Notes: 6/11-6/18

6/15

OPV2V: An open benchmark dataset and fusion pipeline for perception with Vehicle-to-Vehicle Communication

* Utilizing OpenCDA and CARLA simulator
* Proposes an Attentive Intermediate Fusion pipeline

Sensor data consists of

* 4 X cameras: Showcases a 360 degrees FOV at 800 x 600 resolution
* 1 X lidar: 64 channels, capturing at 120 m range with +-2 cm error
* GPS and IMU: 20 mm positional error and 2 degrees heading error

Attentive Intermediate Fusion Pipeline:

* Metadata Sharing: build connection graph and broadcast locations among neighboring CAVs
* Feature Extraction: Extract features based on each detector’s backbone
* Compression: Use Encoder-Decoder to compress/decompress features
* Feature Sharing: share (compressed) features with connected vehicles
* Attentive Fusion: leverage self-attention to learn interactions among features in the same spatial location
* Prediction: generate final object predictions

Benchmarks:

* Four Lidar-based 3D object detectors
* 16 models implemented to evaluate, including single-vehicle settings (meaning it neglects v2v)
* Tested on early, intermediate and late collaborations

6/16

[Collaborative Perception for Connected and Autonomous Driving: Challenges, Possible solutions and Opportunities](https://arxiv.org/pdf/2401.01544v1)

Challenges:

* Asynchronous Information Sharing
  + As CAVs send information to the ego vehicle there will be latency due to the sensors used by each vehicle and model.
* Pose Errors
  + The 6 Degrees of Freedom pose estimated by each CAV’s is not precise.
* Collaboration Security

Possible solutions:

* Asynchronous Info. Sharing:
  + Methods to compensate for the time delay
    - Time delay can be part of the data fed to the convolution network
    - Latency aware framework utilizing past collaboration data to estimate concurrent data
* Pose Errors:
  + Deep learning-based methods
    - Multiscale window attention: Windows of varies sizes, the larger windows hold extensive visual information while smaller windows focus on finer details
    - Split-attention: Merging multi-scale windows
  + Pose correction methods
    - Pose correction before transmitting the information using a pose graph and optimizing it
* Multi-model collaboration:
  + Allow different types of sensors to be used within one agent
    - Usually, one single model agent fuses data of CAV within the same sensors. EX. One agent for Camera data, a separate agent for Lidar data

Their solution:

* Transmission Delay Minimization
  + Adjacent matrix to represent V2V communication graph
  + Shannon Capacity Theorem
  + Higher priority for data transmission to closer vehicles
  + Optimizing the compression ratio and transmission link matrix using gradient descent
* Adaptive Refinement Reconstruction
  + Optimize R-D(rate-distortion) trade off
  + Dynamically adjusts the compression ratio based on real-time channel conditions
  + Convolutional neural network encoder and decoder for data compression and reconstruction
  + Temporal redundancy (repetitive information)
    - Reconstruction network is trained and refined using the raw data which is sent to a roadside edge server
* Domain Alignment
  + Reduce the domain discrepancy (environmental differences) between the perceived images of the ego CAV and other CAVs.
  + Use FFT (fast Fourier transform) to convert the ego vehicle’s images to spectrum space then decouple into amplitude and phase spectrums.
  + Amplitude spectrum are aligned with other vehicles then IFFT( inverse fast Fourier transform) is used to generate the spatial space.

6/17

[OpenCDA: An Open Cooperative Driving Automation Framework Integrated with Co-Simulation](https://arxiv.org/pdf/2107.06260)

Full stack:

* Perception
* Localization
* Planning
* Control
* V2X communication

Simulation with Carla and SUMO Co-simulation

* Carla for 3D environments
* SUMO for large-scale traffic flow modeling.

A diagram of a company

AI-generated content may be incorrect.

Simulating Realistic Rain, Snow, and Fog Variations For Comprehensive Performance Characterization of Lidar Perception

* A realistic simulation model to replicate the effects of rain, snow, and fog on lidar
  + Most datasets are captured in clear conditions, this will give it a more adverse weather
* They simulate rain using ray-traced drop interference, snow by using distribution-based flake scattering(with ray-tracing) and fog with probabilistic attenuation and backscatter.
* Tested object detection using the PointPillars CNN for lidar.