Supplementary: Temporal Object Captioning for Street Scene Videos from LiDAR Tracks

Vignesh Gopinathan
Aptiv and University of Wuppertal

Urs Zimmermann
Aptiv

Michael Arnold
Aptiv

Matthias Rottmann
Osnabrück University

1. LiDAR-based caption generation

In the next section, we detail the implementation of the various thresholds and the corresponding data used to extract the necessary information for constructing the final neighbor captions. Section 1.1 outlines the implementation of host motion and direction tagging, which supports host caption construction. Meanwhile, Sections 1.2, 1.3, 1.4, and 1.5 describe how the relevant information is generated for neighbor caption construction.

1.1. Host tagging

This is the first stage of the rule-based captioning system. Initially, the host's velocity and yaw rate are extracted from the sensor data. At each timestamp, thresholds (see Eq. (1) and Eq. (2)) are applied to assign the corresponding motion and direction tags:

$$\text{Motion tag} = \begin{cases} \text{Stationary}, & v < T_v, \\ \text{Accelerating}, & \frac{dv}{dt} > T_a, \\ \text{Decelerating}, & \frac{dv}{dt} < -T_a, \\ \text{Cruising}, & -T_a \leq \frac{dv}{dt} \leq T_a \end{cases} \tag{1}$$

Here, v, t and $\frac{dv}{dt}$ denote the host vehicle's instantaneous velocity, time between two successive velocity recordings, and the host's instantaneous acceleration, respectively. T_v and T_a represent the thresholds for the velocity and acceleration of the host vehicle.

The direction tag is defined as

$$\text{Direction tag} = \begin{cases} \text{Steering right}, & \omega < -T_\omega, \\ \text{Steering left}, & \omega > T_\omega, \\ \text{Heading straight}, & -T_\omega \leq \omega \leq T_\omega \end{cases} \tag{2}$$

where ω and t represent the yaw rate and the time between two successive yaw rate measurements of the host vehicle. T_ω is the threshold on the yaw rate of the host vehicle.

1.2. Non-Stationary neighbor selection

As explained in the methodology, a key step in caption generation involves extracting object class, lane tags, and motion tags from object tracks, obtained via a SOTA 3D LiDAR–based detector and tracker, on raw LiDAR data. These tags are generated exclusively for non-stationary neighbors, as determined by Eq. (3). A neighbor is deemed stationary by thresholding changes in its bounding box position relative to the host's initial frame, referred to as the host-compensated location. The transformation matrix for computing this host-compensated location is derived from the LiDAR sensor's calibration data and the host vehicle's position relative to its starting location:

$$\text{Stationary} = \begin{cases} \text{True}, & \frac{dh_x}{dt} < T_s \text{ and } \frac{dh_y}{dt} < T_s, \\ \text{False}, & \text{otherwise} \end{cases} \tag{3}$$

Here, h_x and h_y denote the object's center distances from the host vehicle's position at the first frame, measured along and perpendicular to the host's heading at that frame, respectively. The threshold T_s determines whether a neighbor is classified as stationary. Once non-stationary neighbors are identified, the tagging procedures are carried out as described in Sections 1.3, 1.4, and 1.5.

1.3. Lane tagging

In this section, we detail the implementation of the lane tagging algorithm used for caption construction. The algorithm is divided into two parts: the baseline lane tag and the lane tag itself. The baseline step identifies neighbors on oncoming, lateral (lanes perpendicular to the host lane, such as at intersections), or ongoing lanes by thresholding each neighbor's yaw, as shown in Eq. (4). The second step locates each neighbor relative to the host, classifying them as on left, right, or host lane by combining the baseline lane tag with the neighbor's bounding box center in the y-direction, as specified in Eq. (5).

$$\mathcal{L}_B = \begin{cases} \text{oncoming}, & \phi_{ol} < \phi < \phi_{ou}, \\ \text{lateral}, & \phi_{ll} < \phi < \phi_{lu}, \\ \text{ongoing}, & \text{otherwise} \end{cases} \tag{4}$$

where ϕ denotes the neighbor's yaw, and $\mathcal{L}_B$ represents the baseline lane tag, which is refined in a subsequent step to obtain the final lane tag. The parameters ϕ_{ol} and ϕ_{ou} are the lower and upper yaw thresholds for identifying neighbors on the oncoming lane, while ϕ_{ll} and ϕ_{lu} specify the yaw thresholds for identifying neighbors on the lateral lane.

$$\mathcal{L} = \begin{cases} \text{right lateral}, & \mathcal{L}_B = \text{lateral and } y > T_h, \\ \text{left lateral}, & \mathcal{L}_B = \text{lateral and } y < -T_h, \\ \text{host lateral}, & \mathcal{L}_B = \text{lateral and } -T_h \leq y \leq T_h, \\ \text{right}, & \mathcal{L}_B = \text{ongoing and } y > T_h, \\ \text{left}, & \mathcal{L}_B = \text{ongoing and } y < -T_h, \\ \text{host}, & \mathcal{L}_B = \text{ongoing and } -T_h \leq y \leq T_h, \\ \text{oncoming}, & \text{otherwise} \end{cases} \tag{5}$$

Here, $\mathcal{L}$ and $\mathcal{L}_B$ are the refined lane tag and the baseline lane tag respectively. y denotes the distance of the object's center from the host vehicle perpendicular to the host's heading direction and T_h is the threshold for determining whether a neighbor is on the host lane.

1.4. Motion tagging

Another key step is to classify the neighbor's motion relative to the host, using the previously determined lane tags along with a threshold on the neighbor's lateral position relative to the host's heading (Eq. (6)) and neighbor's position along the host's heading Eq. (7) for the lateral and non-lateral lanes respectively:

$$\mathcal{M} = \begin{cases} \text{stationary}, & \mathcal{L} = \text{lateral and } \frac{dy}{dt} = 0 \\ \text{approach}, & \mathcal{L} = \text{left lateral and } \frac{dy}{dt} > T_m, \\ \text{away}, & \mathcal{L} = \text{left lateral and } \left|\frac{dy}{dt}\right| > T_m, \\ \text{constant}, & \mathcal{L} = \text{left lateral and } -T_m < \frac{dy}{dt} < T_m, \\ \text{away}, & \mathcal{L} = \text{right lateral and } \frac{dy}{dt} > T_m, \\ \text{approach}, & \mathcal{L} = \text{right lateral and } \left|\frac{dy}{dt}\right| > T_m, \\ \text{constant}, & \mathcal{L} = \text{right lateral and } -T_m < \frac{dy}{dt} < T_m, \end{cases} \tag{6}$$

$$\mathcal{M} = \begin{cases} \text{stationary}, & \mathcal{L}_\mathcal{B} = \text{ongoing and } \frac{dx}{dt} = 0 \\ \text{away}, & \mathcal{L} = \text{ongoing and } \frac{dx}{dt} > T_m, \\ \text{approach}, & \mathcal{L}_\mathcal{B} = \text{ongoing and } \left|\frac{dx}{dt}\right| > T_m, \\ \text{constant}, & \mathcal{L} = \text{ongoing and } -T_m < \frac{dx}{dt} < T_m, \end{cases} \tag{7}$$

Here $\mathcal{M}$, $\mathcal{L}_\mathcal{B}$ and $\mathcal{L}$ denote the motion, baseline lane and lane tags, respectively. The terms $\frac{dx}{dt}$ and $\frac{dy}{dt}$ represents the neighbor's instantaneous velocity in the host's heading direction and the direction perpendicular to the host's heading respectively. T_m is the speed threshold used to determine whether a neighbor maintains a constant distance from the host.

1.5. Unified tag generation

After extracting the lane and motion tags, we concatenate them across time to form a timeseries of concatenated tags. This time-series is then "unified" into a unique sequence of concatenated tags (retaining their order of occurrence), referred to as the unified tag sequence, which captures the neighbor's action within that time window:

$$\begin{aligned} &c_i = l_i m_i, \mid l_i \in \mathcal{L},\ m_i \in \mathcal{M},\ i \in [0, n], \\ &C = [c_0, c_1, c_2, \ldots, c_n], \\ &U = [c_i \mid c_i \neq c_{i+1},\ i \in [0, n]] \end{aligned} \tag{8}$$

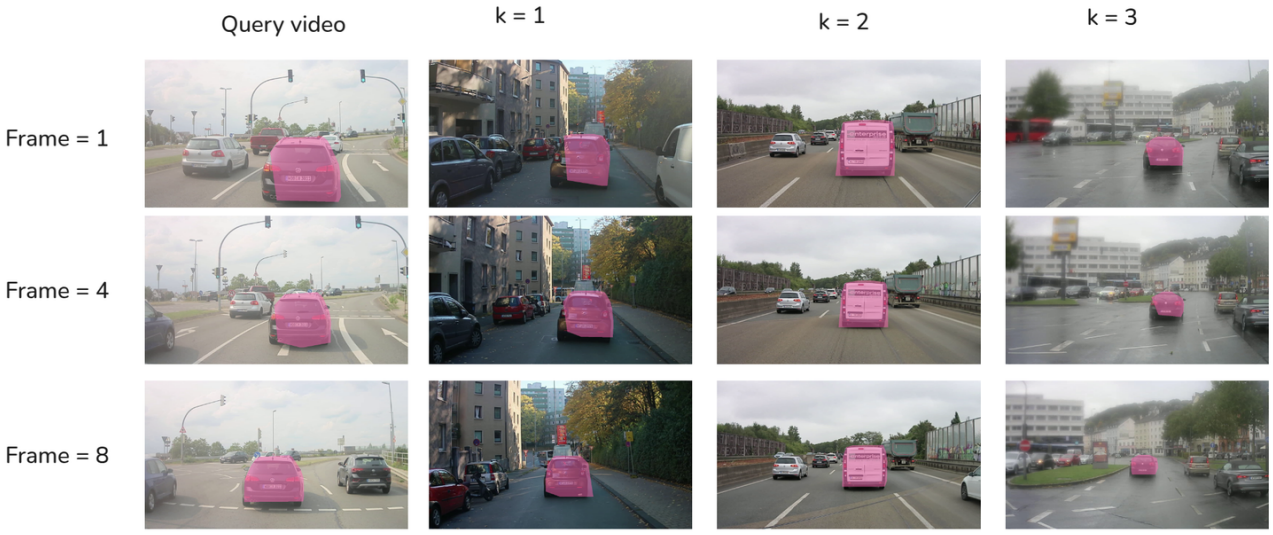
Here, c_i, l_i and m_i are the concatenated, lane and motion tags at timestep i respectively. $\mathcal{L}$, $\mathcal{M}$, C and U are the sequence of the neighbor's lane, motion, concatenated and unified tags respectively.

2. Extended study on embeddings

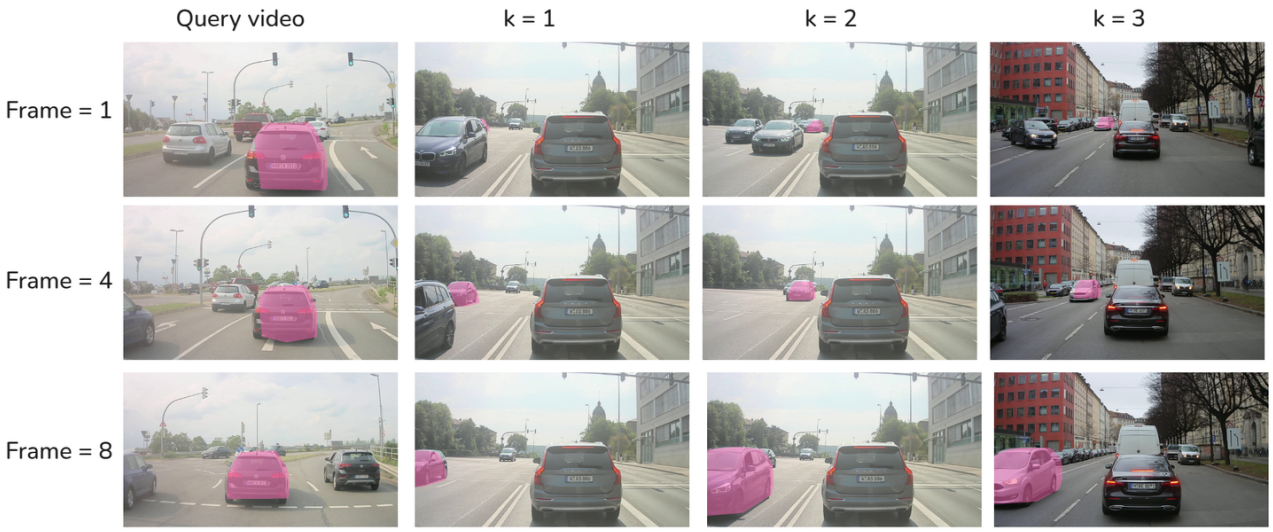
To further examine our model's understanding of temporal dynamics, we qualitatively evaluate its performance in a video-to-video retrieval task and compare it with that of ViCLIP. Figure 1 presents the top three nearest neighbor videos retrieved for a given query, using embeddings from both our model and ViCLIP. In the figure, the first column shows frames from the query video, while the second, third, and fourth columns display frames from the top three retrieved neighbors.

The results reveal a notable difference between the two models. Our model retrieves videos where the object of interest is consistently performing the same action as in the query, even though the surrounding visual context such as lighting conditions, background scenery, and weather varies significantly. This suggests that our model focuses more on the temporal and semantic content of the video, rather than being influenced by low-level visual similarities. In

contrast, ViCLIP's retrieved neighbors often appear more visually similar in appearance but are less aligned in terms of the underlying action. These observations indicate that our model exhibits reduced sensitivity to visual biases and demonstrates a stronger temporal understanding.



(a) Our model



(b) ViCLIP

Figure 1. **Video-to-video retrieval.** This figure shows an example of retrieving the top-three nearest neighbors for a given query video. The first column shows the query frame, and the second, third, and fourth columns present the first, second, and third nearest neighbor frames, respectively. The pink mask highlights the object of interest, which is the focus of each frame's caption. Figure 1a and Figure 1b depict the retrieval results using our model's embeddings and ViCLIP's embeddings, respectively.