Error Concealment of Dynamic LiDAR Point Clouds for Connected and Autonomous Vehicles

Guihua Shi (shigh1102@gmail.com)

Advisor: Cheng-Hsin Hsu

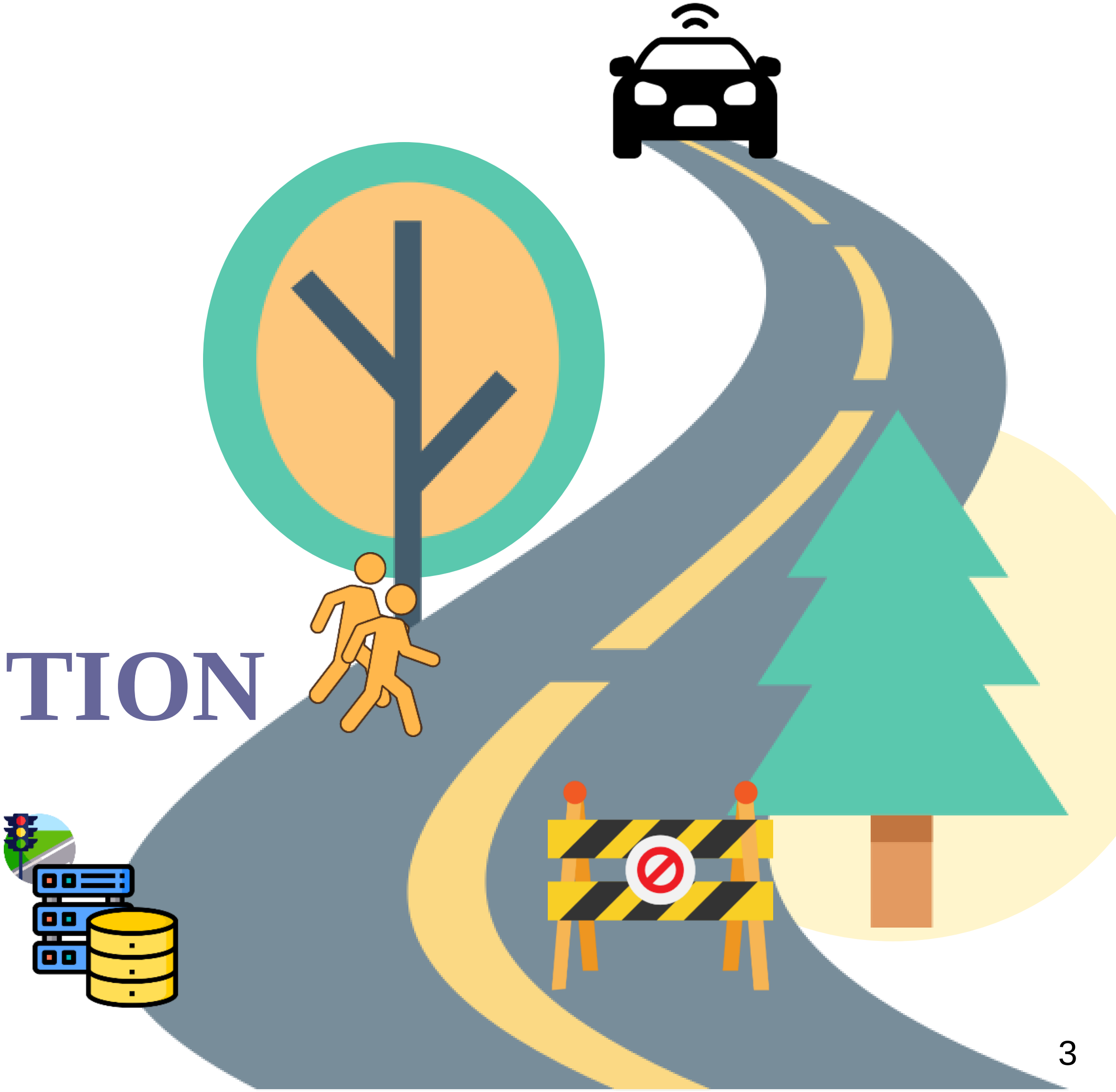
Networking and Multimedia Systems Lab,

ISA, National Tsing Hua University

Outline

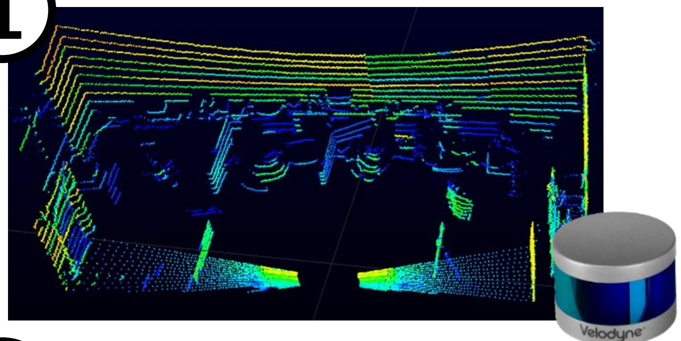
- Introduction
- Motivations
- Related Work
- Problem
- Solutions
- Experimental Setup
- Results
- Conclusion
- Future Work

INTRODUCTION

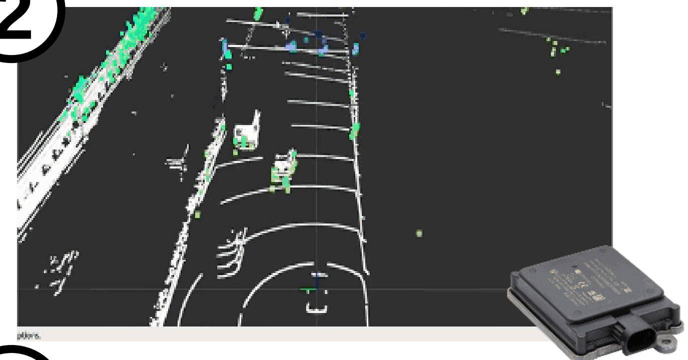


Sensors in CAVs

1

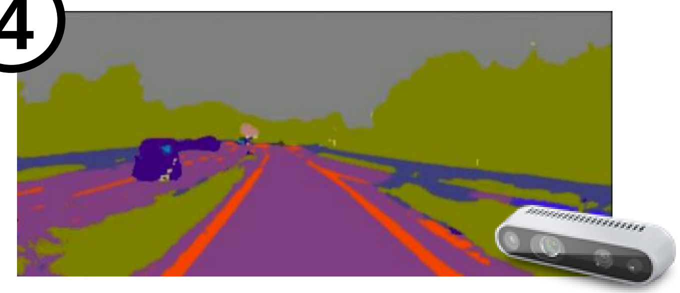


2



3

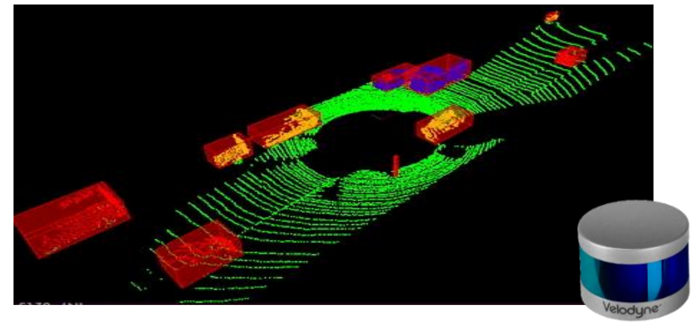
4



Sensor	Max. Distance	H-FoV	V-FoV	FPS
LiDAR	180 m	360°	26.8°	20 Hz
mmWave LiDAR	70 m	90°	30°	17 Hz
RGB Camera	250 m	90°	60°	15 Hz
RGBD Camera	10 m	58°	58°	90 Hz

Applications for Driving Automations

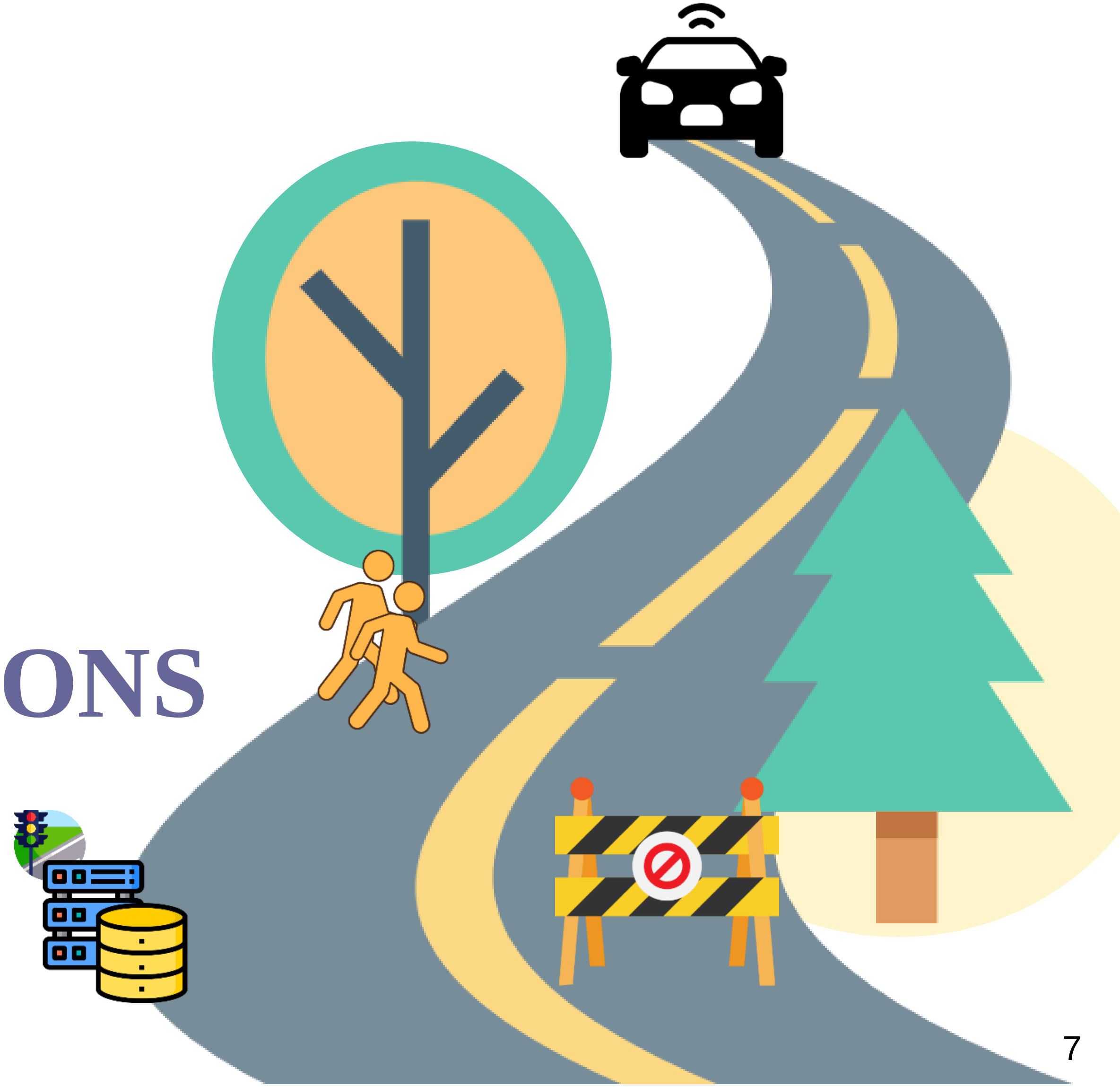
- Object Detection
  - Obstacle detection
  - Congestion analysis
- Semantic Segmentation
  - Refined analysis
  - Obstacle detection
- Lane Detection
  - Lane keeping
  - Route planning



Point Cloud Characteristics

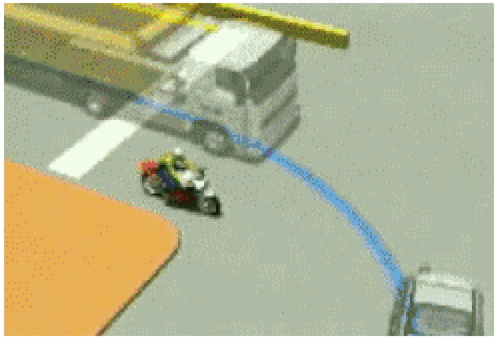
- **Dimension**: A set of points in 3D space, and each point has three coordinates, which are high-dimensional data
- **Unordered**: The points are not in order, and modifying the order will not affect the result
- **Interaction between points**: A single point is meaningless, and the features need to consider its structure and context
- **Invariance under transformations**: For points in the point cloud, their absolute position does not matter, and the overall rotation, transformation, and scaling does not modify the structure

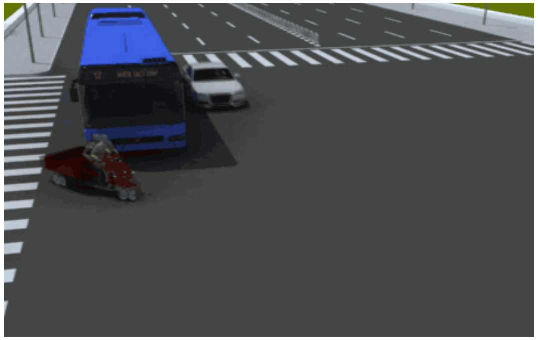
MOTIVATIONS



Cooperative Perception

- The field of view from the single vehicle is always limited:
  - Blind spot
  - Obstacle occlusion
- CAVs can obtain additional information by data sharing:
  - Sensor data [1]
  - Features [2]
  - High-level results [3]

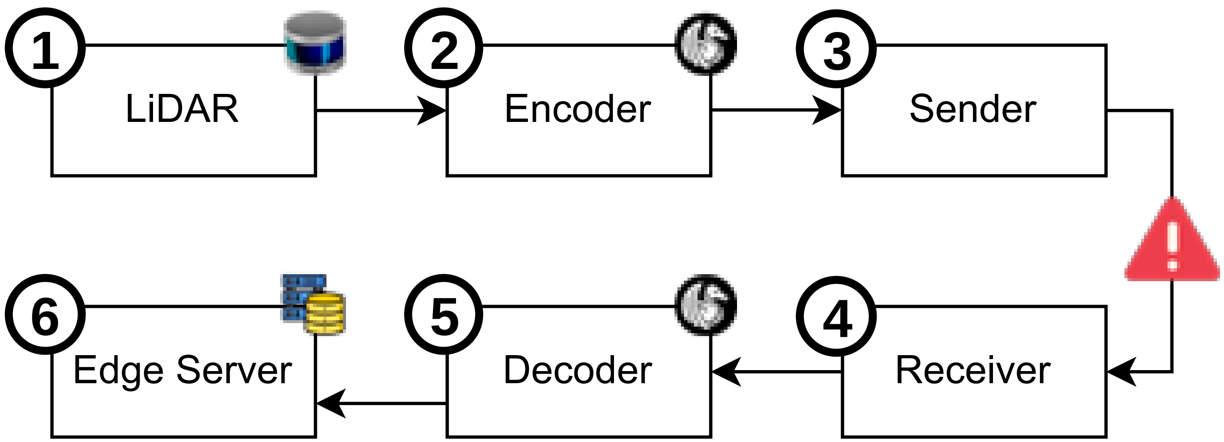




[1] Zhang X, Zhang A, Sun J, et al. Emp: Edge-assisted multi-vehicle perception[C]//Proceedings of the 27th Annual International Conference on Mobile Computing and Networking. 2021: 545-558.
[2] Chen Q, Ma X, Tang S, et al. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds[C]//Proceedings of the 4th ACM/IEEE Symposium on Edge Computing. 2019: 88-100.
[3] Arnold E, Dianati M, de Temple R, et al. Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(3): 1852-1864.

Point Cloud more than 100 Mbps

- **More than 1 million points per second [1]**
- **Streaming uncompressed dynamic point cloud dictates more than 100 Mbps**
- **Difficult to support multiple vehicles [2]**



[1] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: The kitti dataset[J]. The International Journal of Robotics Research, 2013, 32(11): 1231-1237.
[2] Zhang X, Zhang A, Sun J, et al. Emp: Edge-assisted multi-vehicle perception[C]//Proceedings of the 27th Annual International Conference on Mobile Computing and Networking. 2021: 545-558.

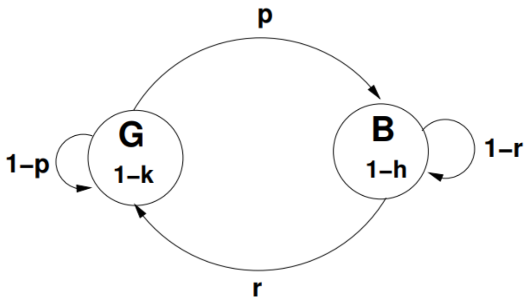
Internet of Vehicles Limitations

- C-V2X (Cellular Vehicle-to-Everything)
  - LTE (4G)
  - **NR (5G)**
- DSRC (Dedicated Short-Range Communication)
  - **5.9 GHz (IEEE 802.11p)**
  - 60 GHz (IEEE 802.11ad)

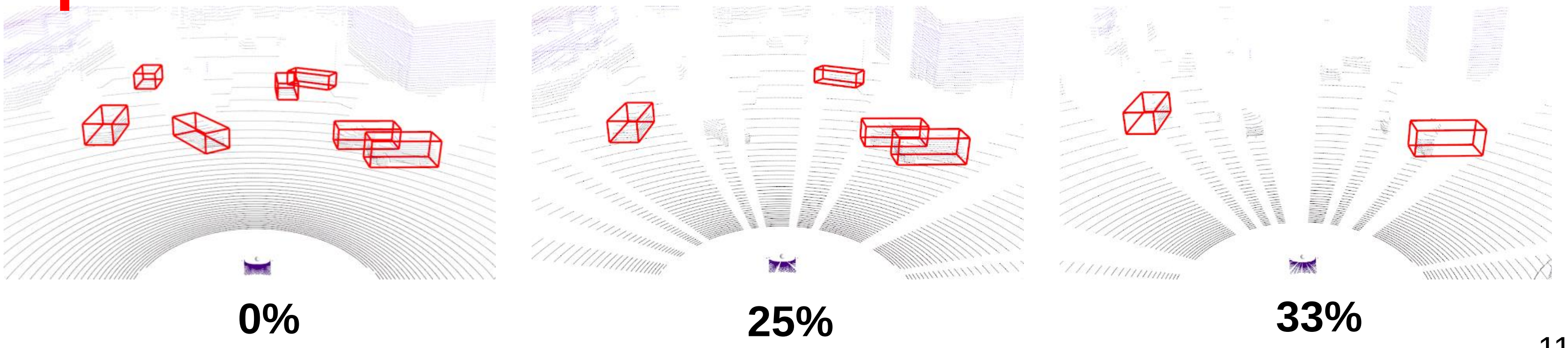
Network	Data Rate	Bandwidth	Latency	Max. Distance
LTE-V2X	<50 Mbps	10 MHz	50 ms	250 m
NR-V2X	<1 Gbps	100 MHz	5 ms	500 m
5.9 GHz	<27 Mbps	10 MHz	150 ms	250 m
60 GHz	<7 Gbps	2 GHz	10 ms	150 m

Packet Loss may lead to misclassification

- Packet loss will generate incomplete point cloud frame
- Incomplete frames may will produce misclassification



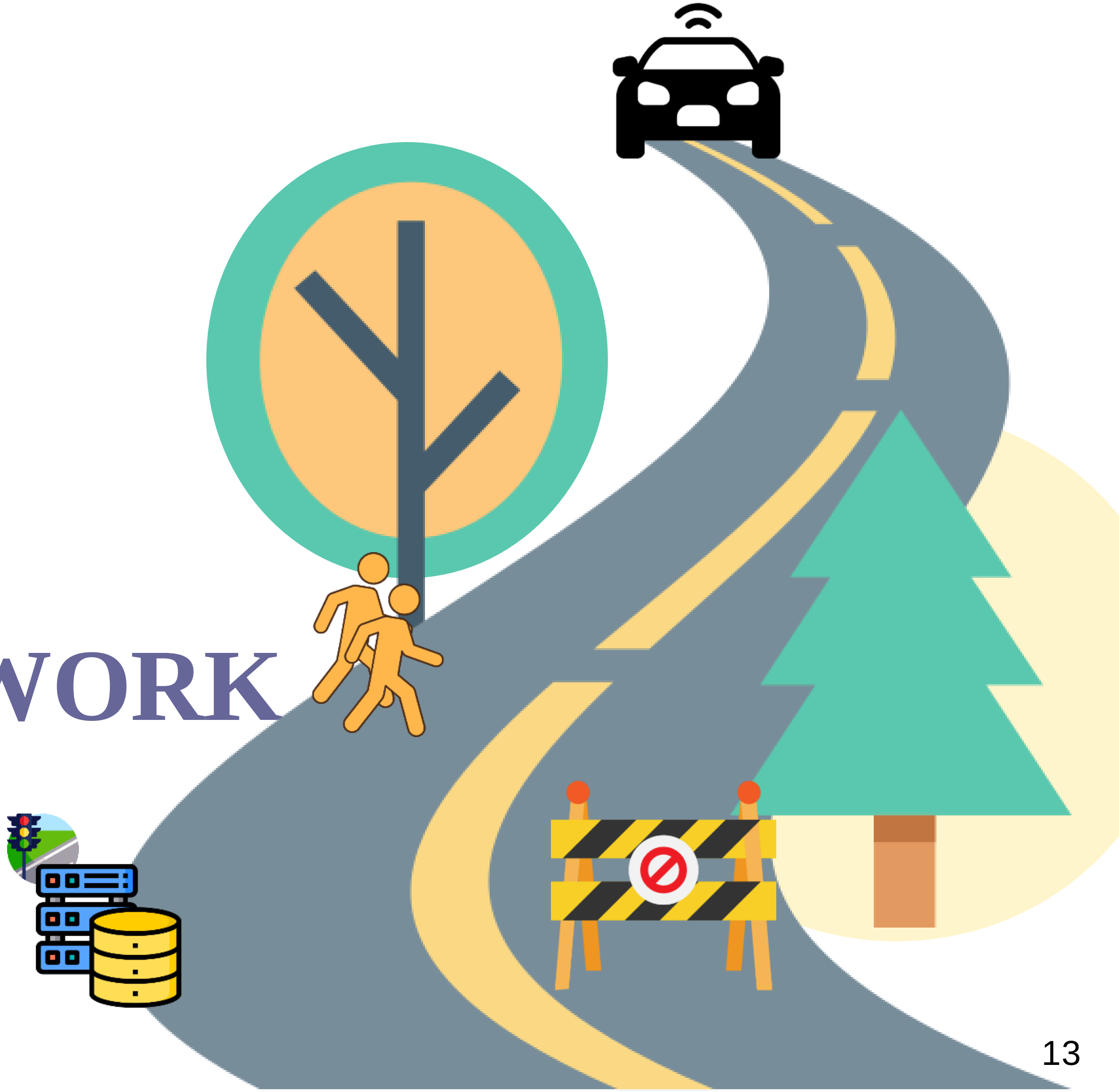
- **Only 71.43% and 42.85% of vehicles are detected when one-fourth and one-third of packets are lost**



Goals and Challenges

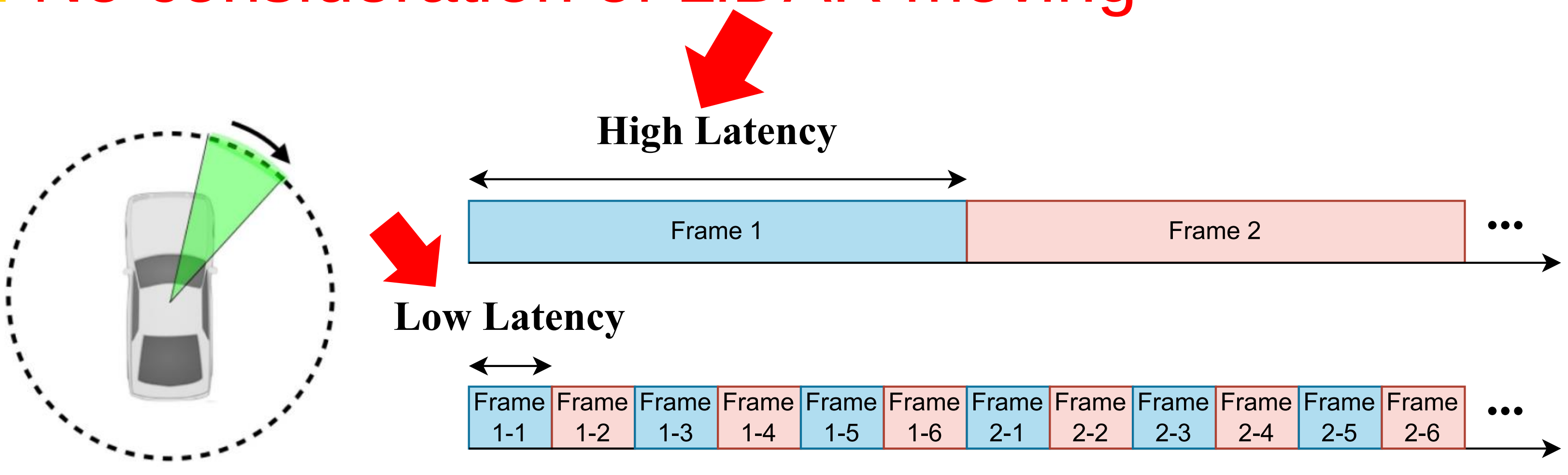
- Goals
  - Minimize the Chamfer distance between the concealed and original point cloud frames
- Challenges
  - Vehicles are moving, which complicates the transformation
  - Some incomplete frames may contain too many lost sectors, Spatial Interpolation (SI) less effective
  - Incomplete frame degrades the performance of such interpolation

RELATED WORK



Point Cloud Caching

- Eliminates high latency due to full-scans
  - Point clouds copy [3]
  - Iterative closest point (ICP) [4]
- No consideration of LiDAR moving



[3] Han W, Zhang Z, Caine B, et al. Streaming object detection for 3-d point clouds[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII. Cham: Springer International Publishing, 2020: 423-441.
[4] Qu C, Shivakumar S S, Liu W, et al. Llol: Low-latency odometry for spinning lidars[C]//2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022: 4149-4155.

Point Cloud Completion

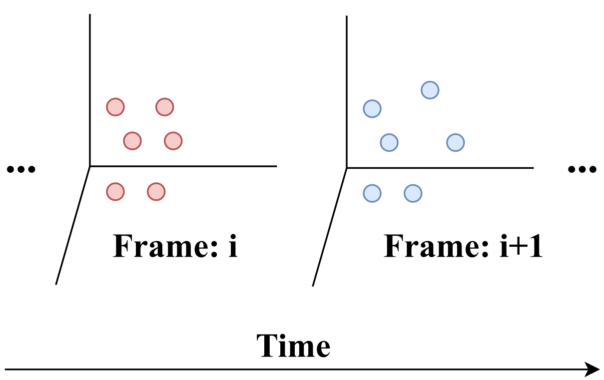
- Focus on upsampling sparse point clouds
  - Estimate the complete geometry of objects and scenes
  - Mostly by deep learning
- No consideration of communication loss
- Rely on semantic labels for object extract

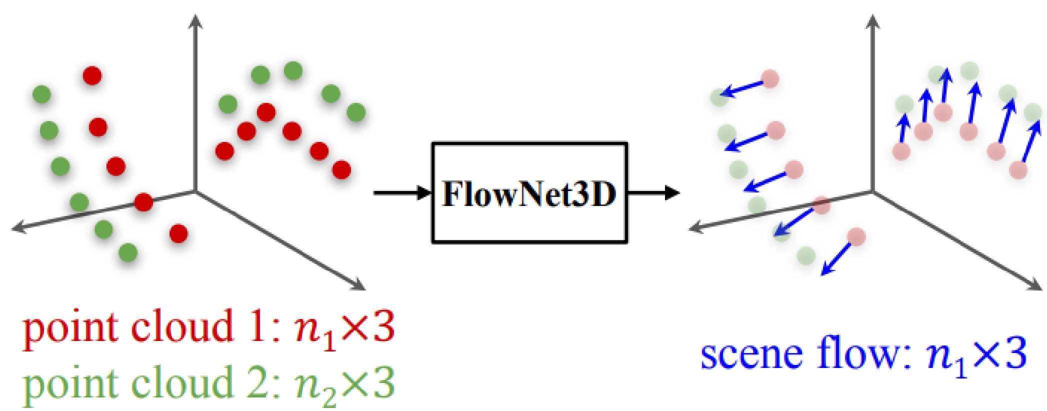
PoinTr [5]

[5] Yu X, Rao Y, Wang Z, et al. Pointr: Diverse point cloud completion with geometry-aware transformers[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 12498-12507.

Point Cloud Interpolation

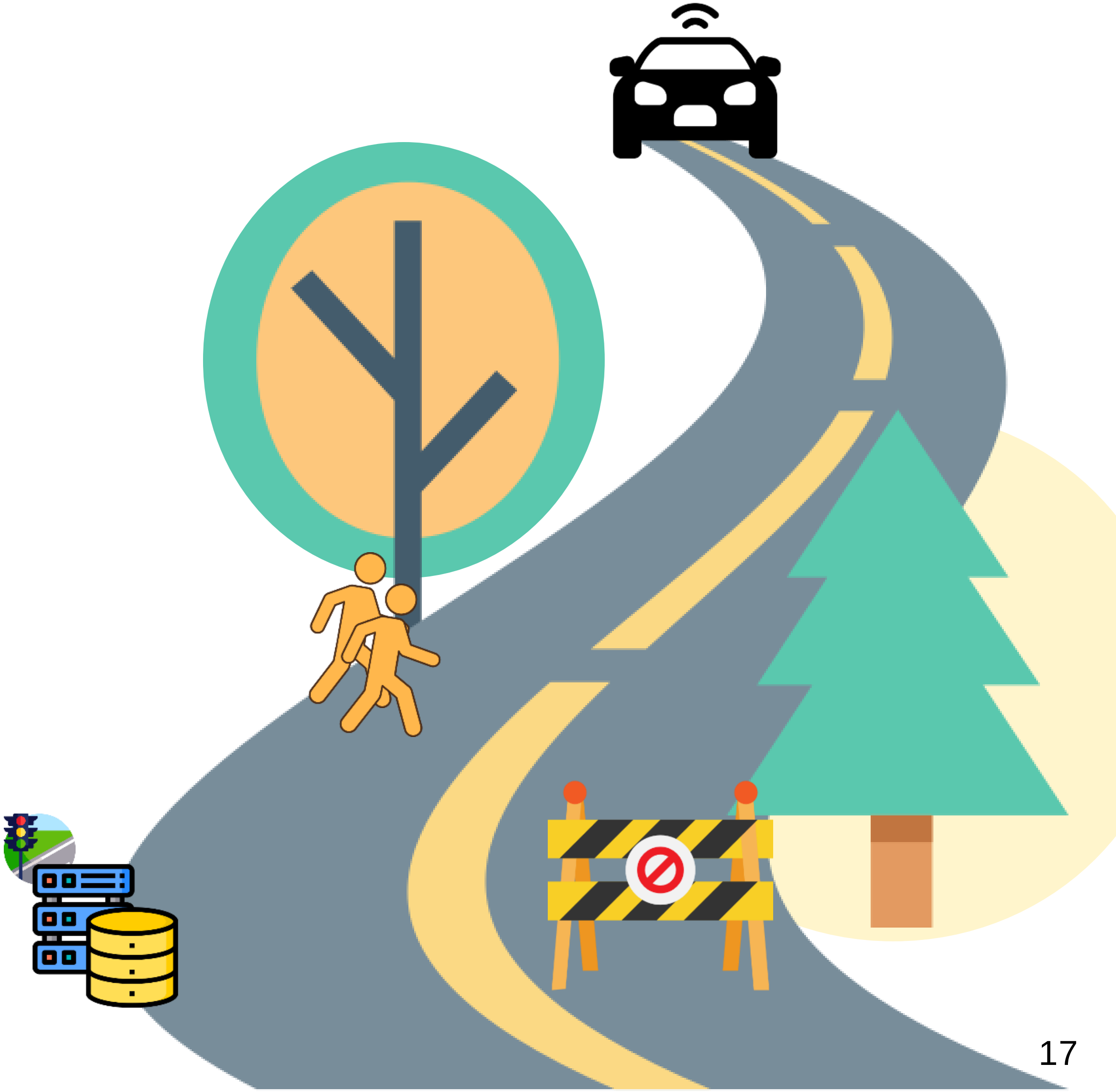
- Use consecutive frames to generate intermediate frame to improve frame rate
  - Nearest-point query (KD-tree)
  - Mid-point prediction
  - Scene flow estimation [5]
- Rely on complete frame



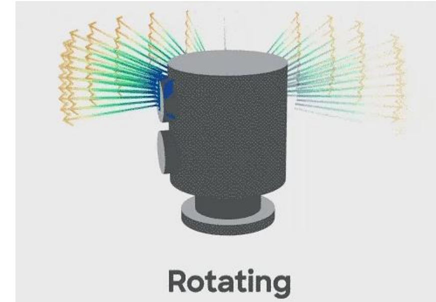


[5] Liu X, Qi C R, Guibas L J. Flownet3d: Learning scene flow in 3d point clouds[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 529-537.

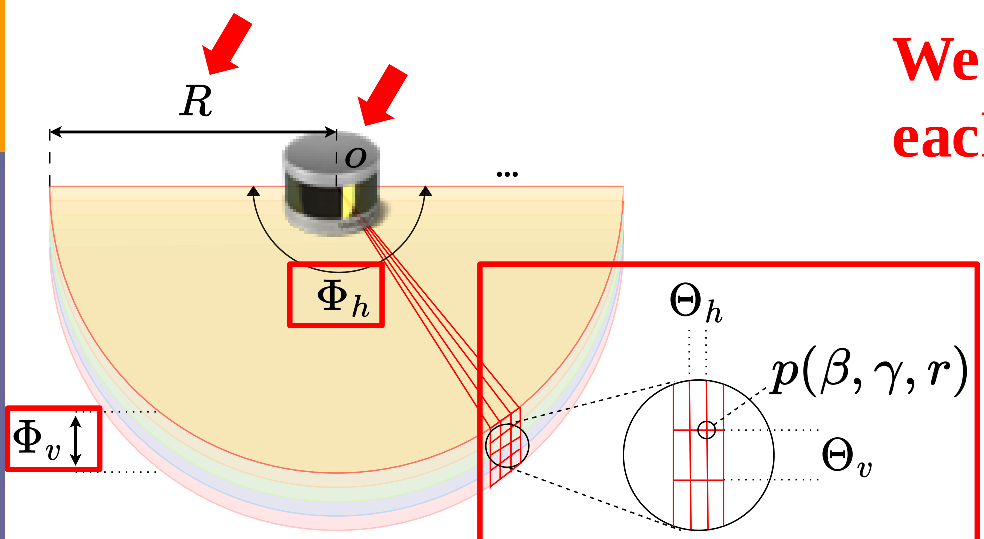
PROBLEM



Measurements of LiDAR



- LiDAR samples evenly in horizontal and vertical directions
- Calculate distance by the response time of reflected laser
- We can transform $p.\beta$, $p.\gamma$, and $p.r$ into $p.x$, $p.y$, and $p.z$, mutually



We can know the orientation of each point before scanning!

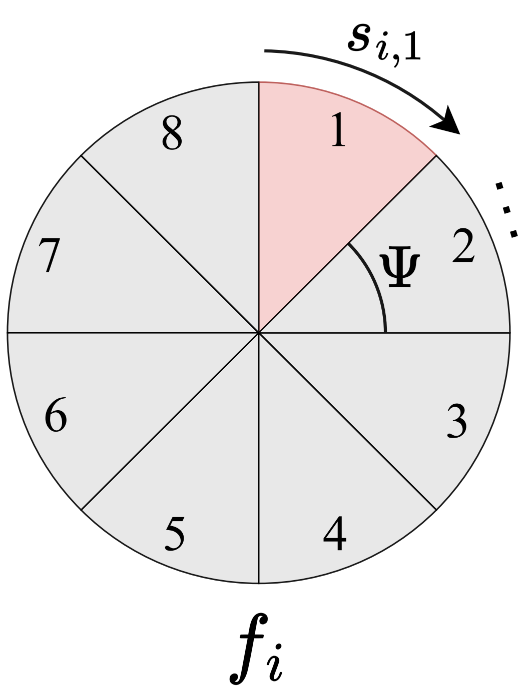
$$p.x = p.r \times cos(p.\beta) \times cos(p.\gamma); \quad (1a)$$

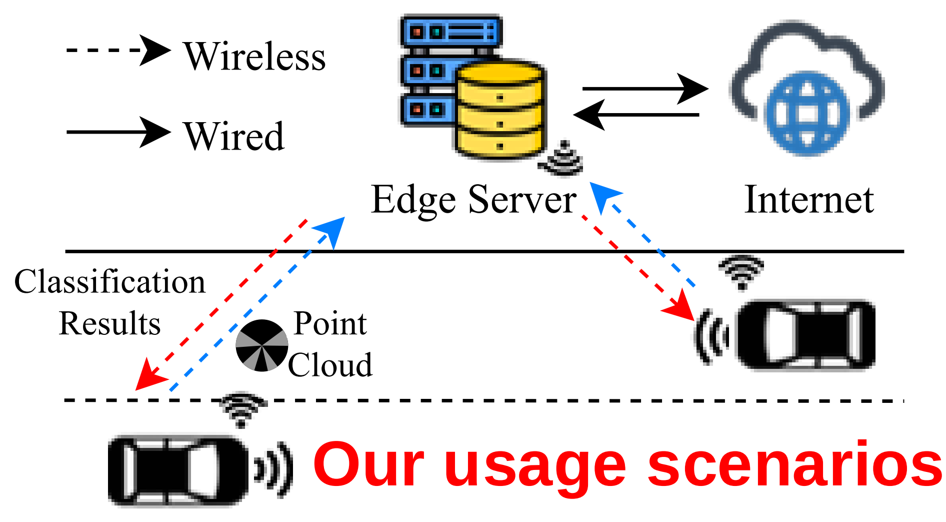
$$p.y = p.r \times cos(p.\beta) \times sin(p.\gamma); \quad (1b)$$

$$p.z = p.r \times sin(p.\beta), \quad (1c)$$

Problem Statement

- A complete frame f_i is split into multiple equal-size sectors $s_{i,j}$ for transmission
- Each sector is encapsulated in one packets before being streamed
- **Packet loss will cause sector loss**

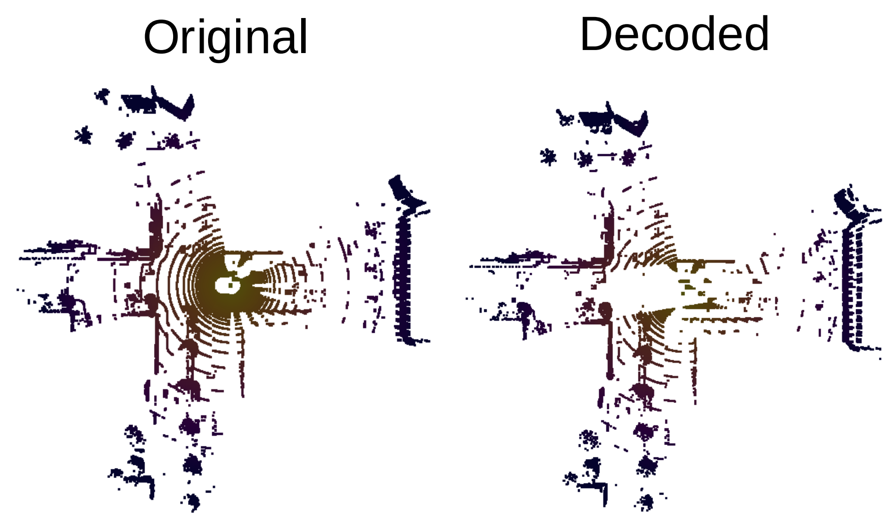


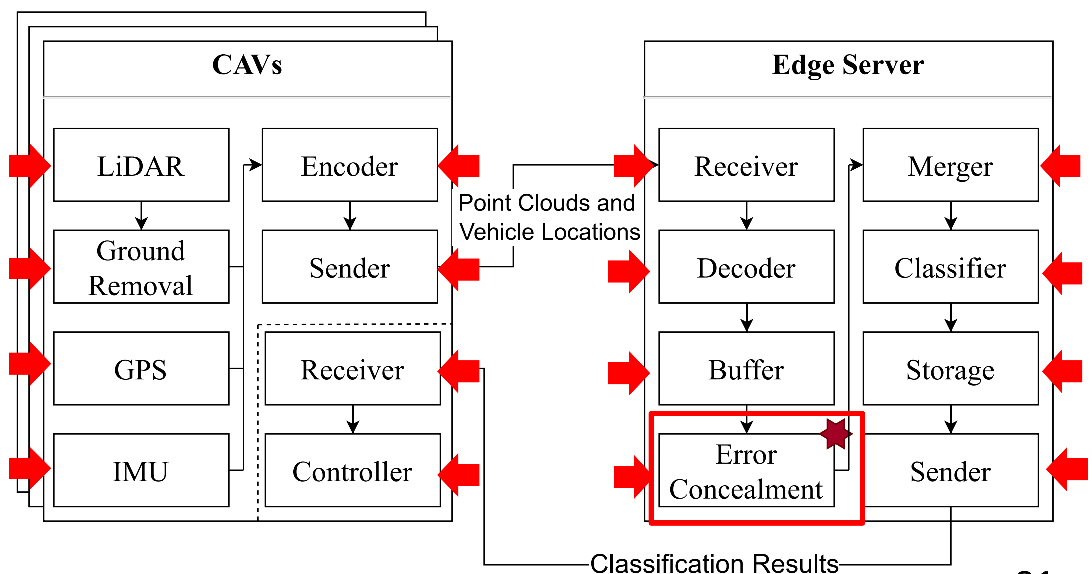


SOLUTIONS

System Overview

- Ground removal
  - RANSAC (Random Sample Consensus) [6]
- Encoder/Decoder
  - Draco [7]
- Classifier
  - Object detection [8]

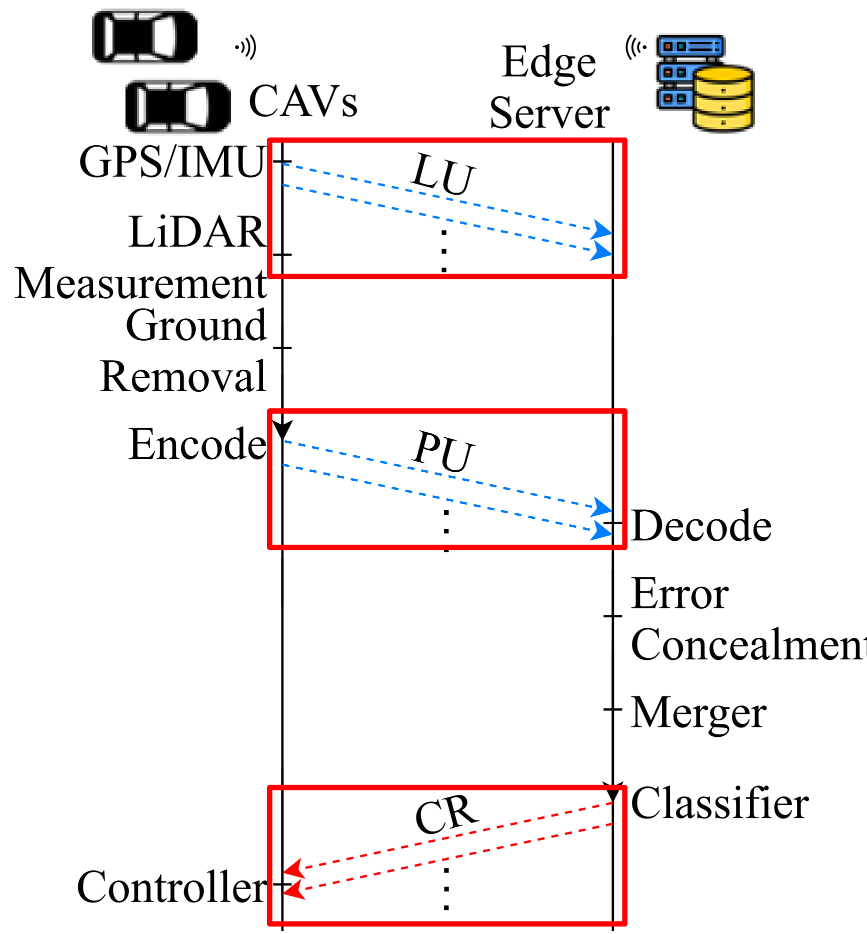




[6] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6): 381-395.
[7] Google. Draco (3D DATA COMPRESSION), 2023. https://github.com/google/draco
[8] Shi S, Wang X, Li H. Pointrcnn: 3d object proposal generation and detection from point cloud[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 770-779.

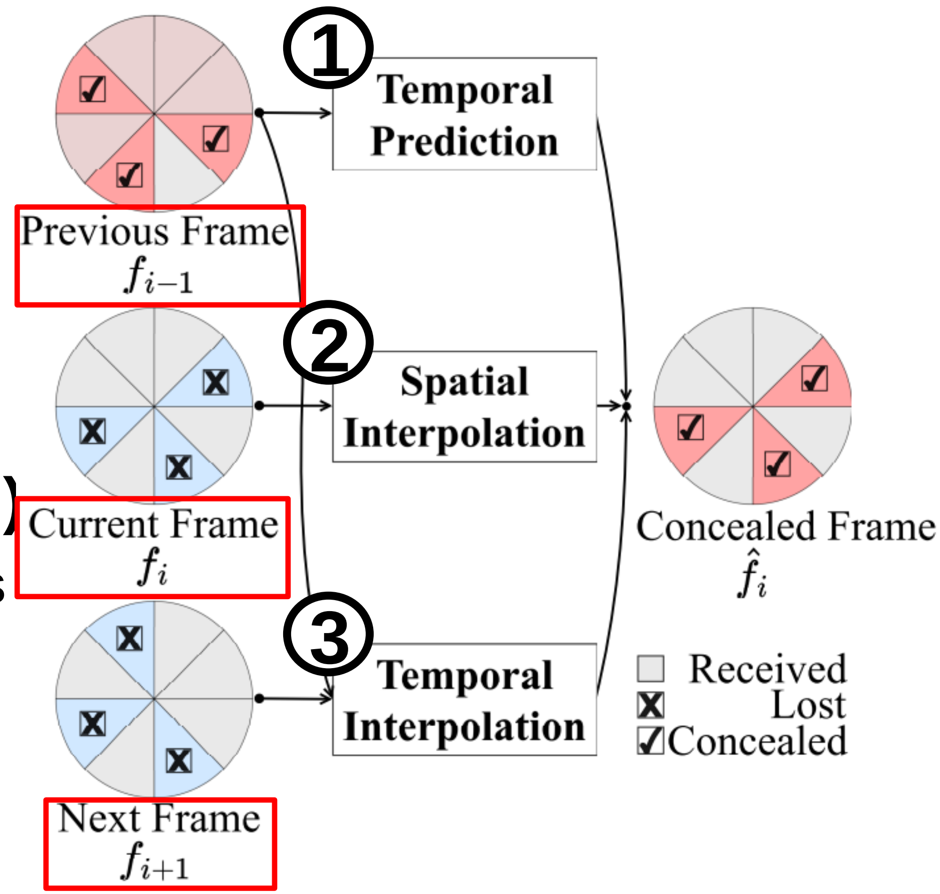
Message Types

- **Location Update (LU):** reports the latest LiDAR center location from GPS and IMU
- **Point Update (PU):** contains the point clouds of a sector, which is sent once the points in that sector are encoded
- **Classification Result (CR):** contains the classification outcomes produced at the edge server



Error Concealment Approaches

- **Temporal Prediction (TP)**
  - Use previous frame
  - Least latency
- **Spatial Interpolation (SI)**
  - Use incomplete current frame
  - Highest applicability
- **Temporal Interpolation (TI)**
  - Use previous and next frames
  - Richest information



Previous frame always complete!

Temporal Prediction (TP)

- Selectively copies points from sectors of previous frame to conceal the lost sectors of current frame.
- Copyover Prediction (CP):
  - Let $\hat{s}_{i,j} = s_{i-1,j}$, for any lost sector $s_{i,j}$
- Motion-compensated Prediction (MP):
  - Consider the location/orientation difference between LiDARs
  - Let M_i be the transformation matrix from f_{i-1} to f_{i+1}
  - Let $\hat{s}_{i,j} = s_{i-1}M_i$, for any lost sector $s_{i,j}$

Spatial Interpolation (SI)

- Employs the points in the current frame to estimate the measured distance $p.r$ for every given pitch $p.\beta$ and yaw $p.\gamma$
- Nearest Neighbor (NN):
  - Find the closest point p^* from all received sectors for each point of lost sector $s_{i,j}$
  - Let $\mathrm{p}.r$ = $p^*.r$
- Least Square (LS):
  - Fit all received points in frame f_i to $p.r = w_1 p.\beta + w_2 p.\gamma + w_3$
  - Use this equation to estimate $p.r$ for all points in lost sectors $s_{i,j}$

Temporal Interpolation (TI) (1/2)

- Analyze frames f_{i-1} and f_{i+1} to locate the closest point, using each pair of points to conceal the lost sectors
- Point Matching (PM):
  - Find the closest point in f_{i+1} for each point in f_{i-1}. Each pair of points is used to estimate a point in the concealed frame f_i
- Iterative Closest Point (ICP):
  - Compute a transform matrix from f_{i-1} to f_{i+1}, denoted as M'_i
  - Let M''_i be the transformation matrix that shifts/rotates half of the displacement/angles of M'_i
  - Any lost sector $s_{i,j}$ can be concealed by $\hat{s}_{i,j} = s_{i-1,j} M''_i$

Temporal Interpolation (TI) (2/2)

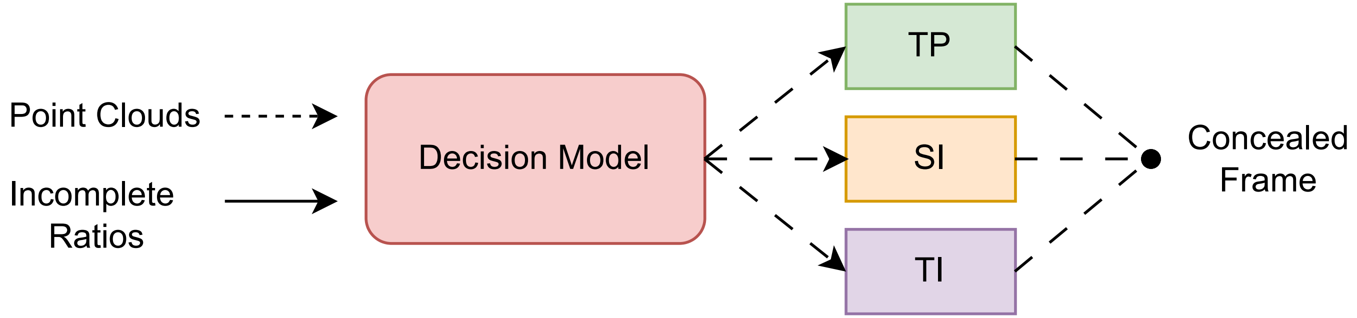
- Scene Flow (SF):
  - Use FlowNet3D [9] to compute scene flows from f_{i-1} to f_{i+1}, denoted as M'_i
  - Let M''_i be the transformation matrix that shifts/rotates half of the displacement/angles of M'_i
  - Any lost sector $s_{i,j}$ can be concealed by $\hat{s}_{i,j} = s_{i-1,j}M''_i$
- Bidirectional Scene Flow (BSF):
  - Use PointINet [10] computes scene flows from f_{i-1} to f_{i+1} and f_{i+1} to f_{i-1}
  - Fuses the two temporally interpolated frames

[9] Liu X, Qi C R, Guibas L J. Flownet3d: Learning scene flow in 3d point clouds[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 529-537.
[10] Lu F, Chen G, Qu S, et al. Pointinet: Point cloud frame interpolation network[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(3): 2251-2259.

LiDAR Error Concealment (LEC)

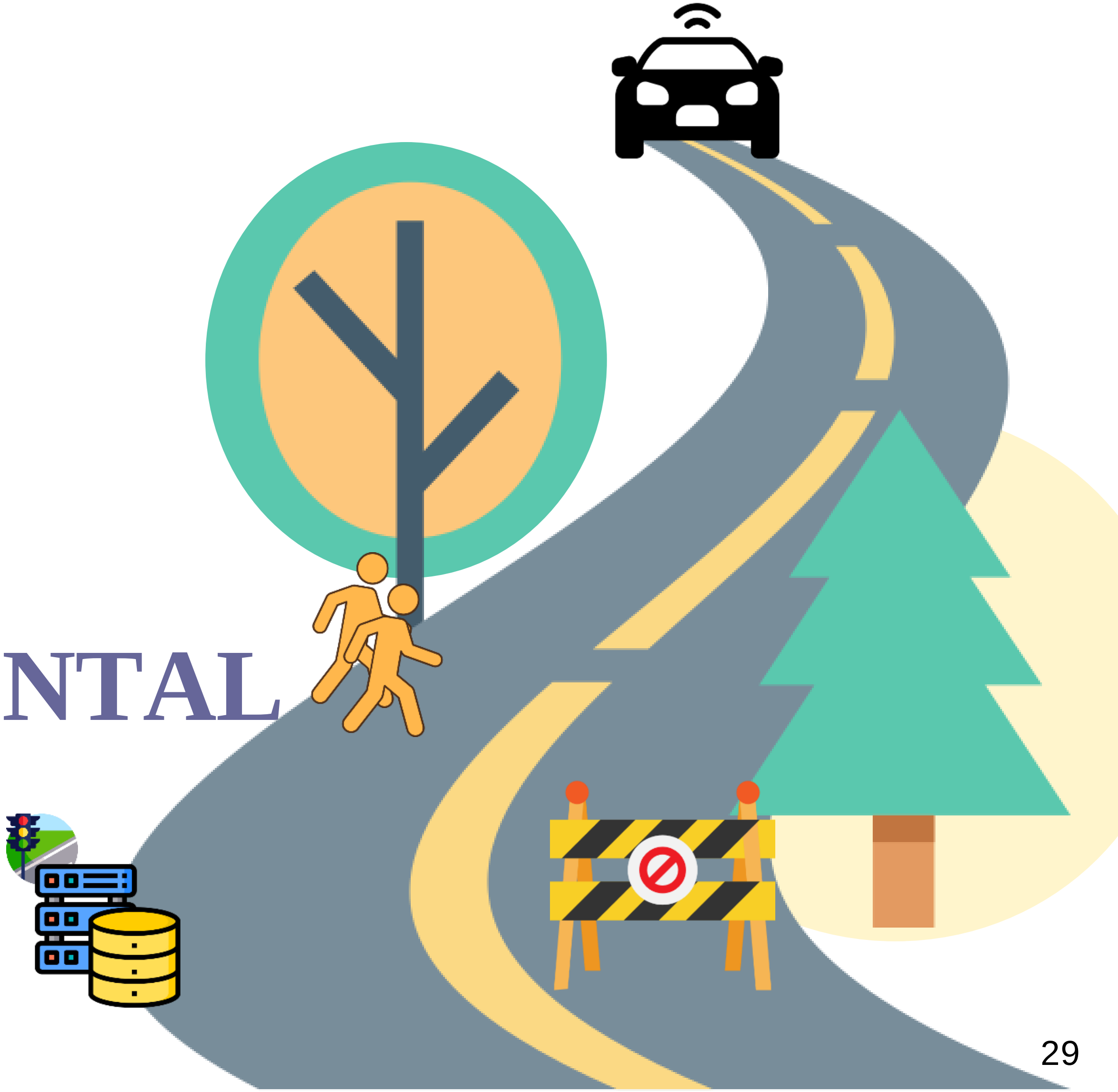
- Adaptively apply one of the three concealment approaches (TP, SI, TI) by ***incomplete ratios***
- Determining the cut-off thresholds is no easy task
- Using ML algorithms
  - Decision Tree (DT)
  - Support Vector Machine (SVM)
  - Random Forest (RF)

Using machine learning algorithms as our decision model

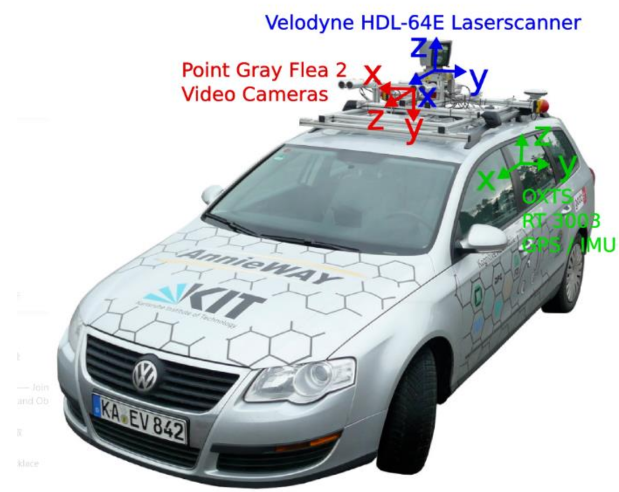


Incomplete Ratios: packet loss rates of previous, current, and next frames

EXPERIMENTAL SETUP



Pre-recorded dataset

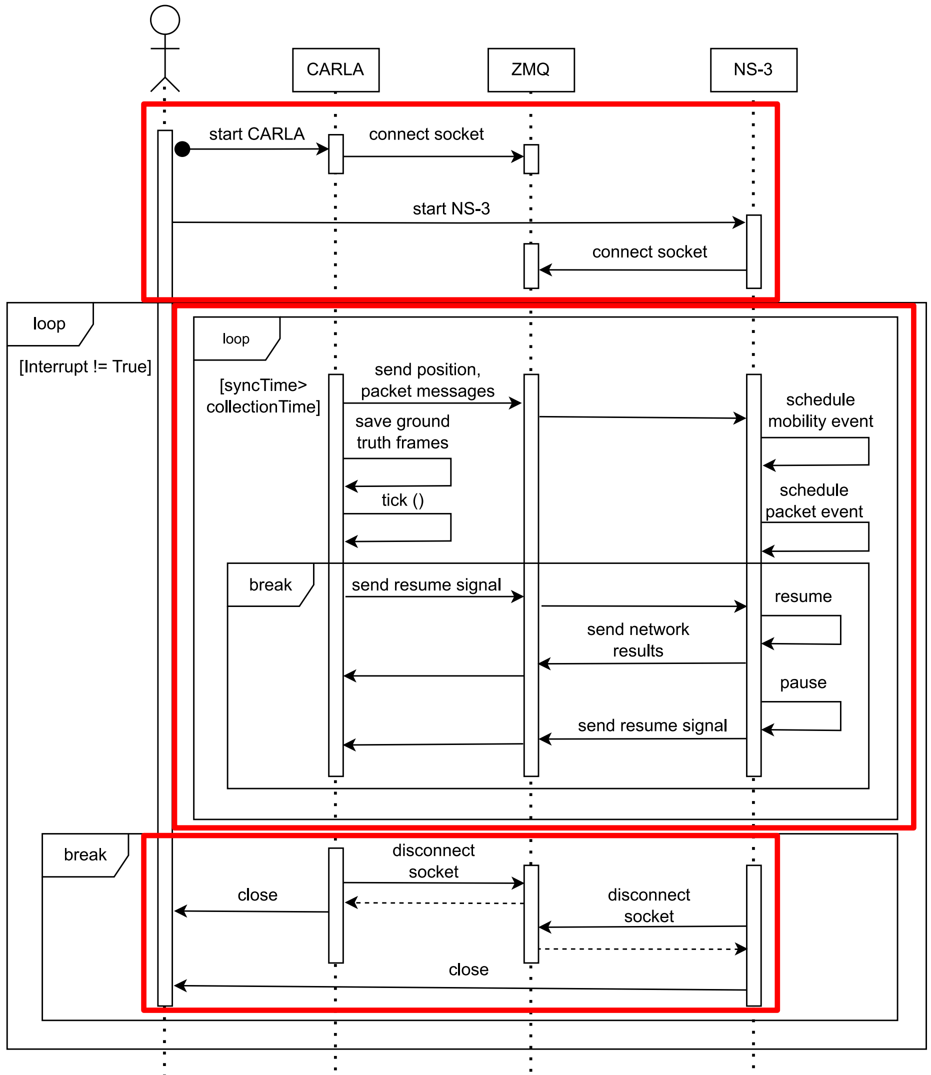


- KITTI Odometry Dataset
  - Captured in real life
  - Only one LiDAR-equipped for each sequence
  - Not propose object detection labels
  - Can not reflect interactions among nearby vehicles
- For 3-vehicles evaluation:
  - Duplicate trajectory in sequence 0 three times
  - Time-shift it by 10 and 20 seconds to create a 3-vehicle dataset
  - Use 4071 frames from sequence 8 for LEC model training

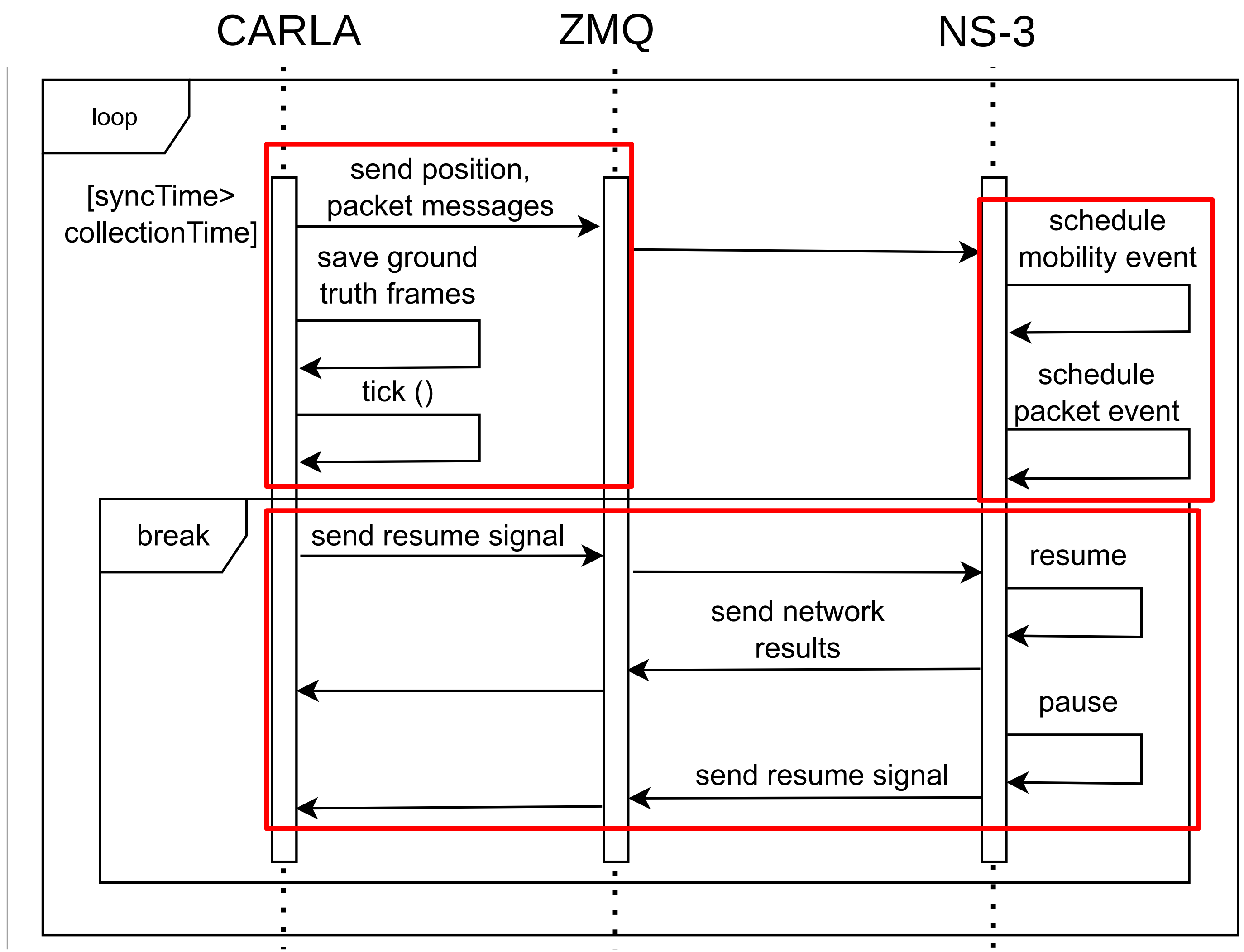
Co-Simulator

- We designed and implemented a co-simulator to evaluate our error concealment algorithms:
  - CARLA
  - NS-3
  - ZeroMQ
- Our co-simulator support:
  - Real-time KITTI-compatible and Semantic3D-compatible ground truth frames
  - V2V, V2X, V2I, I2V, etc. communication modes
  - Multiple network protocol, 5.9GHz DSRC, and NR C-V2X

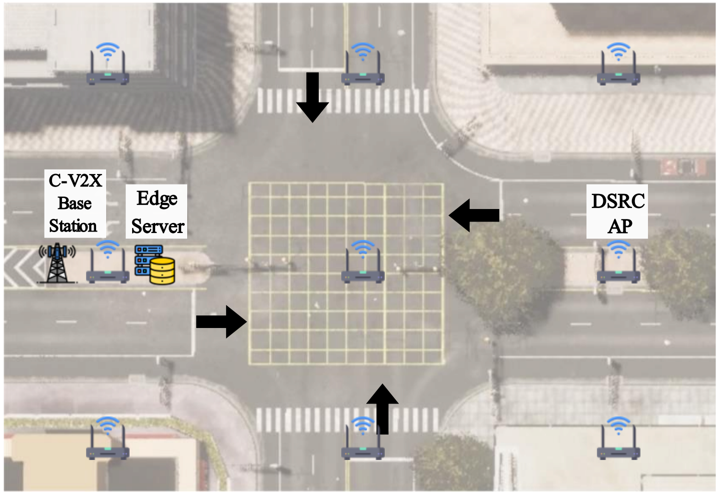
Workflow of Co-Simulator (1/2)



Workflow of Co-Simulator (2/2)



Experimental Setup



- Datasets:
  - Co-simulator
  - KITTI Odometry (real-life)

- Networks:
  - NR C-V2X:
    - Station next to the edge server
  - DSRC:
    - APs separated by 20 m

We repeat each simulation 10 times and report the average results from a random vehicle.

- Vehicles:
  - {1, 3, 5, 7}
  - Velodyne HDL-64E S2 (Ψ=2°)

- Benchmark:
  - Optimal (OPT): selects the smallest Chamfer distance among all TP, SI, and TI algorithms

Performance Metrics

- Low-Level **The lower the better**
  - Chamfer distance (m): The average shortest distance between the points in the target and ground truth frame
  - Hausdorff distance (m): The maximal shortest distance
- High-Level **The higher the better**
  - Intersection-over-Union (%): We use the pre-trained PointRCNN [9] to detect vehicles in front of CAVs
  - Detection Accuracy (%): The fraction of detected vehicles
- Running Time (s) **The lower the better**

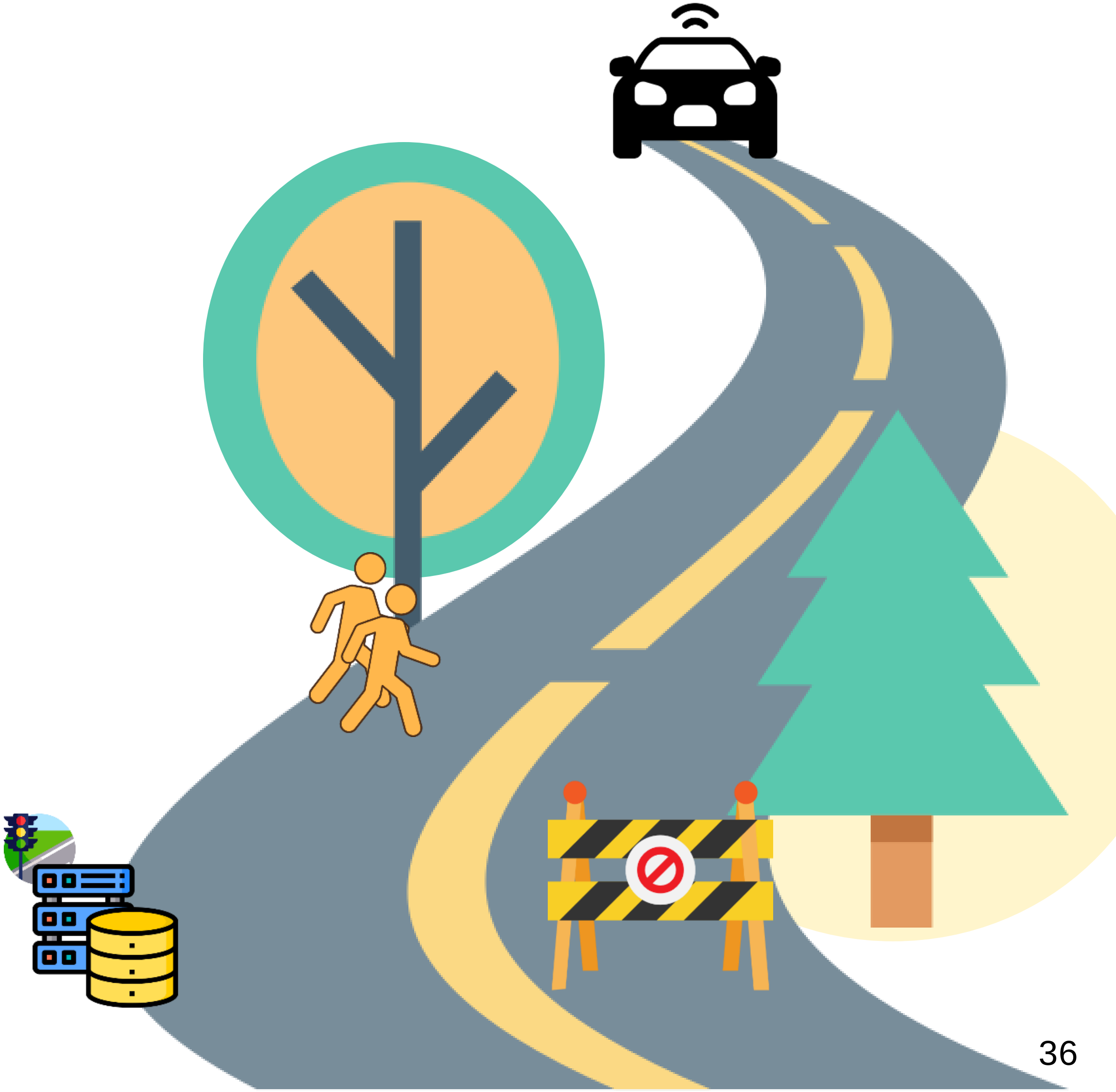
$$d^C(\hat{f}_i, f_i) = \frac{\sum_{p \in \hat{f}_i} \min_{p' \in f_i} \|p - p'\|_2^2}{\hat{n}_i} + \frac{\sum_{p \in f_i} \min_{p' \in \hat{f}_i} \|p - p'\|_2^2}{n_i}$$

$$d^H(\hat{f}_i, f_i) = \max \left\{ \sup_{p \in \hat{f}_i} \inf_{p' \in f_i} \|p - p'\|_2^2, \sup_{p \in f_i} \inf_{p' \in \hat{f}_i} \|p - p'\|_2^2 \right\}$$

We include 95% confidence intervals whenever possible.

[9] Shi S, Wang X, Li H. Pointrcnn: 3d object proposal generation and detection from point cloud[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 770-779.

RESULTS



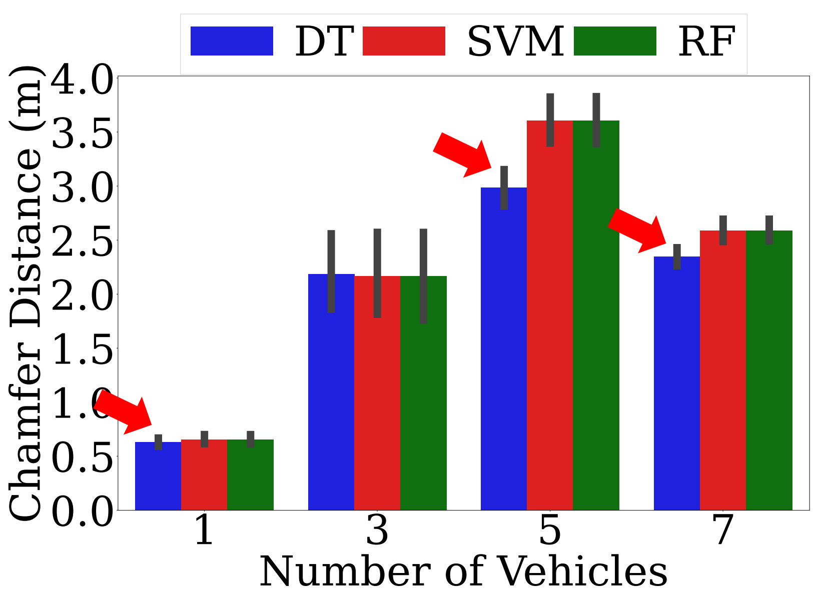
Training of LEC algorithm (1/2)

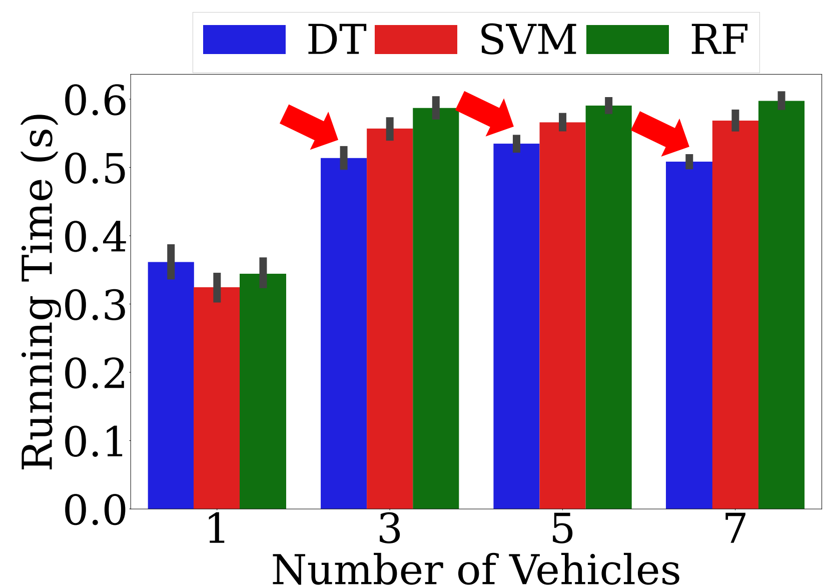
- Use our co-simulator to generate 10,000 consecutive frames and randomly simulate packet loss rates between 0% - 100% for each frame
- Carry out 5-fold cross-validation to find the best hyperparameters
- 80% of the frames are used for training and 20% for validation
- Decision Tree (DT)
  - maximal tree depth: {1, <u>5</u>, 10, 15, 20}
- Support Vector Machine (SVM)
  - Kernel: {linear, poly, <u>rbf</u>, sigmoid, precomputed}
  - Regularization: {1, 5, <u>10</u>, 25, 20}
- Random Forest (RF)
  - maximal tree depth: {1, 5, <u>10</u>, 25, 20}
  - number of trees: {10, 50, <u>100</u>, 150, 200}

	DT	SVM	RF
Training Acc. (%) $\uparrow$	83.12	80.52	84.09
Validation Acc. (%) $\uparrow$	76.14	79.21	78.34

Training of LEC algorithm (2/2)

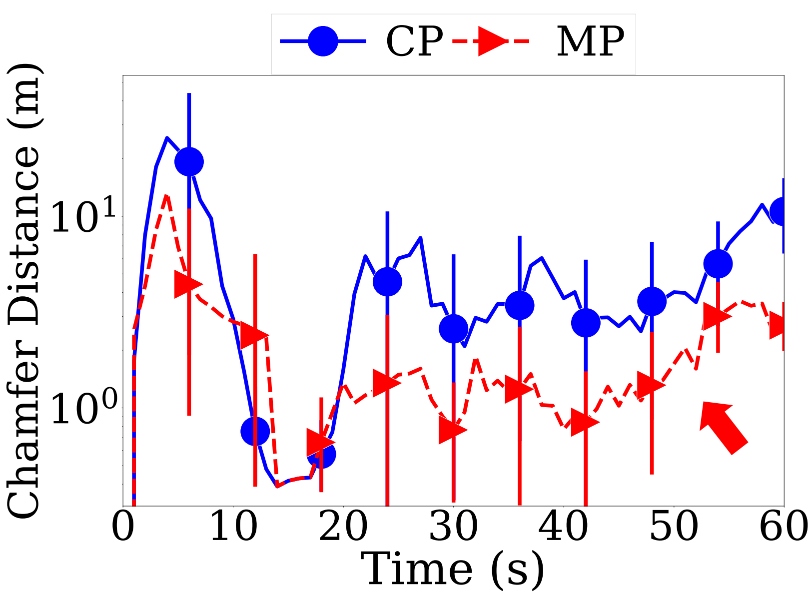
- DT outperforms both SVM and RF by up to 17.17%, and can save 10.53% and 15.00% running time
- Adopts DT as the decision model

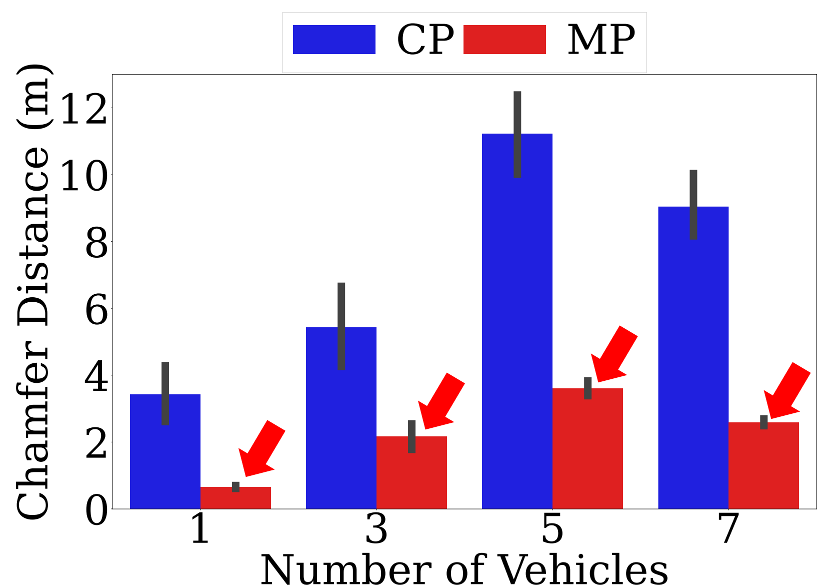




Design decision of concealment approaches (1/3)

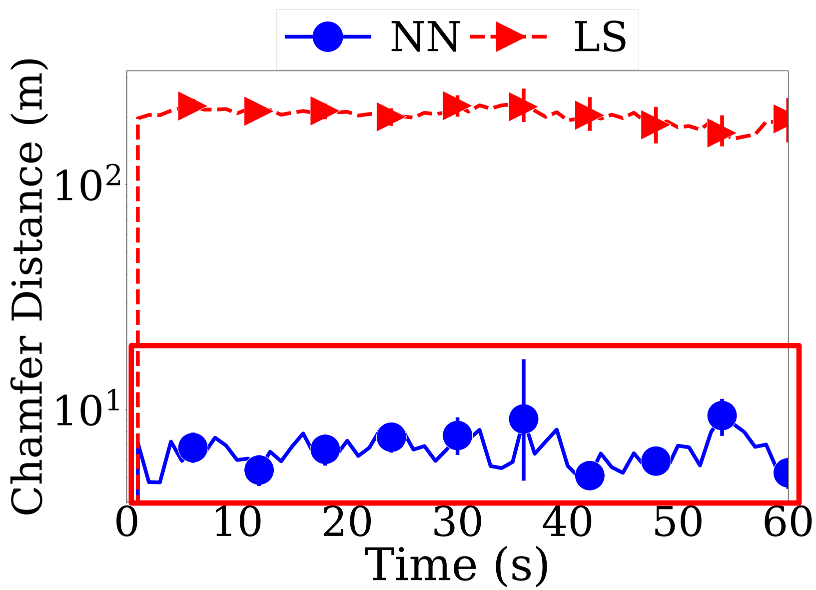
- CP reduce the Chamfer distance by up to 73.28% compared to MP
- We recommend using MP in TP approach

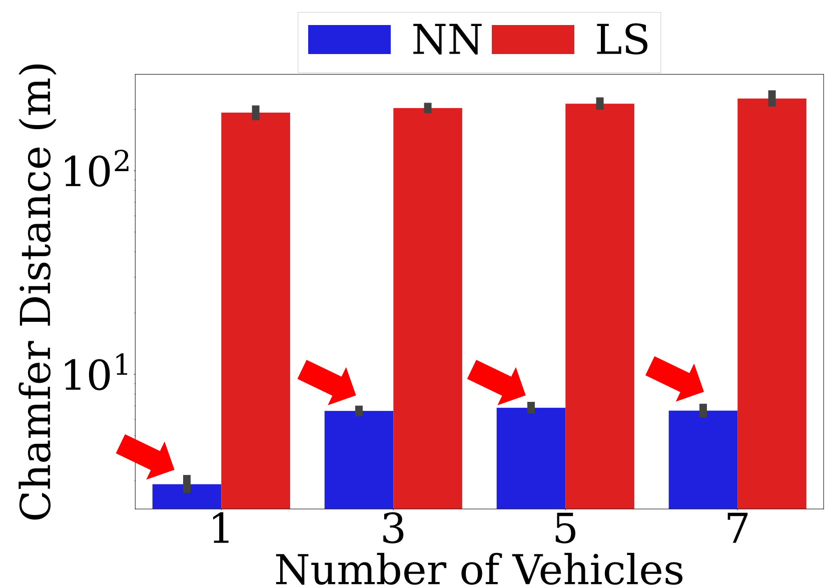




Design decision of concealment approaches (2/3)

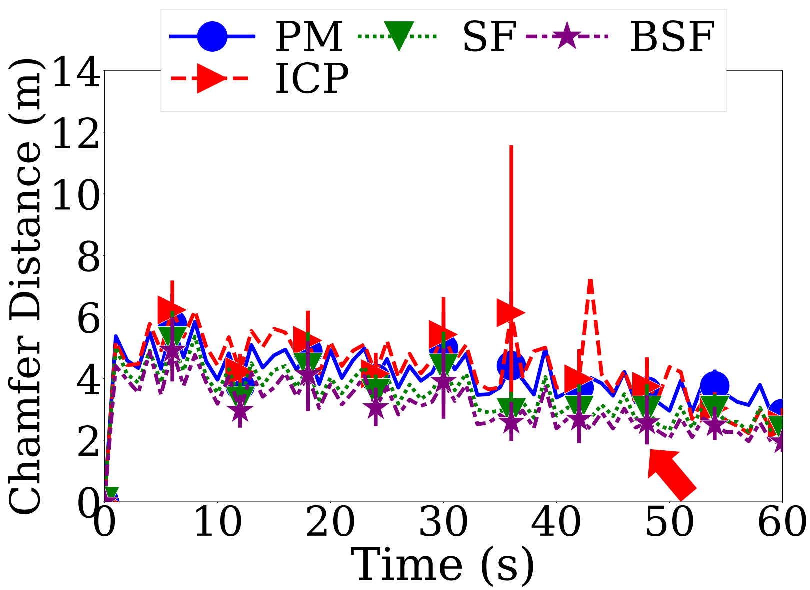
- NN reduce the Chamfer distance by up to 98.48% compared to LS
- We recommend using NN in SI approach

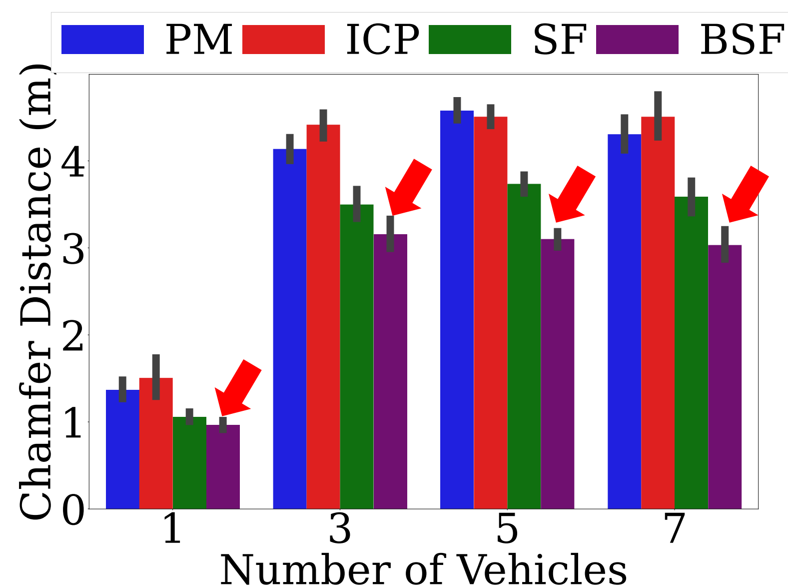




Design decision of concealment approaches (3/3)

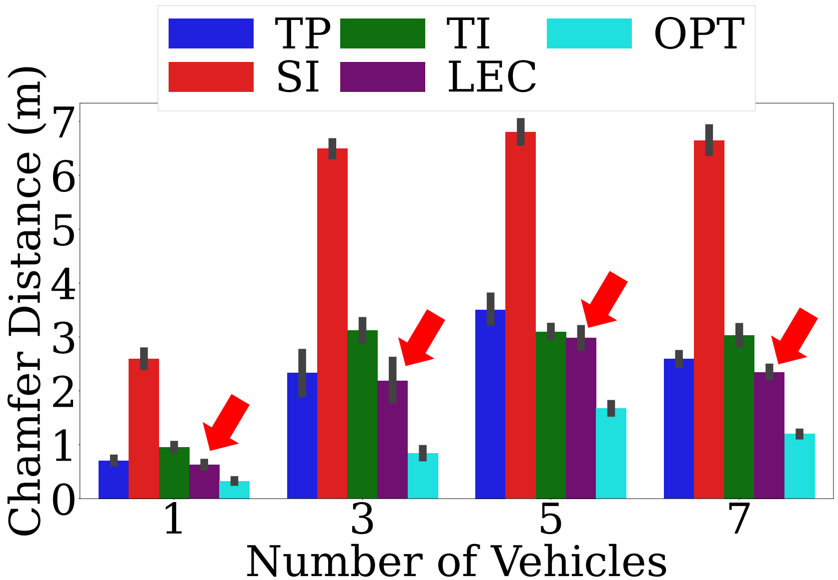
- BSF reduce the Chamfer distance by up to 71.48% compared to other algorithms in TI approach.
- We recommend BSF in TI approach

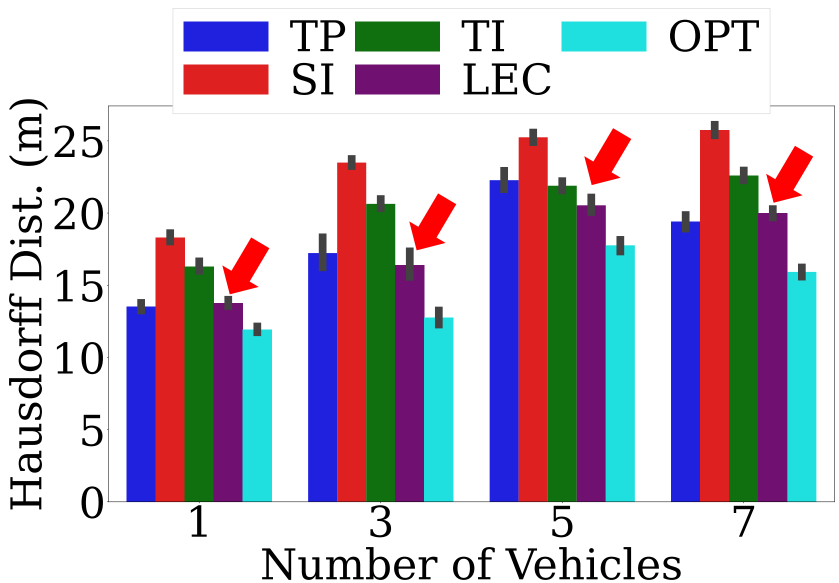




Low-level Performance of LEC with the co-simulator dataset.

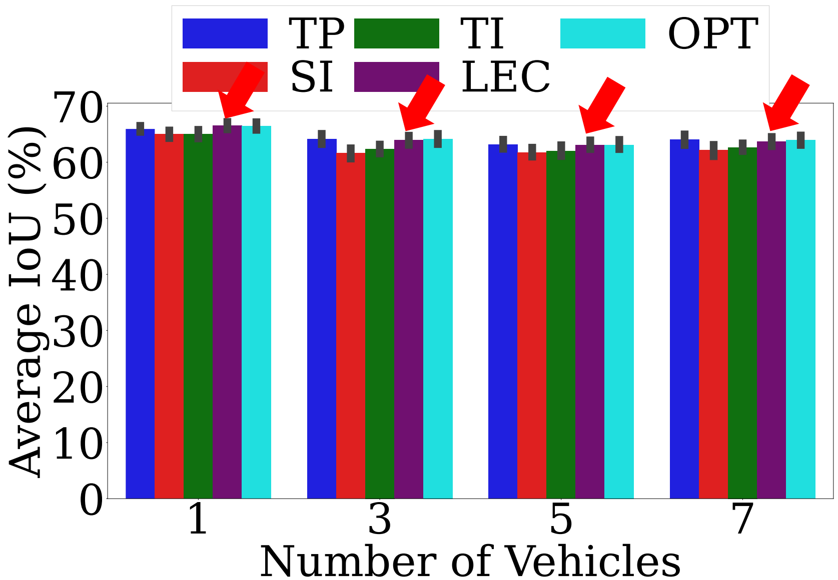
- Our LEC algorithm:
  - Reduce the Chamfer distance by up to 75.77%
  - Cuts the Hausdorff distance by up to 30.17%
  - With a small gap of at most 25.55% in Hausdorff distance

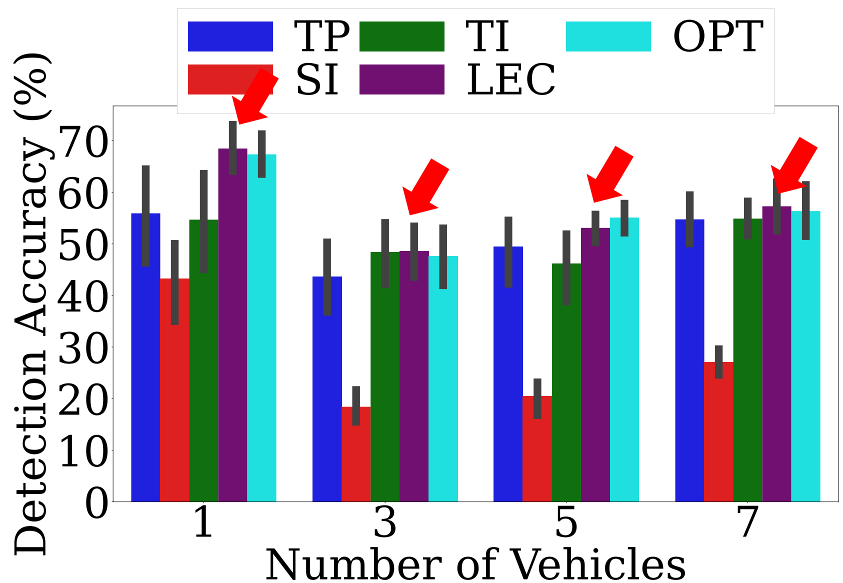




High-level Performance of LEC with the co-simulator dataset.

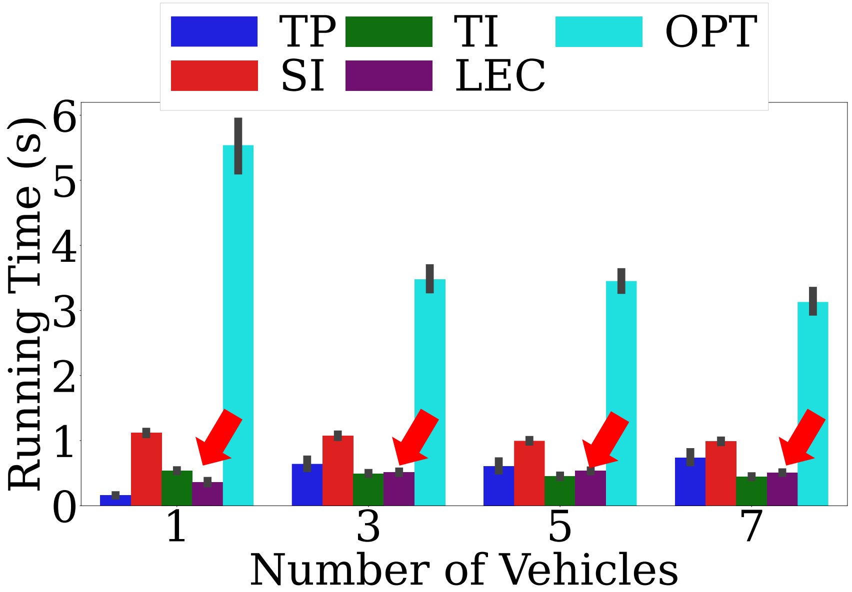
- Our LEC algorithm:
  - Improves the detection accuracy by at most 33.31%
  - With a tiny gap of at small as 0.75% than OPT in detection accuracy
  - With a small gap of at most 0.04% than OPT in average IoU





Running Time of LEC with the co-simulator dataset.

- Our LEC algorithm terminates in 360-570 ms throughout our evaluations.
- Our LEC algorithm can run faster and achieves small gaps from OPT



The running time of OPT is underestimated!

Performance of LEC in a DSRC network

- Compared to C-V2X, DSRC network often causes longer inter-packet intervals
- Our LEC algorithm outperforms TP, SI, and TI in Chamfer and Hausdorff distances by 12.25%-87.43% and 2.46%-66.58%, respectively.

	TP	**SI**	**TI**	**LEC**	**OPT**
Chamfer D. (m) $\downarrow$	0.98	6.84	3.23	0.86	0.79
Hausdorff D. (m) $\downarrow$	8.95	26.12	20.92	8.73	8.55
Run. Time (ms) $\downarrow$	589	1130	570	589	2307
IoU (%) $\uparrow$	66.59	66.34	66.38	66.75	66.79
Accuracy (%) $\uparrow$	52.52	45.96	52.91	53.91	54.36

Performance of LEC in pre-recorded KITTI dataset

- Our LEC algorithm
  - outperforms TP, SI, and TI in Chamfer and Hausdorff distances by 2.56%-92.49% and 0.58%-62.48%, respectively
  - saves 70.72% of the running time compared to OPT, with small gaps of 5.56% and 2.59% in Chamfer and Hausdorff distances

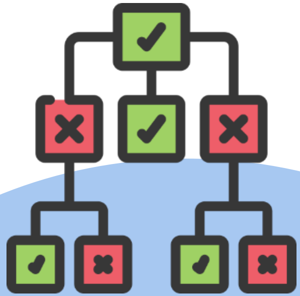
	TP	SI	TI	LEC	OPT
Chamfer D. (m) $\downarrow$	5.06	1.37	0.39	0.38	0.36
Hausdorff D. (m) $\downarrow$	31.93	14.08	12.05	11.98	11.67
Run. Time (ms) $\downarrow$	50	820	420	380	1298

CONCLUSION & FUTURE WORK

Conclusion

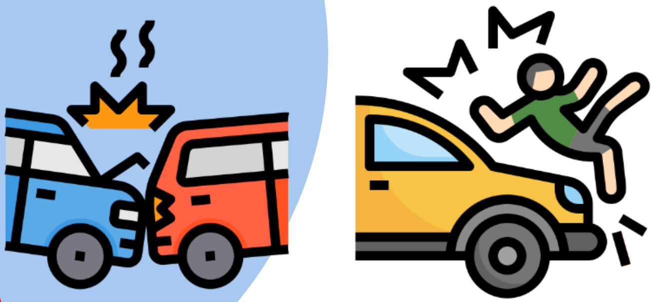
1. Studied the uninvestigated problem of error concealment for dynamic LiDAR point clouds
2. Implemented a comprehensive co-simulator of CARLA and NS-3
   - NR C-V2X and DSRC networks
3. Proposed our LEC algorithm to adaptively select the most promising error concealment approach using an ML model.
4. Significantly outperform the TP, SI, TI and with a small gap for OPT

Future Work



1 Smarter ML models

2 Multi-vehicles feature extraction



3 Large-scale and more simulations

Thank you for listening!

Thanks for the help of Prof. Hsu, Chih-Chun Wu, Ching-Ting Wang and all lab mates.

Publications:

- Guihua Shi, Chih-Chun Wu, Cheng-Hsin Hsu. Error Concealment for Dynamic LiDAR Point Clouds for Connected and Autonomous Vehicles[C]//GLOBECOM 2023-2023 IEEE Global Communications Conference. IEEE, 2023. **(under review)**

Q&A