ReferGPT: Towards Zero-Shot Referring Multi-Object Tracking

Tzoulio Chamiti[1,2*] Leandro Di Bella[1,2*] Adrian Munteanu[1,2] Nikos Deligiannis[1,2]
[1]ETRO Department, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium
[2]imec, Kapeldreef 75, B-3001 Leuven, Belgium
*Equal Contributions
{tzoulio.chamiti, leandro.di.bella, adrian.munteanu, nikos.deligiannis}@vub.be

Abstract

Tracking multiple objects based on textual queries is a challenging task that requires linking language understanding with object association across frames. Previous works typically train the whole process end-to-end or integrate an additional referring text module into a multi-object tracker, but they both require supervised training and potentially struggle with generalization to open-set queries. In this work, we introduce ReferGPT, a novel zero-shot referring multi-object tracking framework. We provide a multi-modal large language model (MLLM) with spatial knowledge enabling it to generate 3D-aware captions. This enhances its descriptive capabilities and supports a more flexible referring vocabulary without training. We also propose a robust query-matching strategy, leveraging CLIP-based semantic encoding and fuzzy matching to associate MLLM generated captions with user queries. Extensive experiments on Refer-KITTI, Refer-KITTIv2 and Refer-KITTI+ demonstrate that ReferGPT achieves competitive performance against trained methods, showcasing its robustness and zero-shot capabilities in autonomous driving. The codes are available on https://github.com/Tzoulio/ReferGPT

1. Introduction

Referring Multi-Object Tracking (RMOT) has emerged as a crucial problem in computer vision, particularly in autonomous driving, surveillance and human-machine interaction. Unlike the standard Multi-Object Tracking (MOT) task [9, 24], which associates detections across frames without explicit user guidance, RMOT introduces textual queries to specify which objects to track [13, 34, 42]. For instance, given a user query such as "the blue car turning right", an RMOT system must identify and track the relevant object across a sequence of frames. This capability is particularly valuable in autonomous driving and intelligent traffic monitoring, where integrating natural language guidance enhances situational awareness and enables more intuitive human-machine interaction in complex environments.

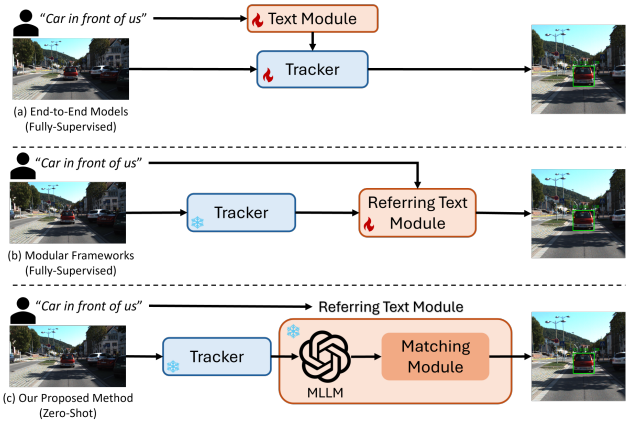

Figure 1. Comparison between ReferGPT and previous RMOT methods. (a) End-to-end models jointly learn tracking and referring. (b) Tracking-by-detection frameworks follow a modular approach and train the referring text module. (c) Our method builds on (b) while eliminating the need for training, enabling zero-shot referring MOT.

Generally, RMOT approaches can be divided into two categories, as seen in Fig.1. The first category (Fig.1(a)) consists of end-to-end transformer-based frameworks, such as TransRMOT [34], where a single supervised unified model performs both object tracking and referring. The second category, depicted in Fig.1(b), includes modular frameworks that follow a tracking-by-detection paradigm, such as iKUN [8], where a dedicated supervised referring module is introduced into an existing tracker. However, while both categories have shown promising results, they remain constrained by two fundamental limitations. First, supervised RMOT methods tend to generalize poorly due to their reliance on a closed-set of predefined queries, limiting their ability to handle open-set ones. Second, 2D-based methods struggle with spatial alignment, particularly for queries that contain explicit spatial information. For ex-

ample, queries such as “cars which are faster than ours” or “the car is moving away from us”, are difficult to refer to using only 2D representations. Additionally, end-to-end joint-detection-and-tracking methods often experience task interference, which degrades association performance. Tracking-by-detection methods remain more reliable, as separating detection and association allows for independent optimization of each task [7, 30].

To address these limitations, we propose ReferGPT, a zero-shot referring MOT framework that combines multi-object tracking through Kalman filtering [31] with a Multi-Modal Large Language Model (MLLM)-based referring modules for query matching. Unlike traditional RMOT methods [8, 34, 42] that operate solely in the 2D space, our approach leverages spatial information from a 3D tracker [35] to enrich the MLLM’s understanding of the scene, allowing it to generate structured object captions, which are in turn matched with the appropriate user queries. Specifically, we incorporate 2D images as visual cues, together with 3D motion and position knowledge through the MLLM prompt, to generate semantically rich yet spatially aware object captions. This is critical for handling queries with depth-dependent constraints, such as ”blue cars that are moving”.

Furthermore, to match the MLLM caption with the user query, we introduce a matching module that combines semantic understanding through CLIP encoding [17] with deterministic data association using fuzzy matching. This allows us to align captions and queries even when they use semantically related terms, like ”the vehicle in front of our car” with ”the automobile ahead of us” providing a more robust matching process. In summary, our contributions are as follows.

- We introduce ReferGPT, a zero-shot Referring Multi-Object Tracking framework capable of handling arbitrary textual queries, without requiring training, for both 2D and 3D inputs. To the best of our knowledge, this is the first work on zero-shot RMOT in autonomous driving.
- We demonstrate the effectiveness of Multi-Modal Large Language Models (MLLMs) in generating spatially grounded text, which enhances query-based tracking performance.
- We perform extensive evaluations on the Refer-KITTI, Refer-KITTIv2 and Refer-KITTI+ datasets, demonstrating that our framework achieves competitive performance without relying on supervised training, highlighting its zero-shot capabilities on the autonomous driving setting.
- We conduct multiple ablation studies showing that each component of our proposed method contributes non-trivially.

2. Related Work

Multi-Object Tracking: Multi-Object Tracking methods can be broadly categorized into Joint Detection and Tracking (JDT) [25], [26], [36], [29], [16], [43] and Tracking-by-Detection (TBD) [18], [32], [35], [4], [28] approaches. JDT methods combine detection and tracking into a unified framework, jointly optimizing object localization and association across frames. Some JDT methods operate in 2D, such as CenterTrack [43], which integrates spatio-temporal memory for short-term association, and PermaTrack [25], which maintains object locations under full occlusions. Others extend JDT to 3D, such as MMF-JDT [29], which integrates object detection and multi-object tracking into a single model, eliminating the traditional data association step by predicting trajectory states and PC-TCNN [36] which generates tracklet proposals, refines them, and associates them to perform multi-object tracking. However, these approaches demonstrate performance limitations compared to TBD methods. TBD methods decouple detection from tracking, allowing for independent optimization and greater flexibility. Specifically, HybridTrack [7] combines deep learning with a learnable Kalman Filter to dynamically adjust motion parameters, MCTrack [39] employs a two-stage matching process that combines bird’s-eye view and image-plane matching to improve robustness against depth errors and PC3T [35] uses a confidence guided data association module for the tracking task. This motivates our choice of TBD as our tracking paradigm, which offers greater flexibility and improved performance.

Referring Multi-Object Tracking: The first work on Referring Multi-Object Tracking (RMOT), TransRMOT [34], introduces an end-to-end transformer framework that uses language expressions as semantic cues to track referred objects frame by frame. TempRMOT [42] extends this by adding a temporal enhancement module to better handle motion across frames, while DeepRMOT [10] incorporates deep cross-modal fusion to improve how visual and linguistic features interact. ROMOT [12] further leverages multi-stage cross-modal attention and vision-language modeling, enabling the tracking of both known and novel objects based solely on descriptive attributes. MLS-Track [15] enhances cross-modal learning by progressively integrating semantic information into visual features at multiple stages of the model. MGLT [5] combines linguistic, temporal, and tracking cues to generate object queries and enhance visual-language alignment. All of the above methods follow an end-to-end approach, integrating tracking and referring into a unified model to leverage cross-modal interactions, but they often lack flexibility.

iKUN [8] was the first method to deviate from the traditional RMOT paradigm by introducing a modular approach. Specifically, iKUN proposed using an existing pre-trained tracker and combining it with a trainable referring text module. MEX [27] later extended this idea and further optimized the cross-modality attention for better computational efficiency of the framework. However, both approaches

train the referring module, which limits their adaptability and requires retraining when faced with out-of-distribution queries. In contrast, ReferGPT operates in a zero-shot setting and removes the need to train the referring text module. This allows for more flexible referring, without the need for task-specific supervision or retraining.

3. Methodology

3.1. Problem Formulation and Method Overview

Given a sequence of K frames $\mathbf{I} = \{I_t\}_{t=1}^{K}$ and a referring text query $\mathbf{Q} = [q_1, q_2, \ldots, q_m] \in \mathbb{R}^m$ where each q_i represents a word in the sequence, our goal is to identify and track objects that match the given query. Using an off-the-shelf object detector $\mathcal{F}_{\text{det}}(\cdot)$, we generate a set of N_t candidate detections per frame. Our method is designed to work with any object detector that produces 3D outputs, regardless of whether the input data comes from LiDAR or RGB images. Next, we employ a tracking-by-detection framework $\mathcal{F}_{\text{trk}}(\cdot)$, as shown in Fig.2, which identifies and keeps track of all true detections. For every identified object, we use an MLLM agent to generate a textual caption D, leveraging 3D motion and position information from the tracker along with the cropped image I_c of the detected object. The generated description and the referring query are then processed by our matching module, which computes a similarity score S_T to determine their alignment. After all frames have been processed, we filter and associate the set of detected objects to the input query, through clustering, based on their computed similarity score.

In the following sections, we describe the usage of $\mathcal{F}_{\text{det}}(\cdot)$ and $\mathcal{F}_{\text{trk}}(\cdot)$ in Sec.3.2. Our MLLM agent is introduced in Sec.3.3. We detail our matching module in Sec.3.4 and we present our filtering process in Sec.3.5.

3.2. 3D Object Detection and Multi-Object Tracking

The tracking-by-detection approach tracks objects by associating detections extracted from a 3D object detector at each timestep t. Formally, given an input frame I_t at timestep t, the detector $\mathcal{F}_{\text{det}}(\cdot)$ produces a set of N_t detections:

$$R_t = \mathcal{F}_{\text{det}}(I_t), \quad R_t = \{\mathbf{r}_t^i\}_{i=1}^{N_t} \in \mathbb{R}^{N_t \times F} \tag{1}$$

where R_t represents the set of detected objects, N_t is the number of detections at timestep t, and F is the number of attributes describing each detection. Each detected object $\mathbf{r}_t^i \in \mathbb{R}^F$ corresponds to a 3D bounding box, parameterized as: $\mathbf{r}_t^i = [x, y, z, w, l, h, \theta] \in \mathbb{R}^7$ where (x, y, z) are the 3D centroid coordinates, (w, l, h) are the width, length, and height, and θ is the heading angle.

These detections serve as the basis for tracking objects over time. To maintain and update object trajectories, we employ a Kalman filter-based [31] multi-object tracking module denoted as $\mathcal{F}_{\text{trk}}(\cdot)$. At each timestep t, $\mathcal{F}_{\text{trk}}(\cdot)$ initializes new trajectories for objects that have been newly detected and do not correspond to any existing track. Additionally, for objects that are already being tracked, it predicts their future states $\hat{X}_t = \{\hat{\mathbf{x}}_t^i\}_{i=1}^{M_t} \in \mathbb{R}^{M_t \times F}$ where M_t is the number of objects currently being tracked at time t. Each trajectory T^i for object i is defined as a sequence of its estimated states $T^i = \left[\mathbf{x}_{t_{\text{init}}}^i, \ldots, \mathbf{x}_t^i\right]$ where t_{init} denotes the timestep when object i was first detected and its trajectory was initialized.

Next, detections R_t are associated with $\hat{X}_t$ using a confidence-guided association strategy [35], which prioritizes high-confidence matches to improve robustness against false positives and occlusions. Once detections are assigned, tracked states, denoted as $X_t = \{\mathbf{x}_t^i\}_{i=1}^{M_t} \in \mathbb{R}^{M_t \times F}$, are updated by incorporating the new associated measurements, refining object localization. Finally, a trajectory management process handles track initiation, termination, and re-identification to ensure stable tracking, even in cases of temporary occlusion or missing detections.

3.3. Multi-Modal Large Language Agent

We define the Multi-Modal Large Language Agent as $\mathcal{F}_{\text{LLM}}(I_c^i, \mathbf{P}, \mathbf{C}_t^i)$ as shown in Fig.2, where I_c^i is the cropped image of the specific object and $\mathbf{P}$ denotes a predefined prompt provided to the MLLM in the form of a text sequence $\mathbf{P} = [p_1, p_2, \ldots, p_n] \in \mathbb{R}^n$ with each p_i being a word in the prompt. $\mathbf{C}_t^i$ is a compact representation of the i-th object's spatial and motion statistics over the most recent T frames. To ensure spatial consistency, we perform a coordinate transformation that converts tracked object states X_t from the global (world) coordinate system to an ego-centric reference frame, aligned with the ego vehicle. Specifically, the tracked states $x_t^i \in X_t$ are transformed into relative positions $\mathbf{p}_t^i \in \mathbb{R}^3$.

In this ego-centric coordinate system, $\mathbf{C}_t^i \in \mathbb{R}^9$ consists of the current position $\mathbf{p} = (x, y, z)$ at t_0, with t_0 denoting the current frame, the average heading angle $\bar{\theta}$ of the past T=5 frames, the Euclidean distance between the current state $\mathbf{x}_{t_0}$ and the state at frame $t_0 - T$, the mean heading angle variation $\bar{\Delta\theta}$, and the spatial variations $\Delta\mathbf{p} = (\Delta x, \Delta y, \Delta z)$ computed across the time window T:

$$\mathbf{C}_t^i = \left[\mathbf{p}, \bar{\theta}, d_{\text{euclid}}, \Delta\mathbf{p}, \bar{\Delta\theta}\right], \tag{2}$$

This compact representation is then flattened into a 1-D text sequence and provided as input to the MLLM. By summarizing both the object's current state and its recent motion over the temporal window T, $\mathbf{C}_t^i$ enables the agent to reason about dynamic behaviors such as being stationary, moving forward, parking, or turning. The agent, in turn, generates a descriptive output text sequence $\mathbf{D}_t^i = [d_1, d_2, \ldots, d_n]$

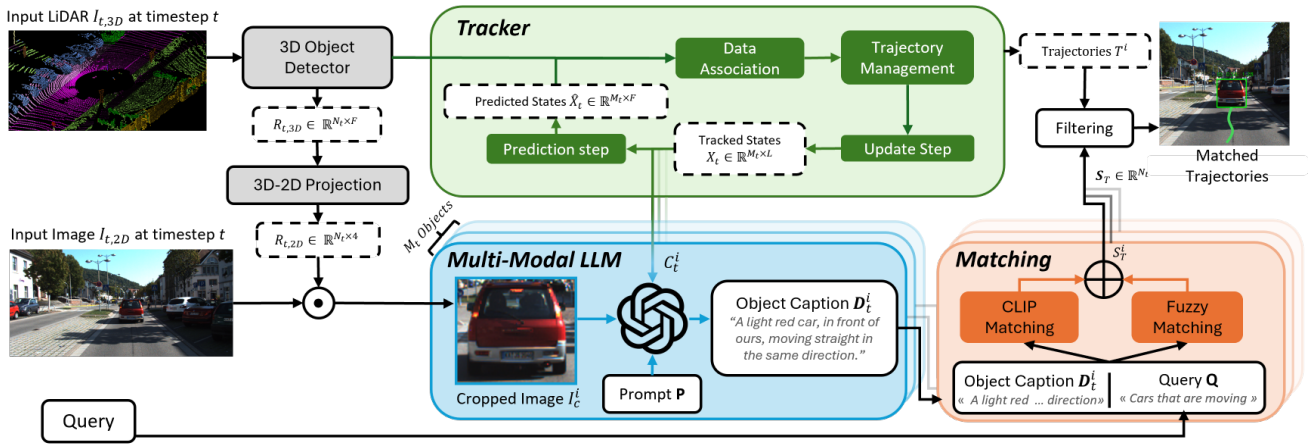

Figure 2. Overview of the proposed ReferGPT framework. Given LiDAR $I_{t,3D}$ and image $I_{t,2D}$ inputs, a 3D object detector extracts object candidates $R_{t,3D}$, which are then tracked using a tracking-by-detection approach with a Kalman filter for trajectory prediction. A Multi-Modal Large Language Model generates descriptive captions $\mathbf{D_t^i}$ for each object by leveraging object coordinates C^i and appearance features I_c^i. These captions are then matched against the referring query $\mathbf{Q}$ using a matching module. The final matched trajectories T^i are filtered and associated with the query to produce the final output.

$\in \mathbb{R}^n$ for each tracked object, where each d_i represents a word in the sequence.

In summary, leveraging the LLM's reasoning and natural language generation capabilities, we translate the spatial and kinematic properties of each object into a natural language description. These descriptions capture attributes such as object color, object type (e.g., car, pedestrian), relative location with respect to the camera, movement status, and directional information. The resulting descriptions serve as intermediate representations of the scene through text and are later used during the matching process, where they are compared with the referring query to identify the corresponding object.

3.4. Matching

As shown in Fig.3, to compute the similarity between each detected object's description generated by the MLLM agent $\mathbf{D}_t^i$ and the referring query $\mathbf{Q}$, we employ a hybrid matching approach that combines fuzzy matching with semantic embedding-based similarity. We first apply the Ratcliff/Obershelp algorithm [19] to compute a fuzzy matching score between each word q_k in the query $\mathbf{Q}$ and each word d_k in the object description $\mathbf{D}_t^i$. The algorithm identifies the longest common contiguous subsequences of characters between word pairs to determine their similarity score. The similarity score for a pair of words (q_k, d_k) is defined as:

$$S_k = \frac{2 \cdot M_k}{|q_k| + |d_k|}, \tag{3}$$

where M_k represents the total number of matching characters in the longest common subsequences, and $|q_k|$ and $|d_k|$ are the lengths of the query word and the description word, respectively. Finally, the overall fuzzy matching score S_F between the query $\mathbf{Q}$ and the description $\mathbf{D}_t^i$ is defined as the sum of all per-word scores: $S_F = \sum_{k=1}^{m} S_k$

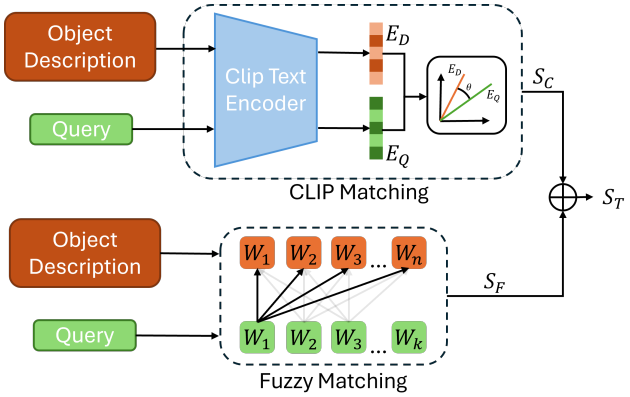

Figure 3. Our Matching Module. Given an object description $\mathbf{D}_t^i$ and a referring query $\mathbf{Q}$, we calculate the total matching score S_T between them.

This technique prioritizes structural similarity by emphasizing shared contiguous substrings. However, fuzzy matching remains limited in handling synonyms and ambiguous terms, as it relies on direct character sequence comparisons rather than true semantic understanding. To address this limitation, we leverage the textual encoder of CLIP to encode both the object description and the referring query. Specifically, we obtain vector embeddings E_D and E_Q for $\mathbf{D}_t^i$ and $\mathbf{Q}$, respectively, and compute their sim-

ilarity using the cosine similarity metric:

$$S_C = \frac{E_D \cdot E_Q}{\|E_D\|\|E_Q\|}, \tag{4}$$

where S_C represents the CLIP similarity score. By incorporating CLIP, we enhance the robustness of the matching process, enabling the model to handle synonyms, contextual variations, and ambiguous phrases that fuzzy matching alone may struggle to resolve. The final matching score is computed as $S_T = S_C + S_F$.

3.5. Filtering Optimization

Upon track termination, each associated detection $\mathbf{x}_t^i$ is characterized by its coordinates and a similarity score S_T^i inferred during the matching step, indicating whether the detected object's description aligns with the given textual query $\mathbf{Q}$. To mitigate the impact of noisy detections and reduce false negatives, we first apply a majority voting strategy: for every tracked object, if the majority of detections within the object trajectory T_i are classified as matching the query, the entire trajectory is inferred to correspond to the query, effectively suppressing false negative matched detections.

Following this filtering step, a key challenge arises from the logit-based similarity scores produced by the matching module. Since the similarity score varies across different queries, depending on query length and the response characteristics of the $\mathcal{F}_{\text{LLM}}$, using a fixed threshold for all queries is suboptimal. An alternative approach could be to select the top-k tracks with the highest similarity scores. However, this strategy assumes prior knowledge of how many detections truly satisfy the query, which is unknown. To address this, we employ an agglomerative hierarchical clustering algorithm [3] to dynamically identify the subset of tracks having the highest similarity scores. This adaptive clustering approach ensures that the most relevant trajectories are selected without relying on a rigid threshold, thereby enhancing the flexibility and generalization of the matching process.

4. Experimental Setup

Dataset and Evaluation Metrics. We conduct our experiments on three benchmark datasets derived from Refer-KITTI. First, we evaluate on the Refer-KITTI-v1 public dataset [34], using its test split, which consists of 3 videos and 150 diverse natural language queries. Second, we assess our method on Refer-KITTI-v2 [42], an extension of v1 featuring a more challenging set of queries. We report results on its pre-defined test split, comprising 4 videos and 859 queries, to demonstrate our method's robustness in more complex settings. Additionally, we evaluate on Refer-KITTI+ [13], following the same split protocol as EchoTrack [13], which includes 3 videos and 154 queries.

We primarily focus on the Higher Order Tracking Accuracy (HOTA) metric [14], which provides a balanced measure of detection, association and localization accuracy in multi-object tracking. It is defined as HOTA $= \sqrt{\text{DetA} \cdot \text{AssA}}$. DetA is the Detection Accuracy [14] emphasizing the accuracy of object detection, while AssA is the Association Accuracy [14] explicitly measuring the association's effectiveness in maintaining track consistency. We also provide the Localization Accuracy (LoCA) which describes how accurately the objects' spatial positions are estimated.

Implementation Details. In our work, we leverage LiDAR input frames for 3D object detection and employ CasA as our object detector [37]. We deploy the model-based PC3T tracker [35] within a tracking-by-detection framework to associate detections across frames. As our MLLM, we employ GPT-4o-mini for its enhanced reasoning and captioning capabilities. For textual encoding, we use CLIP ViT-L/14 [17], which has demonstrated improved representation capabilities due to its transformer-based architecture and larger model size.

5. Experiments

Tab. 1 presents a comprehensive comparison of existing methods on the Refer-KITTI dataset. Our proposed method demonstrates competitive performance despite operating in a zero-shot setting. Unlike other models that rely on task-specific training, ReferGPT generalizes to the referring multi-object tracking task without any training. Notably, by using 3D LiDAR data as input, we achieve a HOTA score of 49.46%. When using 2D as input, our method scores 46.36% HOTA. We also achieve the highest Association Recall (AssRe) scores, highlighting the ability to consistently maintain object identities over time. Despite being zero-shot, ReferGPT delivers competitive results even across detection and localization metrics, with strong DetA and LocA scores.

Tab. 2 and Tab. 3 present the performance of our proposed method on the Refer-KITTIv2 [42] and Refer-KITTI+ [13] datasets, respectively. These datasets introduce additional, more complex queries, posing significant challenges for models not explicitly trained on them. Despite this, ReferGPT demonstrates strong generalization capabilities in handling such open-set referring queries. Specifically, we achieve a competitive HOTA score of 30.12%, closely matching supervised end-to-end approaches such as TransRMOT [34] 31.00%. Notably, ReferGPT scores the highest association accuracy (AssA) of 59.02%, significantly surpassing both TempRMOT [42] and TransRMOT [34]. For the Refer-KITTI+ dataset, ReferGPT sets a new benchmark in HOTA 43.44%, in DetA 29.89% and AssA 63.60%, reaffirming our method's potential in open-set query tracking.

Table 1. Comparison of existing methods on Refer-KITTI dataset [34]. The best is marked in **bold**, and the second-best in underline. 'E' indicates End-to-End methods. The results are reported in %. ReferGPT$_{3D}$ uses LiDAR as input, ReferGPT$_{2D}$ uses Image as input.

Method	E	HOTA ↑	Detection			Association			LocA
			DetA ↑	DetRe ↑	DetPr ↑	AssA ↑	AssRe ↑	AssPr ↑	
Traditional Methods									
FairMOT [40]	×	22.78	14.43	16.44	45.48	39.11	43.05	71.65	74.77
DeepSORT [33]	×	25.59	19.76	26.38	36.93	34.31	39.55	61.05	71.34
ByteTrack [41]	×	24.95	15.50	18.25	43.48	43.11	48.64	70.72	73.90
TransTrack [23]	×	32.77	23.31	32.33	42.23	45.71	49.99	78.74	79.48
Supervised Learning Methods									
iKUN [8]	×	48.84	35.74	51.97	52.26	**66.80**	72.95	87.09	-
MEX [27]	×	45.07	32.81	**62.52**	41.65	-	71.09	-	-
TransRMOT [34]	✓	46.56	37.97	49.69	60.10	57.33	60.02	**89.67**	90.33
EchoTrack [13]	✓	48.86	41.26	53.42	**62.83**	57.59	61.61	89.33	**90.74**
DeepRMOT [10]	✓	39.55	30.12	41.91	47.47	53.23	58.47	82.16	80.49
TempRMOT [42]	✓	**52.21**	**43.73**	55.65	59.25	66.75	71.82	87.76	90.40
ROMOT [12]	✓	35.5	28.30	-	-	46.2	-	-	-
MGLT [5]	✓	49.25	37.09	-	-	65.50	-	-	-
Zero-Shot Learning Methods									
Baseline*	×	21.42	9.48	16.10	18.20	48.82	57.86	72.57	80.72
ReferGPT$_{2D}$ (Ours)	×	46.36	36.58	51.40	52.16	59.00	73.16	69.31	83.26
ReferGPT$_{3D}$ (Ours)	×	49.46	39.43	50.21	58.91	62.57	**73.74**	72.78	81.85

* *Baseline* uses only CLIP-Image encoder for similarity evaluation, without the MLLM-Agent.

Table 2. Comparison of existing methods on Refer-KITTIv2 dataset [42]. The best is marked in **bold**, and the second-best in underline. 'E' indicates End-to-End methods. The results are reported in %.

Method	E	HOTA ↑	Detection			Association			LocA
			DetA ↑	DetRe ↑	DetPr ↑	AssA ↑	AssRe ↑	AssPr ↑	
Traditional Methods									
FairMOT [40]	×	22.53	15.80	20.60	37.03	32.82	36.21	71.94	78.28
ByteTrack [41]	×	24.59	16.78	22.60	36.18	36.63	41.00	69.63	78.00
Supervised Learning Methods									
iKUN [8]	×	10.32	2.17	2.36	19.75	49.77	58.48	68.64	74.56
TransRMOT [34]	✓	31.00	19.40	**36.41**	28.97	49.68	54.59	**82.29**	89.82
TempRMOT [42]	✓	**35.04**	**22.97**	34.23	**40.41**	53.58	59.50	81.29	**90.07**
Zero-Shot Learning Methods									
ReferGPT (Ours)	×	30.12	15.69	21.55	34.41	**59.02**	**74.59**	68.20	79.76

Table 3. Comparison of existing methods on Refer-KITTI+ dataset [13]. The best is marked in **bold**, and the second-best in underline. 'E' indicates End-to-End methods. The results are reported in %.

Method	E	HOTA ↑	Detection			Association			LocA
			DetA ↑	DetRe ↑	DetPr ↑	AssA ↑	AssRe ↑	AssPr ↑	
Supervised Learning Methods									
TransRMOT [34]	✓	35.32	25.61	40.05	38.45	50.33	55.40	81.23	79.44
EchoTrack [13]	✓	37.46	28.83	**39.83**	46.70	50.39	54.14	**82.57**	79.97
Zero-Shot Learning Methods									
ReferGPT (Ours)	×	**43.44**	**29.89**	36.59	**56.98**	**63.60**	**75.20**	73.27	**82.23**

Query: Car in black

Query: Car in front

Query: Car on the left

Query: A woman carrying a bag

Figure 4. Qualitative results of our proposed ReferGPT on refer-KITTIv1 [34]. Each row presents example frames tracked according to the given text query.

5.1. Qualitative Results

In figure 4, we illustrate qualitative examples of our proposed ReferGPT on the Refer-KITTIv1 dataset [34]. Each row presents frames from different video sequences where ReferGPT successfully tracks objects based on the input query. We accurately distinguish between similar objects in complex urban and highway scenes, maintaining robust tracking across frames, even under challenging occlusion cases.

5.2. Ablation Study

Text Encoder Ablation. We experiment with different distilled versions of text encoders to evaluate the trade-off between efficiency and performance in our matching module. As shown in Tab.4, DistilBERT-L6 achieved a HOTA of 38.26% while DistilBERT-L12 significantly improves the results, reaching 49.26% HOTA, showing that increasing model depth enhances the embedding quality of the descriptions and queries. The CLIP encoder achieves the best performance of 49.46% HOTA. These results suggest that distilled models can offer a promising balance between computational efficiency and accuracy.

Table 4. Ablation Study on distilled text encoders. DistilBert-L6 [20] refers to a distilled version of Bert [6] with 6 layers, while Distil-L12 is the 12-layer model. The results are reported in %.

Encoder	HOTA	DetA	AssA
DistilBert-L6 [20]	38.26	24.81	59.50
DistilBert-L12 [20]	49.26	39.35	62.21
CLIP [17]	**49.46**	**39.43**	**62.57**

Matching Module Ablation. We compare different matching module configurations in Tab. 5 to assess their contributions. The results show that using CLIP alone is insufficient achieving 31.13% HOTA. Specifically, the CLIP Image encoder contributes the least, as it must encode a cropped detected object, which provides limited visual context. Additionally, the queries and descriptions often contain complex spatial and motion cues, making it difficult for the image encoder to establish strong similarities between the text and the image. Furthermore, since CLIP is trained on shorter image-caption pairs, the CLIP Text encoder struggles with long descriptions, as they contain words with high semantic weight such as the color, movement, or direction of the object. The best performance is achieved by combining CLIP Text with fuzzy matching, as this balances semantic understanding with token-level precision.

Table 5. Ablation Study on matching components. 'Clip Im' refers to the CLIP image encoder. 'Clip Text' denotes the CLIP text encoder, and 'Fuzzy' indicates the Fuzzy matching module. The results are reported in %.

Clip Im	Clip Text	Fuzzy	HOTA	DetA	AssA
✓	-	-	21.42	9.48	48.82
-	✓	-	30.04	15.51	58.36
✓	✓	-	31.13	15.98	61.00
-	-	✓	47.48	35.94	62.97
✓	-	✓	48.96	38.24	63.00
✓	✓	✓	49.34	**39.41**	62.25
-	✓	✓	**49.46**	39.43	**62.57**

Filtering Components Ablation. We conduct experiments with different filtering strategies to evaluate their impact on performance. Applying a fixed similarity threshold, which renders our method online, achieves 36.61% HOTA. Performing post-processing by clustering alone, increases our results to 46.5%. By combining majority voting and clustering we achieve our best result.

Table 6. Ablation Study on filtering components. **MV** refers to the majority voting process and **C** to the clustering. With **T**, we refer to a fixed threshold filtering. The results are reported in %.

MV	C	T	HOTA	DetA	AssA
-	-	✓	36.61	23.17	58.23
-	✓	-	46.50	36.92	59.10
✓	✓	-	**49.46**	**39.43**	**62.57**

Object Detector Ablation. We experiment with different object detectors, all pre-trained on KITTI, to evaluate their impact on RMOT performance. Our method is designed to be detector-agnostic and works effectively with both 2D and 3D detectors. While CasA achieves the highest 3D detection performance on KITTI among the tested detectors, leading to the best overall results in RMOT, the referring performance gap between CasA and lighter detectors such as QT-3DT (2D input) remains relatively narrow. This suggests that our referring text module is robust enough to compensate for lower-quality object detections.

Table 7. Ablation Study on different 3D Object Detectors. The results are reported in %.

Object Detector	Input	HOTA	DetA	AssA
QT-3DT [11]	2D	46.36	36.58	59.00
PV-RCNN [22]	3D	44.93	34.07	59.86
Second-IOU [38]	3D	45.01	33.97	60.14
Point-RCNN [21]	3D	47.14	36.04	61.92
CasA [37]	3D	**49.46**	**39.43**	**62.57**

Distilled MLLM Ablation. Table 8 shows the impact of different distilled MLLMs on RMOT performance. GPT-4o-mini achieves the highest HOTA, demonstrating its strong prompt-following and spatial reasoning capabilities. While smaller models like Qwen2 [2] and Phi-4 [1] perform worse in detection, their association accuracy remains relatively stable. However, the matching module alone does not suffice. For example, Qwen2 produces suboptimal detected object descriptions (Confusing object colors or pedestrians' genders), showcasing that the quality of the MLLM's descriptive output remains essential for maximizing detection and overall tracking performance. This highlights the potential for even greater performance gains through fine-tuning.

Table 8. Ablation Study on different distilled MLLMs. The results are reported in %.

MLLM	Size	HOTA	DetA	AssA
Qwen2-VL-Instruct [2]	2B	19.97	6.50	61.66
Phi4-Multimodal-Instruct [1]	5B	36.56	21.49	62.56
GPT-4o-mini*	8B*	**49.46**	**39.43**	**62.57**

* *The exact model size has not been officially disclosed. The reported parameter count is based on publicly available information and third-party sources.*

5.3. Limitations

While our method eliminates the need for retraining, the inference pipeline remains computationally expensive. The process of generating natural language descriptions for each detected object, computing similarity scores and performing query matching introduces additional latency. This computational overhead arises from the repeated use of the multimodal large language model, which is resource-intensive both in terms of processing time and memory consumption. However, our work can benefit from advancements in efficient or distilled MLLMs, which promise to reduce inference time and resource demands without sacrificing performance.

Another limitation is that, despite our results demonstrating a flexible vocabulary of referring expressions, our framework is still not fully open-vocabulary. Indeed, the MLLM-generated captions follow the structure and specificity of their prompting. The captions often contain information that may not always align with the level of abstraction required by a given query. As a result, if a user query refers to an aspect that is absent from the generated description, our method may struggle to establish a correct match. Achieving true open-vocabulary refer tracking requires exploring further zero-shot generalization, as it is infeasible to train a model on every possible user query.

6. Conclusion

In this work, we present ReferGPT, a novel framework that performs zero-shot Referring Multi-Object Tracking in 3D. We overcome the limitations of current supervised approaches, that struggle with novel and ambiguous queries, showcasing the scalability and adaptability of our method. We leverage our 3D tracker and feed an MLLM 3D spatial information, enhancing its ability to generate structured, spatially aware descriptions within the tracking-by-detection paradigm. Our extensive evaluations on three different referring datasets based on KITTI traffic scenes, demonstrate that ReferGPT can generalize across diverse and open-set queries, highlighting its potential for tracking in complex, open-world scenarios.

Acknowledgement

The work is supported by the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" programme and by Innoviris within the research project TORRES. N. Deligiannis acknowledges support from the Francqui Foundation (2024-2027 Francqui Research Professorship).

References

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. 8

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. 8

[3] Athman Bouguettaya, Qi Yu, Xumin Liu, Xiangmin Zhou, and Andy Song. Efficient agglomerative hierarchical clustering. *Expert Systems with Applications*, 42(5):2785–2797, 2015. 5

[4] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9686–9696, 2023. 2

[5] Jiajun Chen, Jiacheng Lin, Guojin Zhong, You Yao, and Zhiyong Li. Multi-granularity localization transformer with collaborative understanding for referring multi-object tracking. *IEEE Transactions on Instrumentation and Measurement*, 2025. 2, 6

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 7

[7] Leandro Di Bella, Yangxintong Lyu, Bruno Cornelis, and Adrian Munteanu. Hybridtrack: A hybrid approach for robust multi-object tracking. *arXiv preprint arXiv:2501.01275*, 2025. 2

[8] Yunhao Du, Cheng Lei, Zhicheng Zhao, and Fei Su. ikun: Speak to trackers without retraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19135–19144, 2024. 1, 2, 6

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1

[10] Wenyan He, Yajun Jian, Yang Lu, and Hanzi Wang. Visual-linguistic representation learning with deep cross-modality fusion for referring multi-object tracking. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6310–6314. IEEE, 2024. 2, 6

[11] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular quasi-dense 3d object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1992–2008, 2022. 8

[12] Wei Li, Bowen Li, Jingqi Wang, Weiliang Meng, Jiguang Zhang, and Xiaopeng Zhang. Romot: Referring-expression-comprehension open-set multi-object tracking. *The Visual Computer*, pages 1–13, 2024. 2, 6

[13] Jiacheng Lin, Jiajun Chen, Kunyu Peng, Xuan He, Zhiyong Li, Rainer Stiefelhagen, and Kailun Yang. Echotrack: Auditory referring multi-object tracking for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2024. 1, 5, 6

[14] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129:548–578, 2021. 5

[15] Zeliang Ma, Song Yang, Zhe Cui, Zhicheng Zhao, Fei Su, Delong Liu, and Jingyu Wang. Mls-track: Multilevel semantic interaction in rmot. *arXiv preprint arXiv:2404.12031*, 2024. 2

[16] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8844–8854, 2022. 2

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2, 5, 7

[18] D. Ramanan and D.A. Forsyth. Finding and tracking people from the bottom up. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, pages II–II, 2003. 2

[19] John W Ratcliff, David E Metzener, et al. Pattern matching: The gestalt approach. *Dr. Dobb's Journal*, 13(7):46, 1988. 4

[20] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 7

[21] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 8

[22] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10529–10538, 2020. 8

[23] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 6

[24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1

[25] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10860–10869, 2021. 2

[26] Pavel Tokmakov, Allan Jabri, Jie Li, and Adrien Gaidon. Object permanence emerges in a random walk along memory. *arXiv preprint arXiv:2204.01784*, 2022. 2

[27] Huu-Thien Tran, Phuoc-Sang Pham, Thai-Son Tran, and Khoa Luu. Mex: Memory-efficient approach to referring multi-object tracking. *arXiv preprint arXiv:2502.13875*, 2025. 2, 6

[28] Li Wang, Xinyu Zhang, Wenyuan Qin, Xiaoyu Li, Jinghan Gao, Lei Yang, Zhiwei Li, Jun Li, Lei Zhu, Hong Wang, et al. Camo-mot: Combined appearance-motion optimization for 3d multi-object tracking with camera-lidar fusion. *IEEE Transactions on Intelligent Transportation Systems*, 2023. 2

[29] Xiyang Wang, Chunyun Fu, Jiawei He, Mingguang Huang, Ting Meng, Siyu Zhang, Hangning Zhou, Ziyao Xu, and Chi Zhang. A multi-modal fusion-based 3d multi-object tracking framework with joint detection. *IEEE Robotics and Automation Letters*, pages 1–8, 2024. 2

[30] Xiyang Wang, Shouzheng Qi, Jieyou Zhao, Hangning Zhou, Siyu Zhang, Guoan Wang, Kai Tu, Songlin Guo, Jianbo Zhao, Jian Li, et al. Mctrack: A unified 3d multi-object tracking framework for autonomous driving. *arXiv preprint arXiv:2409.16149*, 2024. 2

[31] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 2, 3

[32] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366. IEEE, 2020. 2

[33] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 6

[34] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14633–14642, 2023. 1, 2, 5, 6, 7

[35] Hai Wu, Wenkai Han, Chenglu Wen, Xin Li, and Cheng Wang. 3d multi-object tracking in point clouds based on prediction confidence-guided data association. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5668–5677, 2021. 2, 3, 5

[36] Hai Wu, Qing Li, Chenglu Wen, Xin Li, Xiaoliang Fan, and Cheng Wang. Tracklet proposal network for multi-object tracking on point clouds. In *IJCAI*, pages 1165–1171, 2021. 2

[37] Hai Wu, Jinhao Deng, Chenglu Wen, Xin Li, Cheng Wang, and Jonathan Li. Casa: A cascade attention network for 3-d object detection from lidar point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022. 5, 8

[38] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 8

[39] Kefu Yi, Kai Luo, Xiaolei Luo, Jiangui Huang, Hao Wu, Rongdong Hu, and Wei Hao. Ucmctrack: Multi-object tracking with uniform camera motion compensation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6702–6710, 2024. 2

[40] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International journal of computer vision*, 129:3069–3087, 2021. 6

[41] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer, 2022. 6

[42] Yani Zhang, Dongming Wu, Wencheng Han, and Xingping Dong. Bootstrapping referring multi-object tracking. *arXiv preprint arXiv:2406.05039*, 2024. 1, 2, 5, 6

[43] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European conference on computer vision*, pages 474–490. Springer, 2020. 2