Appendix: Bandwidth-Adaptive Spatiotemporal Correspondence Identification for Collaborative Perception

Peng Gao[1], Williard Joshua Jose [2], Hao Zhang [2]

I. ALGORITHM

Algorithm 1: The proposed bandwidth-adaptive spatiotemporal CoID approach

Input : Ego spatiotemporal graph $\mathcal{G}$

Output : Identified correspondences $\mathbf{Y}$

1: Initialize the sharing set $\mathcal{S} = \{\}$ and receiving set $\mathcal{M} = \{\}$
2: **while** $Max\_Interaction\_Num$ **do**
3: Receive the query node embedding vectors $\{\mathbf{m}_i'\}^K$, and query graph embedding vector $\mathbf{z}'$ sent from the collaborator
4: Add $\{\mathbf{m}_i'\}^K$ to $\mathcal{M}$
5: Generate ego node embedding vectors $\{\mathbf{m}_i\}^n = \psi(\mathcal{G})$ using Eqs. (1-8)
6: Generate ego graph embedding vector $\mathbf{z}$ using Eqs. (9-10)
7: Compute matching scores $\mathbf{s}$ using Eq. (11-12)
8: Select candidates $\{\mathbf{m}_i\}^K = Top_K(\mathbf{s})$
9: Share ego node embedding vectors $\{\mathbf{m}_i\}^K$ and ego graph embedding vector $\mathbf{z}$ with collaborators
10: Add $\{\mathbf{m}_i\}^K$ in $\mathcal{S}$
11: **end**
12: Compute $\mathbf{Y}$ by performing correspondence identification between the sets $\mathcal{M}$ and $\mathcal{V} \in \mathcal{G}$ given Eqs. (13)
13: **return** $\mathbf{Y}$

The full algorithm is presented in Algorithm 1, which is able to interactively identify the same objects observed by connected vehicles while considering the communication bandwidth constraint. In Lines 3-4, the ego vehicle preserves the shared nodes $\{\mathbf{m}_i'\}^K$ in the receiving set $\mathcal{M}$ and receives the graph embedding vector $\mathbf{z}'$ sent from the collaborator. K denotes the communication bandwidth that can be different across time. Then, in Lines 5-7, the ego vehicle leverages its full observations $\mathcal{G}$ and the shared query to compute the matching scores to indicate the probabilities of ego nodes appearing in the collaborator's observations. In Lines 8-9, the ego vehicle selects top-K candidates to share them with the collaborators. In Line 10, the ego vehicle tracks the nodes to share given the sharing set $\mathcal{S}$ to avoid duplicated sharing. This process runs interactively between the ego-collaborator vehicles until the maximum interaction number is reached. In Line 12, we predict the correspondence between $\mathcal{G}$ and the receiving set $\mathcal{M}$.

[1]North Carolina State University, email: pgao5@ncsu.edu

[2]University of Massachusetts Amherst, email: {wjose, hao.zhang}@umass.edu

TABLE I

DESCRIPTION OF OUR CAD DATASET CONSTRUCTED BASED ON THE HIGH-FIDELITY CARLA AND SUMO SIMULATIONS.

# Instances	9,716 from 4 different scenarios
Training	6,801 data instances for training
Testing	2,914 data instances for testing
Sequence length	5 frames
Bandwidth constraint	Sharing at most 4 nodes in each communication iteration
Maximum interaction number	3 times
Sensor	Color, depth, and GNSS sensors
RGBD specifications	1920×1080 at 10 FPS

II. EXPERIMENTAL SETUP

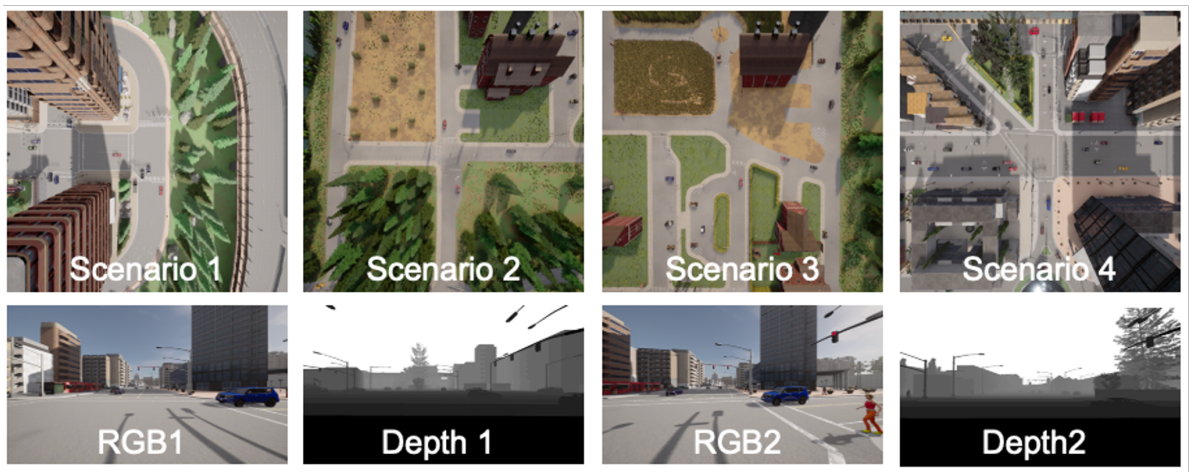

Fig. 1. The first row presents four intersection scenarios that are implemented in our high-fidelity CAD simulations used for experimental evaluation. The second row illustrates the simulated color and depth images obtained by two connected vehicles at the same intersection from different perspectives.

We have developed a high-fidelity connected autonomous driving (CAD) simulator by seamlessly integrating two open-source platforms: CARLA [1] and SUMO [2]. CARLA is a versatile autonomous driving simulator capable of simulating vehicle sensors, driving control, and diverse traffic scenarios. In our experimental setup, we designed four distinct traffic scenarios, with a particular emphasis on street intersections where connected vehicles frequently interact from various driving directions. These four simulated scenarios are visually depicted in Figure 1. Notably, Scenario 4 presents a significantly more challenging environment compared to the others, as it simulates a crowded scene for traffic interactions.

Within the simulations, every connected vehicle is equipped with a suite of sensors, including a front-facing RGB camera, a front-facing depth camera, and a global navigation satellite system (GNSS) sensor. Visual examples of the RGB and depth images captured by a pair of connected vehicles are showcased in Figure 1. The designed

GNSS sensor simulation adheres to the technical specifications of the real SBG Ellipse2-D sensor. The intricate orchestration of traffic patterns, encompassing both pedestrians and vehicles, is expertly managed by SUMO. These virtual entities exhibit behaviors generated randomly but in adherence to real-world traffic rules, such as halting at red lights and yielding to pedestrians, thereby enhancing the realism of the CAD simulator.

We collect a total of 9,716 data instances, of which 6,801 data instances are used for training and 2,914 data instances are used for testing. Each data instance includes a pair of RGBD images observed by two connected vehicles from different perspectives, the GNSS positions and orientations of vehicles, and the ground truth of object correspondences directly obtained from the instance-level segmentation provided by CARLA (each object segmentation has a unique ID). We set the maximum interaction number to 3 and the bandwidth constraint $K = 4$ through all our experiments. The details are presented in Table I

In our graph construction, we use YOLOv5 [3] to detect objects and we extract the ResNet50 feature [4] as each node's appearance feature. We use a sequence of 5 frames for each spatiotemporal graph generation and the detected objects are tracked by using SORT [5]. The positions of nodes are obtained from the depth images. The spatial edges are fully connected with the distance as the attribute. The temporal edges are connected via the object tracking results and temporal edge attributes are identified via the temporal tracking and timestamps.

Our network is implemented based on the PyTorch geometric library. In the spatiotemporal embedding network ψ, we set the number of layers to be 3 and set $\mathbf{W}_v \in \mathbb{R}^{1,000\times256}$, where $1,000$ is the length of ResNet features. We also set $\mathbf{W} \in \mathbb{R}^{256\times128}$ $\mathbf{W}_e \in \mathbb{R}^{1\times128}$ and $\mathbf{W}_b \in \mathbb{R}^{128\times128}$ separately. In addition, we set $\mathbf{q} \in \mathbb{R}^{128\times1}$ defined in Eq. (4) and the multi-head number is 4. In the graph pooling layer, we set the pooling ratio as 0.2 and take the mean operation as the final readout function to generate the graph-level embedding vector. We set the hyperparameters $\delta_p = 0.2$, $\delta_n = 1.4$ and $\gamma = 40$ as defined in the loss function Eq. (15). We also set $\lambda = 0.9$ as defined in Eq. (12). In all the experiments, we use ADAM [6] as the optimization method with the learning rate set to 0.0001. We run 50 epochs to train our approach.

REFERENCES

[1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on Robot Learning*, 2017.

[2] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (simulation of urban mobility)-an open-source traffic simulation," in *The 4th Middle East Symposium on Simulation and Modelling*, 2002.

[3] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, and M. T. Minh, "ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference," Feb. 2022.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *IEEE International Conference on Image Processing*, 2016.

[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv*, 2014.