3EED: Ground Everything Everywhere in 3D

Rong Li[1,*], Yuhao Dong[2,*], Tianshuai Hu[3,*], Ao Liang[4,*], Youquan Liu[5,*], Dongyue Lu[4,*], Liang Pan[6], Lingdong Kong[4,†], Junwei Liang[1,3,‡], Ziwei Liu[2,‡]

[1]HKUST(GZ) [2]NTU [3]HKUST [4]NUS [5]FDU [6]Shanghai AI Laboratory

[*]Equal Contributions [†]Project Lead [‡]Corresponding Authors

Dataset & Toolkit: `project-3eed.github.io`

Figure 1: **Multi-modal, multi-platform 3D grounding from 3EED**. Given a scene and a structured natural language expression, the task is to localize the referred object in 3D space. Our dataset captures diverse embodied viewpoints from `Vehicle`, `Drone`, `Quadruped` platforms, presenting unique challenges in spatial reasoning, scene analysis, and cross-platform 3D generalization.

Abstract

Visual grounding in 3D is the key for embodied agents to localize language-referred objects in open-world environments. However, existing benchmarks are limited to indoor focus, single-platform constraints, and small scale. We introduce **3EED**, a *multi-platform*, *multi-modal* 3D grounding benchmark featuring RGB and LiDAR data from **vehicle**, **drone**, and **quadruped** platforms. We provide over 134,000 objects and 25,000 validated referring expressions across diverse outdoor scenes – **10× larger** than existing datasets. We develop a scalable annotation pipeline combining vision-language model prompting with human verification to ensure high-quality spatial grounding. To support cross-platform learning, we propose platform-aware normalization and cross-modal alignment techniques, and establish benchmark protocols for in-domain and cross-platform evaluations. Our findings reveal significant performance gaps, highlighting the challenges and opportunities of generalizable 3D grounding. The 3EED dataset and benchmark toolkit are released to advance future research in language-driven 3D embodied perception.

Submitted to the 39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks. Do not distribute.

Table 1: **Summary of outdoor 3D grounding benchmarks.** We compare key features from aspects including: [1]**Platform** (`Vehicle`, `Drone`, `Quadruped`), [2]**Area Coverage**, and [3]**Statistics**. Our dataset exhibits advantages on platform diversity, large collections of LiDAR (**L**) and camera (**C**) scenes (**Sce.**), 3D objects (**Obj.**), referring expressions (**Expr.**), and rich elevation variations (**Elev.**).

Dataset	Sensor	Platform			Scene Coverage	Statistics			
						#Sce.	#Obj.	#Expr.	#Elev.
Mono3DRefer [93]	C	✓	✗	✗	140m × 140m	2,025	8,228	41,140	42.8m
KITTI360Pose [35]	L	✓	✗	✗	140m × 140m	-	14,934	43,381	42.8m
CityRefer [55]	L	✗	✓	✗	-	-	5,866	35,196	-
STRefer [43]	L + C	✓	✗	✗	60m × 60m	662	3,581	5,458	-
LifeRefer [43]	L + C	✓	✗	✗	60m × 60m	3,172	11,864	25,380	-
Talk2LiDAR [51]	L + C	✓	✗	✗	140m × 140m	6,419	-	59,207	48.6m
Talk2Car-3D [2]	L + C	✓	✗	✗	140m × 140m	5,534	-	10,169	48.6m
3EED (Ours)	L + C	✓	✓	✓	**280m × 240m**	**23,618**	**134,143**	**25,551**	**80m**

1 Introduction

Grounding free-form language to 3D scenes is a core capability for embodied agents operating in the physical world [1, 12, 6, 7, 42]. By associating natural language expressions with physical objects in 3D space, robots and autonomous systems can interpret high-level human instructions to perform downstream tasks, *e.g.*, navigation, interaction, and situational awareness [59, 83, 92, 19, 58, 84, 82].

Recent advances in 3D visual grounding have primarily focused on indoor benchmarks [31, 3, 30], where sensing is constrained, scenes are small, and objects are limited to household categories [88, 91]. However, real-world applications require models to operate in outdoor environments with greater spatial scale [54, 36], diverse viewpoints [60, 14], and sparse sensor data [5, 37].

While recent datasets have begun addressing outdoor 3D grounding [34, 20, 83, 23], they remain limited by single-platform data (*e.g.*, vehicle-mounted LiDAR), small scale with few objects and expressions, and a lack of multi-modal supervision, often providing only LiDAR or RGB but not both [24, 40, 27, 45, 32, 47]. These gaps limit the development of models that generalize across platforms, modalities, and real-world conditions.

To address these gaps, we introduce **3EED**, a *large-scale, multi-platform, multi-modal* benchmark for 3D visual grounding in outdoor environments (see Fig. 1). Our dataset captures synchronized LiDAR and RGB data from three distinct robotic platforms: `Vehicle`, `Drone`, `Quadruped`. It provides over **134,000 object instances** and **25,000** human-verified **referring expressions**, making it **10× larger** than existing outdoor grounding benchmarks, as compared in Tab. 1.

To enable scalable annotation, we develop a *vision-language model prompting pipeline* combined with *human-in-the-loop verification* to generate high-quality referring expressions. Additionally, we propose **platform-aware normalization** and **cross-modal alignment** techniques to standardize geometric and sensory data while preserving platform-specific characteristics. Based on these contributions, we establish a comprehensive benchmark suite covering in-domain, cross-platform, and multi-object grounding settings. Through extensive experiments with state-of-the-art models [31, 80], we reveal substantial performance gaps across platforms, exposing the challenges of robust and generalizable 3D visual grounding in real-world outdoor environments.

To summarize, the key contributions of this work to the related fields include:

- We present **3EED**, the first large-scale, multi-platform, multi-modal 3D visual grounding benchmark spanning `Vehicle`, `Drone`, `Quadruped` platforms, covering over 134,000 objects and 25,000 human-verified expressions, which is 10× larger than existing outdoor datasets.
- We develop a scalable annotation pipeline combining vision-language model prompting with human validation, enabling high-quality and diverse language supervision.
- We propose *platform-aware normalization* and *cross-modal alignment* to unify sensor geometry and synchronize LiDAR, RGB, and language cues, enabling consistency across diverse platforms.

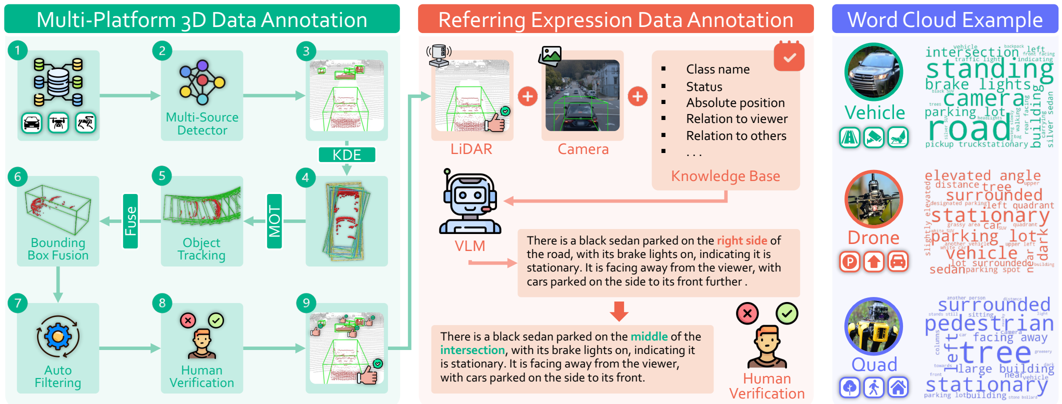


Figure 2: **Overview of annotation workflow. Left:** We collect 3D boxes using multi-detector fusion, tracking, filtering, and manual verification across platforms. **Middle:** Referring expressions are produced by prompting a VLM with structured cues (class, status, position, relations), followed by rule-based rewriting and human refinement. **Right:** Platform-specific word clouds highlight distinct linguistic patterns in descriptions across vehicle, drone, and quadruped agents.

- We establish comprehensive benchmark protocols for in-domain, cross-platform, and multi-object grounding, along with strong baseline evaluations revealing key challenges and future directions.

2 Related Work

3D Visual Grounding. 3D visual grounding localizes objects in 3D scenes from natural language expressions. Early efforts focus on indoor RGB-D datasets like ScanRefer [12] and Nr3D [1], built on ScanNet [15] and ARKitScenes [4], with object categories mostly limited to furniture. Recent datasets such as Multi3DRefer [95] and EmbodiedScan [75] expand to multi-object and egocentric grounding. These resources have driven the development of various models [97, 89, 80, 22, 73, 31, 3, 30, 76, 91, 98, 41, 88] focused on spatial-linguistic alignment in controlled indoor environments.

3D Grounding in the Wild. Grounding language in outdoor 3D scenes introduces challenges such as large spatial scales, sparse point clouds, and diverse object distributions [38, 39, 81, 71, 72, 63]. Talk2Car [17], based on nuScenes [8], is an early benchmark for driving scenarios. STRefer [43] extends this with RGB and LiDAR from mobile agents, focusing on human activities. Mono3DVG [93] studies grounding in monocular images without 3D sensors. KITTI360Pose [35] uses templated language for text-to-position grounding in KITTI-360 [21], targeting positions rather than objects. Talk2LiDAR [51] and CityRefer [55] provide multi-sensor and city-scale grounding tasks. However, all these datasets are limited to **single-platform** data acquisition.

Language-Guided Perception in Embodied Platforms. Language understanding has also been explored in interactive [77, 44, 53, 96, 25] and multi-task perception settings [98, 13, 28, 33, 85, 49, 50, 65, 52, 48, 86]. Refer-KITTI [79] based on KITTI [21] enables tracking multiple objects with a single prompt. nuPrompt [78] employs a language prompt to predict the described object trajectory across views and frames. nuScenes-QA [62] formulates a multi-modal question answering benchmark using nuScenes [8] data. DriveLM [69] formulates driving as a graph-based visual question answering task, leveraging structured visual representations and large language models [56] to answer route-planning and scene-understanding queries. These methods, however, focus on vehicle-based data [21, 8] and semantic-level tasks [66, 29], whereas our dataset enables fine-grained 3D grounding across diverse embodied agents, including drones and legged robots.

3 3EED: Multi-Platform Multi-Modal 3D Grounding Dataset

Existing 3D grounding datasets mainly target small, sensor-fixed indoor spaces, leaving outdoor, multi-platform scenarios underexplored. To bridge this gap, we curate 3EED, the first 3D grounding

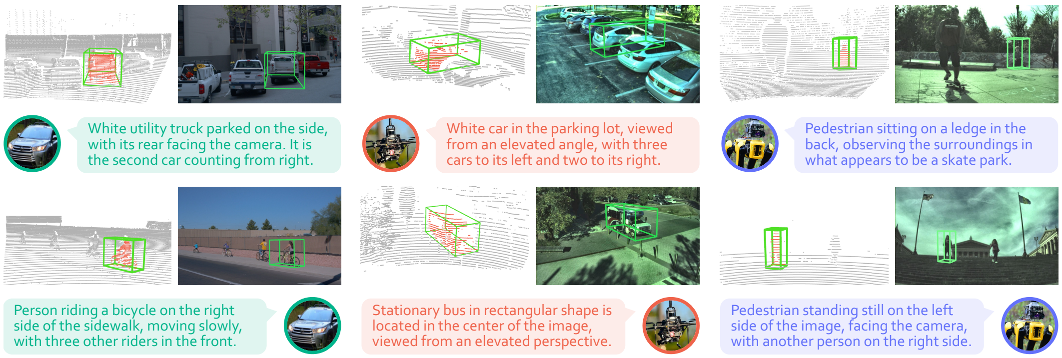


Figure 3: **Examples of multi-platform 3D grounding** from the **3EED** dataset. There are clear discrepancies across both *sensory data* (2D & 3D) and *referring expressions* from the `Vehicle`, `Drone`, and `Quadruped` platforms. For additional examples, kindly refer to the Appendix.

dataset that unifies data from `Vehicle`, `Drone`, `Quadruped` platforms. We formalize the multi-modal, multi-platform 3D grounding task in Sec. 3.1, detail a two-stage annotation pipeline in Sec. 3.2, and present statistics that highlight the scale, diversity, and platform balance in Sec. 3.3.

3.1 Task Formulation: 3D Grounding in the Wild

We define the multi-platform 3D grounding task in our dataset as $\mathcal{F}(\mathcal{P}^{\beta}, I^{\beta}, \mathcal{C}) \rightarrow \mathbf{b}^{\beta}$, where the model $\mathcal{F}$ maps input modalities, optionally including the point cloud $\mathcal{P}^{\beta} = \{\mathbf{p}_i\}_{i=1}^{N^{\beta}}$, image I^{β}, and caption $\mathcal{C}$ to the corresponding 3D bounding box $\mathbf{b}^{\beta} \in \mathbb{R}^7$. Each point $\mathbf{p}_i = (p^x, p^y, p^z) \in \mathbb{R}^3$, and the bounding box is given by its center, dimensions, and orientation angle. β denotes the platform, including the `Vehicle`, `Drone`, and `Quadruped`, and N^{β} is the number of point clouds for platform β. To precisely quantify spatial relationships, we also define the bird's-eye-view distance from *target* to *ego-platform* as ρ and the relative pitch angle as θ^r. In dataset curation and annotation, we explicitly consider **platform-specific factors** caused by inherent geometric differences.

3.2 Dataset Curation & Annotations

Multi-Platform 3D Data Annotation. We collect `Vehicle` sequences from Waymo [70], and `Drone` and `Quadruped` sequences from M3ED [11]. We adopt a uniform **three-stage pipeline** for the `Drone`/`Quadruped` LiDAR–RGB (see Fig. 2, left). ***1)*** *Pseudo-label seeding:* State-of-the-art detectors [67, 68, 16, 94, 90, 87] trained on Waymo [70], nuScenes [8], and Lyft [26] produce platform-agnostic 3D boxes for every frame. ***2)*** *Automatic consolidation:* Kernel-density estimation (KDE) merges detector votes, a 3D multi-object tracker [18] enforces temporal coherence and fills missed detections, and the Tokenize-Anything [57] model is used to project each box onto the RGB view to confirm its class; category conflicts are auto-flagged. ***3)*** *Human refinement:* Annotators polish the flagged boxes in the user interface, cross-validating to equalize accuracy across platforms. This hybrid scheme yields consistent annotations while limiting manual effort to roughly 100s per frame.

Referring Expression Data Annotation. After collecting the 3D boxes, we attach platform-invariant language supervision through a parallel procedure (see Fig. 2, middle). ***1)*** *Structured prompting:* Each 3D box is projected onto its RGB view, together with a knowledge base with five template slots *category*, *status*, *absolute location*, *egocentric position*, *relation*, to a vision language model [74]. Few-shot expression examples in the prompt are used to guide the model to output a single, well-formed referring sentence. Platform-specific terms are normalized by platform-invariant rewriting rules to ensure consistent wording across vehicle, drone, and quadruped views. ***2)*** *Human verification:* Annotators inspect the image, projected box, and caption in an interactive UI, checking semantic correctness, spatial fidelity, absence of ambiguity, and platform-consistency. Cases that are unsatisfactory will be discarded. This staged pipeline delivers concise, unambiguous expressions across vehicle, drone, and quadruped views, providing high-quality language targets for 3D visual grounding.

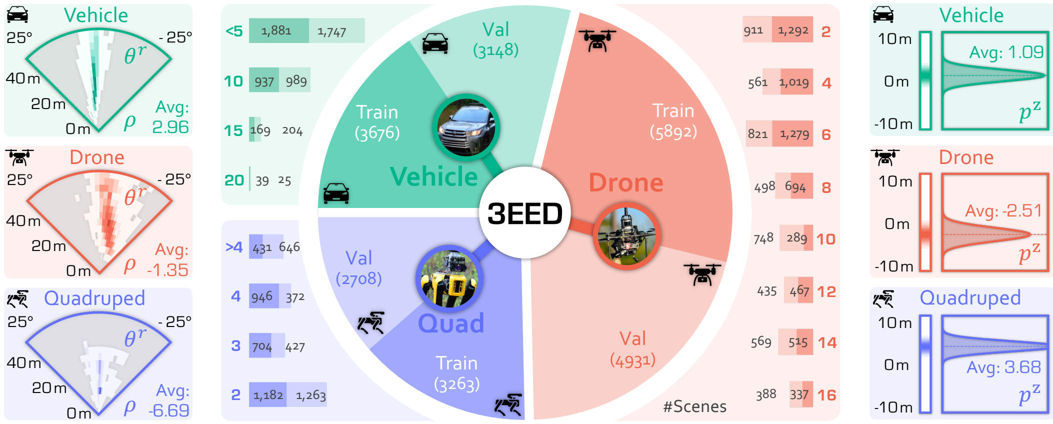


Figure 4: **Dataset statistics** of the three platforms in **3EED**. **Left:** Target bounding box distributions in polar coordinates. Color intensity indicates the frequency of targets in each (ρ, θ^r) bin. **Middle:** Scene distribution for train/val splits on each platform, along with per-scene object count histograms. **Right:** Elevation distributions of input point cloud, p^z, reflecting view-dependent elevation biases.

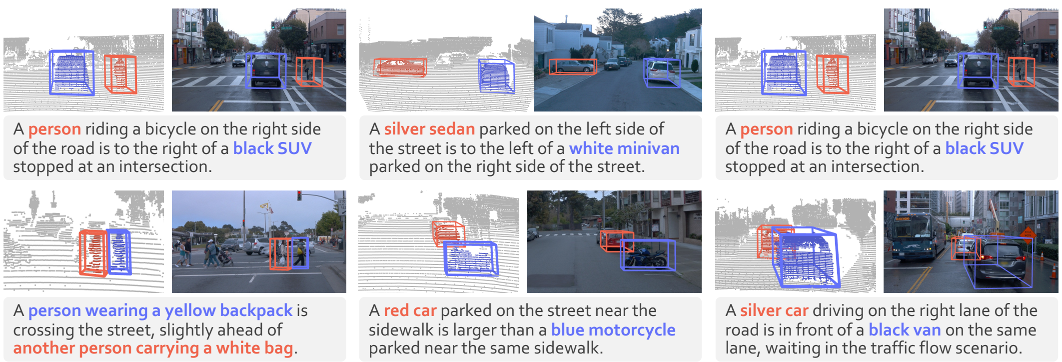


Figure 5: **Examples of multi-object 3D grounding** from the **3EED** dataset. Given a scene and a multi-object expression, the goal of this task is to localize the 3D bounding box of each referred object by reasoning over both semantic attributes and inter-object spatial relationships.

3.3 Dataset Statistics & Analysis

Benchmark Comparisons. **3EED** is, to our knowledge, the *first* outdoor 3D visual grounding benchmark that standardizes sensing across three embodied platforms `Vehicle`, `Drone`, and `Quadruped` by using synchronized LiDAR–RGB acquisition. As summarized in Tab. 1, our dataset provides $\mathbf{134{,}143}$ object bounding boxes and $\mathbf{25{,}551}$ human-verified referring expressions over $\mathbf{23{,}618}$ tightly time-aligned frames, focusing on the two safety-critical classes *Vehicle* and *Pedestrian*. Spatially, our scenes span up to $\mathbf{280\,m} \times \mathbf{240\,m}$ horizontally and exceed $\mathbf{80\,m}$ in elevation, with an order of magnitude larger than any previous outdoor corpus, making it uniquely suited for studying long-range, cross-platform grounding. The train/val split is carefully balanced. As shown in Fig. 4 (middle), containing 3.7k/3.1k vehicle, 5.9k/4.9k drone, and 3.3k/2.7k quadruped scenes, enabling rigorous analysis of both platform-specific challenges and cross-platform generalization.

Platform-Specific Analysis. To illuminate how **3EED** supports **robust multi-platform downstream tasks**, we dissect the sensing geometry and scene composition of each agent in three dimensions:

***1)* Viewpoint geometry of targets:** Fig.4 (left) shows the distribution of pitch angleθ^r and BEV range ρ for each 3D box. `Vehicle` data clusters at mid-range with near-zero pitch, typical of level driving. `Drone` covers larger ρ with steep negative θ^r from top-down views. `Quadruped` stays close in ρ but varies widely in pitch due to ground-level perspective. These patterns expose models to varied spatial cues like "behind" and "under", improving generalization to novel viewpoints.

2) **Per-platform object density:** Fig. 4 (middle) shows object density per platform. `Drone` captures the busiest scenes due to its wide view, `Vehicle` records moderate density, and `Quadruped` sees fewer but closer objects. This range enables 3EED to test the ability to disambiguate crowded scenes, maintain situational awareness, and localize small, nearby targets – offering a challenging testbed for robust 3D grounding. ***3)*** **Input point-cloud geometry:** Fig.4 (right) shows the vertical distribution of LiDAR points p^z per platform. `Vehicle` scans center around the sensor height, `Drone` captures top-down views, and `Quadruped` looks upward toward obstacles. These elevation biases affect how spatial terms like "above" or "below" are grounded, offering rich vertical language diversity across viewpoints.

4 Benchmark Establishment

The scale and heterogeneity of **3EED**, including the three embodied platforms, synchronized LiDAR–RGB sensing, and densely annotated outdoor scenes, allow us to benchmark diverse grounding tasks: [1]*Single-platform, single-object grounding*: follow the conventional setup and serve as a sanity check. [2]*Cross-platform transfer*: train on the data-rich vehicle data and evaluate on the scarcer drone and quadruped data, reflecting real-world constraints where labeling drone or quadruped is costly yet generalization is crucial. [3]*Multi-object grounding*: requires locating all described targets in a frame-crucial outdoors, where autonomous systems must track multiple objects rather than a single cup on a table. Fig. 5 illustrates several scenes that involve multiple objects. [4]*Multi-platform grounding*: unified training on all platforms to build a general and robust grounding model.

4.1 Challenges for Existing Methods

Most 3D grounding models are designed for indoor RGB-D data, with dense, uniform points and small, consistent object sizes. On **3EED**, they face three key challenges: ***1)*** **Range-dependent sparsity:** LiDAR points thin out with distance, breaking indoor assumptions of dense neighborhoods. ***2)*** **Extreme scale variation:** Outdoor targets range from small cones to large vehicles, invalidating fixed-size anchors. ***3)*** **Cross-platform gaps:** Different viewpoints and sensor heights cause shifts in density and field of view unseen in indoor settings. As we will illustrate in the next section, these challenges reveal the need for outdoor- and platform-aware model designs.

4.2 Unified Cross-Platform Baseline

To kick-start research on *cross-platform transfer* and *multi-object grounding*, we present a scale-adaptive and agent-invariant baseline model tailored to **3EED**. It effectively addresses these challenges and serves as a strong reference point for future work in robust, general 3D visual grounding.

Baseline Overview. We adapt previous work [31] to our dataset: a scale-adaptive PointNet++ [61] backbone encodes LiDAR, a frozen RoBERTa [46] encodes language, and a Transformer predicts every referenced 3D box in one shot. Training blends box-regression, token-alignment, and contrastive multimodal losses. In the multi-object grounding setting, each target object is associated with a distinct positive map. We apply Hungarian matching to assign each query to a specific target object, enabling supervised learning via one-to-one loss computation.

Multi-Scale Sampling (MSS). Each PointNet++ layer gathers neighborhoods at radii from 0.6 m to 4.8 m, dynamically capturing sharp local details nearby and broad contextual structure far away. This range-aware sampling effectively counters LiDAR sparsity and object-size extremes, thereby letting the backbone reliably localize both tiny traffic cones and massive buses.

Scale-Aware Fusion (SAF). Backbone features from all radii are passed through a lightweight MLP that learns dynamic, per-point weights, thus strongly emphasizing whichever scale best explains the local geometry. SAF automatically adapts to dramatic density shifts across platforms, yielding highly scale-robust, agent-agnostic embeddings at essentially negligible computational cost.

Table 2: **Benchmark results of state-of-the-art models** on the **3EED** dataset. The performances are measured under both [1]*Single-platform* and [2]*Cross-platform* settings across three platforms: `Vehicle`, `Drone`, and `Quadruped`. All scores are given in percentage (%).

Method	Platform Adaptation	Vehicle Acc@25	Vehicle Acc@50	Drone Acc@25	Drone Acc@50	Quadruped Acc@25	Quadruped Acc@50	Union Acc@25	Union Acc@50
• **Training Platform:** Vehicle									
BUTD-DETR [31]	✗	59.53	45.34	8.66	3.68	19.20	8.27	27.54	18.30
EDA [80]	✗	60.47	51.34	9.03	4.12	12.91	6.85	27.26	20.40
Ours	✓	67.81	65.03	18.84	13.03	35.29	27.10	39.17	33.43
Improve ↑	-	**+7.34**	**+13.69**	**+9.81**	**+8.91**	**+16.09**	**+18.83**	**+11.63**	**+13.03**
• **Training Platform:** Drone									
BUTD-DETR [31]	✗	1.81	0.24	19.37	8.93	12.50	2.77	11.21	4.02
EDA [80]	✗	2.59	0.19	22.01	14.31	9.17	2.13	12.59	6.70
Ours	✓	11.30	1.43	23.00	16.32	17.18	4.23	17.45	8.21
Improve ↑	-	**+8.71**	**+1.19**	**+0.99**	**+2.01**	**+4.68**	**+1.46**	**+4.86**	**+1.51**
• **Training Platform:** Quadruped									
BUTD-DETR [31]	✗	10.71	4.16	5.27	1.51	27.25	15.98	12.12	5.76
EDA [80]	✗	9.32	4.06	7.52	2.16	25.26	18.60	12.25	6.09
Ours	✓	21.68	9.11	9.39	5.13	37.76	28.59	19.99	12.05
Improve ↑	-	**+10.97**	**+4.95**	**+1.87**	**+2.97**	**+10.51**	**+9.99**	**+7,74**	**+5.96**
• **Training Platform: Union** (Vehicle + Drone + Quadruped)									
BUTD-DETR [31]	✗	55.50	42.01	24.01	16.83	37.84	27.07	37.29	26.66
EDA [80]	✗	58.63	49.85	24.68	17.71	37.23	26.16	39.47	30.18
Ours	✓	66.23	61.69	28.51	23.55	46.01	41.33	46.03	40.98
Improve ↑	-	+7.60	+11.84	+3.83	+5.84	+8.17	+14.26	+6.56	+10.80

Cross-Platform Alignment (CPA). Before feature extraction, each scan is rotated to cancel roll and pitch, thus consistently aligning gravity with the global z-axis; drones additionally receive an altitude-normalizing height offset. This simple, one-shot normalization removes viewpoint bias, enabling models trained on vehicles to generalize smoothly to other agents without additional retraining.

5 Experiments

5.1 Experimental Setups

Implementation Details. Our method is implemented in PyTorch, following the training schedule and optimization settings of previous work [31], but optimized for efficiency. Raw LiDAR from any platform is uniformly down-sampled to 16,384 points and encoded by a PointNet++ backbone [61] trained from scratch; its final layer yields 1,024 visual tokens. An MLP assigns each token an objectness score, and the top 256 tokens are input into a six-layer Transformer decoder. Objectness is supervised with focal loss by labeling the four nearest points to every ground-truth center as positives. We freeze RoBERTa, use a learning rate of 1×10^{-3} for the visual encoder and 1×10^{-4} for all other layers, and train for 100 epochs on two NVIDIA RTX 4090 GPUs. See **Appendix** for more details.

Evaluation Metrics. Following [12, 1, 43], we report *Top-1 Acc*, counting a success when the top box exceeds a chosen IoU. We evaluate at `Acc@25` (lenient) and `Acc@50` (strict), and report mean IoU (`mIoU`) for overall quality. In multi-object setup, all objects must meet the IoU threshold, penalizing misses and false positives. Results are averaged over official train/val splits for fair comparison.

Baselines. We adapt two representative baselines. *EDA* [80] is a prior art on indoor datasets by decoupling sentences into object, attribute, relation, and pronoun tokens, enforcing dense token-point alignment. However, it relies on dense scenes and grammar-consistent text, making it fragile under sparse LiDAR, large object-size variation, and diverse viewpoints. *BUTD-DETR* [31] uses a DETR-style decoder [9] with ScanNet box proposals and synthetic prompts but struggles on drone and quadruped data due to its dependence on indoor detectors. Neither baseline addresses range-dependent sparsity, scale variation, or cross-platform biases, motivating our scale-adaptive, agent-invariant baseline. Due to space limits, additional details are provided in the **Appendix**.

Table 3: **Benchmark results of state-of-the-art models** on the **3EED** dataset. The performances are measured under the *multi-object* setting on the `Vehicle` platform. We report the class-wise performance on `Acc@25`, `Acc@50`, and `mIoU` metrics. All scores are given in percentage (%).

Method	Car			Pedestrian			Average		
	`Acc@25`	`Acc@50`	`mIoU`	`Acc@25`	`Acc@50`	`mIoU`	`Acc@25`	`Acc@50`	`mIoU`
BUTD-DETR [31]	30.92	19.83	52.39	26.56	18.75	37.28	25.40	17.91	47.88
EDA [80]	29.58	26.21	56.73	28.15	14.75	38.37	26.91	25.92	51.07
Ours	37.21	33.14	59.28	32.81	20.31	54.21	32.32	29.89	56.40
Improve ↑	**+7.63**	**+14.63**	**+6.89**	**+4.66**	**+1.56**	**+15.84**	**+5.41**	**+3.97**	**+5.33**

Table 4: **Ablation study on components.** The performances are measured under the *multi-platform* setting. SAF: The scale-aware fusion module. MSS: The multi-scale sampling method.

Method	`Vehicle`		`Drone`		`Quadruped`	
	`Acc@25`	`Acc@50`	`Acc@25`	`Acc@50`	`Acc@25`	`Acc@50`
Base	55.50	42.01	24.01	16.83	37.84	27.07
– MSS	61.34	50.21	24.85	19.73	43.53	30.69
– SAF	63.45	56.75	28.24	22.94	45.42	40.98
Full	66.23	61.69	28.51	23.55	46.01	41.33

Table 5: **Ablation study on scene complexity.** The performances are measured under the *multi-platform* setting. Here, we split scenes based on the number of objects per scene.

Object Count	`Vehicle`		`Drone`		`Quadruped`	
	`Acc@25`	`Acc@50`	`Acc@25`	`Acc@50`	`Acc@25`	`Acc@50`
$1-3$	70.86	65.42	42.53	33.63	57.51	50.69
$4-6$	64.82	59.71	39.71	33.73	35.45	33.57
$7-9$	52.48	47.77	27.84	24.88	28.26	23.37
>9	58.11	55.41	18.16	15.66	0.00	0.00

5.2 Comparative Study

Cross-Platform Generalization. Tab. 2 compares existing 3D grounding backbones under in-distribution (*single-platform*) and out-of-distribution (*cross-platform*) settings.

1) Single-Platform vs. Cross-Platform. When trained on `Vehicle` data, BUTD-DETR [31] achieves `Acc@25` of 59.53 on the vehicle test split, but drops to 8.66 on drone and 19.20 on quadruped, exposing severe generalization gaps due to differing viewpoints, object scales, and LiDAR densities.

2) Cross-Platform Transfer Gains. Our scale-adaptive backbone with platform alignment substantially narrows this gap. For example, training on `Drone` and evaluating on `Vehicle` boosts `Acc@25` by +8.71 over the baseline, demonstrating stronger transfer from aerial to ground perspectives.

3) Unified Multi-Platform Training. A unified model trained jointly on all three platforms delivers balanced performance, with `Acc@25` of 66.23, 28.51, and 46.01 on vehicle, drone, and quadruped, respectively, yielding an average gain of +6.56 over the best method. This confirms the critical role of **3EED** in providing diverse supervision for building truly generalizable 3D grounding systems.

Coherent Object Co-grounding. Tab. 3 presents the evaluation results on our dataset for the *multi-object grounding* task. Notably, in this setting, `Acc@25` is a strict metric that requires all objects mentioned in the description to be correctly grounded, while `mIoU` captures the average IoU across individual predicted-ground truth pairs. Existing methods such as BUTD-DETR achieve moderate `mIoU` (47.88) but low joint grounding (`Acc@25` = 25.40), revealing their tendency to localize objects in isolation rather than reason about them collectively. In contrast, our baseline leverages multi-scale sampling and dynamic feature fusion to build discriminative representations that capture both fine details and broad context, essential for disambiguating multiple objects of varying size and distance. These design choices deliver substantial improvements in both metrics, demonstrating markedly stronger multi-object reasoning and tighter language-to-3D alignment in complex outdoor scenes.

Qualitative Assessments. Fig. 6 showcases representative *multi-platform grounding* results on vehicle, drone, and quadruped data. Our unified model consistently outputs precise, tightly aligned 3D boxes despite drastic shifts in viewpoint, object scale, and point-cloud density. In contrast, baseline methods like BUTD-DETR [31] and EDA [80] often yield misaligned or fragmented predictions, especially under challenging aerial and low-angle quadruped perspectives. These comparisons underscore our ability to learn genuine cross-platform invariance and deliver reliable grounding across diverse embodied sensing scenarios.

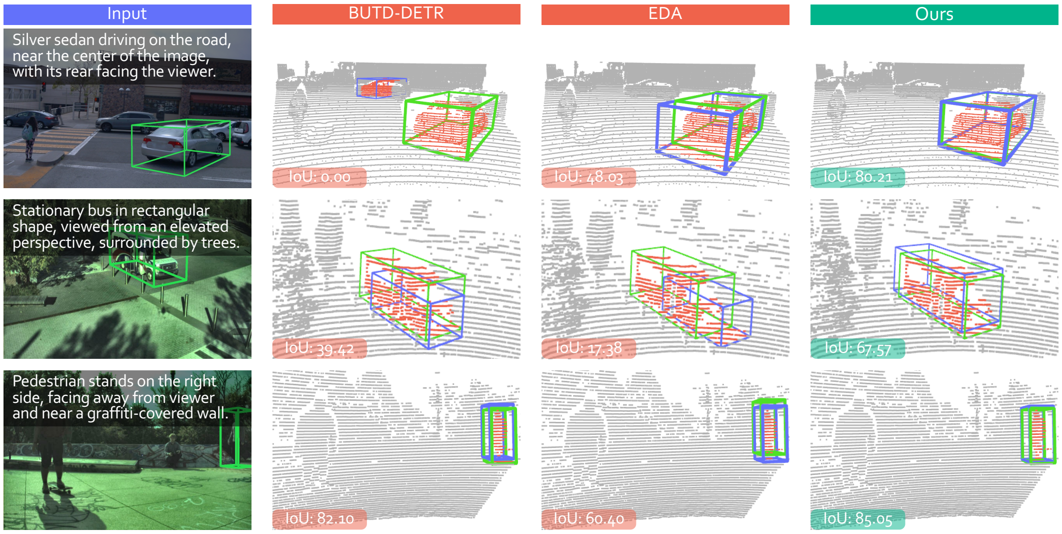


Figure 6: **Qualitative comparisons** of 3D grounding approaches on the **3EED** dataset. We show the comparisons under the *multi-platform* setting. The three examples are from the `Vehicle`, `Drone`, and `Quadruped` platforms, respectively. Kindly refer to the appendix for additional results.

Table 6: **Comparisons of platform-level 3D grounding statistics.** We report the average number of *annotated objects per scene* and *LiDAR points per object*. All scores are given in percentage (%).

Platform	Average #Objects / Scene	Average #Points / Object	`Vehicle` `Acc@25`	`Vehicle` `Acc@50`	`Drone` `Acc@25`	`Drone` `Acc@50`	`Quadruped` `Acc@25`	`Quadruped` `Acc@50`
`Vehicle`	4.74	**1452.83**	67.81	65.03	18.84	13.03	35.29	27.10
`Drone`	**8.57**	93.27	11.30	1.43	23.00	16.32	17.18	4.23
`Quadruped`	3.36	207.25	21.68	9.11	9.39	5.13	37.76	28.59

5.3 Ablation Study

Component Analysis. Tab. 4 presents an ablation of our two core modules. **MSS** alone raises Vehicle `Acc@25` from 55.50% to 61.34% (+5.84%), while **SAF** alone achieves 63.45%, confirming their complements. MSS samples neighborhood radii from a larger range, capturing both fine local edges and broad context to mitigate LiDAR sparsity and object-size variation. SAF fuses multi-scale features through a lightweight MLP that learns per-point weights, highlighting the most informative scale and adapting to dramatic density shifts. Together, they deliver the strongest overall performance.

Object Density Impact. We analyze how referential grounding performance varies with the object density per scene. We divide test samples into bins based on the number of annotated 3D bounding boxes (1-3, 4–6, 7–9, 10+), and compute the average `Acc@25` for each bin. As shown in Tab. 5, accuracy consistently drops as object count increases. On the `Vehicle` platform, `Acc@25` drops from 70.86 in scenes with 1–3 objects to 52.48 in scenes with 7–9 objects. This reflects the increased difficulty of resolving referential ambiguity in cluttered environments.

Platform Complexity Impact. Tab. 6 breaks down grounding performance by platform alongside two key scene statistics: mean LiDAR points per object and mean object count per scene. `Drone` scenes suffer the lowest `Acc@50`, driven by extreme sparsity (just 93 points/object *vs.*1,452 for `Vehicle` and 207 for `Quadruped`) and the highest object density (8.57 objects/scene), which together amplify distractors and hinder precise localization. Quadruped data, with moderate density (207 points/object) but fewer objects, sits between drone and vehicle performance. These disparities, including ultra-sparse returns and elevated clutter, explain the pronounced aerial performance gap.

6 Conclusion

We introduced 3EED, a large-scale, multi-platform, multi-modal benchmark for outdoor 3D visual grounding, featuring $134{,}000$ objects and $25{,}000$ expressions, which is $10\times$ larger than existing datasets. We proposed scalable annotation, platform-aware normalization, and cross-modal alignment to support robust grounding. Our benchmark reveals cross-platform performance gaps, highlighting challenges for generalizable 3D grounding. We release our dataset and baseline models, hoping to advance the future development of language-driven embodied 3D perception.

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European Conference on Computer Vision*, pages 422–440. Springer, 2020.

[2] Yeong-Seung Baek and Heung-Seon Oh. Lidarefer: Outdoor 3d visual grounding for autonomous driving with transformers. *arXiv preprint arXiv:2411.04351*, 2024.

[3] Eslam Mohamed Bakr, Yasmeen Alsaedy, and Mohamed Elhoseiny. Look around and refer: 2d synthetic semantics knowledge distillation for 3d visual grounding. In *Advances in Neural Information Processing Systems*, volume 35, pages 37146–37158, 2022.

[4] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *Advances in Neural Information Processing Systems*, 2021.

[5] Stefan Andreas Baur, Frank Moosmann, and Andreas Geiger. Liso: Lidar-only self-supervised 3d object detection. In *European Conference on Computer Vision*, pages 253–270, 2024.

[6] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

[7] Alexandre Boulch, Corentin Sautier, Björn Michele, Gilles Puy, and Renaud Marlet. Also: Automotive lidar self-supervision by occupancy estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13455–13465, 2023.

[8] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[11] Kenneth Chaney, Fernando Cladera, Ziyun Wang, Anthony Bisulco, M Ani Hsieh, Christopher Korpela, Vijay Kumar, Camillo J Taylor, and Kostas Daniilidis. M3ed: Multi-robot, multi-sensor, multi-environment event dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4016–4023, 2023.

[12] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision*, pages 202–221. Springer, 2020.

[13] Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. In *IEEE/CVF International Conference on Computer Vision*, 2023.

[14] Huixian Cheng, Xianfeng Han, and Guoqiang Xiao. Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving. In *IEEE International Conference on Multimedia and Expo*, pages 1–6, 2022.

[15] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[16] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 1201–1209, 2021.

[17] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. Talk2car: Taking control of your self-driving car. *arXiv preprint arXiv:1909.10838*, 2019.

[18] Lue Fan, Yuxue Yang, Yiming Mao, Feng Wang, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Once detected, never lost: Surpassing human performance in offline lidar based 3d object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 19820–19829, 2023.

[19] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7:3795–3802, 2022.

[20] Biao Gao, Yancheng Pan, Chengkun Li, Sibo Geng, and Huijing Zhao. Are we hungry for 3d lidar data for semantic segmentation? a survey of datasets and methods. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6063–6081, 2021.

[21] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[22] Zoey Guo, Yiwen Tang, Ray Zhang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. Viewrefer: Grasp the multi-view knowledge for 3d visual grounding. In *IEEE/CVF International Conference on Computer Vision*, pages 15372–15383, 2023.

[23] Xiaoshuai Hao, Mengchuan Wei, Yifan Yang, Haimei Zhao, Hui Zhang, Yi Zhou, Qiang Wang, Weiming Li, Lingdong Kong, and Jing Zhang. Is your hd map constructor reliable under sensor corruptions? In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[24] Fangzhou Hong, Lingdong Kong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Unified 3d and 4d panoptic segmentation via dynamic shifting networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3480–3495, 2024.

[25] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[26] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, pages 409–418. PMLR, 2021.

[27] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020.

[28] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *International Conference on Machine Learning*, 2024.

[29] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 1610–1618, 2021.

[30] Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3d visual grounding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15524–15533, 2022.

[31] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *European Conference on Computer Vision*, pages 417–433. Springer, 2022.

[32] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12605–12614, 2020.

[33] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. *arXiv preprint arXiv:2401.09340*, 2024.

[34] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3d dataset: Data, benchmarks and analysis. In *IEEE International Conference on Robotics and Automation*, pages 1110–1116, 2021.

[35] Manuel Kolmet, Qunjie Zhou, Aljoša Ošep, and Laura Leal-Taixé. Text2pos: Text-to-point-cloud cross-modal localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6687–6696, 2022.

[36] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 228–240, 2023.

[37] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023.

[38] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21705–21715, 2023.

[39] Lingdong Kong, Xiang Xu, Jiawei Ren, Wenwei Zhang, Liang Pan, Kai Chen, Wei Tsang Ooi, and Ziwei Liu. Multi-modal data-efficient 3d scene understanding for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3748–3765, 2025.

[40] Li Li, Hubert PH Shum, and Toby P Breckon. Rapid-seg: Range-aware pointwise distance distribution networks for 3d lidar segmentation. In *European Conference on Computer Vision*, pages 222–241, 2024.

[41] Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. *arXiv preprint arXiv:2412.04383*, 2024.

[42] Ye Li, Lingdong Kong, Hanjiang Hu, Xiaohao Xu, and Xiaonan Huang. Is your lidar placement optimized for 3d scene understanding? In *Advances in Neural Information Processing Systems*, volume 37, pages 34980–35017, 2024.

[43] Zhenxiang Lin, Xidong Peng, Peishan Cong, Ge Zheng, Yujin Sun, Yuenan Hou, Xinge Zhu, Sibei Yang, and Yuexin Ma. Wildrefer: 3d object localization in large-scale dynamic scenes with multi-modal visual data and natural language. In *European Conference on Computer Vision*, pages 456–473. Springer, 2024.

[44] Xiongkun Linghu, Jiangyong Huang, Xuesong Niu, Xiaojian Ma, Baoxiong Jia, and Siyuan Huang. Multi-modal situated reasoning in 3d scenes. In *Advances in Neural Information Processing Systems*, 2024.

[45] Venice Erin Liong, Thi Ngoc Tho Nguyen, Sergi Widjaja, Dhananjai Sharma, and Zhuang Jie Chong. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. *arXiv preprint arXiv:2012.04934*, 2020.

[46] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[47] Youquan Liu, Runnan Chen, Xin Li, Lingdong Kong, Yuchen Yang, Zhaoyang Xia, Yeqi Bai, Xinge Zhu, Yuexin Ma, Yikang Li, et al. Uniseg: A unified multi-modal lidar segmentation network and the openpcseg codebase. In *IEEE/CVF International Conference on Computer Vision*, pages 21662–21673, 2023.

[48] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *Advances in Neural Information Processing Systems*, volume 36, pages 37193–37229, 2023.

[49] Youquan Liu, Lingdong Kong, Xiaoyang Wu, Runnan Chen, Xin Li, Liang Pan, Ziwei Liu, and Yuexin Ma. Multi-space alignments towards universal lidar segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14648–14661, 2024.

[50] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H Hsu. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.0468*, 2021.

[51] Yuhang Liu, Boyi Sun, Guixu Zheng, Yishuo Wang, Jing Wang, and Fei-Yue Wang. Talk to parallel lidars: A human-lidar interaction method based on 3d visual grounding. *arXiv preprint arXiv:2405.15274*, 2024.

[52] Anas Mahmoud, Jordan SK Hu, Tianshu Kuai, Ali Harakeh, Liam Paull, and Steven L Waslander. Self-supervised image-to-point distillation via semantically tolerant contrastive loss. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7102–7110, 2023.

[53] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352, 2022.

[54] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4213–4220, 2019.

[55] Taiki Miyanishi, Fumiya Kitamori, Shuhei Kurita, Jungdae Lee, Motoaki Kawanabe, and Nakamasa Inoue. Cityrefer: geography-aware 3d visual grounding dataset on city-scale point cloud data. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[56] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[57] Ting Pan, Lulu Tang, Xinlong Wang, and Shiguang Shan. Tokenize anything via prompting. In *European Conference on Computer Vision*, pages 330–348. Springer, 2024.

[58] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IEEE Intelligent Vehicles Symposium*, pages 687–693, 2020.

[59] Scott Drew Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Hong Eng, Daniela Rus, and Marcelo H Ang. Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1):6, 2017.

[60] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Using a waffle iron for automotive point cloud semantic segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 3379–3389, 2023.

[61] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[62] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *AAAI Conference on Artificial Intelligence*, volume 38, pages 4542–4550, 2024.

[63] Ryan Razani, Ran Cheng, Ehsan Taghavi, and Liu Bingbing. Lite-hdseg: Lidar semantic segmentation using lite harmonic dense convolutions. In *IEEE International Conference on Robotics and Automation*, pages 9550–9556, 2021.

[64] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[65] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9891–9901, 2022.

[66] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105*, 2022.

[67] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[68] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, 131(2):531–551, 2023.

[69] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European Conference on Computer Vision*, pages 256–274. Springer, 2024.

[70] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[71] Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. Torchsparse: Efficient point cloud inference engine. In *Conference on Machine Learning and Systems*, 2022.

[72] Haotian Tang, Shang Yang, Zhijian Liu, Ke Hong, Zhongming Yu, Xiuyu Li, Guohao Dai, Yu Wang, and Song Han. Torchsparse++: Efficient training and inference framework for sparse convolution on gpus. In *IEEE/ACM International Symposium on Microarchitecture*, 2023.

[73] Ozan Unal, Christos Sakaridis, Suman Saha, and Luc Van Gool. Four ways to improve verbo-visual fusion for dense 3d visual grounding. In *European Conference on Computer Vision*, pages 196–213. Springer, 2024.

[74] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[75] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19757–19767, 2024.

[76] Yuan Wang, Yali Li, and Shengjin Wang. G3-lq: Marrying hyperbolic alignment with explicit semantic-geometric modeling for 3d visual grounding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13917–13926, 2024.

[77] Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Move as you say interact as you can: Language-guided human motion generation with scene affordance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[78] Dongming Wu, Wencheng Han, Yingfei Liu, Tiancai Wang, Cheng-zhong Xu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving. In *AAAI Conference on Artificial Intelligence*, volume 39, pages 8359–8367, 2025.

[79] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring multi-object tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14633–14642, 2023.

[80] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19231–19242, 2023.

[81] Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, and Ling Shao. Polarmix: A general data augmentation technique for lidar point clouds. *Advances in Neural Information Processing Systems*, 35:11035–11048, 2022.

[82] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *AAAI Conference on Artificial Intelligence*, pages 2795–2803, 2022.

[83] Aoran Xiao, Jiaxing Huang, Dayan Guan, Xiaoqin Zhang, Shijian Lu, and Ling Shao. Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):11321–11339, 2023.

[84] Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, and Eric P Xing. 3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9382–9392, 2023.

[85] Xiang Xu, Lingdong Kong, Hui Shuai, and Qingshan Liu. Frnet: Frustum-range networks for scalable lidar segmentation. *IEEE Transactions on Image Processing*, 34:2173–2186, 2025.

[86] Xiang Xu, Lingdong Kong, Hui Shuai, Wenwei Zhang, Liang Pan, Kai Chen, Ziwei Liu, and Qingshan Liu. 4d contrastive superflows are dense 3d representation learners. In *European Conference on Computer Vision*, pages 58–80, 2024.

[87] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[88] Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. In *IEEE International Conference on Robotics and Automation*, pages 7694–7701, 2024.

[89] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. Sat: 2d semantics assisted training for 3d visual grounding. In *IEEE/CVF International Conference on Computer Vision*, pages 1856–1866, 2021.

[90] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021.

[91] Zhihao Yuan, Jinke Ren, Chun-Mei Feng, Hengshuang Zhao, Shuguang Cui, and Zhen Li. Visual programming for zero-shot open-vocabulary 3d visual grounding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20623–20633, 2024.

[92] Changyu Zeng, Wei Wang, et al. Self-supervised learning for point cloud data: A survey. *Expert Systems with Applications*, 237:121354, 2024.

[93] Yang Zhan, Yuan Yuan, and Zhitong Xiong. Mono3dvg: 3d visual grounding in monocular images. In *AAAI Conference on Artificial Intelligence*, volume 38, pages 6988–6996, 2024.

[94] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022.

[95] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *IEEE/CVF International Conference on Computer Vision*, pages 15225–15236, 2023.

[96] Zhuofan Zhang, Ziyu Zhu, Junhao Li, Pengxiang Li, Tianxu Wang, Tengyu Liu, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Siyuan Huang, and Qing Li. Task-oriented sequential grounding and navigation in 3d scenes. *arXiv preprint arXiv:2408.04034*, 2024.

[97] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *IEEE/CVF International Conference on Computer Vision*, pages 2928–2937, 2021.

[98] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *IEEE/CVF International Conference on Computer Vision*, pages 2911–2921, 2023.

Appendix

Table of Contents

A The 3EED Dataset

In this section, we provide a comprehensive overview of the 3EED dataset, including its motivation, collection methodology, and unique characteristics. We describe the design choices made to ensure diversity in sensor platforms, scene composition, and language annotation, and highlight the potential to support research in 3D visual grounding across real-world embodied platforms.

Table 7: Statistics of the 3EED dataset across platforms and splits.

Platform	# Scenes	# Captions	# Objects
Training			
`Vehicle`	3,676	4,380	14,042
`Quadruped`	3,263	3,259	10,510
`Drone`	5,892	5,827	44,193
Total	**12,831**	**13,466**	**68,745**
Validation			
`Vehicle`	3,148	4,447	14,051
`Quadruped`	2,708	2,707	9,107
`Drone`	4,931	4,931	42,240
Total	**10,787**	**12,085**	**65,398**
Summary	**23,618**	**25,551**	**134,143**

A.1 Overview

Our dataset is built on top of two existing autonomous driving and robotics datasets: **Waymo Open Dataset** [70] and **M3ED** [11]. Our dataset includes point cloud and image data collected from three distinct embodied platforms – `Vehicle`, `Drone`, and `Quadruped` – capturing scenes from street-level, aerial, and low-ground perspectives, respectively. The referring expressions are generated by Qwen-VL-72B [74], covering five aspects: *category*, *status*, *absolute location*, *egocentric position*, and *spatial relation*, with human verification.

The full dataset contains 23,618 multi-modal scenes, 25,551 referring expressions, and 134,143 annotated 3D object instances across three sensor platforms. The **training set** consists of 12,831 scenes, with 13,466 captions and 68,745 objects, while the **validation set** includes 10,787 scenes, 12,085 captions, and 65,398 objects.

Breaking down by platform: the `Vehicle` split provides 6,824 scenes and 28,093 objects; the `Quadruped` split includes 5,971 scenes and 19,617 objects; and the `Drone` split contributes the largest portion with 10,823 scenes and 86,433 objects. This distribution reflects the platform diversity and scale of our dataset, supporting cross-platform and cross-viewpoint grounding evaluation.

This cross-platform, cross-viewpoint composition allows our dataset to serve as a unified benchmark for 3D grounding under varying spatial configurations, sensor geometries, and linguistic descriptions. It enables the evaluation of platform-agnostic language understanding in real-world conditions.

A.2 Dataset Curation Details

This section details the data sourcing, 3D bounding box annotation pipeline, and referring expression generation process used to construct the 3EED dataset. We describe how annotated 3D boxes are curated across platforms using a combination of pretrained detectors, tracking, and manual refinement, and how language expressions are generated and verified to ensure grounding quality and consistency across scenes.

A.2.1 Data Sources

The dataset is built on top of two large-scale real-world 3D perception datasets: **Waymo Open Dataset** [70] and **M3ED** [11].

Waymo Open Dataset [70] provides high-resolution LiDAR and RGB data collected from vehicle-mounted sensors in urban and suburban driving environments. We use a subset of Waymo annotated scenes to construct the `Vehicle` portion of our dataset, leveraging its high-quality 3D bounding boxes as ground truth. Our annotations are built independently on top of their publicly available sequences.

M3ED Dataset [11] is a multi-platform dataset, featuring synchronized RGB and LiDAR streams from both quadruped robots and aerial drones operating in various outdoor scenes. The `Drone` and `Quadruped` portions of our dataset are derived from M3ED. Since M3ED does not contain pre-annotated 3D bounding boxes, we adopt a semi-automatic annotation pipeline that combines multiple pretrained detectors, trajectory tracking, and human refinement to generate high-quality 3D boxes.

A.2.2 Annotation Details on 3D Bounding Boxes

The 3D bounding box annotations in **3EED** are obtained through a combination of high-quality existing labels and a carefully designed cross-platform annotation pipeline.

Vehicle Platform. For the `Vehicle` platform, we adopt 3D object annotations directly from the official Waymo Open Dataset [70], which provides dense, high-accuracy bounding boxes for traffic participants such as vehicles, pedestrians, and cyclists etc.. These annotations are widely regarded as reliable and are used without further modification.

Drone and Quadruped Platforms. For the `Drone` and `Quadruped` platform, the original M3ED Dataset [11] does not contain pre-annotated 3D bounding boxes and require custom 3D bounding box annotations. We establish an annotation pipeline introduced in Figure 2 of the main paper. The process is composed of three stages:

- *Pseudo-label seeding.* We first pretrain a diverse set of state-of-the-art 3D detectors: PV-RCNN [67], PV-RCNN++ [68], Voxel-RCNN [16], IA-SSD [94], CenterPoint [90], and SECOND [87], on large-scale external datasets (*e.g.*, Waymo [70], nuScenes [8], Lyft [26]). These models are then used to infer pseudo-labels on our data, covering a variety of sensor configurations and scene layouts.
- *Automatic consolidation.* To consolidate predictions, we apply a kernel density estimation (KDE) approach to fuse overlapping boxes and improve consistency. A 3D multi-object tracking algorithm (CTRL [18]) is used to propagate detections over time and interpolate missing instances. To further validate category correctness, we employ the Tokenize Anything model [57] to project pseudo-boxes onto RGB images and cross-check the detected objects with open-vocabulary tags (see Figure 7). Boxes with mismatched semantics are flagged for review, reducing semantic drift across modalities.
- *Human refinement.* Finally, we manually refine each box on a per-frame basis. Three trained annotators iteratively verify, correct, and cross-validate all annotations to ensure high-quality outputs. Despite the assistance from automation, the sparsity and noise of real-world point clouds require human oversight.

This multi-stage toolkit integrates detection, filtering, image-level verification, and annotation interfaces. It enables scalable and accurate labeling for mobile platforms where no prior annotations exist, contributing to the high consistency and realism of our dataset.

A.2.3 Annotation Details on Referring Expressions

To evaluate grounding performance under natural and unambiguous language, we annotate referring expressions for each 3D bounding box in our dataset. These expressions are designed to support both single-object and multi-object grounding across diverse platforms, and are generated via a hybrid automatic–manual pipeline.

Generation with Vision-Language Models. We use the Qwen-VL-72B [74] vision-language model to automatically generate initial referring expressions. For each annotated 3D bounding box, we first project it onto the corresponding RGB image frame, then provide both the image and a task-specific

Figure 7: Automatic pseudo-label screening interface powered by the Tokenize Anything model.

prompt to the model. The prompts are carefully designed to guide the model to produce detailed, visually grounded, and unambiguous expressions.

For the *single-object grounding* setting, we use a structured prompt (see Table 8) that elicits descriptions covering the object's class, status, absolute position, spatial relationships, and motion. For the *multi-object grounding* setting, we adopt a more compositional prompt (see Table 9) that encourages descriptions of two objects and their semantic relationships in temporal 3D scenes, covering appearance, motion, and relative spatial configuration.

Manual Verification and Filtering. All generated referring expressions undergo human verification to ensure semantic correctness, referential clarity, and linguistic fluency. To facilitate this process, we develop a custom annotation interface, as shown in Figure 8. Annotators review each expression in the context of the full scene, with the target object visualized via its projected 3D bounding box overlaid on the RGB image. If an expression is partially inaccurate or omits essential details, it may be directly edited. If the description is fundamentally flawed – such as containing hallucinated attributes or being referentially ambiguous – the sample is discarded. This verification process is conducted by a team of five trained annotators to ensure consistency and overall annotation quality.

Platform-Aware Annotation Alignment. To support fair and consistent evaluation across diverse platforms, we adopt a unified annotation protocol for `Vehicle`, `Drone`, and `Quadruped` scenes. Specifically, the same instruction prompt is used across all platforms, ensuring that the generation process follows identical linguistic and visual grounding expectations, regardless of the underlying sensor configuration or viewpoint.

All spatial descriptions in referring expressions are written from the *observer's perspective*, *i.e.*, relative to the camera view that captured the scene. This design allows language like "on the left", "facing away", or "in the front" to remain intuitive and unambiguous to models operating on image-grounded or LiDAR-centered input. Rather than using global scene-relative coordinates (*e.g.*, "north-east corner"), we ensure all position statements are grounded in the visual evidence available from the sensor's viewpoint.

A.3 Examples of Single-Object 3D Grounding

Figure 9, Figure 10, and Figure 11 present representative examples of single-object 3D grounding from the `Vehicle`, `Drone`, and `Quadruped` platforms in our dataset. Each example displays the fused RGB image and LiDAR point cloud, along with a natural language referring expression and its corresponding 3D bounding box.

These examples highlight several key characteristics of the **3EED** dataset:

Table 8: Prompt for Single-Object Grounding

You are an assistant designed to generate fine-grained descriptions for 3D objects grounded in images.

Given a single object highlighted by a bounding box and its class label, please generate a detailed and unambiguous description focusing on the following aspects:

- **1. Class**: Specify the object's type and visual features (e.g., color, shape, vehicle model, clothing of pedestrians).
- **2. Status**: Indicate whether the object is static or in motion, and describe its speed or behavioral state.
- **3. Absolute Position**: Describe the object's location within the image (e.g., bottom-left, center).
- **4. Viewer Perspective**: Explain the object's orientation relative to the camera or viewer (e.g., facing the camera, viewed from behind).
- **5. Spatial Relations**: Outline how the object is situated relative to nearby elements in the scene.
- **6. Moving Direction** (if applicable): Specify whether the object is moving toward or away from the viewer, or turning in a particular direction.

After addressing each aspect, **compose a fluent summary sentence (less than 100 words)** that uniquely identifies the object within the scene.

Response Format:

```
1. class: [...]
2. status: [...]
3. position in the image: [...]
4. relation to the viewer: [...]
5. relationships with other objects: [...]
6. moving direction: [...]
7. Summary: [complete descriptive sentence]
```

Important: Your description should be as specific and detailed as possible. Ensure the response is uniquely aligned with the given object and avoids ambiguity.

- *Cross-platform diversity.* `Vehicle` scenes often feature structured road layouts with multiple traffic participants, such as cars, pedestrians, and motorcycles. `Drone` scenes offer wide-area top-down coverage with more cluttered object distributions, including overlapping vehicles, elevated viewpoints, and richer spatial context. `Quadruped` scenes are recorded from a low-altitude, ground-level perspective, focusing on close-range human interactions and sidewalk-level details.
- *Natural language variation.* Referring expressions reflect platform-specific visibility and spatial reasoning. For example, `Vehicle` -mounted viewpoints encourage descriptions like "on the left side of the street", while `Drone`-based annotations describe objects "in the upper right quadrant" or "viewed from above". `Quadruped` expressions capture nuanced positional cues (*e.g.*, "facing the camera", "walking away on the path") and often describe subtle behaviors or clothing.
- *Scene conditions.* Our dataset includes scenes captured under diverse environmental conditions, including both daytime and nighttime settings. This is evident in the `Vehicle` and `Drone` examples, where objects may be illuminated by streetlights or appear in low-light settings, adding realism and complexity to the grounding task.

Table 9: Prompt for Multi-Object Grounding

You are a multimodal assistant tasked with describing and comparing two objects in a temporal 3D scene.

You are provided with a sequence of images where two objects are marked with green bounding boxes. You will also be given:

- The class label of each object
- A predefined semantic relationship between them

Your task is to describe **each object individually**, and then articulate the relationship between them. Ensure your descriptions are **precise, grounded in visual evidence**, and cover the following perspectives:

- **1. Appearance**: Describe the object's color, texture, size (small, medium, large), shape, category, and material.
- **2. State**: Specify whether the object is moving or static, and describe its current action (e.g., turning, accelerating).
- **3. Spatial Relationship**: Explain its location and relation to nearby scene elements.
- **4. Temporal Movement**: Summarize how the object's position changes across the image sequence.
- **5. Other**: Include any other details that can aid recognition.

Then, describe the relationship between the two objects based on their relative spatial or temporal behavior (e.g., "the car is overtaking the cyclist", "the robot is approaching the chair").

Response Format:

```
Object A:
1. appearance: [...]
2. state: [...]
3. spatial relationship: [...]
4. temporal movement: [...]
5. other: [...]

Object B:
1. appearance: [...]
2. state: [...]
3. spatial relationship: [...]
4. temporal movement: [...]
5. other: [...]

Relationship: [description of how Object A relates to Object B]
```

Important: Focus only on the two marked objects. Your response must be detailed and unambiguous, and should accurately reflect both visual and temporal information.

- *Multi-modal alignments.* Despite differences in viewpoint and density, all annotations maintain strong visual-language grounding. Each expression unambiguously describes a target object with sufficient detail for model disambiguation, including appearance, position, context, and motion when applicable.

These examples demonstrate the richness and difficulty of grounding in our dataset: models must generalize across platforms, lighting conditions, and spatial perspectives while maintaining consistent

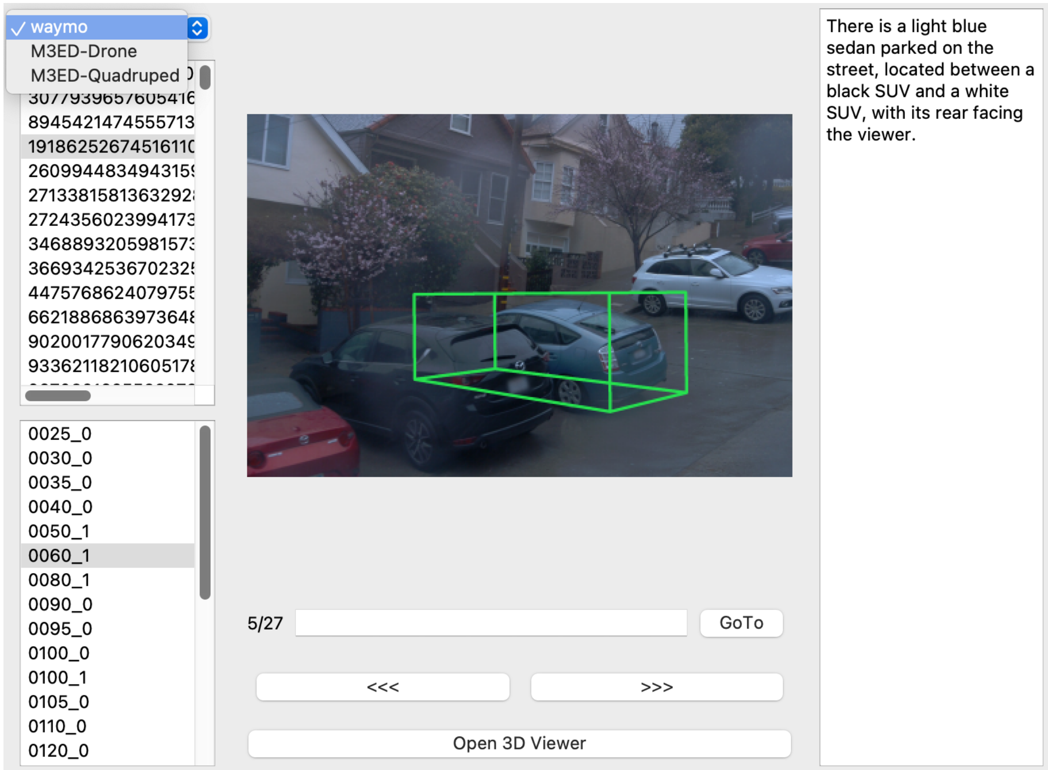


Figure 8: Graphical user interface used during the human refinement phase. Annotators inspect each scene by viewing the 3D bounding box projected onto the RGB image, alongside the automatically generated referring expression. Annotators verify or revise the description to ensure it uniquely and accurately identifies the target object. Scenes failing this verification are discarded.

language understanding. The platform-aware yet prompt-consistent annotation pipeline ensures comparability while preserving diversity.

A.4 Examples of Multi-Object 3D Grounding

Figure 12 presents representative examples from the multi-object grounding subset of our dataset. In this setting, each scene contains two target objects annotated with distinct 3D bounding boxes and described through interrelated referring expressions. These expressions not only characterize each object individually (*e.g.*, class, appearance, motion), but also explicitly capture their spatial, temporal, or semantic relationships.

The examples span a variety of real-world outdoor scenarios involving pedestrians, cyclists, and vehicles. Referring expressions encode rich visual-semantic grounding cues, such as:

- **Relative positioning**: "in front of", "to the right of", "ahead of", "shorter than".
- **Comparative reasoning**: "is larger than", "is taller than", "is shorter than".
- **Temporal context and motion state**: "driving on the road", "stopped at the traffic light", "moving forward".

A.5 Statistics and Analyses

In this section, we present detailed statistics and analyses that characterize the **3EED** dataset across platforms and splits. We examine the distribution of scene complexity, defined by the number of annotated objects per scene, and show how this varies significantly between the `Vehicle`, `Drone`, and `Quadruped` platforms. Additionally, we analyze point-level density within 3D

Figure 9: **Additional examples of 3D grounding** from the `Vehicle` platform in **3EED** dataset. The data shown include the LiDAR point clouds, the RGB frames, and the associated referring expressions. Best viewed in colors and zoomed in for more details.

bounding boxes, highlighting strong differences in LiDAR sampling resolution across platforms. These statistics provide important context for interpreting grounding performance and understanding platform-specific challenges in 3D perception and language grounding.

A.5.1 Scene Complexity Statistics across Platforms

Table 10 presents detailed statistics of the training and validation splits across the three platforms in the **3EED** dataset – `Vehicle`, `Drone`, `Quadruped` platforms. Each scene is categorized by the number of objects it contains, providing insight into the distribution of scene complexity. These statistics are collected on the single-object grounding subset, where only one referred object is annotated per scene.

We observe that `Quadruped` scenes are predominantly sparse, with over 95% of both training and validation scenes containing fewer than 4 objects. Such low-density settings simplify the localization task and reduce ambiguity during reference resolution. In contrast, `Drone` data features a much higher proportion of crowded scenes: over 55% of the training scenes and 60% of the validation

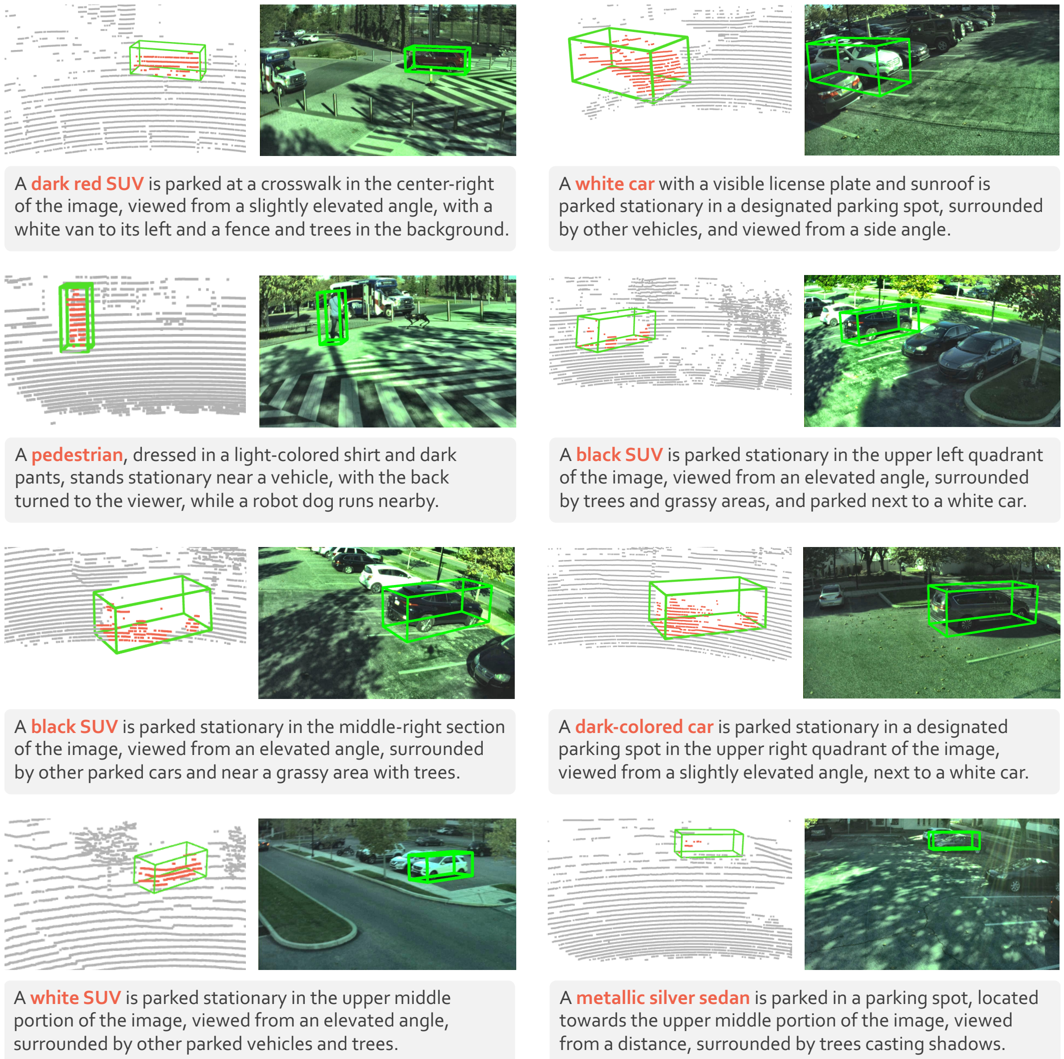


Figure 10: **Additional examples of 3D grounding** from the `Drone` platform in **3EED** dataset. The data shown include the LiDAR point clouds, the RGB frames, and the associated referring expressions. Best viewed in colors and zoomed in for more details.

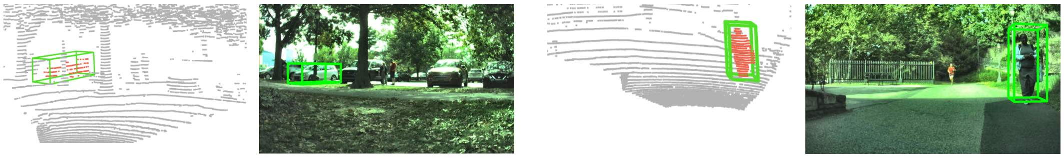
scenes contain 7 or more objects. This reflects the broader aerial perspective and wider field of view, which captures more complex environments and increases grounding difficulty.

The `Vehicle` platform lies between the two, exhibiting a relatively balanced distribution of scene complexities. This makes Vehicle data a valuable middle ground for learning models that generalize across both sparse and dense settings.

Overall, these statistics highlight the diverse spatial configurations in our dataset and provide context for the performance variations discussed in the experiment section of the main paper, particularly in the cross-platform grounding evaluation.

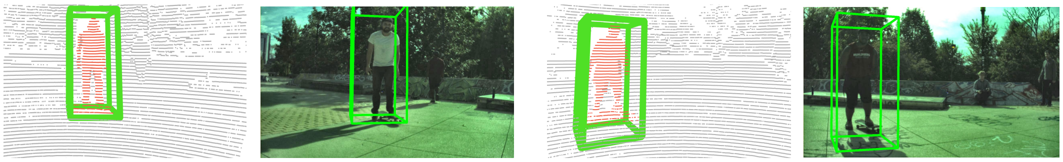
A.5.2 Box Density Statistics

Figure 13 illustrates the distribution of 3D bounding boxes by the number of LiDAR points contained within each box, across the three platforms. The `Drone` platform features extremely sparse boxes, with over 60% containing fewer than 100 points. This is a result of its high-altitude viewpoint and long-range perception, which leads to sparser spatial sampling. Conversely, the `Vehicle` platform



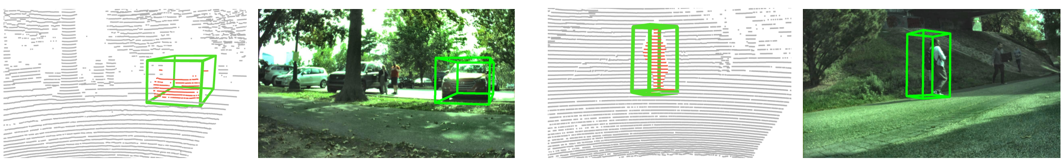
A **white compact hatchback car** is parked stationary in the middle-left portion of the image, surrounded by trees and grass, with pedestrians nearby on a sidewalk.

A **pedestrian** dressed in a light-colored top and dark pants is walking on the right side of the image, near a lamp post and a fence, with another person visible in the background.



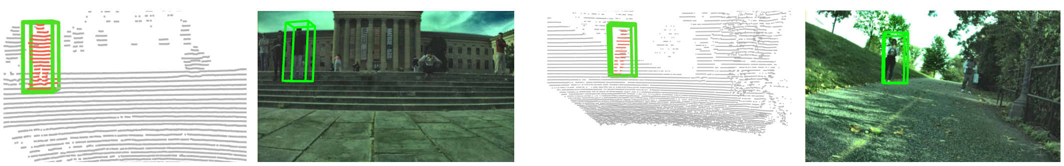
A **person** dressed in a white t-shirt and dark pants stands on a skateboard in the center of the image, facing the camera, with a wall of graffiti and trees in the background.

A **man** wearing a brown top, shorts, and a cap stands on a skateboard in the foreground, facing the camera directly, near a graffiti-covered wall and other skateboarders.



A **stationary Hyundai SUV** is parked on the right side of the image, near the edge of the road and surrounded by trees, with vehicles and pedestrians in the background.

A **pedestrian**, dressed casually with a backpack, is walking away from the viewer along a path that leads towards a bridge or overpass, surrounded by other distant pedestrians.



A **pedestrian**, dressed in a dark shirt and light pants, stands still on the left side of the image, near the top of a staircase, facing away from the viewer towards a large building.

A **pedestrian** is standing on the left side of a path, facing the camera with their body slightly turned to the right, surrounded by trees and greenery.

Figure 11: **Additional examples of 3D grounding** from the `Quadruped` platform in **3EED** dataset. The data shown include the LiDAR point clouds, the RGB frames, and the associated referring expressions. Best viewed in colors and zoomed in for more details.

has more than 28% of boxes with over 900 points, reflecting the dense coverage typical in street-level LiDAR. The `Quadruped` platform occupies a middle ground but still exhibits noticeable sparsity, with a third of its boxes containing fewer than 100 points.

These density differences strongly affect 3D feature quality and grounding performance, especially in low-point regimes where accurate object localization becomes more challenging.

A.6 License

The **3EED** dataset and its associated toolkit are released under the Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)[1] license.

[1] https://creativecommons.org/licenses/by-sa/4.0/legalcode.

Figure 12: **Additional examples of multi-object 3D grounding** from the 3EED dataset.

B Benchmark Construction Details

In this section, we describe how we construct benchmark settings for evaluating 3D language grounding using our dataset. All tasks are formulated in a proposal-free setting, where models must directly predict 3D bounding boxes from point clouds and referring expressions. We also detail the baseline models, training configurations, and evaluation metrics used throughout our experiments.

Table 10: Scene count grouped by number of objects per scene across platforms and splits.

Platform	1–3	4–6	7–9	10+	Total
Training					
Vehicle	1,373	1,058	387	208	3,026
Drone	1,292	1,361	1,631	1,608	5,892
Quadruped	1,886	1,377	0	0	3,263
Total	**4,551**	**3,796**	**2,018**	**1,816**	**12,181**
Validation					
Vehicle	1,290	1,013	433	229	2,965
Drone	911	1,034	846	2,140	4,931
Quadruped	1,690	834	184	0	2,708
Total	**3,891**	**2,881**	**1,463**	**2,369**	**10,604**
Summary	**8,442**	**6,677**	**3,481**	**4,185**	**22,785**

Our goal is to enable fair, controlled, and reproducible comparison across grounding tasks with varying spatial and linguistic complexity.

B.1 Single-Object Grounding Baselines

We compare our approach against two 3D visual grounding baselines adapted to the outdoor point cloud domain: **BUTD-DETR** [31] and **EDA** [80]. Both models were originally proposed for grounding in 3D indoor scenes [15], and we adapt them to our benchmark with raw point cloud input. In all comparisons, we follow a unified setting that does not rely on pre-computed object proposals; each model directly predicts 3D bounding boxes from the raw point cloud and query language.

BUTD-DETR [31] is a transformer-based grounding model that fuses top-down language cues and bottom-up visual features for referential localization. In our setting, we remove the use of region proposals entirely and adapt the model to operate on raw point clouds. The point cloud is encoded using a PointNet++ backbone [61], producing a sequence of 3D-aware visual tokens. The language input is processed by a frozen RoBERTa-base encoder [46], generating contextualized word embeddings. The encoder module uses separate self-attention and cross-attention layers to jointly process language and visual streams. The decoder is composed of transformer layers, where non-parametric queries are derived from the top-K visual tokens based on confidence scores. Each query outputs a 3D bounding box via a regression head that predicts box center and size relative to the anchor point. It supervises the model using a Hungarian matching algorithm that assigns queries to ground-truth boxes. We retain the original box regression and token-level soft alignment loss. The contrastive loss is also included, with a symmetric formulation that aligns all predicted queries to token embeddings and vice versa, following their *not-mentioned* augmentation strategy for unmatched queries.

EDA [80] decomposes each language query into semantic components and explicitly aligns them with point-level features. The model uses the same point encoder as BUTD-DETR [31]. The language input is encoded via a frozen RoBERTa-base model and parsed into three components: object type, visual attributes, and spatial relations. Each component attends to the point features via separate alignment branches, predicting soft attention masks over the point cloud. The decoder aggregates these aligned components through cross-attention and predicts the final 3D bounding box via a regression head. The model is trained with a combination of L1 and GIoU losses for box prediction, along with a multi-branch semantic alignment loss that supervises the consistency between each language component and its corresponding spatial region.

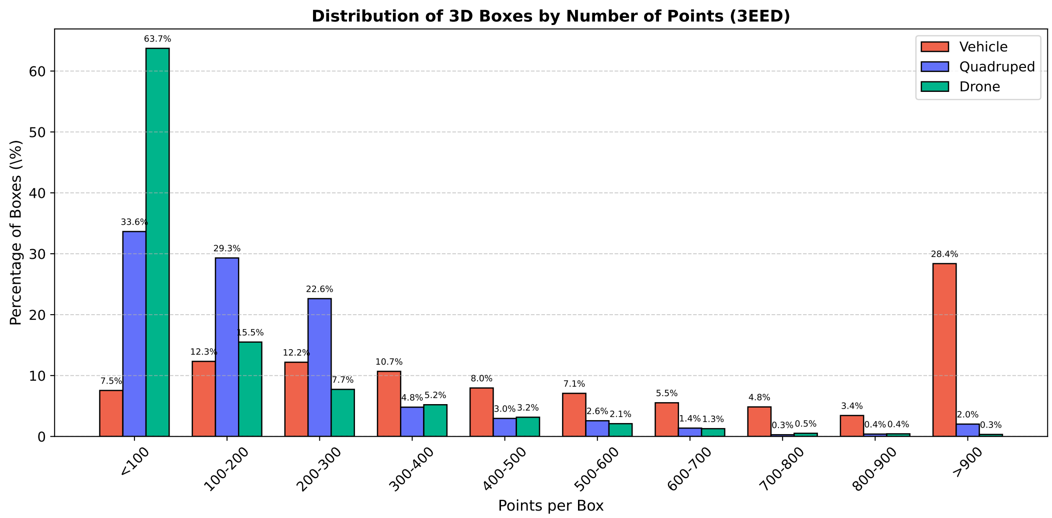


Figure 13: **Distribution of 3D boxes by number of points contained in each box**, across 🚘 `Vehicle`, 🚁 `Drone`, and 🐕 `Quadruped` platforms. Drone boxes are significantly sparser, while Vehicle boxes are generally denser, indicating strong variations in point cloud density across platforms.

B.2 Multi-Object Grounding Baselines

We extend the single-object grounding paradigm to handle multiple objects. Given a natural language utterance and a 3D scene, the model aims to localize all target objects referred to in the input. The core challenge lies in resolving the correspondence between multiple referred entities and their textual descriptions within the utterance.

To address this, we construct a token-level association map that aligns each target object to its corresponding span in the language input. Each object is linked to a binary mask over the token sequence, indicating which words describe it. These masks are normalized to ensure balanced supervision across all objects during training.

Hungarian matching is used to assign predictions to ground-truth boxes. In the single-object case, each scene involves a single reference box. In the multi-object case, matching is performed for each target object separately, with losses computed and averaged across targets.

During inference, the model processes a single utterance that refers to multiple target objects. For each object, we compute the semantic similarity between the candidate boxes and the relevant language span, and select top-ranked boxes based on these similarity scores.

B.3 Implementation Details

Encoder-Decoder. Our model processes raw LiDAR point clouds, which are uniformly downsampled to $16,384$ points per scene. The point cloud is encoded using a four-layer point-based encoder with multi-scale sampling (MSS) and semantic-aware fusion (SAF) modules. The model is trained from scratch without any pretraining. The radius settings for MSS are `[[`$0.2, 0.8$`], [`$0.8, 1.6$`], [`$1.6, 3.2$`], [`$1.6, 4.8$`]]`. Text features are extracted using a frozen RoBERTa-base [37] model, and projected to a 288-dimensional space via a linear projection layer to match the point cloud feature dimension. Language and visual tokens interact through three layers of bidirectional cross-attention. A total of $1,024$ keypoints are sampled from the output of the cross-attention encoder and used as input queries to the decoder. The decoder consists of six transformer layers that iteratively refine 3D box predictions. All boxes are predicted directly from point cloud and language input.

Loss Function. During training, predictions are matched to ground-truth boxes via Hungarian matching as DETR [10], using a cost that combines box ℓ_1 distance, 3D generalized IoU [64], and a soft token-level classification score. The model is supervised using a combination of classification loss, box regression loss, GIoU loss, and a contrastive alignment loss. The contrastive loss is computed between projected visual queries and language tokens using temperature-scaled cosine similarity, with supervision applied in both query-to-token and token-to-query directions. All losses are applied at the decoder outputs.

Training Details. We use AdamW for optimization. For single-object grounding, the learning rate is set to 1×10^{-3} for the point encoder and 1×10^{-4} for all other modules. Training is conducted for 100 epochs on two NVIDIA RTX 4090 GPUs (24 GB each), with a batch size of 12 per GPU. For multi-object grounding, the learning rate is set to 1×10^{-4} for all modules. Training is conducted for 200 epochs on a single RTX 4090 GPU, also with a batch size of 12.

B.4 Evaluation Metrics

To assess grounding performance, we adopt standard IoU-based metrics including **Acc@δ** and **mean IoU (mIoU)**.

Accuracy@IoUδ. Following prior works [12, 1], we compute the percentage of predicted 3D bounding boxes whose Intersection over Union (IoU) with the ground-truth box exceeds a threshold $\delta \in \{0.25, 0.50\}$:

$$\mathrm{Acc@}\delta = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left[\mathrm{IoU}(\hat{b}_i, b_i^{\mathrm{gt}}) > \delta \right],$$

where N is the number of queries, $\hat{b}_i$ is the predicted box, and b_i^{gt} is the ground truth.

Mean IoU (mIoU). To provide a finer-grained measure of localization quality, we also report the mean IoU between the predicted and ground-truth boxes across all queries:

$$\mathrm{mIoU} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{IoU}(\hat{b}_i, b_i^{\mathrm{gt}}) \; .$$

Unlike Acc@δ, which thresholds the overlap, mIoU captures continuous localization precision and is sensitive to small alignment errors. Together, these metrics provide a comprehensive view of grounding performance under both strict and relaxed criteria.

B.5 Evaluation Protocol

To ensure fair and reproducible comparison across models, we standardize the evaluation protocol across four benchmark settings.

- *Single-platform, single-object grounding.* Models are trained and evaluated on the same platform (`Vehicle`, `Drone`, and `Quadruped`), enabling assessment of in-domain performance under consistent sensor geometry and point cloud density. A prediction is considered correct if the predicted bounding box has an Intersection over Union (IoU) above a predefined threshold with the ground-truth box.
- *Cross-platform transfer.* In this setting, models are trained on one platform and evaluated on a disjoint target platform (e.g., train on `Vehicle`, test on `Drone`). The evaluation protocol mirrors that of the single-object setting, enabling controlled assessment of cross-platform generalization.
- *Multi-object grounding.* For queries referring to multiple objects within a scene, the model must predict all corresponding 3D bounding boxes. A prediction is deemed correct only if *all* referred objects are correctly localized with IoU above the threshold. This setting tests the model's ability to handle complex referential expressions and object-object relationships.
- *Multi-platform grounding.* Models are trained jointly on data from all three platforms and evaluated separately on each one. This setting examines the model's robustness to diverse spatial distributions, sensor configurations, and environmental conditions in a unified training regime.

Table 11: **Ablation on Cross-Platform Alignment (CPA).** We train the model on the `Vehicle` platform and evaluate it on `Drone` and `Quadruped` platforms.

CPA	`Drone`		`Quadruped`	
	Acc@25	Acc@50	Acc@25	Acc@50
✗	15.53	9.24	34.54	22.46
✓	18.84	13.03	35.29	27.10
Improve ↑	**+3.31**	**+3.79**	**+0.75**	**+4.64**

Reproducibility. All evaluations are conducted on a fixed validation split with no overlap between training and evaluation scenes. The evaluation pipeline is standardized across all settings, and we release our full codebase and configuration files to support reproducible benchmarking and future comparisons.

C Additional Experimental Results

In this section, we provide extended experimental results to complement the main paper.

C.1 Effectiveness of Cross-Platform Alignment

To evaluate the effectiveness of our Cross-Platform Alignment (CPA) module, we conduct an ablation where the model is trained on the `Vehicle` platform and tested on two unseen platforms: `Drone` and `Quadruped`. As shown in Table 11, removing CPA leads to a noticeable performance drop across both platforms, highlighting the challenge of viewpoint and elevation discrepancies in cross-platform transfer. Specifically, accuracy on the `Drone` platform drops from $18.84/13.03$ to $15.53/9.24$, while on the `Quadruped` platform it decreases from $35.29/27.10$ to $34.54/22.46$. These results validate the importance of aligning gravity and normalizing altitude prior to feature extraction, enabling the model to better generalize to novel embodied viewpoints.

D Additional Visual Comparisons

In this section, we provide more qualitative examples to complement the main results. These visualizations illustrate the strengths and failure patterns of different methods across sensor platforms and grounding settings.

D.1 Qualitative Results for Single-Object 3D Grounding

Figure 14, Figure 15, and Figure 16 present single-object grounding results from the `Vehicle` `Drone` and `Quadruped` platforms, respectively. These comparisons reveal several key insights:

- *Vehicle Platform* (Figure 14). Our method consistently localizes referred objects more accurately, particularly in crowded scenes. For instance, in examples involving parked or moving vehicles near intersections, our model correctly resolves spatial descriptions like "moving forward on the street, positioned near the crosswalk" or "parked on the right side of the street", whereas baseline methods often misplace the box or miss the object entirely.
- *Drone Platform* (Figure 15). Despite the elevated perspective and sparse point clouds, our method produces robust results by leveraging cross-platform cues. Notably, in scenes with occlusions or dense parking lots, our model successfully grounds phrases like "black SUV with grassy area to its left" and "white car with sunroof", demonstrating resilience to complex layouts and ambiguous references. In contrast, EDA and BUTD-DETR frequently fail to produce any box or yield inaccurate boundaries.

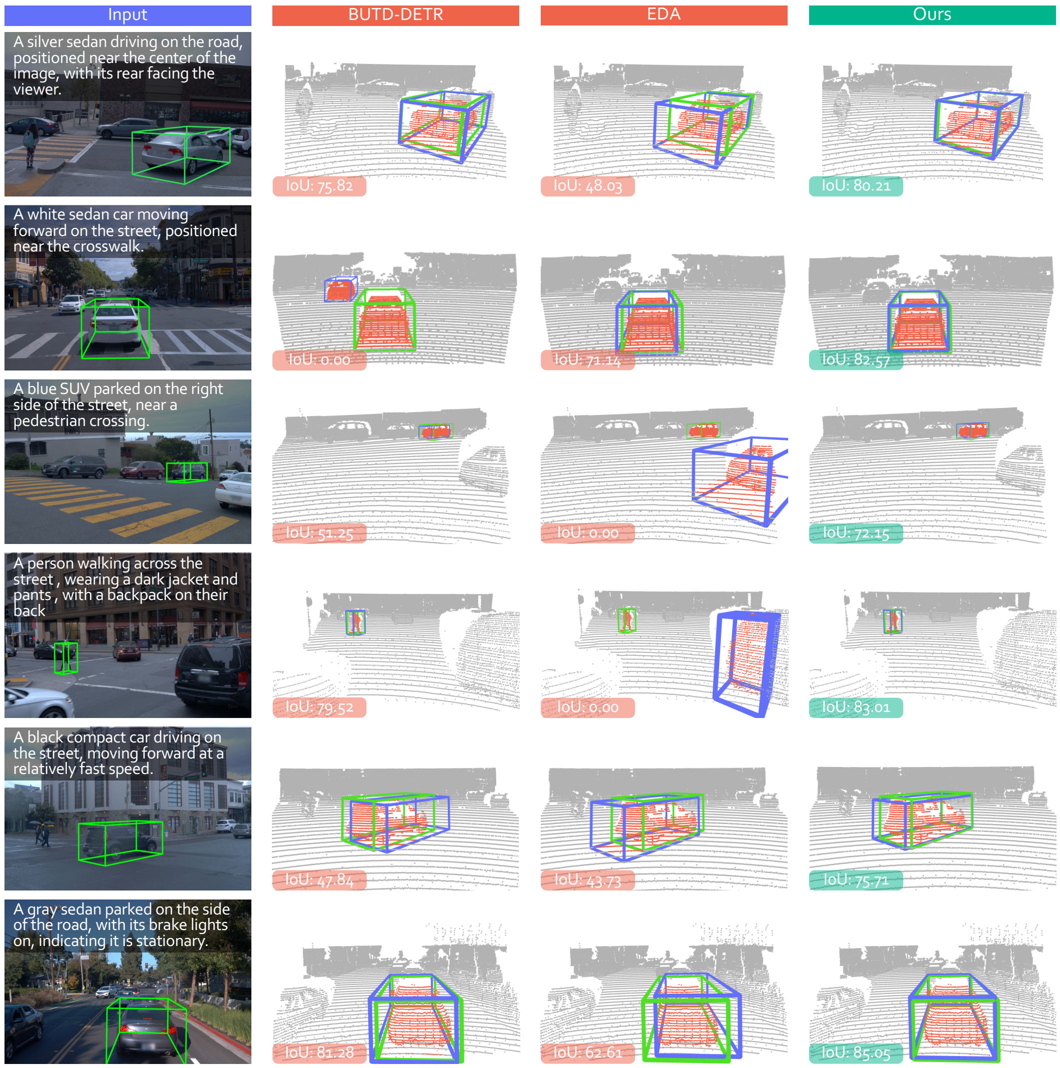


Figure 14: **Additional qualitative comparisons** of single-object 3D grounding on the 🚘 `Vehicle` platform from the **3EED** dataset. The data shown include the RGB frames, the LiDAR point clouds, and the associated referring expressions. The ground truth and predicted boxes are shown in green and blue, respectively. Best viewed in colors and zoomed in for more details.

- *Quadruped Platform* (Figure 16). Grounding from the quadruped perspective introduces unique challenges due to low-angle views and close-range objects. Our method shows clear improvements, accurately grounding pedestrians and vehicles even when facing away from the camera or interacting with the environment. For example, descriptions such as “moving towards a bridge” and “near the edge of the parking lot” are correctly localized only by our approach. Baselines either regress coarse boxes or misinterpret perspective cues.

These qualitative comparisons validate the platform-agnostic design of our approach and demonstrate the ability to disambiguate fine-grained language in diverse visual-spatial contexts.

D.2 Qualitative Results for Multi-Object 3D Grounding

Figure 17 illustrates representative examples from the multi-object grounding setting. Here, each scene contains two referred objects and a complex expression that captures both individual characteristics and inter-object relationships.

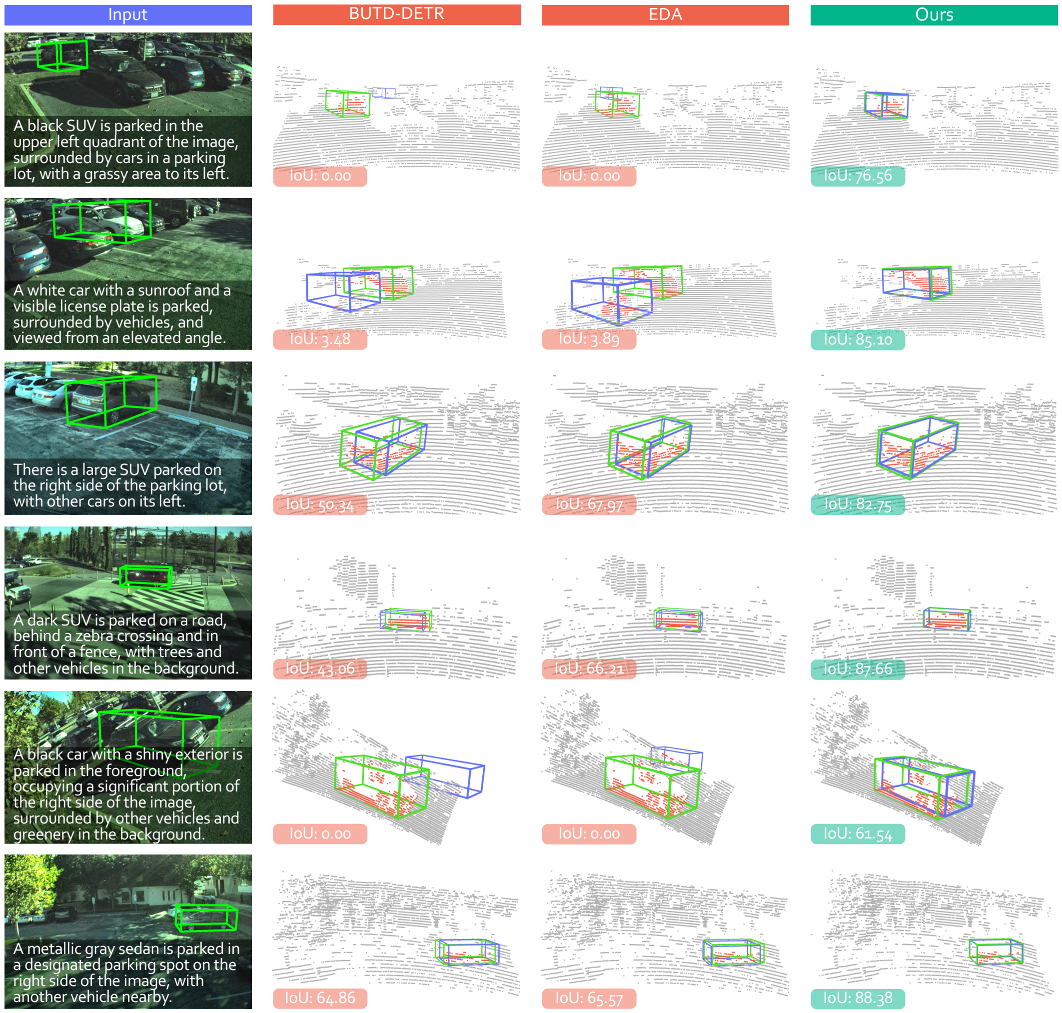


Figure 15: **Additional qualitative comparisons** of single-object 3D grounding on the `Drone` platform from the 3EED dataset. The data shown include the RGB frames, the LiDAR point clouds, and the associated referring expressions. The ground truth and predicted boxes are shown in green and blue, respectively. Best viewed in colors and zoomed in for more details.

Our method shows notable advantages in:

- *Capturing relative semantics:* In expressions like "a white oistal truck is taller than a yellow car" or "a silver sedan is to the left of a red car", our model localizes both objects with high precision and correct relative positioning.
- *Handling comparatives and prepositions:* Even in cases with overlapping objects or subtle distinctions, our method interprets spatial relations (*e.g.*, "to the left of", "is behind") more reliably than baselines.
- *IoU consistency:* The paired IoU scores (IoU1/IoU2) of our predictions are consistently higher, reflecting better localization and object differentiation.

In contrast, BUTD-DETR [31] often fails to detect one of the objects, while EDA [80] tends to confuse spatial hierarchy, misplace referred instances, or miss the relationships altogether.

Overall, these visual results demonstrate that our model excels not only in individual object grounding but also in multi-entity reasoning, which is crucial for real-world applications requiring collaborative spatial understanding.

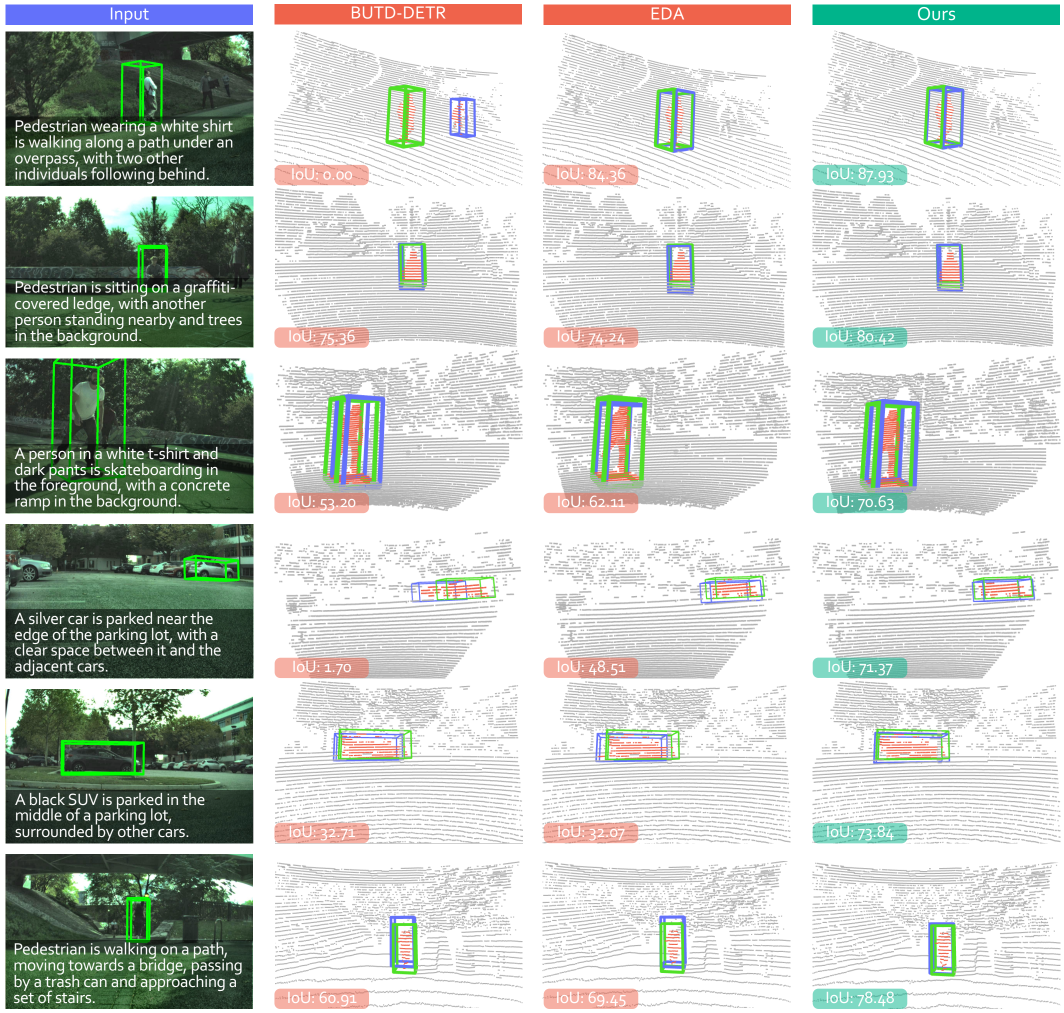


Figure 16: **Additional qualitative comparisons** of single-object 3D grounding on the `Quadruped` platform from the **3EED** dataset. The data shown include the RGB frames, the LiDAR point clouds, and the associated referring expressions. The ground truth and predicted boxes are shown in green and blue, respectively. Best viewed in colors and zoomed in for more details.

E Broader Impact & Limitations

In this section, we elaborate on the broader impact, societal influence, and potential limitations.

E.1 Broader Impact

This work introduces a new benchmark and methodology for 3D visual grounding across diverse robotic platforms, including vehicles, drones, and quadrupeds. By addressing cross-platform perception and grounding under real-world sparsity, we hope to inspire future research in robust, generalizable spatial language understanding. The dataset and evaluation settings reflect realistic conditions encountered by embodied agents in autonomous driving, inspection, and delivery. We expect this work to benefit the development of safe, context-aware decision-making systems that can interpret human intent across environments. All data collection and annotation followed privacy-compliant and publicly accessible sources.

Figure 17: **Additional qualitative comparisons** of multi-object 3D grounding approaches on the 3EED dataset. The data shown include the RGB frames, the LiDAR point clouds, and the associated referring expressions. The ground truth and predicted boxes in the prediction results are shown in green and blue, respectively. Best viewed in colors and zoomed in for more details.

E.2 Societal Influence

The ability to ground language in 3D scenes is critical for real-world human-robot interaction, especially in complex outdoor scenarios. Our benchmark enables evaluating such capabilities beyond indoor or single-device assumptions, pushing toward a more inclusive and scalable understanding. Potential downstream applications include collaborative navigation, voice-based robotics control, and assistive technologies in search-and-rescue operations. While our dataset promotes progress in these areas, we note that grounding models trained on limited sensory conditions may inadvertently inherit biases from pretrained language models or overlook vulnerable populations in data-scarce environments.

E.3 Potential Limitations

Despite its scale and diversity, our dataset may still suffer from platform-specific biases (*e.g.*, drone views emphasizing sparse or elevated contexts), which could limit generalization. The current version focuses primarily on static scenes with one or more referred objects, without modeling temporal dynamics or dialogue-based interaction. In addition, our evaluation settings assume accurate text descriptions and do not yet account for ambiguous, contradictory, or noisy language input. Furthermore, while our benchmark covers three robotic platforms, generalization to other types of sensors or modalities (*e.g.*, thermal, event cameras) remains unexplored.

F Public Resource Used

In this section, we acknowledge the use of the public resources, during the course of this work:

F.1 Public Datasets Used

- M3ED[2] .... CC BY-SA 4.0
- Waymo Open Dataset[3] .... Apache License 2.0

F.2 Public Implementation Used

- BUTD-DETR[4] .... CC BY-SA 4.0 License
- EDA[5] .... CC BY-SA 4.0 License
- Open3D[6] .... MIT License
- PyTorch[7] .... BSD License
- Pointnet2 PyTorch [8] .... UNLICENSE
- PointNet++[9] .... MIT License
- xtreme1[10] .... Apache License 2.0
- WildRefer [11] .... CC BY-SA 4.0 License

[2] https://m3ed.io.
[3] https://github.com/waymo-research/waymo-open-dataset.
[4] https://github.com/nickgkan/butd_detr.
[5] https://github.com/yanmin-wu/EDA.
[6] http://www.open3d.org.
[7] https://pytorch.org.
[8] https://github.com/erikwijmans/Pointnet2_PyTorch.
[9] https://github.com/charlesq34/pointnet2.
[10] https://github.com/xtreme1-io/xtreme1.
[11] https://github.com/4DVLab/WildRefer.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Both contributions and scope have been discussed in abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The detailed analysis on limitations have been discussed in the appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This is an empirical study that excludes theory assumptions and proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All information needed to reproduce the experimental results have been disclosed. To ensure reproducibility, code and data are committed to be publicly available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The detailed implementation procedures have been included in the appendix. To ensure reproducibility, code and data are committed to be publicly available.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and test details have been discussed in either main body or appendix. To ensure reproducibility, code and data are committed to be publicly available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Sufficient information about experiment settings have been discussed.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The details on computing resources have been discussed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This research follows the NeurIPS Code of Ethics properly.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The discussion on societal impacts has been included in the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The discussion on safeguards has been included in the appendix.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The acknowledgments on licenses have been included in the appendix.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The discussions on new assets have been included in the appendix.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.