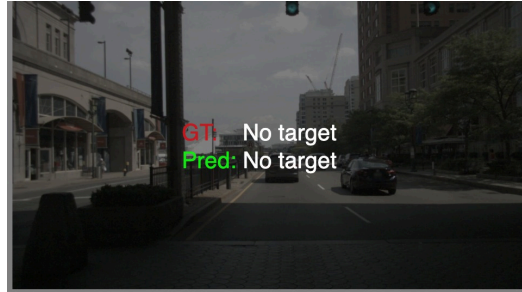
(iii) x_{inst} : “Slow down a bit, that bus might pull out into traffic.”



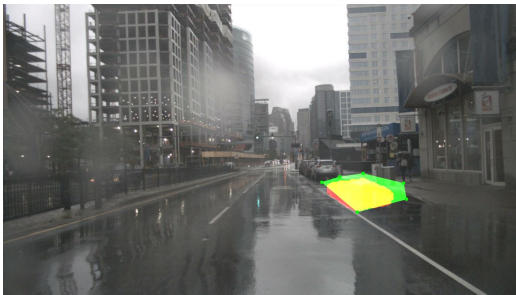
(iv) x_{inst} : “Stop near those to the left.”



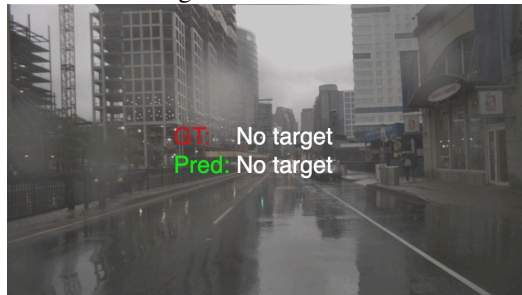
(v) x_{inst} : “Park behind that truck that is way up ahead.”



(vi) x_{inst} : “Pull up in front of that gate left of the green trash bin.”



(vii) x_{inst} : “Pull up to that girl on the right closer to the road.”



(viii) x_{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.



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.

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.

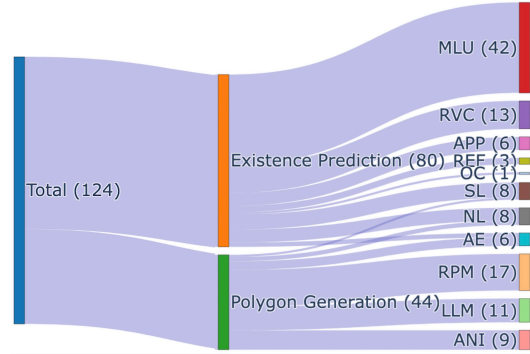


Figure 11: Categorization of failure modes. Error analysis results categorized into Cases (i) Existence Prediction and (ii) Polygon Generation.

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

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

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

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

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

271 **Landmark location misunderstanding (LLM).** This refers to cases where the model misunder-
272 stands the location of one or more landmarks, leading to inaccurate polygon generation.

273 **Ambiguous navigation instruction (ANI).** This refers to cases where the navigation instructions
274 are ambiguous, because they lack sufficient clarity or specificity to uniquely identify the target re-
275 gion.

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

279 As shown in Table 11, the main bottleneck in Case (i) was MLU, presumably because of an insuf-
280 ficient semantic understanding of landmarks. A possible solution to address this issue is to overlay
281 masks generated by more robust semantic segmentation models (e.g., Grounded SAM [A10], De-
282 tic [A11], Grounding DINO [A12] and Open-Vocabulary SAM [A13]) onto the input images before
283 handling them into LAPM. This approach aims to enhance the semantic understanding of land-
284 marks and thereby mitigate the bottleneck in MLU. By contrast, the main bottleneck for Case (ii)
285 was RPM, which was likely to have arisen from an inadequate understanding of the 3D structure of
286 environments, including vehicles. A possible solution is to incorporate tasks such as vehicle orienta-
287 tion classification [A14] into the pre-training process. These additions would strengthen the model’s
288 capability to understand 3D structural representations, thereby alleviating RPM-related limitations.

289 G Details of Real-World Experiments

290 G.1 Experimental Setup Details

291 We used GPT-4o to generate navigation instructions. Specifically, we used Grounding DINO [A12]
292 to detect candidate landmarks based on the categories defined in the Talk2Car dataset, and gener-
293 ated visual inputs by overlaying bounding boxes on the identified landmarks. We then provided
294 these images as input to GPT-4o, along with prompts instructing it to refer to detected landmarks
295 when generating navigation instructions. To evaluate whether the generated instructions incorpo-
296 rated the designated landmarks appropriately, we conducted manual inspections under both single-
297 target and multi-target conditions. Additionally, for the no-target condition, we constructed samples
298 by swapping instructions across different samples, following the approach used for the GRiN-Drive
299 benchmark, to simulate scenarios in which no valid target landmarks were present. In these cases,
300 we manually verified that the generated instructions did not reference any specific landmarks in-
301 appropriately. The instructions typically included referring expressions grounded in the visualized
302 landmarks and described navigation tasks for directing a mobile agent to a designated destination.

303 G.2 Additional Quantitative Results

304 Table 6 shows the overall quantitative comparison between GENNAV and baseline methods in the
305 real-world experiment. The values in the table are the average and standard deviation over five trials.

Table 6: 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. [A1]	224×224	pixel	25.95 ±3.96	8.50 ±2.60	5.83 ±3.28	41.83 ±8.09
LAVT [A5]	224×224	pixel	22.84 ±3.35	10.00 ±3.06	6.00 ±2.05	55.00 ±5.17
TNRSM [A6]	224×224	pixel	23.11 ±5.88	24.25 ±4.11	13.25 ±4.56	56.83 ±5.08
Rufus et al. [A1]	640×640	pixel	21.68 ±5.95	3.50 ±2.97	2.00 ±2.54	40.42 ±6.02
LAVT [A5]	640×640	pixel	30.44 ±1.63	11.73 ±10.69	5.00 ±5.80	32.67 ±19.74
TNRSM [A6]	640×640	pixel	28.94 ±4.98	7.25 ±12.10	4.50 ±8.69	47.43 ±18.60
Gemini [A7]	1600×900	bbox	4.81 ±0.40	4.25 ±0.69	3.00 ±0.69	61.67 ±1.44
Qwen2-VL [A8]	1600×900	bbox	20.48 ±2.33	5.75 ±2.59	1.50 ±1.05	66.33 ±3.36
Qwen2-VL [A8]	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

306 The "Type" column specifies the segmentation approach of each method. The msIoU of GENNAV
307 and the baseline methods were as follows, grouped by input resolution:

308 In the 224×224 setting, the method by Rufus et al. [A1], LAVT, and TNRSM achieved msIoU
309 scores of 25.95, 22.84, and 23.11, respectively. Under the 640×640 condition, GENNAV achieved
310 an msIoU of 34.32, whereas the method by Rufus et al., LAVT, and TNRSM achieved scores of
311 21.68, 30.44, and 28.94, respectively. Notably, GENNAV outperformed the MLLMs that had high-
312 resolution inputs. These results demonstrate that GENNAV achieved the highest msIoU among the
313 baseline methods.

314 Similarly, the P@0.1 scores of the baseline methods by Rufus et al., LAVT, and TNRSM in the
315 224×224 setting were 8.50, 10.00, and 24.25, respectively. At the 640×640 resolution, the P@0.1
316 scores of GENNAV and the method by Rufus et al., LAVT, and TNRSM were 28.75, 3.50, 11.73,
317 and 7.25, respectively. Furthermore, the MLLM baselines, that is, Gemini, Qwen2-VL (bbox), and
318 Qwen2-VL (polygon), achieved P@0.1 scores of 4.25, 5.75, and 0.00, respectively. These results
319 demonstrate that GENNAV outperformed the baseline methods in terms of P@0.1, demonstrating
320 a 4.5 point improvement over the best-performing baseline, TNRSM (224×224). Overall, GEN-
321 NAV achieved higher accuracy (67.50%) than the baseline methods in the real-world experiments
322 conducted in a zero-shot manner.

323 G.3 Additional Qualitative Results

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



(i) x_{inst} : “Park next to the black car moving on the left lane.”



(ii) x_{inst} : “Stop behind a moving vehicle in the traffic lane.”



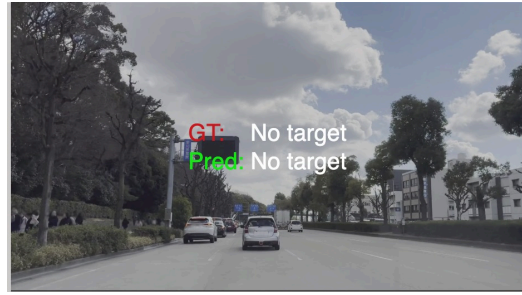
(iii) x_{inst} : “Stop around the cyclist crossing the street.”



(iv) x_{inst} : “Sstop to the right of the large truck with the open bed.”



(v) x_{inst} : “Park behind that truck that is way up ahead.”



(vi) x_{inst} : “Stop to the left of the silver vehicle in the left lane.”



(vii) x_{inst} : “Park near the person standing near the intersection.”



(viii) x_{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.

References

- [A1] N. Rufus, K. Jain, K. Nair, V. Gandhi, and M. Krishna. Grounding Linguistic Commands to Navigable Regions. In *IROS*, pages 8593–8600, 2021.
- [A2] Y. Zhang, D. Wu, W. Han, and X. Dong. Bootstrapping Referring Multi-Object Tracking. *arXiv preprint arXiv:2406.05039*, 2024.
- [A3] J. Achiam, S. Adler, S. Agarwal, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.
- [A4] N. Fiedler, M. Bestmann, and N. Hendrich. ImageTagger: An Open Source Online Platform for Collaborative Image Labeling. In *RoboCup*, pages 162–169, 2019.
- [A5] 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.
- [A6] N. Hosomi, S. Hatanaka, Y. Iioka, W. Yang, K. Kuyo, T. Misu, K. Yamada, and K. Sugiyura. Trimodal Navigable Region Segmentation Model: Grounding Navigation Instructions in Urban Areas. *IEEE RA-L*, 9(5):4162–4169, 2024.
- [A7] 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.
- [A8] 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.
- [A9] 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.
- [A10] 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.
- [A11] 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.
- [A12] 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.
- [A13] 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.
- [A14] A. Kumar, T. Kashiyaama, H. Maeda, and Y. Sekimoto. Citywide Reconstruction of Cross-Sectional Traffic Flow from Moving Camera Videos. In *Big Data*, pages 1670–1678, 2021.