Language Prompt for Autonomous Driving

Dongming Wu[1*†], Wencheng Han[2*], Yingfei Liu[3], Tiancai Wang[3], Cheng-zhong Xu[2], Xiangyu Zhang[3,4], Jianbing Shen[2‡]

[1] Beijing Institute of Technology,
[2] SKL-IOTSC, CIS, University of Macau,
[3] MEGVII Technology,
[4] Beijing Academy of Artificial Intelligence
{wudongming97, wencheng256}@gmail.com, wangtiancai@megvii.com, jianbingshen@um.edu.mo

Abstract

A new trend in the computer vision community is to capture objects of interest following flexible human command represented by a natural language prompt. However, the progress of using language prompts in driving scenarios is stuck in a bottleneck due to the scarcity of paired prompt-instance data. To address this challenge, we propose the first object-centric language prompt set for driving scenes within 3D, multi-view, and multi-frame space, named NuPrompt. It expands nuScenes dataset by constructing a total of 40,147 language descriptions, each referring to an average of 7.4 object tracklets. Based on the object-text pairs from the new benchmark, we formulate a novel prompt-based driving task, *i.e.*, employing a language prompt to predict the described object trajectory across views and frames. Furthermore, we provide a simple end-to-end baseline model based on Transformer, named PromptTrack. Experiments show that our PromptTrack achieves impressive performance on NuPrompt. We hope this work can provide some new insights for the self-driving community.

1 Introduction

Leveraging natural language description in visual tasks is one of the recent trends in the vision community (Radford et al. 2021; Kirillov et al. 2023). It has garnered significant interest for its potential applications in various downstream tasks, such as embodied intelligence and human-robot interactions (Deruyttere et al. 2019; Chen et al. 2023; Gupta and Kembhavi 2023; Hu et al. 2023; Wu et al. 2023a,c; Bai et al. 2024). Its core idea is to predict the desired answer by shifting human instruction inputs but not updating model weights, delivering high adaptability in response to varying human demands. A key factor for the progress in 2D scenes is the availability of large-scale image-text pairs (Lin et al. 2014; Changpinyo et al. 2021; Schuhmann et al. 2021). However, this success is hard to replicate in self-driving scenarios due to the scarcity of 3D instance-text pairs.

Pioneering works like Talk2Car (Deruyttere et al. 2019), Cityscapes-Ref (Vasudevan, Dai, and Van Gool 2018) have started to incorporate natural language into object detection tasks in driving scenes. Unfortunately, these datasets only allow each expression to refer to a single object within an individual image, restricting their usage in scenarios with multiple referred objects or changing temporal states. Furthermore, Refer-KITTI (Wu et al. 2023b) addressed this issue by extending the KITTI dataset (Geiger, Lenz, and Urtasun 2012) to include expressions that ground multiple video objects. This work mainly focuses on modular images and 2D detection, thereby leaving room for improvement in 3D driving scenes. A recent advancement, namely NuScenes-QA (Qian et al. 2023), offers numerous question-answer pairs for 3D multi-view driving scenes, making significant strides in the use of language prompts. However, it primarily contributes to scene-level understanding and overlooks the direct and fine-grained semantic correspondence between 3D instances and natural language expression.

To advance the research of prompt learning in driving scenarios, we propose a new large-scale benchmark, named **NuPrompt**. The benchmark is built on the popular multi-view 3D object detection dataset nuScenes (Caesar et al. 2019). We assign a language prompt to a collection of objects sharing the same characteristics for grounding them. Essentially, this benchmark provides lots of 3D instance-text pairings with three primary attributes: ❶ *Real-driving descriptions.* Different from existing benchmarks that only represent 2D objects from modular images, the prompts of our dataset characterize a diverse range of driving-related objects from 3D, panoramic views, and long-temporal space. Fig. 1 shows a typical example, *i.e.*, a car surpasses our car from behind towards the front across multiple views. ❷ *Instance-level prompt annotations.* Every prompt indicates a fine-grained and discriminative object-centric description, as well as enabling it to cover an arbitrary number of driving objects. ❸ *Large-scale language prompts.* From a quantitative perspective, NuPrompt has 40,147 language prompts.

Along with NuPrompt, we present a new and challenging prompt-based driving perception and prediction task, whose main idea is to track and forecast the trajectory for 3D objects based on a specified language prompt. The chal-

[*] These authors contributed equally.
[†] The work is done during Dongming Wu interned at MEGVII.
[‡] Corresponding author: *Jianbing Shen*.

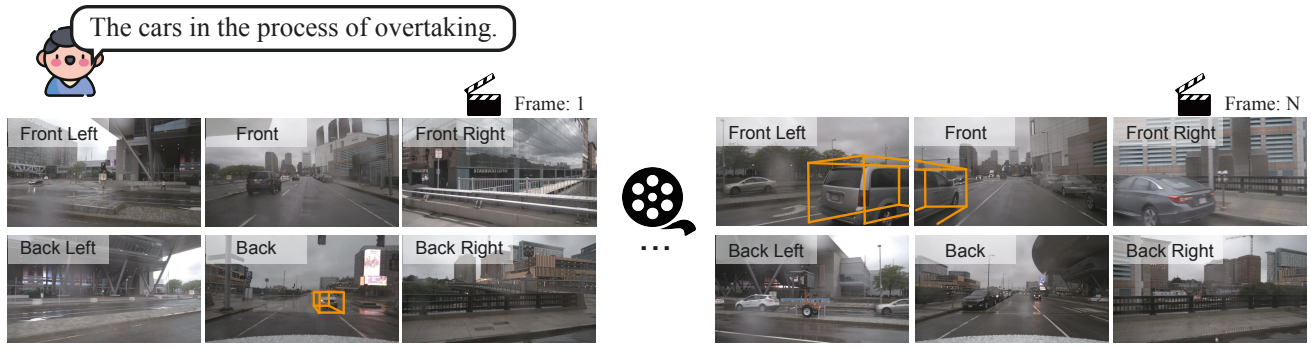

Figure 1: A representative example from NuPrompt. The language prompt "the cars in the process of overtaking" is precisely annotated and corresponded to the objects within the 3D, multi-frame, and multi-view driving space. NuPrompt contains 40,147 object-prompt pairs with fine-grained semantic alignment. Each language prompt refers to multiple object trajectories.

Dataset	Basic Task	3D	#Views	#Videos	#Frames	#Prompts	# Instances per-prompt
RefCOCO (Yu et al. 2016)	Det&Seg	✗	1	-	26,711	142,209	1
Talk2Car (Deruyttere et al. 2019)	Det	✓	1	-	9,217	11,959	1
Cityscapes-Ref (Vasudevan, Dai, and Van Gool 2018)	Det	✗	1	-	4,818	30,000	1
Refer-KITTI (Wu et al. 2023b)	MOT	✗	1	18	6,650	818	10.7
NuScenes-QA (Qian et al. 2023)	VQA	✓	6	850	34,149	459,941	-
DriveLM-nuScenes (Sima et al. 2024)	Det&VQA	✗	6	-	4,871	445,209	-
NuPrompt (Ours)	MOT	✓	6	850	34,149	40,776	7.4

Table 1: Comparison of our NuPrompt with existing prompt-based datasets. '-' means unavailable. NuPrompt provides the nature and complexity of driving scenes, *i.e.*, 3D, multi-view space, and multi-frame domain. Besides, it focuses on object-centric understanding by pairing a language prompt with multiple targets of interest.

lenges of this task lie in two aspects: temporal association across frames and cross-modal semantic comprehension. To address the challenges, we propose an end-to-end baseline built on camera-only 3D tracker PF-Track (Pang et al. 2023), named **PromptTrack**. Note that PF-Track has exhibited excellent spatial-temporal modeling through its past and future reasoning branches. We additionally involve cross-attention between prompt embedding and visual features, and then add one prompt reasoning branch to ground prompt-referred objects. Furthermore, we also evaluate the motion prediction of these prompt-referred objects in our experiments. In addition to the prompt-based prediction tasks, we expect that our annotations can facilitate future research in multi-model LLM of autonomous driving.

In summary, our contributions are three-fold:

- We propose a new large-scale language prompt set for driving scenes, named NuPrompt. As far as we know, it is the first dataset specializing in multiple 3D objects of interest from video domain.
- We construct a new prompt-based driving perception and prediction task, which requires using a language prompt as a semantic cue to predict object trajectories.
- We develop a simple baseline model that unifies prompt-based object tracking and motion prediction in a single framework, called PromptTrack.

2 Related Work

Language Prompt in Driving Scenes. The utilization of human commands within driving scenes allows the system to understand driving systems from the human perspective, thereby facilitating human control over driving procedures. Talk2Car (Deruyttere et al. 2019), the pioneering benchmark featuring language prompts for autonomous vehicles, is constructed on the base of nuScenes (Caesar et al. 2019). However, its annotation only comprises keyframes that catch the eye of annotators. However, the prompts deployed in both Talk2Car tend to represent an individual object. To solve this problem, Refer-KITTI (Wu et al. 2023b) further develops KITTI (Geiger, Lenz, and Urtasun 2012), where each prompt can refer to a collection of referent objects. More recently, NuScenes-QA (Qian et al. 2023) opens a new avenue, Visual Question Answering (VQA), for understanding scene-level driving scenarios. DriveLM (Sima et al. 2024) considers 2D crucial objects for graph visual question answering. Although there are other natural language sets used in the driving area, like BDD-X (Kim et al. 2018), DRAMA (Malla et al. 2023), TalkBEV (Dewangan et al. 2024), and Rank2Tell (Sachdeva et al. 2024), they are completely different from ours because of using language as captions for improving driving explanations. A thorough comparison between existing prompt-based driving datasets and ours is summarized in Table 1.

Referring Expression Understanding. Given a language prompt, the goal of referring expression understanding is to

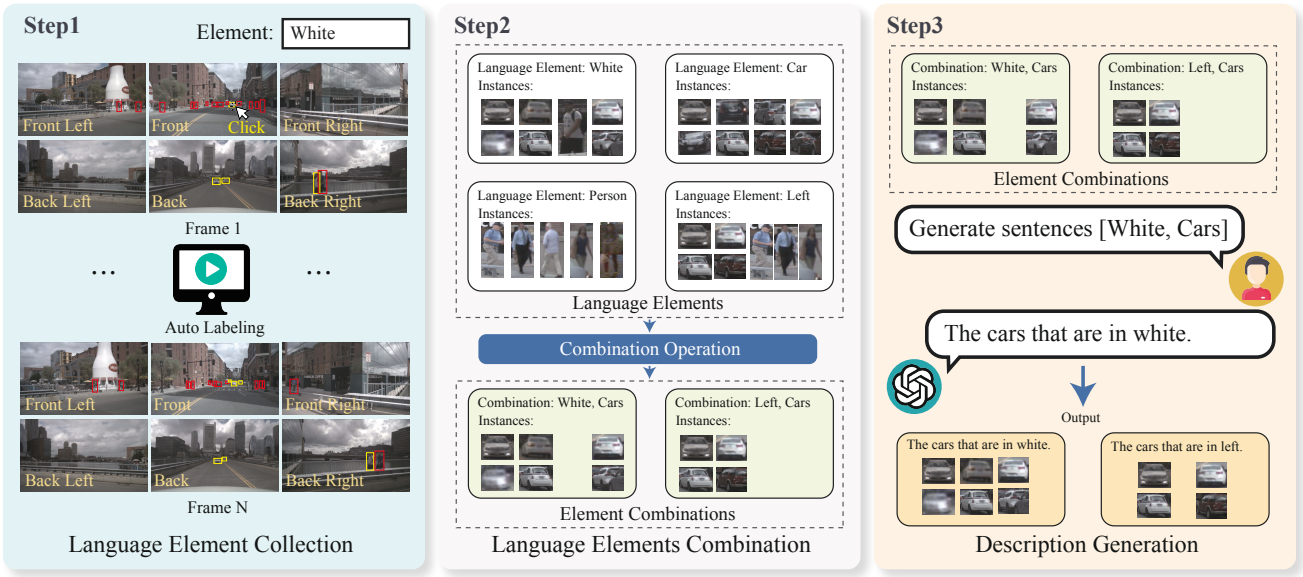

Figure 2: Pipeline of language prompt annotation procedure, which includes three steps: language element collection, language element combination, and description generation. Firstly, we pair each language tag with referent objects during the language element collection phase. Following this, certain language elements are selected and combined in the language element combination stage. Finally, with the combinations obtained, we employ LLM to create language descriptions.

localize the described objects using boxes or masks, which shares a similar idea with our prompt-based driving benchmark. The initiation of datasets like RefCOCO/+/g (Yu et al. 2016) has helped stimulate interest in this field. These datasets use succinct yet unambiguous natural language expressions to ground a visual region within an image. Some follow-up works further improve this dataset by allowing it to support expressions that refer to unlimited target objects (Liu, Ding, and Jiang 2023; Chen et al. 2019). Besides, Refer-DAVIS$_{16/17}$ (Khoreva, Rohrbach, and Schiele 2019) and Refer-Youtube-VOS (Seo, Lee, and Han 2020) are another two popular video-based referring expression understanding benchmarks supporting video object segmentation. A recent work in the field called GroOT (Nguyen et al. 2023) expands the large-scale multi-object tracking dataset TAO (Dave et al. 2020) to support referring understanding.

3 Dataset Overview

3.1 Data Collection and Annotation

Our NuPrompt is built on one of the most popular datasets for multi-view 3D object detection, nuScenes (Caesar et al. 2019). While the original nuScenes dataset includes visual images and point cloud data, we here focus solely on visual images for NuPrompt. As shown in Fig. 2, the cars collecting the data are equipped with six different cameras: Front, Front Left, Front Right, Back Right, Back Left, and Back. These cameras overlap in some areas. Therefore, NuPrompt provides 360° of 3D space for each scene.

To efficiently generate training labels for the new dataset, we designed a three-step semi-automatic labeling pipeline (see Fig. 2). The first step aims at identifying language elements and associating them with 3D bounding boxes. The second step is to combine the language elements using certain rules. In the third step, we base the language element combinations to produce various language prompts using a large language model (LLM). Detailed information about these three steps is provided as follows.

Step 1: Language Element Collection. This paper uses the term "language element" to refer to a basic attribute of objects. Examples of language elements include colors (e.g. red, yellow, and black), actions (e.g. running, stopping, and crossing the road), locations (e.g. left, right, and back), and classes (e.g. car and pedestrian), which cover diverse descriptions of driving scenes. The key problem is how to label the bounding boxes with the corresponding language elements. To solve this, we design a labeling system to manually collect and match language elements with bounding boxes in a video sequence, as demonstrated in Fig. 2. Annotators type the language element texts and click on the corresponding bounding boxes. When the target changes status and no longer belongs to the language elements, annotators need to click on the target again and remove it from the list. This procedure can efficiently reduce the amount of human labor required. To ensure a variety of expressions, each video is assigned to five independent annotators who manually create descriptive expressions to formulate query sentences. Two other annotators then carefully check the matching between the expressions and the video targets.

Step 2: Language Elements Combination. As mentioned earlier, language elements are basic attributes of objects. By combining these attributes, we can create various descriptions for different groups of objects. There is one logical relationship we can use to merge the attributes: AND. We use

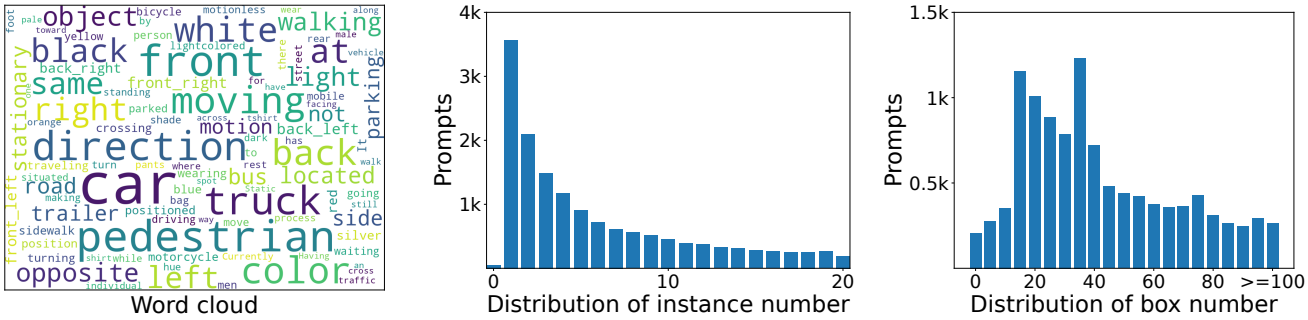

Figure 3: Left: Word cloud of top 100 words in NuPrompt. Middle: Distribution of instance number per prompt, where instance number also represents object tracklet number. Right: Distribution of box number per prompt.

this operation to combine sets of bounding boxes and their language elements, resulting in a new set with the merged attributes. In our dataset, we manually choose some meaningful attribute combinations and randomly generate many combinations for the objects.

Step 3: Prompt Generation. After Step 2, we are able to determine the correspondence between combinations of language elements and a group of bounding boxes. However, getting valid natural language sentences to describe the objects can be expensive using human labor, and there is no guarantee of the desired variety. Large language model (LLM) have recently shown great potential in understanding logistics and producing sentences similar to those generated by humans. Therefore, we determine GPT3.5 (OpenAI 2023) as our language model. We prompt it with a request like "Generate a sentence to describe the objects based on the following descriptions: *pedestrians, moving, red, not in the left*" where the italicized words represent the combination of elements. The LLM can respond with a meaningful description of the objects, such as "The objects are red pedestrians, currently in motion, not situated on the left side." To guarantee accuracy, we will ask annotators to filter out any incorrect descriptions. We also prompt the LLM multiple times to generate multiple descriptions.

3.2 Dataset Statistics

Thanks to nuScenes (Caesar et al. 2019), our language prompts provide a number of comprehensive descriptions for the objects in the 3D, surrounding, and temporal space. Besides, they cover diverse environments comprising pedestrian streets, public roads, and highways in the cities of Boston and Singapore. Furthermore, they encompass different weather conditions (*e.g.*, rain and sun), and illumination (*e.g.*, day and night). To offer deeper insights into NuPrompt, we next present more quantitative statistics.

Language Prompt. We manually label 23,368 language elements and utilize them in the combination of 16,761 unified descriptions through LLM. Among them, there are 12,001 unique language elements and 12,001 unique language combinations. In total, the NuPrompt has 40,147 language prompts. On average, each video within the dataset contains 47 language prompts. We show the word cloud of the top 100 words in Fig. 3. From the left figure, we can observe that NuPrompt dataset has a large number of words that describe driving object appearances, like 'black', 'white' and 'red', and locations, like 'front', 'left' and 'right'. Besides, some motion words like 'walking', 'moving', and 'crossing' are also common descriptions.

Referent Objects. In contrast to previous benchmarks that refer to 2D objects in modular images, another feature of NuPrompt is its surrounding 3D space. This indicates that there are lots of objects crossing different views, presenting improved simulation being closer to real driving scenes. More importantly, NuPrompt is designed to involve an arbitrary number of predicted objects. The left of Figure 3 indicates that the majority of prompts describe between 1 and 10 instances, and can sometimes even exceed 20. According to Table 1, the average number of instances referred to by each prompt in our dataset is 7.4. In addition, the distribution of box is displayed on the right of Fig. 3.

3.3 Benchmark Protocols

Task Definition. Together with NuPormpt, we formulate a straightforward yet challenging task that employs a language prompt as a meaningful signal to anticipate the pathways of objects. This task specifically includes multi-object tracking and movement prediction, based on prompts.

Evaluation Metrics. To evaluate the similarity between the predicted tracklet and ground truth tracklet, we use Average Multiple Object Tracking Accuracy (AMOTA) as a primary metric (Bernardin and Stiefelhagen 2008). However, unlike the original multi-object tracking task that averages AMOTA across different categories, the evaluation on NuPrompt is class-agnostic. Hence, we calculate AMOTA for all prompt-video pairs and take the average of all these AMOTA values. For a more detailed analysis, we also utilize Average Multi-Object Tracking Precision (AMOTP) and Identity Switches (IDS) metrics. Additionally, we evaluate trajectory prediction performance using Average Displacement Error (ADE) and Final Displacement Error (FDE), following the works (Pang et al. 2023; Gu et al. 2023).

Data Split. The NuPrompt contains a total of 850 videos along with language prompts. Following nuScenes (Caesar et al. 2019), we split NuPrompt into training and validation set, which contain 700 videos and 150 videos, respectively.

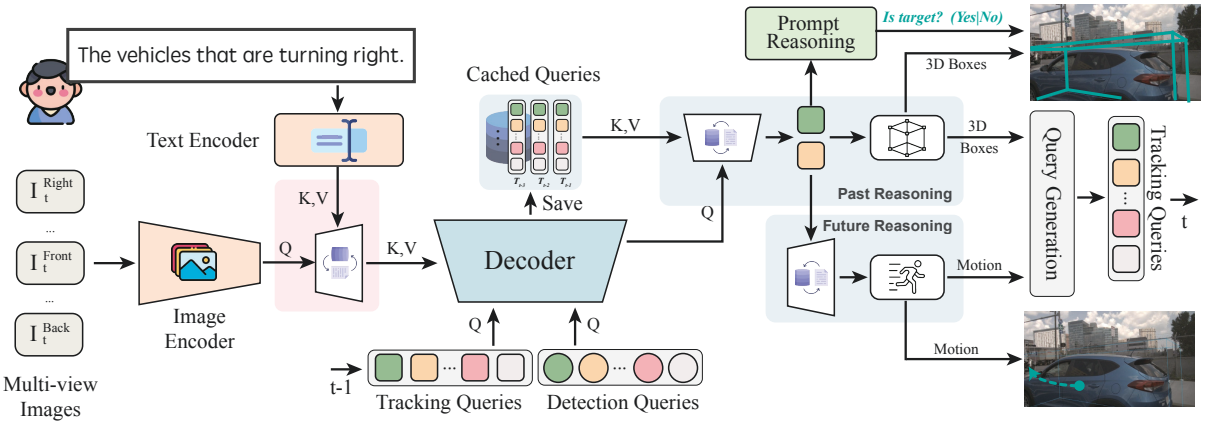

Figure 4: Overall architecture of PromptTrack. For each frame, the visual features and language prompt embedding are fused before being fed into the Transformer decoder. Then past reasoning enhances and refines tacks by attending to cached historical queries, and the future reasoning benefits cross-frame query propagation using predicted position. Lastly, the prompt reasoning branch predicts the prompt-referred tracks using binary classification.

4 Method

Given multi-frame multi-view images and a natural language prompt, our goal is to track the described object. To accomplish this, we propose PromptTrack, an end-to-end framework. It modifies the query-based method PF-Track (Pang et al. 2023) to adapt to the prompt input. Fig. 4 shows the overall pipeline of PromptTrack. Note that PF-Track incorporates a past reasoning branch and a future reasoning branch which are based on the decoded queries. These branches aim to refine tack prediction using cached historical queries and improve cross-frame query propagation using motion localization prediction, respectively. In addition to these two branches, we propose a new prompt reasoning branch to predict the prompt-referred tracks.

4.1 Overall Architecture

Formally, let $\boldsymbol{F}_t$ denote the extracted visual features at timestamp t, and $\boldsymbol{S}$ denote the encoded linguistic features. To enrich the information on visual features, we first incorporate the visual features with the linguistic features in a multiplication way and form enhanced visual feature maps. Specifically, we flatten two kinds of features and use cross-modal attention to encourage the feature fusion between $\boldsymbol{F}_t$ and $\boldsymbol{S}$, generating prompt-aware visual features $\boldsymbol{F}'_t$:

$$\boldsymbol{F}'_t = \textbf{CrossModalAttn}(\text{Q} = \boldsymbol{F}_t, \text{K,V} = \boldsymbol{S}, \text{PE} = \text{Pos}(\boldsymbol{S})), \quad (1)$$

where Pos means position embedding (Vaswani et al. 2017).

To capture different views and stereo information, we follow PETR (Liu et al. 2022) to add 3D position embedding on $\boldsymbol{F}'_t$. For notion simplicity, the position-augmented visual feature is also represented as $\boldsymbol{F}'_t$. Then a set of 3D queries $\boldsymbol{Q}_t$ interact with $\boldsymbol{F}'_t$ via a stack of Transformer decoder layers, outputting updated queries $\boldsymbol{Q}^D_t$ and bounding boxes $\boldsymbol{B}^D_t$:

$$\boldsymbol{Q}^D_t, \boldsymbol{B}^D_t = \textbf{Decoder}\left(\boldsymbol{Q}_t, \boldsymbol{F}'_t\right), \quad (2)$$

where each input query $\boldsymbol{q}^i_t \in \boldsymbol{Q}_t$ means an object with a feature vector $\boldsymbol{f}^i_t$ and a 3D localization $\boldsymbol{c}^i_t$, *i.e.*, $\boldsymbol{q}^i_t = \{\boldsymbol{f}^i_t, \boldsymbol{c}^i_t\}$.

To automatically link objects across different frames, the input queries $\boldsymbol{Q}_t$ merge track queries $\boldsymbol{Q}^{track}_t$ from the last frame. The box information from the last frame also provides an excellent spatial position prior to the current frame, benefiting the model for accurately inferring the same object. Besides, to capture new-born objects, a set of fixed 3D queries $\boldsymbol{Q}^{fixed}_t$, also called detection queries, are concatenated with track queries $\boldsymbol{Q}^{track}_t$ to generate $\boldsymbol{Q}_t$. Following the work (Pang et al. 2023), the number of fixed queries is set to 500. As the first frame has no previous frames, we only utilize the fixed queries to detect objects.

Past and Future Reasoning. After the Transformer Decoder, past and future reasoning are sequentially conducted for attending to historical embeddings and predicting future trajectory, respectively. Formally, the past reasoning $\mathcal{F}^p$ integrates two decoded outputs $\boldsymbol{Q}^D_t$ and $\boldsymbol{B}^D_t$ as well as cached historical queries $\boldsymbol{Q}_{t-\tau_h:t-1}$ from past τ_h frames to produce refined queries $\boldsymbol{Q}^R_t$ and refined bounding boxes $\boldsymbol{B}^R_t$:

$$\boldsymbol{Q}^R_t, \boldsymbol{B}^R_t = \mathcal{F}^p\left(\boldsymbol{Q}^D_t, \boldsymbol{B}^D_t, \boldsymbol{Q}_{t-\tau_h:t-1}\right), \quad (3)$$

where $\mathcal{F}^p$ has a cross-frame attention module for promoting history information integration across τ_h frames per object. Moreover, it also includes a cross-object attention module to encourage discriminative feature representation for each individual object. The sequential cross-frame attention and cross-object attention modules lead to $\boldsymbol{Q}^R_t$. A multi-layer perceptron (MLP) is used to predict coordinate residuals and adjust the object boxes, leading to $\boldsymbol{B}^R_t$.

Based on the refined results from past reasoning, the future reasoning $\mathcal{F}^f$ uses a cross-frame attention module to predict long-term trajectories $\boldsymbol{M}_{t:t+\tau_f}$ for next τ_f frames:

$$\boldsymbol{Q}^{track}_{t+1}, \boldsymbol{M}_{t:t+\tau_f} = \mathcal{F}^f\left(\boldsymbol{Q}^R_t, \boldsymbol{Q}_{t-\tau_h:t-1}\right), \quad (4)$$

where the position vectors of the refined queries $\boldsymbol{Q}^R_t$ is updated to generate $\boldsymbol{Q}^{track}_{t+1}$ according to the single-step movement $\boldsymbol{M}_{t:t+1}$. The main motivation of future reasoning is that as the ego-car goes forward, the reference position of all

Method	Decoder	AMOTA ↑	AMOTP↓	RECALL↑	MOTA↑	IDS↓	ADE↓	FDE↓
CenterPoint tracker	DETR3D	0.079	1.820	19.6%	0.093	350	-	-
CenterPoint tracker	PETR	0.178	1.650	29.1%	0.197	174	-	-
DQTrack	DETR3D	0.186	1.641	30.7%	0.208	160	2.51	2.73
DQTrack	Stereo	0.198	1.625	30.9%	0.214	103	2.40	2.61
DQTrack	PETRv2	0.234	1.545	33.2%	0.269	78	2.28	2.36
ADA-Track	PETR	0.249	1.538	35.3%	0.270	67	2.20	2.31
PromptTrack (Ours)	DETR3D	0.202	1.615	31.0%	0.222	95	2.38	2.50
PromptTrack (Ours)	PETR	**0.259**	**1.513**	**36.6%**	**0.280**	**26**	**2.17**	**2.21**

Table 2: Main results on NuPrompt. CenterPoint tracker (Yin, Zhou, and Krähenbühl 2021) is a heuristic-based tracking algorithm that utilizes different decoders such as DETR3D and PETR. DQTrack, ADA-Track and our PromptTrack are end-to-end frameworks, which can also equip various decoders. ↑ and ↓ represent the direction of better performance about each metric.

Prompt Fusion	AMOTA ↑	AMOTP↓	RECALL↑
w/o Prompt Fusion	0.073	**1.398**	**43.5%**
Two-way Fusion	0.249	1. 530	36.7%
Prompt as Query	0.054	1.723	21.6%
Ours	**0.259**	1.513	36.6%

Table 3: Different prompt-visual fusion methods. *w/o* Prompt Fusion' represents removing prompt reasoning and using all tracking instances as prompt prediction. 'Two-way Fusion' and 'Prompt as Query' are fusion variants.

Past & Future	AMOTA ↑	AMOTP↓	RECALL↑
w/o Past Reason	0.239	1.543	33.1%
w/o Future Reason	0.247	1.523	34.9%
Ours	**0.259**	1.513	36.6%

Table 4: Ablation study on past and future reasoning.

objects from the last frame has to be adjusted to align with the new ego-coordinates. In summary, using past and future information can improve the quality of visible object tracking. To further identify the object referred to in the prompt, we outline the new prompt reasoning as follows.

4.2 Prompt Reasoning

Based on the past reasoning that tracks all visible objects, prompt reasoning focuses on grounding prompt-referred objects. Since the refined queries $\boldsymbol{Q}_t^R$ from the past reasoning branch have integrated prompt embeddings, our prompt reasoning $\mathcal{F}^l$ (l for 'language') can directly output the prompt-referred object probability $\boldsymbol{P}_t$:

$$\boldsymbol{P}_t = \mathcal{F}^l\left(\boldsymbol{Q}_t^R\right), \tag{5}$$

where $\mathcal{F}^l$ is an MLP with two fully-connected layers. $\boldsymbol{P}_t$ is a binary probability that indicates whether the output embedding is a prompt-referred object.

4.3 Instance Matching and Loss

Our method views the 3D detection as a set prediction problem following query-based methods (Carion et al. 2020; Wu et al. 2023d), so it requires one-to-one matching before calculating loss. The tracking queries $\boldsymbol{Q}_t^{track}$ and their ground-truth have been matched when propagating queries. As for the fixed queries $\boldsymbol{Q}_t^{fixed}$, we match them with newborn objects using a bipartite graph matching, following MOTR (Zeng et al. 2022). During the process of matching, we only use the queries $\boldsymbol{Q}_t^D$ from decoder outputs, and then implement the correspondence in the overall loss:

$$\begin{aligned}\mathcal{L} = \lambda_{cls}^D\mathcal{L}_{cls}^D + \lambda_{box}^D\mathcal{L}_{box}^D + +\lambda_f\mathcal{L}_f \\ +\lambda_{cls}^R\mathcal{L}_{cls}^R + \lambda_{box}^R\mathcal{L}_{box}^R + \lambda_p\mathcal{L}_p,\end{aligned} \tag{6}$$

where $\mathcal{L}_{cls}$s are Focal loss (Lin et al. 2017) for classification, $\mathcal{L}_{box}$s are L1 loss for bounding box regression, $\mathcal{L}_f$ is L1 loss for motion prediction. The settings of their weights λs follow PF-Track (Pang et al. 2023). Besides, our prompt reasoning loss $\mathcal{L}_p$ is also Focal loss, weighted by λ_p.

5 Experiment

In this section, we conduct a set of experiments on our proposed benchmark NuPrompt. For implementation details, we follow the settings of PF-Track (Pang et al. 2023).

5.1 Main Results

Since there are no existing methods for the new prompt-based prediction task, we modify the existing **open-sourced** camera-only tracking models for state-of-the-art comparison. First, most previous camera-only tracking methods follow a heuristic-based pattern (Liu et al. 2023b; Wang et al. 2023), which include two parts: basic detectors and trackers. Following them, we utilize two state-of-the-art single-frame object detectors, *i.e.*, DETR3D (Wang et al. 2022) and PETR (Liu et al. 2022), and a popular tracker, CenterPoint tracker (Yin, Zhou, and Krähenbühl 2021). The powerful tracker has been used in many state-of-the-art detection methods from the leaderboard of nuScenes tracking task (Liu et al. 2023b; Wang et al. 2023). To distinguish the prompt-referred objects, the basic detectors incorporate the implementation of our prompt fusion module. Second, we incorporate two end-to-end camera-based 3D multi-object tracking models DQTrack (Li et al. 2023) and ADA-Track (Ding et al. 2024), which utilizes diverse decoders like DETR3D (Wang et al. 2022), Stereo (Li et al.

Prompt Type	AMOTA ↑	AMOTP↓	RECALL↑
Language Elements	0.271	1.498	38.9%
Unified Descriptions	0.242	1.523	35.7%
Total Prompts	0.259	1.513	36.6%

Table 5: Ablation studies on different prompt types.

Prompt Number	AMOTA ↑	AMOTP↓	RECALL↑
20%	0.210	1.471	31.3%
50%	0.247	1.511	36.1%
100%	0.259	1.513	36.6%

Table 6: Ablation studies on different prompt numbers.

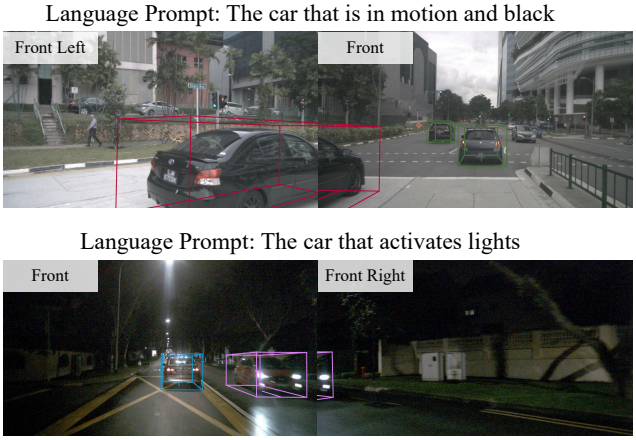

Figure 5: Qualitative results of PromptTrack on NuPrompt.

2023), PETRv2 (Liu et al. 2023a), or PETR (Liu et al. 2022). This approach stands apart from the previously mentioned heuristic-based methods that rely on a two-step inference process. We seamlessly integrate our prompt reasoning module within DQTrack to enable the prediction of objects specifically referred to in prompts. For all other configurations, we adhere to the official settings and guidelines. Third, we evaluate our PromptTrack framework by substituting its fundamental detector with DETR3D. On top of NuPrompt, we test the proposed PromptTrack and these competitors, and the main results are shown in Table 2. As seen, PromptTrack achieves 0.259 on AMOTA, outperforming other counterparts across the majority of metrics.

In addition to prompt-refereed tracking evaluation, we assess the performance of motion prediction by forecasting over eight frames. As object motion cannot be effectively tested by single-frame object detectors, PromptTrack is the only model we examine. From Table 2, PromptTrack with PETR detector results in an ADE of 2.17 and a FDE of 2.21, showing impressive performance. For inference speed, we test PromptTrack using VOV backbone on one Nividia A100 GPU over the validation set. PromptTrack achieves 7.7 FPS.

5.2 Ablation Studies

Prompt Fusion Methods. Table 3 showcases various prompt-visual fusion strategies. The first is to eliminate the prompt fusion module and use all predicted tracking instances as our prompt prediction (*w/o* Prompt Fusion). Second, we can improve upon our current approach of utilizing language prompt features solely to augment visual features (one-way fusion) by introducing a two-way fusion method. Specifically, before enhancing the visual features, we generate visual-enhanced language features, which bring a bidirectional interaction between the two modalities. Third, we evaluate another variant of prompt-visual fusion, only adding prompt embedding with the decoder queries (represented as 'Prompt as Query'). These variants, though implemented, do not yield notable performance gains. In summary, our prompt-visual fusion strategy proves to be highly effective for the given task.

Past & Future Reasoning. In Table 4, it is evident that neglecting either past or future reasoning leads to diminished performance, emphasizing the importance of incorporating both reasoning branches.

Prompt Type & Number. We test our PromptTrack on language elements and unified descriptions (see details in §2), as presented in Table 5. Notably, the model performance on the unified descriptions exhibits a decline compared to that on the language elements, suggesting that the integration of the combined prompts poses certain challenges. We additionally experiment to examine the impact of varying prompt numbers by randomly sampling 20%, 50%, and 100% from the training set, as shown in Table 6. A clear trend emerges: as the number of prompts increases, there is a corresponding enhancement in model performance, highlighting the positive correlation between prompt quantity and model efficacy.

5.3 Qualitative Results

We visualize two typical qualitative results in Fig. 5. As seen, our PromptTrack can detect and track prompt-referred targets accurately under various challenging situations, like crossing different views and varying object numbers.

6 Conclusion

In this work, we presented NuPrompt, the first large-scale language prompt set designed specifically for 3D perception in autonomous driving. NuPrompt provides numerous precise and fine-grained 3D object-text pair annotations. As a result, we designed a simple yet challenging prompt-driven object trajectory prediction task, *i.e.*, tracking objects and forecasting their motion using a language prompt as a semantic cue. To solve this problem, we further proposed an end-to-end prompt-based tracking model with prompt reasoning modification on PF-Track, called PromptTrack. After conducting a set of experiments on NuPrompt, we verified the effectiveness of our algorithm.

Acknowledgments

This work was supported by the FDCT grants 0102/2023/RIA2, 0154/2022/A3, and 001/2024/SKL, the SRG2022-00023-IOTSC grant and the MYRG-CRG2022-00013-IOTSC-ICI grant.

References

Bai, Y.; Wu, D.; Liu, Y.; Jia, F.; Mao, W.; Zhang, Z.; Zhao, Y.; Shen, J.; Wei, X.; Wang, T.; et al. 2024. Is a 3D-Tokenized LLM the Key to Reliable Autonomous Driving? *arXiv preprint arXiv:2405.18361*.

Bernardin, K.; and Stiefelhagen, R. 2008. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2019. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*.

Changpinyo, S.; Sharma, P.; Ding, N.; and Soricut, R. 2021. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*.

Chen, L.; Wu, P.; Chitta, K.; Jaeger, B.; Geiger, A.; and Li, H. 2023. End-to-end Autonomous Driving: Challenges and Frontiers. *arXiv preprint arXiv:2306.16927*.

Chen, Z.; Ma, L.; Luo, W.; and Wong, K.-Y. K. 2019. Weakly-supervised spatio-temporally grounding natural sentence in video. *arXiv preprint arXiv:1906.02549*.

Dave, A.; Khurana, T.; Tokmakov, P.; Schmid, C.; and Ramanan, D. 2020. Tao: A large-scale benchmark for tracking any object. In *ECCV*.

Deruyttere, T.; Vandenhende, S.; Grujicic, D.; Van Gool, L.; and Moens, M.-F. 2019. Talk2car: Taking control of your self-driving car. *arXiv preprint arXiv:1909.10838*.

Dewangan, V.; Choudhary, T.; Chandhok, S.; Priyadarshan, S.; Jain, A.; Singh, A. K.; Srivastava, S.; Jatavallabhula, K. M.; and Krishna, K. M. 2024. Talk2BEV: Language-enhanced Bird's-eye View Maps for Autonomous Driving. In *ICRA*.

Ding, S.; Schneider, L.; Cordts, M.; and Gall, J. 2024. ADA-Track: End-to-End Multi-Camera 3D Multi-Object Tracking with Alternating Detection and Association. In *CVPR*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*.

Gu, J.; Hu, C.; Zhang, T.; Chen, X.; Wang, Y.; Wang, Y.; and Zhao, H. 2023. ViP3D: End-to-end visual trajectory prediction via 3d agent queries. In *CVPR*.

Gupta, T.; and Kembhavi, A. 2023. Visual programming: Compositional visual reasoning without training. In *CVPR*.

Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. 2023. Planning-oriented autonomous driving. In *CVPR*.

Khoreva, A.; Rohrbach, A.; and Schiele, B. 2019. Video object segmentation with language referring expressions. In *ACCV*.

Kim, J.; Rohrbach, A.; Darrell, T.; Canny, J.; and Akata, Z. 2018. Textual explanations for self-driving vehicles. In *ECCV*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *ICCV*.

Li, Y.; Yu, Z.; Philion, J.; Anandkumar, A.; Fidler, S.; Jia, J.; and Alvarez, J. 2023. End-to-end 3d tracking with decoupled queries. In *ICCV*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *ICCV*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Liu, C.; Ding, H.; and Jiang, X. 2023. GRES: Generalized referring expression segmentation. In *CVPR*.

Liu, Y.; Wang, T.; Zhang, X.; and Sun, J. 2022. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*.

Liu, Y.; Yan, J.; Jia, F.; Li, S.; Gao, A.; Wang, T.; and Zhang, X. 2023a. Petrv2: A unified framework for 3d perception from multi-camera images. In *ICCV*.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D. L.; and Han, S. 2023b. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*.

Malla, S.; Choi, C.; Dwivedi, I.; Choi, J. H.; and Li, J. 2023. DRAMA: Joint Risk Localization and Captioning in Driving. In *WCCV*.

Nguyen, P.; Quach, K. G.; Kitani, K.; and Luu, K. 2023. Type-to-Track: Retrieve Any Object via Prompt-based Tracking. *arXiv preprint arXiv:2305.13495*.

OpenAI. 2023. https://chat.openai.com.

Pang, Z.; Li, J.; Tokmakov, P.; Chen, D.; Zagoruyko, S.; and Wang, Y.-X. 2023. Standing Between Past and Future: Spatio-Temporal Modeling for Multi-Camera 3D Multi-Object Tracking. In *CVPR*.

Qian, T.; Chen, J.; Zhuo, L.; Jiao, Y.; and Jiang, Y.-G. 2023. NuScenes-QA: A Multi-modal Visual Question Answering Benchmark for Autonomous Driving Scenario. *arXiv preprint arXiv:2305.14836*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Sachdeva, E.; Agarwal, N.; Chundi, S.; Roelofs, S.; Li, J.; Kochenderfer, M.; Choi, C.; and Dariush, B. 2024. Rank2tell: A multimodal driving dataset for joint importance ranking and reasoning. In *WACV*.

Schuhmann, C.; Vencu, R.; Beaumont, R.; Kaczmarczyk, R.; Mullis, C.; Katta, A.; Coombes, T.; Jitsev, J.; and Komatsuzaki, A. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Seo, S.; Lee, J.-Y.; and Han, B. 2020. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*.

Sima, C.; Renz, K.; Chitta, K.; Chen, L.; Zhang, H.; Xie, C.; Luo, P.; Geiger, A.; and Li, H. 2024. Drivelm: Driving with graph visual question answering. In *ECCV*.

Vasudevan, A. B.; Dai, D.; and Van Gool, L. 2018. Object referring in videos with language and human gaze. In *CVPR*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wang, S.; Liu, Y.; Wang, T.; Li, Y.; and Zhang, X. 2023. Exploring Object-Centric Temporal Modeling for Efficient Multi-View 3D Object Detection. In *ICCV*.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*.

Wu, D.; Chang, J.; Jia, F.; Liu, Y.; Wang, T.; and Shen, J. 2023a. Topomlp: An simple yet strong pipeline for driving topology reasoning. *arXiv preprint arXiv:2310.06753*.

Wu, D.; Han, W.; Wang, T.; Dong, X.; Zhang, X.; and Shen, J. 2023b. Referring Multi-Object Tracking. In *CVPR*.

Wu, D.; Jia, F.; Chang, J.; Li, Z.; Sun, J.; Han, C.; Li, S.; Liu, Y.; Ge, Z.; and Wang, T. 2023c. The 1st-place solution for cvpr 2023 openlane topology in autonomous driving challenge. *arXiv preprint arXiv:2306.09590*.

Wu, D.; Wang, T.; Zhang, Y.; Zhang, X.; and Shen, J. 2023d. Onlinerefer: A simple online baseline for referring video object segmentation. In *ICCV*.

Yin, T.; Zhou, X.; and Krähenbühl, P. 2021. Center-based 3D Object Detection and Tracking. In *CVPR*.

Yu, L.; Poirson, P.; Yang, S.; Berg, A. C.; and Berg, T. L. 2016. Modeling context in referring expressions. In *ECCV*.

Zeng, F.; Dong, B.; Zhang, Y.; Wang, T.; Zhang, X.; and Wei, Y. 2022. Motr: End-to-end multiple-object tracking with transformer. In *ECCV*.