VisionTrap: Vision-Augmented Trajectory Prediction Guided by Textual Descriptions

Seokha Moon[1], Hyun Woo[1], Hongbeen Park[1], Haeji Jung[1],
Reza Mahjourian[2], Hyung-gun Chi[3], Hyerin Lim[4], Sangpil Kim[1], and
Jinkyu Kim[1]

[1] Korea University, Seoul 02841, Republic of Korea
[2] The University of Texas at Austin, Texas 78712, USA
[3] Perdue University, West Lafayette 95008, USA
[4] Hyundai Motor Company, Seongnam 13529, Republic of Korea

Abstract. Predicting future trajectories for other road agents is an essential task for autonomous vehicles. Established trajectory prediction methods primarily use agent tracks generated by a detection and tracking system and HD map as inputs. In this work, we propose a novel method that also incorporates visual input from surround-view cameras, allowing the model to utilize visual cues such as human gazes and gestures, road conditions, vehicle turn signals, etc, which are typically hidden from the model in prior methods. Furthermore, we use textual descriptions generated by a Vision-Language Model (VLM) and refined by a Large Language Model (LLM) as supervision during training to guide the model on what to learn from the input data. Despite using these extra inputs, our method achieves a latency of 53 ms, making it feasible for real-time processing, which is significantly faster than that of previous single-agent prediction methods with similar performance. Our experiments show that both the visual inputs and the textual descriptions contribute to improvements in trajectory prediction performance, and our qualitative analysis highlights how the model is able to exploit these additional inputs. Lastly, in this work we create and release the nuScenes-Text dataset, which augments the established nuScenes dataset with rich textual annotations for every scene, demonstrating the positive impact of utilizing VLM on trajectory prediction. Our project page is at https://moonseokha.github.io/VisionTrap.

Keywords: Motion Forecasting · Trajectory Prediction · Autonomous Driving · nuScenes-Text Dataset

1 Introduction

Predicting agents' future poses (or trajectories) is crucial for safe navigation in dense and complex urban environments. To achieve such task successfully, it is required to model the following aspects: (i) understanding individual's behavioral contexts (*e.g.*, actions and intentions), (ii) agent-agent interactions, and

[*] Corresponding author: J. Kim (jinkyukim@korea.ac.kr)

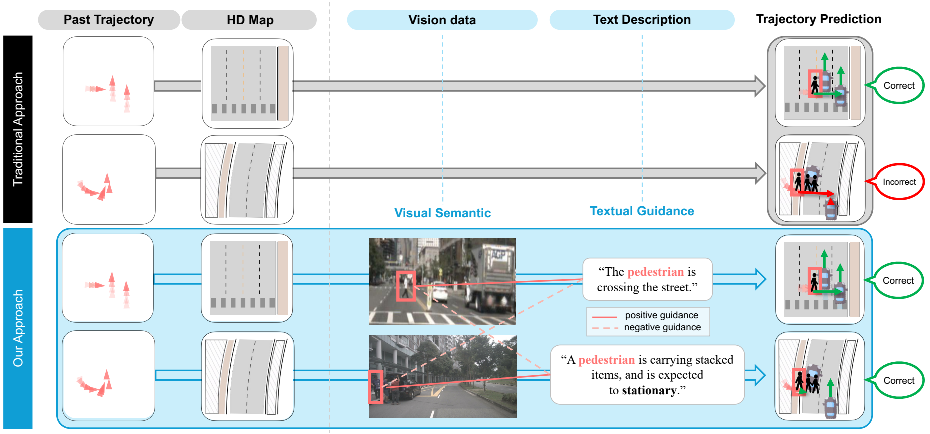


Fig. 1: Existing approaches are often conditioned only on agents' past trajectories and HD map to predict future trajectories. Here, we want to explore leveraging camera images and textual descriptions obtained from images to better learn the agent's behavioral context and agent-environment interactions by incorporating high-level semantic information into the prediction process, such as "a pedestrian is carrying stacked items, and is expected to stationary."

(iii) agent-environment interactions (*e.g.*, pedestrians on the crosswalk). Recent works [5, 12, 13, 24, 25, 31, 49, 50] have achieved remarkable progress, but their inputs are often limited – they mainly use a high-definition (HD) map and agents' past trajectories from a detection and tracking system as inputs.

HD map is inherently static, and only provide pre-defined information that limits their adaptability to changing environmental conditions like traffic near construction areas or weather conditions. They also cannot provide visual data for understanding agents' behavioral context, such as pedestrians' gazes, orientations, actions, gestures, and vehicle turn signals, all of which can significantly influence agents' behavior. Therefore, scenarios requiring visual context understanding may necessitate more than non-visual input for better and more reliable performance.

In this paper, we advocate for leveraging visual semantics in the trajectory prediction task. We argue that visual inputs can provide useful semantics, which non-visual inputs may not provide, for accurately predicting agents' future trajectories. Despite its potential advantages, only a few works [10, 23, 27, 36–39] have used vision data to improve the performance of trajectory prediction in autonomous driving domain. Existing approaches often utilize images of the area where the agent is located or the entire image without explicit instructions on what information to extract. As a result, these methods tend to focus only on salient features, leading to sub-optimal performance. Additionally, because they typically rely solely on frontal-view images, it becomes challenging to fully recognize the surrounding driving environment.

To address these limitations and harness the potential of visual semantics, we propose **VisionTrap**, a vision-augmented trajectory prediction model that efficiently incorporates visual semantic information. To leverage visual semantics obtained from surround-view camera images, we first encode them into a composite Bird's Eye View (BEV) feature along with map data. Given this vision-aware BEV scene feature, we use a deformable attention mechanism to extract scene information from relevant areas (using predicted agents' future positions), and augment them into per-agent state embedding, producing scene-augmented state embedding. In addition, recent works [4,15,23,27,36] have shown that classifying intentions can improve model performance by helping predict agents' instantaneous movements. Learning with supervision of each agent's intention helps avoid training restrictions and oversimplified learning that may not yield optimal performance. However, annotating agents' intentions by dividing them into action categories involves inevitable ambiguity, which can be costly and hinder efficient scalability. Moreover, creating models that rely on these small sets can limit the model's expressiveness. Thus, as shown in Fig. 1, we leverage textual guidance as supervision to guide the model in leveraging richer visual semantics by aligning visual features (*e.g.*, an image of a pedestrian nearby a parked vehicle) with textual descriptions (*e.g.*, "a pedestrian is carrying stacked items, and is expected to stationary."). While we use additional input data, real-time processing is crucial in autonomous driving. Therefore, we designed VisionTrap based on a real-time capable model proposed in this paper. VisionTrap efficiently utilizes visual semantic information and employs textual guidance only during training. This allows it to achieve performance comparable to high-accuracy, non-real-time single-agent prediction methods [7, 29] while maintaining real-time operation.

Since currently published autonomous driving datasets do not include textual descriptions, we created the nuScenes-Text dataset based on the large-scale nuScenes dataset [3], which includes vision data and 3D coordinates of each agent. The nuScenes-Text dataset collects textual descriptions that encompass high-level semantic information, as shown in Fig. 8: "A man wearing a blue shirt is talking to another man, expecting to cross the street when the signal changes." Automating this annotation process, we utilize both a Vision-Language Model (VLM) and a Large-Language Model (LLM).

Our extensive experiments on the nuScenes dataset show that our proposed text-guided image augmentation is effective in guiding our trajectory prediction model successfully to learn individuals' behavior and environmental contexts, producing a significant gain in trajectory prediction performance.

2 Related Work

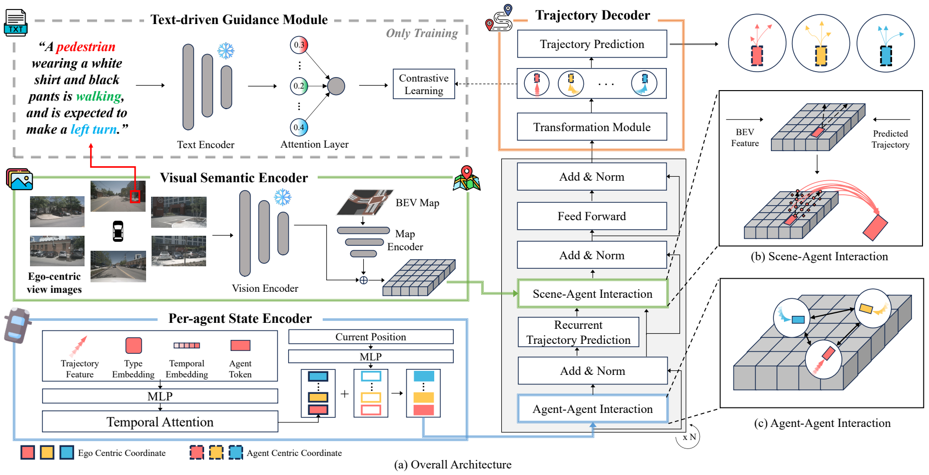
Encoding Behavioral Contexts for Trajectory Prediction. Recent works in trajectory prediction utilize past trajectory observations and HD map to provide static environmental context. Traditional methods use rasterized Bird's Eye View (BEV) maps with ConvNet blocks [5,12,34,41,44], while recent approaches employ vectorized maps with graph-based attention or convolution layers for better understanding complex topologies [11, 13, 14, 21, 40, 41]. However, HD maps are static and cannot adapt to changes, like construction zones affecting agent behavior. To address this, some works [10, 27, 37–39] aim to address these issues by utilizing images. To obtain meaningful visual semantic information about the situations an agent faces in a driving scene, it is necessary to utilize environmental information containing details from the objects themselves and from the environments they interact with. However, [27,37,39] focus solely on extracting information about agents' behavior using images near the agents, while [10,38] process the entire image at once and focus only on information about the scene without considering the parts that agents need to interact with. Therefore, in this paper, we propose an effective way to identify relevant parts of the image that each agent should focus on and efficiently learn semantic information from those parts.

Scene-centric vs. Agent-centric. Two primary approaches to predicting road agents' future trajectories are scene-centric and agent-centric. Scene-centric methods [32,42,47] encode each agent within a shared scene coordinate system, ensuring rapid inference speed but may exhibit slightly lower performance than agent-centric methods. Agent-centric approaches [2,8,24,25,50] standardize environmental elements and separately predict agents' future trajectories, offering improved predictive accuracy. However, their inference time and memory requirements are linearly scaled with the number of agents in the scene, posing a scalability challenge in dense urban environments with hundreds of pedestrians and vehicles. Thus, in this paper, we focus on scene-centric approaches.

Multimodal Contrastive Learning. With the increasing diversity of data sources, multimodal learning has become popular as it aims to effectively integrate information from various modalities. One of the common and effective approaches for multimodal learning is to align the modalities in a joint embedding space, using contrastive learning [18,35,45]. Contrastive Learning (CL) pulls together the positive pairs and pushes away the negative pairs, constructing an embedding space that effectively accommodates the semantic relations among the representations. Although CL is renowned for its ability to create a robust embedding space, its typical training mechanism introduces sampling bias, unintentionally incorporating similar pairs as negative pairs [6]. Debiasing strategies [6,16,17,30,48] have been introduced to mitigate such false-negatives, and it is particularly crucial in autonomous driving scenarios where multiple agents within a scene might have similar intentions in their behaviors. In our work, we carefully design our contrastive loss by filtering out the negative samples that are considered to be false-negatives. Inspired by [30,48], we do this by utilizing the sentence representations and their similarities, and finally achieve debiased contrastive learning in multimodal setting.

3 Method

This paper explores leveraging high-level visual semantics to improve the trajectory prediction quality. In addition to conventionally using agents' past trajectories and their types as inputs, we advocate for using visual data as an additional input to utilize agents' visual semantics. As shown in Fig. 2, our model consists of four main modules: (i) Per-agent State Encoder, (ii) Visual Semantic Encoder, (iii) Text-driven Guidance module, and (iv) Trajectory Decoder. Our *Per-agent*



(b) Scene-Agent Interaction

(c) Agent-Agent Interaction

(a) Overall Architecture

Fig. 2: An overview of VisionTrap, which consists of four main steps: (i) Per-agent State Embedding, which produces per-agent context features given agents' state observations; (ii) Visual Semantic Encoder, which transforms multi-view images with an HD map into a unified BEV feature, updating agents' state embedding via a deformable attention layer; (iii) Text-driven Guidance Module, which supervises the model to reason about detailed visual semantics and (iv) Trajectory Decoder, which predicts agents' the future poses in a fixed time horizon.

State Encoder takes as an input a sequence of state observations (which are often provided by a detection and tracking system), producing per-agent context features (Sec. 3.1). In our *Visual Semantic Encoder*, we encode multi-view images (capturing the surrounding view around the ego vehicle) into a unified Bird's Eye View (BEV) feature, followed by concatenation with a dense feature map of road segments. Given this BEV feature, the per-agent state embedding is updated in the Scene-Agent Interaction module (Sec. 3.2). We utilize *Text-driven Guidance module* to supervise the model to understand or reason about detailed visual semantics, producing richer semantics (Sec. 3.3). Lastly, given per-agent features with rich visual semantics, our *Trajectory Decoder* predicts the future positions for all agents in the scene in a fixed time horizon (Sec. 3.4).

3.1 Per-agent State Encoder

Encoding Agent State Observations. Following recent trajectory prediction approaches [31, 50], we first encode per-agent state observations (*e.g.*, agent's observed trajectory and semantic attributes) provided by object detection and tracking systems. We utilize the geometric attributes with relative positions (instead of absolute positions) by representing the observed trajectory of agent i as $\{p_i^t - p_i^{t-1}\}_{t=1}^{T}$ where $p_i^t = (x_i^t, y_i^t)$ is the location of agent i in an ego-centric coordinate system at time step $t \in \{1, 2, \ldots, T\}$. T denotes the observation time horizon. Note that we use an ego-centric (scene-centric) coordinate system where a scene is centered and rotated around the current ego-agent's location and orientation. Given these geometric attributes and their semantic attributes a_i (*i.e.*, agent types, such as cars, pedestrians, and cyclists), per-agent state embedding

$s_i^t \in \mathbb{R}^{d_s}$ for agent i at time step t is obtained as follows:

$$s_i^t = f_{\text{geometric}}(p_i^t - p_i^{t-1}) + f_{\text{type}}(a_i) + f_{\text{PE}}(e^t), \tag{1}$$

where $f_{\text{geometric}} : \mathbb{R}^2 \to \mathbb{R}^{d_s}$, $f_{\text{type}} : \mathbb{R}^1 \to \mathbb{R}^{d_s}$, and $f_{\text{PE}} : \mathbb{R}^{d_{pe}} \to \mathbb{R}^{d_s}$ are MLP blocks. Note that we use the learned positional embeddings $e^t \in \mathbb{R}^{d_{pe}}$, guiding the model to learn (and utilize) the temporal ordering of state embeddings.

Encoding Temporal Information. Following existing approaches [46,50], we utilize a temporal Transformer encoder to learn the agent's temporal information over the observation time horizon. Given the sequence of per-agent state embeddings $\{s_i^t\}_{t=1}^T$ and an additional learnable token $s^{T+1} \in \mathbb{R}^{d_s}$ stacked into the end of the sequence, we feed these input into the temporal (self-attention) attention block, producing per-agent spatio-temporal representations $s_i' \in \mathbb{R}^{d_s}$.

Encoding Interaction between Agents. We further use the cross-attention-based agent-agent interaction module to learn the relationship between agents. Further, as our model depends on the geometric attributes with relative positions, we add embeddings of the agents' current position p_i^T to make the embeddings spatially aware, producing per-agent representation $z_i = s_i' + f_{loc}(p_i^T)$ where $f_{loc} : \mathbb{R}^2 \to \mathbb{R}^{d_s}$ is another MLP block. This process is performed at once within the ego-centric coordinate system to eliminate the cost of recalculating correlation distances with other agents for each individual agent. The agent state embedding z_i is used as the query vector, and those of its neighboring agents are converted to the key and the value vectors as follows:

$$q_i^{\text{Interact}} = W_Q^{\text{Interact}} z_i, \quad k_j^{\text{Interact}} = W_K^{\text{Interact}} z_j, \quad v_j^{\text{Interact}} = W_V^{\text{Interact}} z_j, \tag{2}$$

where $W_Q^{\text{Interact}}, W_K^{\text{Interact}}, W_V^{\text{Interact}} \in \mathbb{R}^{d_{\text{Interact}} \times d_s}$ are learnable matrices.

3.2 Visual Semantic Encoder

Vision-Augmented Scene Feature Generation. Given ego-centric multi-view images $\mathcal{I} = \{\mathcal{I}_j\}_{j=1}^{n_I}$, we feed them into Vision Encoder using the same architecture from BEVDepth [20], to produce the BEV image feature as $B_I \in \mathbb{R}^{h \times w \times d_{\text{bev}}}$. Then, we incorporate the rasterized map information into the BEV embeddings to align B_I. We utilize CNN blocks with Feature Pyramid Network (FPN) [22] to produce another BEV feature $B_{\text{map}} \in \mathbb{R}^{h \times w \times d_{\text{map}}}$. Lastly, we concatenate all generated BEV features into a composite BEV scene feature $B = [B_I; B_{\text{map}}] \in \mathbb{R}^{h \times w \times (d_{\text{bev}} + d_{\text{map}})}$. In this process, we compute map aligned around the current location and direction of the ego vehicle only once, even in the presence of n agents, as we adopt an ego-centric approach. This significantly reduces computational costs compared to agent-centric approaches, which require reconstructing and encoding map for each of the n agents.

Augmenting Visual Semantics into Agent State Embedding. When given the vision-aware BEV scene feature B, we use deformable cross-attention [51] module to augment map-aware visual scene semantics into the per-agent state

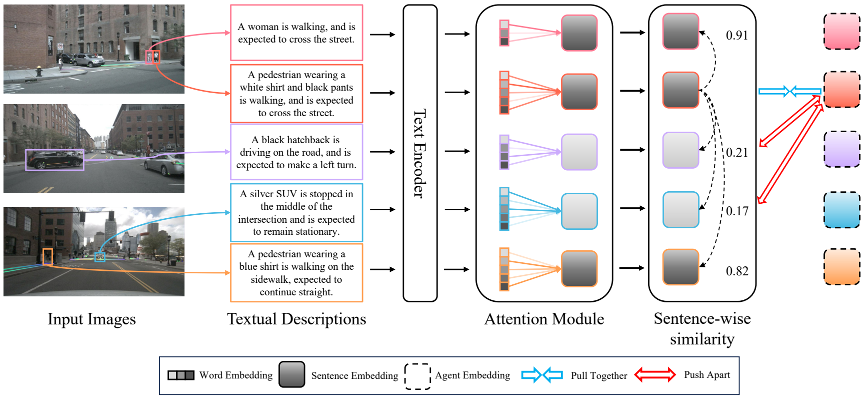

A woman is walking, and is expected to cross the street.
A pedestrian wearing a white shirt and black pants is walking, and is expected to cross the street.
A black hatchback is driving on the road, and is expected to make a left turn.
A silver SUV is stopped in the middle of the intersection and is expected to remain stationary.
A pedestrian wearing a blue shirt is walking on the sidewalk, expected to continue straight.
Text Encoder
0.91
0.21
0.17
0.82
Input Images
Textual Descriptions
Attention Module
Sentence-wise similarity
Word Embedding
Sentence Embedding
Agent Embedding
Pull Together
Push Apart

Fig. 3: An overview of our Text-driven Guidance Module. We extract word-level embeddings using pretrained BERT [9] as a text encoder, and then we use an attention module to aggregate these per-word embeddings into a composite sentence-level embedding. Based on the cosine similarity between these embeddings, we apply contrastive learning loss to ground textual descriptions into the agent's state embedding.

embedding z_i, as illustrated in Fig. 2 (b). This allows for the augmentation of agent state embedding z_i. Compared to commonly used ConvNet-based architectures [5, 12, 34], our approach leverages a wide receptive field and can selectively focus on scene feature, explicitly extracting multiple areas where each agent needs to focus and gather information. Additionally, as the agent state embedding is updated for each block, the focal points for the agent also require repeated refinement. To achieve this, we employ a Recurrent Trajectory Prediction module, which utilizes the same architecture as the main trajectory decoder(explained in Sec. 3.4). This module refines the agent's future trajectory $u^{\text{aux}} = \{u_i^{\text{aux}}\}_{i=1}^{T_f}$ by recurrently improving the predicted trajectories. These refined trajectories serve as reference points for agents to focus on in the Scene-Agent Interaction module, integrating surrounding information around the reference points into the agent's function. Our module is defined as follows:

$$z_i^{\text{scene}} = z_i^{\text{interact}} + \sum_{h=1}^{H} W_h \left[\sum_{o=1}^{O} \left(\alpha_{hio} W_h' \mathbf{B}_{\left(u_i^{\text{aux}} + \triangle u_{hio}^{\text{aux}}\right)} \right) \right], \tag{3}$$

where H denotes the number of attention heads and O represents the number of offset points for every reference point u_i^{aux} where we use an auxiliary trajectory predictor and use the agent's predicted future positions as reference points. Note that W_h and W_h' are learnable matrices, and α_{hio} is the attention weight for each learnable offset $\triangle u_{hio}^{\text{aux}}$ in each head. The number of attention points is typically set fewer than the number of surrounding road elements, reducing computational costs.

3.3 Text-driven Guidance Module

We observe that our visual semantic encoder simplifies visual reasoning about a scene to focus on salient visible features, resulting in sub-optimal performance in trajectory prediction. For instance, the model may primarily focus on the vehicle itself, disregarding other semantic details, such as “a vehicle waiting in front of the intersection with turn signals on, expected to turn left.” Therefore, we introduce the Text-driven Guidance Module to supervise the model, allowing the model to understand the context of the agents using detailed visual semantics. For this purpose, we employ multi-modal contrastive learning where positive pair is pulled together and negative pairs are pushed farther. However, the textual descriptions for prediction tasks in the driving domain are diverse in expression, posing an ambiguity in forming negative pairs between descriptions.

To address this, as shown in Fig. 3, we extract word-level embeddings using BERT [9], and then we use a attention module to aggregate these per-word embeddings into a composite sentence-level embedding $\mathcal{T}_i$ for agent i. Given $\mathcal{T}_i$, we measure cosine similarity with other agents' sentence-level embeddings $\mathcal{T}_j$ for $j \neq i$, and we treat as negative pairs if $\mathrm{sim}_{\cos}(\mathcal{T}_i, \mathcal{T}_j) < \theta_{\mathrm{th}}$ where θ_{th} is a threshold value (we set $\theta_{\mathrm{th}} = 0.8$ in our experiments). Further, we limit the number of negative pairs within a batch for stable optimization, which is particularly important as the number of agents in a given scene varies. Specifically, given an agent i, we choose top-k sentence-level embeddings from $\{\mathcal{T}_j\}$ sorted in ascending order for $j \neq i$. Subsequently, we form a positive pair between the agent's state embedding z_i^{scene} and corresponding textual embedding $\mathcal{T}_i$, while negative pairs as z_i^{scene} and $\{\mathcal{T}_j\}_{j=1}^{k}$. Ultimately, we use the following InfoNCE loss [33] to guide agent's state embedding with textual descriptions:

$$\mathcal{L}_{\mathrm{cl}} = -\log \frac{e^{\mathrm{sim}_{\cos}(z_i^{\mathrm{scene}}, \mathcal{T}_i)/\tau}}{\sum_{j=1}^{k} e^{\mathrm{sim}_{\cos}\left(z_i^{\mathrm{scene}}, \mathcal{T}_j\right)/\tau}}, \tag{4}$$

where τ is a temperature parameter used in the attention layer, enabling biasing the distribution of attention scores.

3.4 Trajectory Decoder

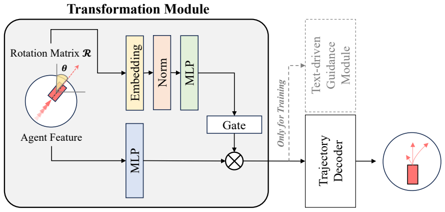


Fig. 4: An overview of transformation module, which standardizes agents' orientation.

Transformation Module. For fast inference speed and compatibility with ego-centric images, we adopt ego-centric approach in the State Encoder and Scene Semantic Interaction. However, as noted by Su *et al.* [42], ego-centric approaches typically underperform compared to agent-centric approaches due to the need to learn invariance for transformations and rotations between scene elements. This implies that the features of agents with similar future movements are not standardized. Thus, prior to utilizing the Text-driven Guidance Module and predicting each agent's future trajectory, we employ the Transformation Module to standardize each agent's orientation, aiming to mitigate the complexity associated with learning rotation invariance. This allows us to effectively apply the Text-driven Guidance Module, as we can make the features of agents in similar situations similar. As depicted in Fig. 4, the Transformation Module takes the agent's feature and rotation matrix $\mathcal{R}$ as input and propagates the rotation matrix to the agent's feature using a Multi-Layer Perceptron (MLP). This transformation enables the determination of which situations the agent's features face along the y-axis.

Trajectory Decoder. Similar to [5,31,34,43], we use a parametric distribution over the agent's future trajectories $u = \{u_i\}_{i=1}^{T_f}$ for $u_i \in \mathbb{R}^2$ as Gaussian Mixture Model (GMM). We represent a mode at each time step t as a 2D Gaussian distribution over a certain position with a mean $\mu_t \in \mathbb{R}^2$ and covariance $\Sigma_t \in \mathbb{R}^{2\times 2}$. Our decoder optimizes a weighted set of a possible future trajectory for the agent, producing full output distribution as

$$p(u) = \sum_{m=1}^{M} \rho_m \prod_{t=1}^{T_f} \mathcal{N}(u_t - \mu_m^t, \Sigma_m^t), \tag{5}$$

where our decoder produces a softmax probability ρ over mixture components and Gaussian parameters μ and Σ for M modes and T_f time steps.

Loss Functions. We optimize trajectory predictions and their associated confidence levels by minimizing $\mathcal{L}_{\text{traj}}$ to train our model in an end-to-end manner. We compute $\mathcal{L}_{\text{traj}}$ by minimizing the negative log-likelihood function between actual and predicted trajectories and the corresponding confidence score, and it can be formulated as follows:

$$\mathcal{L}_{\text{traj}} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\sum_{m=1}^{M} \frac{\rho_{i,m}}{\sqrt{2b^2}} \exp\left(-\frac{(\mathbf{Y}_i - \hat{\mathbf{Y}}_{i,m})^2}{2}\right)\right). \tag{6}$$

Here, b and $\mathbf{Y}$ represent the scale parameters and the real future trajectory, respectively. We denote predicted future positions as $\hat{\mathbf{Y}}_{i,m}$ and the corresponding confidence scores as $\rho_{i,m}$ for agent i at future time step t across different modes $m \in M$. Furthermore, we minimize an auxiliary loss function $\mathcal{L}_{\text{traj}}^{\text{aux}}$ similar to $\mathcal{L}_{\text{traj}}$ to train the trajectory decoder used by the Recurrent Trajectory Prediction module. Ultimately, our model is trained by minimizing the following loss $\mathcal{L}$, with $\lambda_{\text{traj}}^{\text{aux}}$ and λ_{cl} controlling the strength of each loss term:

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \lambda_{\text{traj}}^{\text{aux}}\mathcal{L}_{\text{traj}}^{\text{aux}} + \lambda_{\text{cl}}\mathcal{L}_{\text{cl}}. \tag{7}$$

4 nuScenes-Text Dataset

To our best knowledge, currently available driving datasets for prediction tasks lack textual descriptions of the actions of road users during various driving events. While the DRAMA dataset [26] offers textual descriptions for agents in driving scenes, it only provides a single caption for one agent in each scene alongside the corresponding bounding box. This setup suits detection and captioning tasks but not prediction tasks. To address this gap, we collect the textual

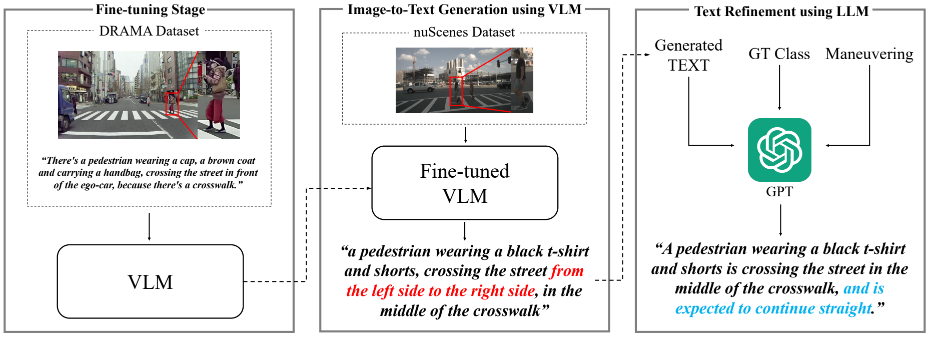
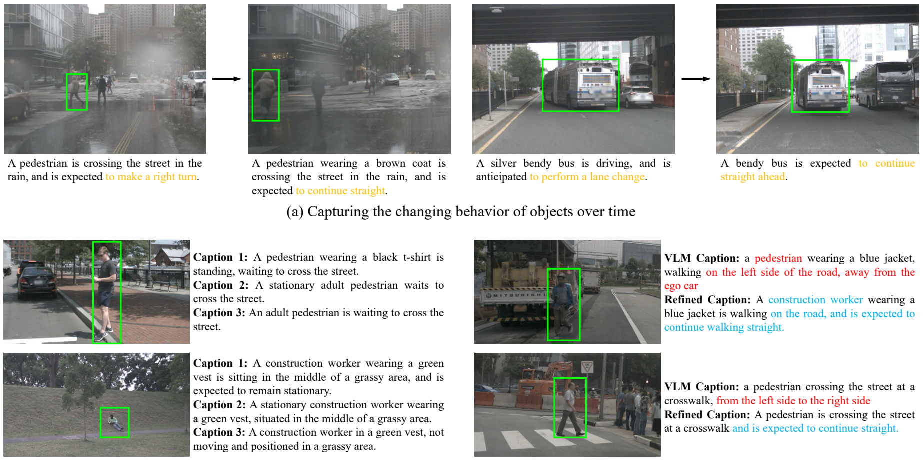

Fine-tuning Stage
DRAMA Dataset
"There's a pedestrian wearing a cap, a brown coat and carrying a handbag, crossing the street in front of the ego-car, because there's a crosswalk."
VLM
Image-to-Text Generation using VLM
nuScenes Dataset
Fine-tuned VLM
"a pedestrian wearing a black t-shirt and shorts, crossing the street from the left side to the right side, in the middle of the crosswalk"
Text Refinement using LLM
Generated TEXT
GT Class
Maneuvering
GPT
"A pedestrian wearing a black t-shirt and shorts is crossing the street in the middle of the crosswalk, and is expected to continue straight."

Fig. 5: To create the nuScenes-Text Dataset, three main steps are involved: (i) Fine-tuning stage using DRAMA Dataset [26], (ii) Image-to-Text Generation stage applying the fine-tuned VLM to the nuScenes Dataset [3], and (iii) Text Refinement process using ground truth information (*e.g.* GT class, Maneuvering) along with generated text and GPT [1]. The red color indicates that needs to be filtered out, while the cyan color indicates additional content related to the intention.

descriptions for the nuScenes dataset [3], which provides surround-view camera images, trajectories of road agents, and map data. With its diverse range of typical road agents activities, nuScenes is widely used in prediction tasks.

Textual Description Generation. We employ a three-step process for generating textual descriptions of agents from images, as illustrated in Fig. 5. Initially, we employ a pre-trained Vision-Language Model (VLM) BLIP-2 [19]. However, it often underperforms in driving-related image-to-text tasks. To address this, we fine-tune the VLM with the DRAMA dataset [26], containing textual descriptions of agents in driving scenes. We isolate the bounding box region representing the agent of interest, concatenate it with the original image (Fig. 5), and leverage the fine-tuned VLM to generate descriptions for each agent separately in the nuScenes dataset [3] as an image captioning task. However, the generated descriptions often lack correct action-related details, providing unnecessary information for prediction. To address shortcomings, we refine generated texts using GPT [1], a well-known Large Language Model (LLM). Inputs include the generated text, agent type, and maneuvering. Rule-based logic determines the agent's maneuvering (*e.g.*, stationary, lane change, turn right). We use prompts to correct inappropriate descriptions, aiming to generate texts that provide prediction-related information on agent type, actions, and rationale. Examples are provided in Fig. 6, with additional details (*e.g.*, rule-based logic, GPT prompt) in the supplemental material.

Coverage of nuScenes-Text Dataset. In this section, we demonstrate how well our created nuScene-Text Dataset encapsulates the context of the agent, as depicted in Fig. 6, and discuss the coverage and benefits of this dataset. Fig. 6a represents the contextual information of the agent changing over time in text form. This attribute assists in accurately predicting object trajectories under behavioral context changes. We also demonstrate in Fig. 6b that distinctive characteristics of each object can be captured (*e.g.*, "A pedestrian waiting



A pedestrian is crossing the street in the rain, and is expected to make a right turn.

A pedestrian wearing a brown coat is crossing the street in the rain, and is expected to continue straight.

A silver bendy bus is driving, and is anticipated to perform a lane change.

A bendy bus is expected to continue straight ahead.

(a) Capturing the changing behavior of objects over time

Caption 1: A pedestrian wearing a black t-shirt is standing, waiting to cross the street.
Caption 2: A stationary adult pedestrian waits to cross the street.
Caption 3: An adult pedestrian is waiting to cross the street.

Caption 1: A construction worker wearing a green vest is sitting in the middle of a grassy area, and is expected to remain stationary.
Caption 2: A stationary construction worker wearing a green vest, situated in the middle of a grassy area.
Caption 3: A construction worker in a green vest, not moving and positioned in a grassy area.

(b) Diversity of generated textual descriptions

VLM Caption: a pedestrian wearing a blue jacket, walking on the left side of the road, away from the ego car
Refined Caption: A construction worker wearing a blue jacket is walking on the road, and is expected to continue walking straight.

VLM Caption: a pedestrian crossing the street at a crosswalk, from the left side to the right side
Refined Caption: A pedestrian is crossing the street at a crosswalk and is expected to continue straight.

(c) Refinement of textual description with LLM

Fig. 6: Examples of our generated textual descriptions

to cross the street.", "A construction worker sitting on the lawn.") and generate three unique textual descriptions for each object, showcasing diverse perspectives. Additionally, to enhance text descriptions when the VLM generates incorrect agent types, behavior predictions, or harmful information, such as "from the left side to the right side", which can be misleading due to the directional variation in BEV depending on the camera's orientation, we refine the text using an LLM. This refinement process aims to improve text quality for identifying driving scenes through surround images. Fig. 6c illustrates this improvement process, ensuring the relevance and accuracy of text by removing irrelevant details (indicated by red) and adding pertinent information (indicated by cyan).

Dataset Statistics. Our created dataset contains 1,216,206 textual descriptions for 391,732 objects (three for each object), averaging 13 words per description. In Fig. 7, we visualize frequently used words, highlighting the dataset's rich vocabulary and diversity. Further, we conduct a human evaluation using Amazon Mechanical Turk (Mturk) to quantitatively evaluate image-text alignments. 5 human evaluators are recruited, and it is performed on a subset of 1,000 randomly selected samples. Each evaluator is presented with the full image, cropped object image, and corresponding text and asked the question: "Is the image well-aligned with the text, considering the reference image?". The results show that 94.8% of the respondents chose 'yes', indicating a high level of accuracy in aligning images with texts. All results are aggregated through a majority vote. Further details on the nuScenes-Text Dataset are provided in the supplemental material.

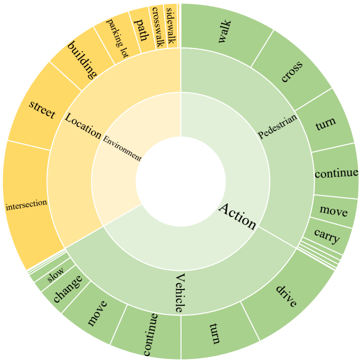


Fig. 7: Frequency of words

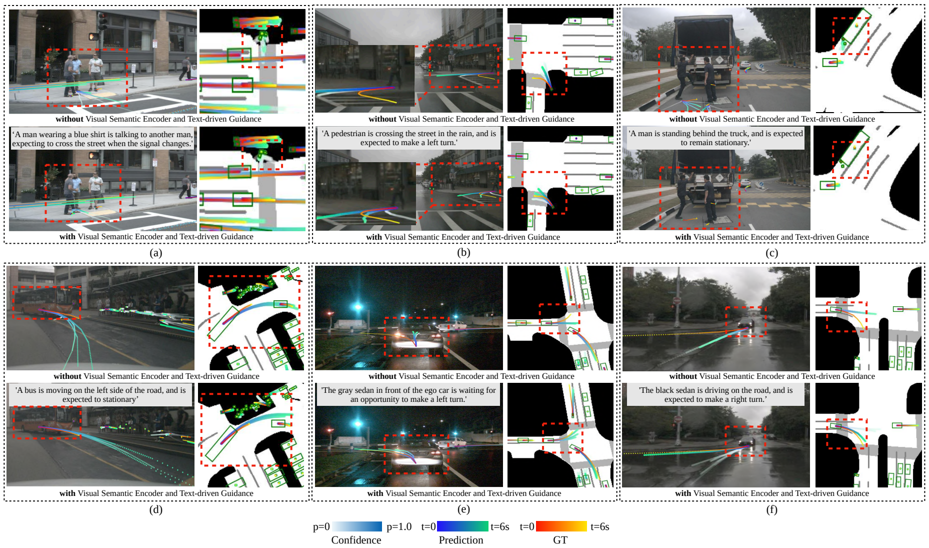

without Visual Semantic Encoder and Text-driven Guidance
'A man wearing a blue shirt is talking to another man, expecting to cross the street when the signal changes.'
with Visual Semantic Encoder and Text-driven Guidance
(a)
without Visual Semantic Encoder and Text-driven Guidance
'A pedestrian is crossing the street in the rain, and is expected to make a left turn.'
with Visual Semantic Encoder and Text-driven Guidance
(b)
without Visual Semantic Encoder and Text-driven Guidance
'A man is standing behind the truck, and is expected to remain stationary.'
with Visual Semantic Encoder and Text-driven Guidance
(c)
without Visual Semantic Encoder and Text-driven Guidance
'A bus is moving on the left side of the road, and is expected to stationary'
with Visual Semantic Encoder and Text-driven Guidance
(d)
without Visual Semantic Encoder and Text-driven Guidance
'The gray sedan in front of the ego car is waiting for an opportunity to make a left turn.'
with Visual Semantic Encoder and Text-driven Guidance
(e)
without Visual Semantic Encoder and Text-driven Guidance
'The black sedan is driving on the road, and is expected to make a right turn.'
with Visual Semantic Encoder and Text-driven Guidance
(f)
p=0 p=1.0
Confidence
t=0 t=6s
Prediction
t=0 t=6s
GT

Fig. 8: Examples of trajectory prediction outputs in six different scenarios. The examples on the top row represent scenarios with pedestrians, while those on the bottom row have vehicles. We also provide ground truth textual descriptions about an object in a red box, which were not seen during inference.

5 Experiments

Dataset. We conduct experiments using the nuScenes dataset [3], which offers two versions: (i) a dataset dedicated to a trajectory prediction task and (ii) a whole dataset. While the former focuses solely on single-agent prediction tasks, the latter is more suitable for our purposes. Therefore, we provide scores for both datasets in our experiments. Further implementation, evaluation, and dataset details can be found in the supplemental material.

Qualitative Analysis. Fig. 8 presents the results of VisionTrap on nuScenes dataset [3], demonstrating the impact of Visual Semantic Encoder and Text-driven Guidance Module on agent trajectory prediction.

The top row shows improved results of pedestrians. For (a), while the result without visual information predicts the man will cross the crosswalk, the prediction with visual information indicates the man will remain stationary due to red traffic light and people talking to each other rather than trying to cross the road. (b) presents how gaze and body orientation help in predicting the pedestrian's intention to walk towards the crosswalk, and (c) provides visual context of the man getting on a stationary vehicle, implying the trajectory of the man would remain stationary as well. The following row exhibits the improved prediction results of vehicles. In (d), understanding that the people are standing at a bus stop enables the model to make a reasonable prediction for the bus. (e) gives a visual cue of turn signal, indicating the vehicle's intention of turning left. Lastly, visual context in (f) leads to a more stable prediction of the vehicle turning right, as the image clearly shows the vehicle is directed towards its right.

Table 1: Trajectory prediction performance comparison on nuScenes [3] dataset regarding ADE_{10}, MR_{10}, and FDE_1. Inference times are reported in milliseconds (msec), measured based on 12 agents using a single RTX 3090 Ti GPU.

Model	Prediction Method	Using Map Data	Time ↓ (msec)	ADE_{10} ↓	MR_{10} ↓	FDE_1 ↓	
Multipath [5]	single	✔	87	1.50	0.74	-	
MHA-JAM [29]	single	✔	-	1.24	0.46	8.57	
P2T [7]	single	✔	116	1.16	0.46	10.5	
PGP [8]	single	✔	215	1.00	0.37	7.17	
LAformer [24]	single	✔	115	0.93	0.33	-	
Trajectron++ [41]	multi	✔	38	1.51	0.57	9.52	
AgentFormer [46]	multi	✔	107	1.45	-	-	Average
VisionTrap baseline	multi		13	1.48	0.56	10.75	improvement:
+ Map Encoder	multi	✔	21	1.40	0.53	10.41	4.65%
+ Visual Semantic Encoder	multi	✔	53	1.23	0.36	9.32	21.97%
+ Text-driven Guidance (Ours)	multi	✔	53	1.17	0.32	8.72	27.56%

These examples highlight the crucial role of visual data in improving trajectory prediction accuracy, offering insights that cannot obtained from non-visual data. Further qualitative analysis details are available in the supplemental material.

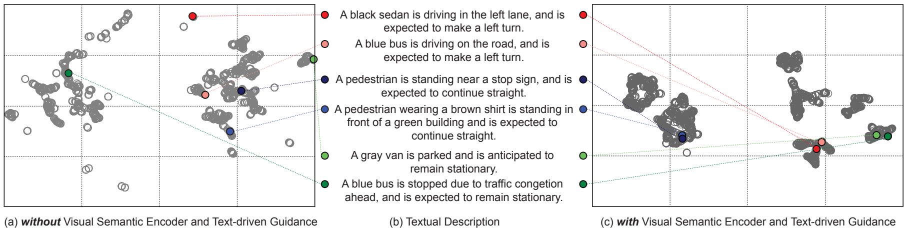
Quantitative Analysis. Tab. 1 compares our model with other methods for single and multi-agent prediction. Our query-based prediction model designed to effectively utilize visual semantic information and Text-driven Guidance Module, which we use as baseline, achieves the fastest inference speed. We also demonstrate that the Visual Semantic Encoder significantly improves performance, especially when combined with the Text-driven Guidance Module, yielding comparable results to existing single-agent prediction methods with better miss rate performance, while still maintaining real-time operation. These results suggest that vision data provides additional information inaccessible to non-vision data, and textual descriptions derived from vision data effectively guide the model.

Since our method employs egocentric surround-view images, it is feasible to effectively predict for all observed agents in the scene. We utilize the nuScenes dataset covering all scenes, enabling comprehensive evaluation of all observed agents (refer to Tab. 2). This demonstrates the contributions of all proposed components to predicting all agents in the scene.

Table 2: Ablation study of variant models on nuScenes [3] whole dataset.

Method	ADE_{10} ↓	FDE_{10} ↓	MR_{10} ↓
VisionTrap baseline	0.425	0.641	0.081
+ Map Encoder	0.407	0.601	0.075
+ Visual Semantic Encoder	0.382	0.551	0.056
+ Text-driven Guidance (Ours)	**0.368**	**0.535**	**0.051**

Finally, we emphasize that the purpose of this study is not to achieve state-of-the-art performance. Instead, our aim is to demonstrate that vision information, often overlooked in trajectory prediction tasks, can provide additional insights. These insights are inaccessible from non-vision data, thereby enhancing performance in trajectory prediction tasks. This is our original motivation for this task, and the results in Fig. 8, Tab. 1 and Tab. 2 provide justification for our method.



(a) ***without*** Visual Semantic Encoder and Text-driven Guidance (b) Textual Description (c) ***with*** Visual Semantic Encoder and Text-driven Guidance

Fig. 9: UMAP [28] visualizations for per-agent state embeddings from models (a) without and (c) with leveraging visual and textual semantics. (b) We also provide corresponding ground truth textual descriptions.

UMAP Visualization. We observe an overall improvement in clustering of agent state embeddings when leveraging visual and textual semantics in Fig. 9. Furthermore, extracting textual descriptions of agents within the same cluster group is shown to exhibit similar situations. This indicates that state embeddings for agents in similar situations are located in a similar embedding space.

Analyzing the Text-driven Guidance Module. To analyze the effect of each component of the proposed Text-Based Guidance Module, we removed each factor to see how the model performs, as shown in Tab. 3. In the case of A, we use simple symmetric contrastive loss that is used in [35]. However, our loss adopts asymmetric form of contrastive loss that only calculates softmax probabilities in one direction. B gives the result of incorporating symmetric loss in our loss design. C shows the result of removing the stage of negative pair refinement, allowing potential false-negatives. In D, we skip the process of ascending sorting and limiting the number of negative pairs. Removing these steps causes variance in number of agents considered each scene, leading to different scales of loss. In the end, our asymmetric contrastive loss with negative pairs refined and its number constrained demonstrated the best performance across all metrics.

Table 3: Performance comparison to analyze the effect of each component of Text-Based Guidance Module on the nuScenes [3] all dataset.

Method	$\mathrm{ADE}_6 \downarrow$	$\mathrm{FDE}_6 \downarrow$	$\mathrm{MR}_6 \downarrow$
A. CLIP loss	0.51	0.79	0.10
B. Ours w/ symmetric loss	0.50	0.76	0.10
C. Ours w/o refining negative pair	0.49	0.72	0.09
D. Ours w/o top-k algorithm	0.46	0.67	0.08
E. Ours	**0.44**	**0.66**	**0.07**

6 Conclusion

In this paper, we introduced an novel approach called VisionTrap to trajectory prediction by incorporating visual input from surround-view cameras. This enables the model to leverage visual semantic cues, which were previously inaccessible to traditional trajectory prediction methods. Additionally, we utilize text descriptions produced by a VLM and refined by a LLM to provide supervision, guiding the model in learning from the input data. Our thorough experiments demonstrate that both visual inputs and textual descriptions contribute to enhancing trajectory prediction performance. Furthermore, our qualitative analysis shows how the model effectively utilizes these additional inputs.

Acknowledgment. This work was supported by Autonomous Driving Center, Hyundai Motor Company R&D Division. This work was partly supported by IITP under the Leading Generative AI Human Resources Development(IITP-2024-RS-2024-00397085, 10%) grant, IITP grant (RS-2022-II220043, Adaptive Personality for Intelligent Agents, 10% and IITP-2024-2020-0-01819, ICT Creative Consilience program, 5%). This work was also partly supported by Basic Science Research Program through the NRF funded by the Ministry of Education(NRF-2021R1A6A1A13044830, 10%). This work also supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2024((International Collaborative Research and Global Talent Development for the Development of Copyright Management and Protection Technologies for Generative AI, RS-2024-00345025, 4%),(Research on neural watermark technology for copyright protection of generative AI 3D content, RS-2024-00348469, 25%)), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT)(RS-2019-II190079, 1%). We also thank Yujin Jeong and Daewon Chae for their helpful discussions and feedback.

References

1. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
2. Buhet, T., Wirbel, E., Bursuc, A., Perrotton, X.: Plop: Probabilistic polynomial objects trajectory planning for autonomous driving. arXiv preprint arXiv:2003.08744 (2020)
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
4. Casas, S., Luo, W., Urtasun, R.: Intentnet: Learning to predict intention from raw sensor data. In: Conference on Robot Learning. pp. 947–956. PMLR (2018)
5. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. arXiv preprint arXiv:1910.05449 (2019)
6. Chuang, C.Y., Robinson, J., Lin, Y.C., Torralba, A., Jegelka, S.: Debiased contrastive learning. Advances in neural information processing systems **33**, 8765–8775 (2020)
7. Deo, N., Trivedi, M.M.: Trajectory forecasts in unknown environments conditioned on grid-based plans. arXiv preprint arXiv:2001.00735 (2020)
8. Deo, N., Wolff, E., Beijbom, O.: Multimodal trajectory prediction conditioned on lane-graph traversals. In: Conference on Robot Learning. pp. 203–212. PMLR (2022)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Fang, L., Jiang, Q., Shi, J., Zhou, B.: Tpnet: Trajectory proposal network for motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6797–6806 (2020)

11. Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., Schmid, C.: Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11525–11533 (2020)
12. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., Moutarde, F.: Home: Heatmap output for future motion estimation. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). pp. 500–507. IEEE (2021)
13. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., Moutarde, F.: Thomas: Trajectory heatmap output with learned multi-agent sampling. arXiv preprint arXiv:2110.06607 (2021)
14. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., Moutarde, F.: Gohome: Graph-oriented heatmap output for future motion estimation. In: 2022 international conference on robotics and automation (ICRA). pp. 9107–9114. IEEE (2022)
15. Girase, H., Gang, H., Malla, S., Li, J., Kanehara, A., Mangalam, K., Choi, C.: Loki: Long term and key intentions for trajectory prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9803–9812 (2021)
16. Hwang, I., Lee, S., Kwak, Y., Oh, S.J., Teney, D., Kim, J.H., Zhang, B.T.: Selecmix: Debiased learning by contradicting-pair sampling. Advances in Neural Information Processing Systems **35**, 14345–14357 (2022)
17. Jang, T., Wang, X.: Difficulty-based sampling for debiased contrastive representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 24039–24048 (June 2023)
18. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021)
19. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)
20. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1477–1485 (2023)
21. Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., Urtasun, R.: Learning lane graph representations for motion forecasting. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 541–556. Springer (2020)
22. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
23. Liu, B., Adeli, E., Cao, Z., Lee, K.H., Shenoi, A., Gaidon, A., Niebles, J.C.: Spatiotemporal relationship reasoning for pedestrian intent prediction. IEEE Robotics and Automation Letters **5**(2), 3485–3492 (2020)
24. Liu, M., Cheng, H., Chen, L., Broszio, H., Li, J., Zhao, R., Sester, M., Yang, M.Y.: Laformer: Trajectory prediction for autonomous driving with lane-aware scene constraints. arXiv preprint arXiv:2302.13933 (2023)
25. Liu, Y., Zhang, J., Fang, L., Jiang, Q., Zhou, B.: Multimodal motion prediction with stacked transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7577–7586 (2021)
26. Malla, S., Choi, C., Dwivedi, I., Choi, J.H., Li, J.: Drama: Joint risk localization and captioning in driving. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1043–1052 (2023)

27. Malla, S., Dariush, B., Choi, C.: Titan: Future forecast using action priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11186–11196 (2020)
28. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
29. Messaoud, K., Deo, N., Trivedi, M.M., Nashashibi, F.: Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation (2020)
30. Miao, P., Du, Z., Zhang, J.: Debcse: Rethinking unsupervised contrastive sentence embedding learning in the debiasing perspective. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 1847–1856 (2023)
31. Nayakanti, N., Al-Rfou, R., Zhou, A., Goel, K., Refaat, K.S., Sapp, B.: Wayformer: Motion forecasting via simple & efficient attention networks. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 2980–2987. IEEE (2023)
32. Ngiam, J., Caine, B., Vasudevan, V., Zhang, Z., Chiang, H.T.L., Ling, J., Roelofs, R., Bewley, A., Liu, C., Venugopal, A., et al.: Scene transformer: A unified architecture for predicting multiple agent trajectories. arXiv preprint arXiv:2106.08417 (2021)
33. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
34. Phan-Minh, T., Grigore, E.C., Boulton, F.A., Beijbom, O., Wolff, E.M.: Covernet: Multimodal behavior prediction using trajectory sets. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14074–14083 (2020)
35. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
36. Rasouli, A., Kotseruba, I., Kunic, T., Tsotsos, J.K.: Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In: ICCV (2019)
37. Rasouli, A., Rohani, M., Luo, J.: Bifold and semantic reasoning for pedestrian behavior prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15600–15610 (October 2021)
38. Rasouli, A., Yau, T., Lakner, P., Malekmohammadi, S., Rohani, M., Luo, J.: Pepscenes: A novel dataset and baseline for pedestrian action prediction in 3d. arXiv preprint arXiv:2012.07773 (2020)
39. Rasouli, A., Yau, T., Rohani, M., Luo, J.: Multi-modal hybrid architecture for pedestrian action prediction. In: 2022 IEEE Intelligent Vehicles Symposium (IV). pp. 91–97. IEEE (2022)
40. Rowe, L., Ethier, M., Dykhne, E.H., Czarnecki, K.: Fjmp: Factorized joint multi-agent motion prediction over learned directed acyclic interaction graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13745–13755 (2023)
41. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16. pp. 683–700. Springer (2020)

42. Su, D.A., Douillard, B., Al-Rfou, R., Park, C., Sapp, B.: Narrowing the coordinate-frame gap in behavior prediction models: Distillation for efficient and accurate scene-centric motion forecasting. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 653–659. IEEE (2022)
43. Varadarajan, B., Hefny, A., Srivastava, A., Refaat, K.S., Nayakanti, N., Cornman, A., Chen, K., Douillard, B., Lam, C.P., Anguelov, D., et al.: Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 7814–7821. IEEE (2022)
44. Wu, D., Wu, Y.: Air^2 for interaction prediction. arXiv preprint arXiv:2111.08184 (2021)
45. Yuan, X., Lin, Z., Kuen, J., Zhang, J., Wang, Y., Maire, M., Kale, A., Faieta, B.: Multimodal contrastive training for visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6995–7004 (2021)
46. Yuan, Y., Weng, X., Ou, Y., Kitani, K.: Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
47. Zeng, W., Liang, M., Liao, R., Urtasun, R.: Lanercnn: Distributed representations for graph-centric motion forecasting. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 532–539. IEEE (2021)
48. Zhou, K., Zhang, B., Zhao, X., Wen, J.R.: Debiased contrastive learning of unsupervised sentence representations (2022)
49. Zhou, Z., Wang, J., Li, Y.H., Huang, Y.K.: Query-centric trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17863–17873 (2023)
50. Zhou, Z., Ye, L., Wang, J., Wu, K., Lu, K.: Hivt: Hierarchical vector transformer for multi-agent motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8823–8833 (2022)
51. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)