Slim-FCP: Lightweight Feature-Based Cooperative Perception for Connected Automated Vehicles

Jingda Guo, Dominic Carrillo, Qi Chen, Qing Yang, Song Fu, Hongsheng Lu, Rui Guo
Department of Computer Science and Engineering
University of North Texas, Denton, TX
Toyota Motor North America, R&D InfoTech Labs
{jingdaguo, dominiccarrillo, qichen}@my.unt.edu {qing.yang, song.fu}@unt.edu
{hongsheng.lu, rui.guo}@toyota.com

***Abstract*—Cooperative perception provides a novel way to conquer the sensing limitation on a single automated vehicle and potentially improves driving safety. To reduce the transmission data volume, existing solutions use the intermediate data generated by convolutional neural network (CNN) models, namely feature maps, to achieve cooperative perception. The feature maps are however too large to be transmitted by the current V2X technology. We propose a novel approach, called Slim-FCP, to significantly reduce the transmission data size. It enables a channel-wise feature encoder to remove irrelevant features for a better compression ratio. In addition, it adopts an intelligent channel selection strategy through which only representative channels of feature maps are selected for transmission. To evaluate the effectiveness of Slim-FCP, we further define a recall-to-bandwidth (RB) ratio metric to quantitatively measure how the recall of object detection changes with respect to the available network bandwidth. Experiment results show that Slim-FCP reduces the transmission data size by 75%, compared with the best state-of-the-art solution, with a subtle loss on object detection's recall.**

***Index Terms*—Automated vehicles, cooperative perception, 3D object detection, feature fusion.**

I. Introduction

Perception system on automated vehicles (AV) allows a vehicle to collect information and extract relevant knowledge from the environment, e.g., detecting objects [1]. Cooperative perception, on the other hand, enables vehicles to share local perception data with each other [2]. The prime reason for developing cooperative perception is maximizing the line of sight and field of view of automated vehicles. The extended field of view on automated vehicles will significantly improve the recall performance on object detection on automated vehicles. The major challenges of achieving this goal on connected and automated vehicles (CAVs) lie in transmitting massive amounts of rich sensor data between vehicles.

A. Main Challenges

There are two technical challenges we need to conquer in designing an efficient and effective cooperative perception solution. The first challenge is to diminish the feature map on spatial and channel domains with less sacrifice on performance. While not all features are meaningful for the object detection task, the irrelevant features can be removed before transmission to save communication resources. Moreover, if irrelevant features can be accurately identified and replaced with a reference number (e.g., zero), feature maps can be further compressed to save network bandwidth. The perturbation on feature maps caused by removing irrelevant features, however, may affect the detection recall of an object detection model, which degrades the performance of cooperative perception. In other words, feature removal should not excessively sacrifice performance, leaving us a big challenge on designing a proper feature compression and removing mechanism.

The second challenge lies in the difficulty of channel selection of feature maps to exchange amongst vehicles for cooperative perception. While F-Cooper [3] reduces the transmission data volume by only transmitting a subset of feature maps, how to select the best set of channels for cooperative perception remains an open question. The selection of improper channels may lead to a significant drop in detection performance, as the semantic information provided by the received feature maps could become insufficient. The volume of semantic information carried by different channels varies, and identifying representative channels of feature maps helps us achieve better performance with less resources needed.

B. Proposed Solution

To address the above-mentioned challenges, we propose a lightweight feature-based cooperative perception (Slim-FCP) for connected automated vehicles. The architecture of Slim-FCP is shown in Fig. 1 which contains three major components: channel-wise feature encoder and decoder, irrelevant feature remover, and channel selection on feature maps. With Slim-FCP, the feature maps produced by a convolutional neural network (CNN) model will be encoded and significantly compressed before they are shared. Here we assume two CAVs to utilize the same detection model, as many works have indicated that models on devices can be up-to-date periodically from cloud or edge servers [4], [5].

While F-Cooper [3] replaces original raw data with feature maps to achieve cooperative perception, we argue that

Copyright (c) 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

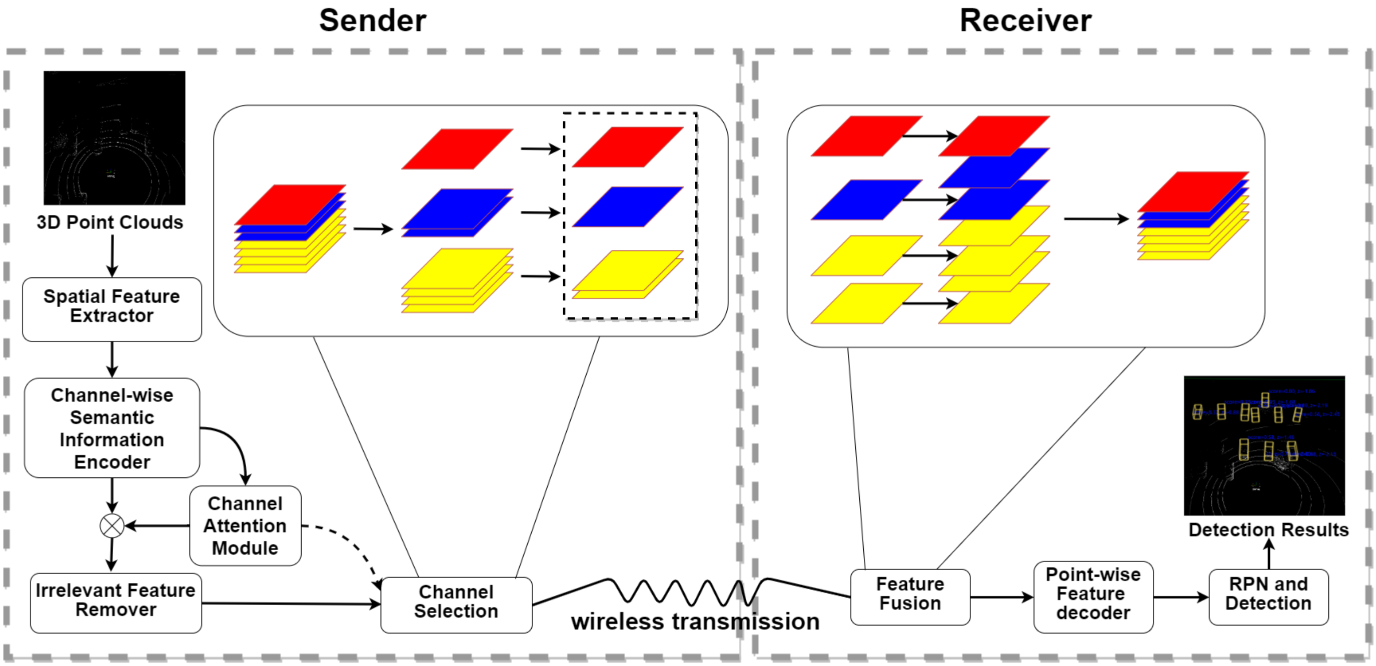


Figure 1: Overall structure of Slim-FCP

feature maps can be further encoded by machine-learning approaches. The parameters of a feature map can significantly be reduced by feature encoding with negligible effect on detection performance. To this end, we propose the semantic information based encoder and decoder to reduce/recover the parameters of transmitted feature maps. The encoder and decoder design considers the channel dependencies that exist on feature maps, i.e., we adopt channel-wise convolution on the encoder to remove channel dependencies to facilitate the following channel selection operation. By doing so, we avoid the lacking channels problem from F-Cooper, which alters the fusion results in many cases. For the decoder (on the receiver side), we employ a deconvolutional layer and a point-wise convolutional layer to decode the original semantic information, by rebuilding the shape and channel dependencies of the received feature maps.

Because not all features/parameters on a feature map are meaningful for object detection, we propose a learning-based irrelevant feature remover, which dynamically replaces irrelevant features with a reference number, increasing the sparsity of a feature map and improving feature map compression ratio. To deal with the situations of network congestion, or limited network bandwidth, we propose a channel selection mechanism to enable AVs to exchange fewer channels of feature maps but achieve similar cooperative perception performance. Unlike F-Cooper's channel selection strategy, our solution considers how much a channel can contribute to the object detection task and how much redundant information a channel carries, compared to already-selected channels. The proposed channel selection strategy can eliminate the issues caused by transmitting a random subset of channels in F-Cooper.

C. *Contributions*

The main contributions of this paper can be summarized as follows. First, we propose a lightweight feature-based cooperative perception framework for connected automated vehicles, which significantly reduces transmission data volume between CAVs, compared with the state-of-the-art feature-based cooperative perception solutions. Second, to the best of our knowledge, we are the first to explore what semantic information contributes more to cooperative perception. Experiment results show that a large portion of a feature map is useless for object detection and can be removed for transmission efficiency. Third, for extreme network conditions, e.g., with network congestion, the proposed channel selection strategy can achieve a similar detection performance by only transmitting partial channels of feature maps. We verify that Slim-FCP introduces negligible computation overhead on CAVs, and the sacrifice on detection recall caused by channel selection is acceptable. As proven in our experiments, both processing and transmission times of Slim-FCP are tiny. Moreover, Slim-FCP is generic and applicable to other systems, such as a vehicular edge system, to enable a cooperative perception system with the assistance of edge computing.

II. PRELIMINARIES AND BACKGROUND

With the rapid grown of edge computing [6], [7], more complicated computing tasks are now implemented in a distributed manner. It was shown that the sensing and detection ability of automated vehicles could be enhanced by enabling the communication between AVs to achieve so-called cooperative perception [2], [3], [8], [9]. Objects that cannot be detected by a single vehicle can easily be located, considering the information from other vehicles.

Three types of information can be shared between vehicles to achieve cooperative perception: (1) raw sensor data, (2) feature maps generated from detection models, and (3) final detection results. Cooper [2] allows CAVs to share raw LiDAR data with others, thus significantly improves sensing range and detection recall. However, raw data sharing can be a huge burden to wireless vehicular networks. An automated vehicle can generate LiDAR data at a speed 100 MB/s, and sharing such a huge amount of data among vehicles is unrealistic, even for a high-speed vehicular network. To save network bandwidth, sharing final object detection results to achieve cooperative perception introduces less traffic onto the network. The performance of result-based cooperative perception, however, is limited by the detection effectiveness of individual vehicles. For example, a receiver vehicle can still not detect the objects that have not been detected by other vehicles.

To address this issue, the currently existing solutions utilize the intermediate results processed by a detection model, namely feature maps, to realize feature-based cooperative perception [3], [10]. As most object detection models on automated vehicles are CNN-based, feature maps produced by a model's feature extractors become accessible. A feature extractor usually consists of several convolutional layers which extract abstract features from raw data. The extracted features, stored in a feature map, represent important semantic information of the input data. Only features are treated as the input for high-level machine learning tasks, e.g., object detection and classification. By sharing feature maps among vehicles, cooperative perception can be achieved with less network traffic introduced. F-Cooper [3] allows vehicles to share feature maps and fuse received and local ones to realize a more precise object detection. When the receiving feature maps complement their own ones, new semantic information is added to improve the detection model's performance. The fused feature maps are generated by employing the $maxout$ function on two feature maps, which can be regarded as the merging of these feature maps. As stated in F-Cooper [3], for objects that cannot be detected on the individual vehicle, they can be detected from fused feature maps, which is also the main contribution of raw-data cooperative perception [2].

III. SLIM-FCP: LIGHTWEIGHT FEATURE BASED COOPERATIVE PERCEPTION SOLUTION

Vehicular applications have less tolerance on high frame processing delays [11], [12]; therefore, finding the right balance between performance and network resourcing and pushing data processing tasks from vehicle to edge is a significant research task [13]–[15]. While F-Cooper enables cooperative perception on CAVs by sharing feature maps, the size of feature maps must be reduced before transmission to save network bandwidth and decrease networking delay. To this end, we propose to design (1) a channel-wise autoencoder, (2) an irrelevant feature remover (IFR), and (3) a channel selection mechanism, to reduce the size of the transmitting feature maps.

A. Channel-Wise Semantic Feature Encoder and Decoder

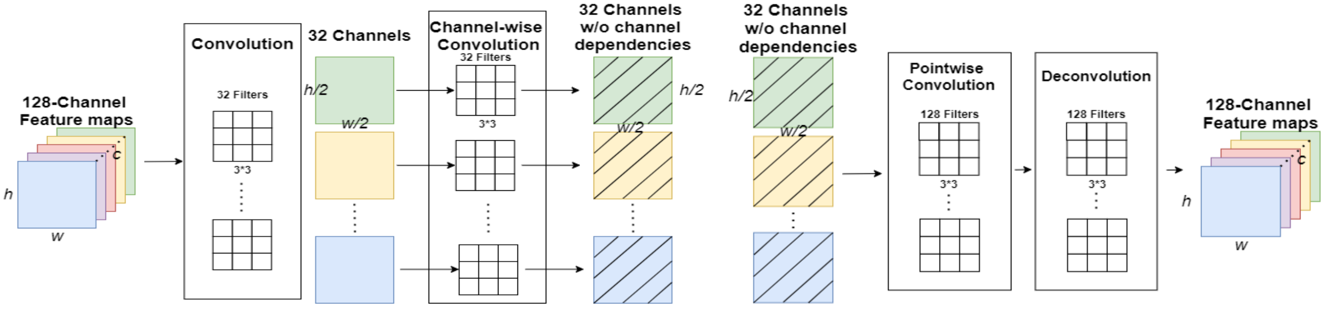
As feature maps are usually sparse and useful semantic information is generally represented by a small portion of a feature map, we propose to employ a channel-wise semantic information encoder (on sender side), to encode feature maps, and a point-wise convolution decoder (on receiver side), to recover the original semantic information. The proposed autoencoder's architecture is shown in Fig. 2 where 128-channel feature maps are converted into 32 independent channels. Although the proposed autoencoder can be configured to support a higher conversion ratio, e.g., 128 to 16 channels, we find converting (128 channels) to 32 channels provides the best tradeoff between object detection recall and transmitted data volume.

It is well-known that the semantic information within a feature map is represented in both the spatial ($h * w$) and channel (c) domains, where h and w are the height and width of feature maps, and c is the total number of channels. By an autoencoder [16], the original 128-channel feature maps can be converted into 32 channels, as shown in Fig. 2a. Introducing the autoencoder aims to generate a concise representation of the original feature maps, with reduced dimensionality, by training the network to ignore noise/background information. As the involved convolutional operations are applied to all channels, the resulting features in 32 channels are correlated, complicating the channel selection (which will be discussed later). To mitigate this issue, making future operations on independent channels, we further propose the channel-wise feature autoencoder, which removes the correlations among channels to produce 32 channels without channel dependency. Different from traditional autoencoder [17], in our channel-wise feature autoencoder, we set the group number equal to the channel number in the convolution operations to ensure one channel is processed by only one filter. Interested readers can refer to [18] for more details on channel-wise convolution.

Correspondingly, we design a point-wise convolution decoder on the receiver side to recover the shape and semantic information of received feature maps. To avoid the potential issues caused by removing the channel dependencies from the encoder, we adopt the point-wise convolution [18] to rebuild the channel dependencies by using 1×1 kernels in convolution operations. It is then followed by a deconvolution layer to recover the spatial size of the received feature maps, as shown in Fig. 2b. Our point-wise convolution decoder, together with the channel-wise feature encoder, makes preparations for future channel selection, and significantly reduces the size of transmitted feature maps.

B. Channel Selection on Feature Maps

While different channels represent different semantic information after channel-wise convolution encoder, selecting representative channels for transmission could be challenging. Improper channel selection may cause a significant drop in detection performance. To select the best channels for transmission, we consider both attention weights and the uniqueness of semantic information contained in various channels.



(a) Channel-wise Feature Encoder

(b) Point-wise Feature Decoder

Figure 2: Semantic feature encoding and decoding. (a) On the sender side, the to-be-transmitted feature maps first go through a convolutional layer, composed of 32 3×3 filters, which reduce the size of each feature map from $w \times h$ to $w/2 \times h/2$. The resulting feature maps are then processed by a channel-wise convolutional layer (3×3 kernels) to produce 32 independent feature maps. The 32 feature maps are finally transmitted to the receiver. (b) When a set of feature maps are received, the receiver adopts a point-wise convolution layer and a deconvolution layer to recover the semantic information carried in the original feature maps.

Attention weight based channel selection. To weigh the importance of channels, we produce the SENet channel attention module [19] as shown in Fig. 3. Literature also states that the magnitude of feature value on feature maps can represent the strength of features [20] for 3D detection tasks. As the final output feature maps are the element-wise multiplication product of attention weight and input feature maps, we indicate that channels with higher attention weight contain more important semantic information than other channels. The channel attention module is adopted after the channel-wise feature encoder, and the output feature maps are computed as follows,

$$\mathbf{f^i} = F_{scale}(\mathbf{f_0^i}, w^i) = \mathbf{f_0^i} \otimes w_i, \forall i = 1, 2, \cdots, 32, \tag{1}$$

where $\mathbf{f_0^i}$ is the $i-th$ channel of input feature map, and F_{scale} denotes the element-wise multiplication, and $\mathbf{f^i}$ is the final output feature map of the attention module.

Semantic information based channel selection. However, channels selected by attention weight have its limitations. The semantic information carried by different channels may be repetitive. E.g., since laser becomes sparse when distance increase, near objects are more likely to have a prominent feature on the feature map, and channels that carry semantic information from the near region are more likely to have higher attention weight. Simply selecting channels with high attention weights may lead to a drop-down of detection performance on a certain region, e.g., a relatively far region.

For convolutional feature maps, if several channels represent repetitive semantic information, the diversity of these channels should be small; therefore, the distance between channels is also small. By contrast, a channel's semantic information is irreplaceable if the channel is different from other channels. Based on the above finding, we consider the redundancy of semantic information carried on different channels by measuring its norm distance to all neighbor channels. Similar to [21], we name the distance measuring process as feature map entropy, which helps us select representative channels of feature maps. Specifically, we construct a channel distance matrix d_i for $i-th$ channel on feature map, in which d_i contains the distance to k nearest neighbor channels. In this case, we measure the distance between channels by Euclidean distance $d_{ij} = \|\mathbf{f^i} - \mathbf{f^j}\|$. Then we compute the average distance of $i-th$ channel to its k nearest neighbors as follows,

$$A_i = \sum_{k=1}^{i} d_{ik}/k. \tag{2}$$

The larger the average distance A_i is, the channel is away from other channels, and more irreplaceable semantic information is carried by channel i. That is to say, channel i can be considered a representative channel for transmission. Therefore, the corresponding channel contains more semantic information and should be selected for transmission. With the consideration of the semantic information redundancy across channels, we avoid the potential semantic information lack in a specific region, e.g., a far area with fewer point clouds been collected.

In the experiment, we jointly consider the attention weight and uniqueness of semantic information for channel selection, cover the whole physical region with fewer channels, and avoid excessive sacrifices on the detection performance after fusion. We discuss our detailed implementation in the experiment section.

C. Irrelevant Feature Remover

In addition to channel selection, classic data compression solutions can also be leveraged to reduce the size of transmitted feature maps, and increase the communication efficiency and network capacity [22]. A key observation is that irrelevant features in feature maps can be removed with negligible effect on CNN-based models [23], [24]. If we remove those irrelevant features from feature maps and set their values to a constant number, a much better compression ratio on feature maps can be achieved. A common approach to removing irrelevant features is the mask-based feature remover [23], which computes a mask associated with each pixel in a feature map. The remover mask replaces irrelevant information with

Scenario	Dataset	Cooper [2]		F-Cooper [3]		Slim-FCP w/o CS		Slim-FCP	
		Near	Far	Near	Far	Near	Far	Near	Far
Multi-lane Roads	KITTI	76.89	64.19	72.91	59.14	72.47	58.34	70.83	54.37
Road Intersections	T&J	69.27	57.33	65.50	52.61	64.72	52.15	61.75	48.02
Parking Lots	T&J	65.03	52.42	61.76	46.12	61.38	45.70	57.84	40.56

Table I: Recall comparison among Cooper [2], F-Cooper [3], Slim-FCP without Channel Selection, and Slim-FCP (%).

the pre-defined perturbation, e.g., a reference number or noise. The remover is defined as the follows,

$$[\Phi(f_0;m)(p)] = \ m(p)f_0(p) + (1-m(p))\xi, \tag{3}$$

where p is one pixel of input feature map f_0, $m(p) \in [0,1]^{C*H*W}$ is the corresponding mask value associated with pixel p. The choices of a reference number can be a constant value ξ, as stated in Eq. (1).

To seek a higher compression ratio, we use the constant number "zero" as the reference in the mask. Taking zero as the reference also saves the computational effort and contributes to the consistent output of the model, as stated in [24]. With the irrelevant feature remover, the background and irrelevant area on the feature map is changed to zero. Meanwhile, the prominent area on the feature map is nearly unchanged, allowing us to compress feature maps with a high compression ratio.

As the irrelevant feature remover perturbs the feature maps with masks, we still hope to keep the consistency of the detection output of our model. Therefore, we define a distance metric $\rho(f, f_0)$, where the f represents the output feature maps of irrelevant feature remover, to measure the cosine distance between two feature maps. The distance must be small if two feature maps are similar, which means a relatively consistent detection output; otherwise, it means a deviation on outputs. Therefore, this distance metric can ensure the output consistency of our model, and be optimized along with the model during the training process. The overall mask operation and optimization for irrelevant feature remover can be summarized as follows.

$$f(p) = m(p)f_0(p), \tag{4}$$

$$m(p) = \underset{m(p)}{\arg\min}\{\rho(f, f_0) + \lambda \cdot ||m(p)||_1\}. \tag{5}$$

Here, the L1 norm of $m(p)$ keeps the sparsity of the mask, and most pixels are zero, leading to the output feature maps being sparse. Only prominent and crucial semantic features remain on feature maps, and those areas on the feature map correspond to the large values on the generated feature masks.

IV. Experiment and Result Evaluation

In this section, we evaluate the performance of Slim-FCP compared with the baseline feature-based cooperative perception solution, F-Cooper [3].

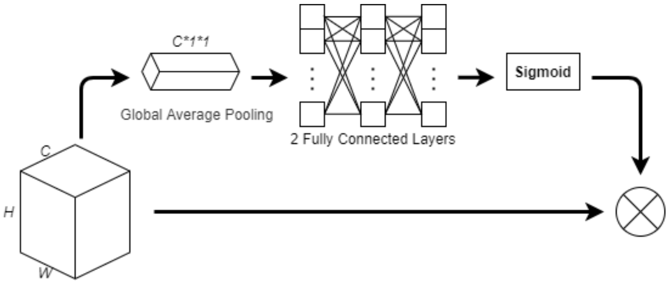


Figure 3: Attention module from [19]. The global average pooling operation generates the channel-wise statistics of input feature maps. The following two fully connected layers build the channel dependencies of the output weight, and the Sigmoid function maps the weight to $[0, 1]$. The attention weight scales with the input feature maps at the end of the attention module.

A. Experimental Setup

We evaluate our Slim-FCP on both KITTI [25] and T&J datasets [3]. KITTI is a well-known vision benchmark dataset; however, as it is not created for cooperative vision processing tasks, it offers a limited amount of data to evaluate cooperative perception solutions. In KITTI, the data collected by one vehicle at two different time instances are considered as if generated from two different vehicles. To evaluate cooperative perception solutions in a more realistic setting, we extend the T&J dataset provided by F-Cooper by including more driving scenarios. The T&J dataset contains more new scenarios, e.g., road intersection and parking lots. In total, we use approximately $1,600$ and 800 sets of data for evaluation from the T&J and KITTI datasets, respectively. In experiments, the detection region of the automated vehicle (equipped with a Velodyne VLP-16 LiDAR sensor) is $[0, 70.4]$, $[-40, 40]$ and $[-3, 1]$ meters along the x, y, and z axles. We define objects within 20 meters from the receiver vehicles are in the "near" category, and those beyond this range are in the "far" category.

B. Cooperative Detection Recall

We dive into the details of how Slim-FCP performs on detecting 3D objects, against the baseline F-Cooper [3], and the raw data fusion solution, Cooper [2]. We report our results with the IoU (Intersection over Union) threshold equaling to 0.7, and the detection confidence score threshold is 0.5. We take the top 10 unique channels for channel selection, according to two selection strategies, namely attention weight, and semantic information channel selections. We set nearest neighbor number $k = 5$ for distance computation among channels. The total number of the selected channels for transmission is 20.

The comparison results are shown in Table I where "Slim-FCP w/o CS" represents a simplified version of Slim-FCP which does not implement the channel selection mechanism. For the "near" category, we observe that the recall of Slim-FCP w/o CS is similar to F-Cooper's on the KITTI dataset. The recall gap of two approaches is less than 1%. For road intersection and parking lots cases, the recall decrease is not as apparent as on the multi-lane road cases. The main reason is that the road intersection and parking lots cases are from the T&J dataset, which is low-resolution data compared to the KITTI dataset and is less sensitive to the tiny changes in detection performance. The recall decrease is also inconspicuous for the "far" category, which is about 0.5% in all cases. In summary, the detection recall is very similar between F-Cooper and Slim-FCP w/o CS, meaning that the effective representation of semantic information on feature maps is not apparently affected by semantic feature encoder and irrelevant feature remover. The slight decrease in the recall is mainly caused by encoding, as slimming feature maps reduce a large number of parameters and inevitably drop a tiny part of semantic information.

To verify the performance of Slim-FCP when it enables channel selection, we show the recall comparison results in Table I. For the "near" category, the recall decrease of Slim-FCP on the KITTI dataset is approximately 2%, while on the T&J dataset, it is about 4%, compared to F-Cooper. It makes sense that a high-end LiDAR sensor is used to collect the KITTI dataset; therefore, the corresponding feature maps contain more semantic information and have better resistance capacity against channel selection. For the "far" category, Slim-FCP performs well on open-area cases, such as multi-lane roads and intersections, with about 5% recall decrease. In the parking lots cases, the decrease is slightly larger. Due to the occlusion, features from far areas become inconspicuous. The Slim-FCP further partially discards the features by channel selection, leaving us a slightly larger decrease on recall. However, even so, the recall of Slim-FCP in the parking lots cases is still over 40%. Most mis-detected vehicles are either very far from the source vehicles or highly occluded in a certain position. These objects are considered lower priority objects and can be easily detected with streaming feature sharing.

Moreover, we argue that our channel selection keeps Slim-FCP running when network bandwidth is limited, while other approaches may not work in such a situation. Due to the transmission data volume required by non-selection approaches, receivers may not receive messages or receive outdated messages that are useless. Meanwhile, our Slim-FCP with channel selection performs well with a slight sacrifice on recall even in this extreme situation. More importantly, Slim-FCP still keeps the cooperative perception working with limited bandwidth resources. Moreover, slimming messages by channel selection enables more AVs to participate in cooperative perception, compensating the performance gap between non-selection and selection approaches. For situations with smooth network communication, CAVs should adopt the Slim-FCP w/o CS for the best detection result.

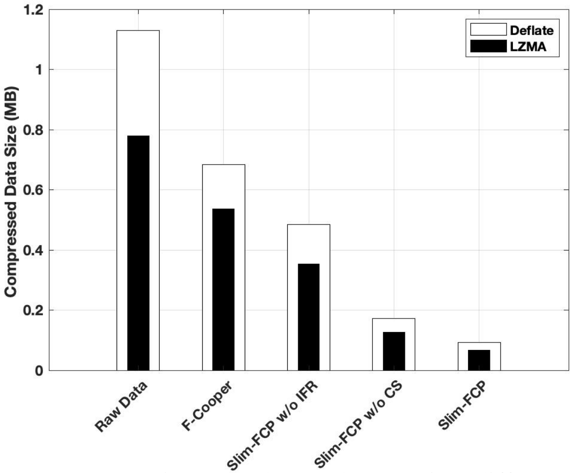


Figure 4: Comparison on data volume using different compression approaches

C. Transmission Data Size

Fig. 4 shows the comparison of different cooperative perception approaches on transmission data size. F-Cooper utilizes a lossless data compression method Deflate [26], and the compressed data size is about two-thirds of the raw data. For Slim-FCP, the transmission data size is reduced to $0.485\ MB$ even without the irrelevant feature remover (IFR). The compressed data size of Slim-FCP is significantly reduced to approximately $0.17\ MB$ with IFR enabled, which is only one-fifth of the raw data. The large decrease in data size indicates that a significant proportion of features on feature maps are irrelevant features and can be removed with negligible effect on detection. The data size can be further reduced to about $93\ KB$ when Slim-FCP enables the channel selection mechanism. In our experience, we replace Deflate with the LZMA algorithm [27], another widely used lossless compression algorithm, and increase the compression ratio. By comparison, the transmission data size of Slim-FCP can be as small as $67\ KB$ with channel selection for extreme cases such as network congestion and computational resource limitation.

D. Recall/Bandwidth Ratio

As indicated in [28], [29], bandwidth efficiency is one of the most important factors for edge-based applications. To demonstrate how Slim-FCP outperforms other approaches on bandwidth efficiency, we introduce a term called **Recall/Bandwidth ratio** (**RB ratio**). Here, we define the bandwidth as the size of data shared by a certain approach every second to enable cooperative perception. As shown in Fig 5, Cooper, the raw data cooperative perception solution requires the most bandwidth among all approaches, leading to a low RB ratio. By sharing feature maps instead of raw data, F-Cooper performs better on bandwidth efficiency than Cooper, which is

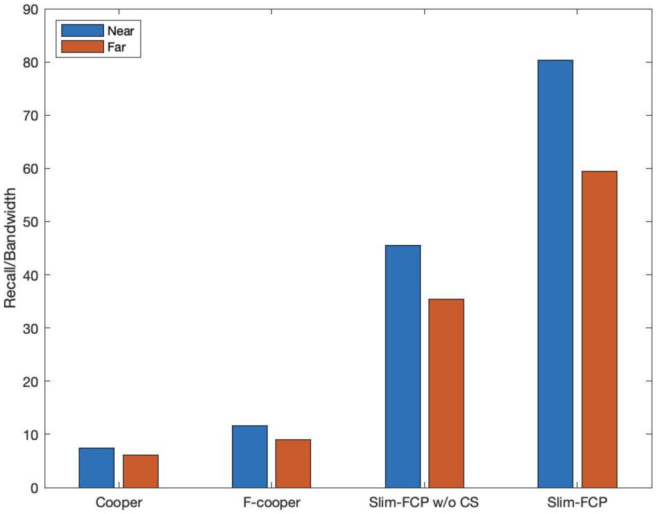


Figure 5: Comparison on Recall/Bandwidth ratio among different approaches

about 50% RB ratio improvement on both near and far categories. Meanwhile, Slim-FCP greatly outperforms F-Cooper on bandwidth efficiency. The RB ratio is approximately 3 times better than F-Cooper even without channel selection and it increases to 6 times with channel selection enabled. This dramatic difference shows the great improvement in the bandwidth efficiency of Slim-FCP and the robustness of detection performance when limited network bandwidth is available.

E. Processing and Transmission Delay

The computation burden introduced by Slim-FCP is negligible for processing units on CAVs, making our processing time very close to F-Cooper, which is about 20 fps. For transmission delay among CAVs, we compute the delay based on the 5G NR based C-V2X [30], the future networking protocol for V2X communication. Suppose the LiDAR sensor collected data at 10 Hz, and the conservative transmission data rate is 100 $Mbps$ in the 5G NR based C-V2X [30]. Taking the Slim-FCP with channel selection as an example, the total time required to transmit one piece of LiDAR data from a vehicle to an edge server (or another vehicle) is about 50 ms, and this delay introduced by C-V2X is acceptable for cooperative perception on CAVs.

F. Channel Selection Strategy

Our channel selection module produces two strategies for selecting representative channels: attention weight and semantic information based channel selection. To find the best combination of channels selected by two strategies, we hope to identify how performance evolves with different channel selection strategies. To this end, we compare the performance of five different selection groups, in which each option picks a different number of channels by two strategies. Channels are first picked up based on attention weight, and then the semantic information to ensure no repetitive channels exist in the selection results. As shown in Fig. 7, the recall of Slim-FCP increases when more channels are selected for transmission. The recall of groups with 15 channels are backward to groups with 20 channels on both “near” and “far” categories since the volume of semantic information is in direct ratio to the number of selected channels. For group 3 to 5, which contain the same number of channels, groups with more channels selected by attention weight (group 3) prioritize detecting “near” objects. As discussed above, channels representing semantic information from “near” objects are more likely to have large attention weights. By comparison, groups with more channels selected by semantic information have better detection recall on “far” objects. Two strategies seem to compensate for detecting objects with different distances. When we increase the number of channels selected by semantic information based strategy,

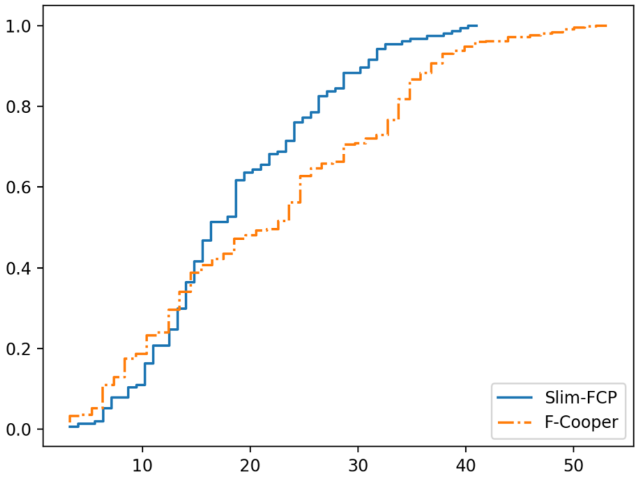


Figure 6: Cumulative Distribution Function vs. Range of detected objects in meters

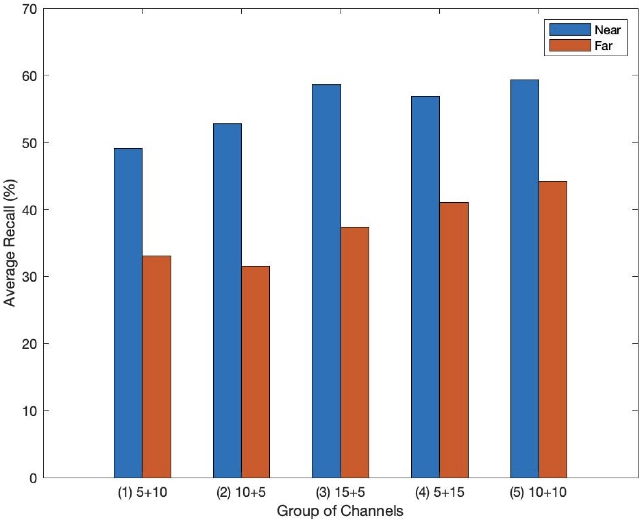


Figure 7: Recall comparison of Slim-FCP among different channel selection strategies. The first and second numbers are the number of channels selected by attention weight and semantic information, respectively.

the recall on the "far" category increases accordingly. To leverage the benefit of both selection strategies, we take group 5 as our primary evaluated approach, which picks ten unique channels by each selection strategy.

G. Effective Detection Range

As the increase of detection range is the main benefit of cooperative perception on CAVs, we compare F-Cooper to our Slim-FCP with channel selection enabled. Here we omit the comparison between F-Cooper and Slim-FCP without channel selection since they are very similar in detection range. We illustrate this comparison in Fig. 6, which shows the difference in the detection range of the two approaches. As we can see, 89% of detected objects by Slim-FCP are within 30 meters, while F-Cooper is 72%. This illustrates that our approach performs similarly with F-Cooper to detect objects within 30 meters, which is the high priority area for autonomous driving. The difference becomes apparent in detecting distant objects (over 30 meters). The maximum detection range of Slim-FCP with channel selection is about 42 meters, and only 11% of detected objects have a distance of over 30 meters. In contrast, F-Cooper performs better in detecting these objects and has a maximum detection range of 53.4 meters. This is another evidence to show that the channel selection strategy affects more on objects that do not have a prominent feature on feature maps. However, since channel selection potentially enables more AVs to participate in cooperative perception, this weakness can be compensated by feature maps from other nearby CAVs. Some feature enhancement strategies also provide a novel way to improve the detection performance on those distant objects [20].

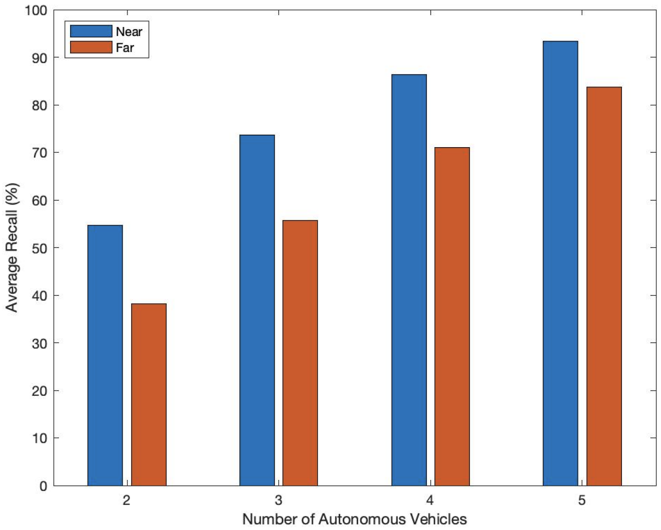


Figure 8: Recall improvement of Slim-FCP with multiple AVs

H. Slim-FCP on Multiple CAVs

We discuss how the number of participating AVs affect the performance of cooperative perception in this section, and compare the recall with multiple-AV cases on Fig. 8. Cases are evaluated on Slim-FCP with channel selection enabled, and the distance between sender and receiver is at least 12 meters. As shown in Fig. 8, the recall increases significantly when more AVs participate. This drastic difference proves that the missing semantic information caused by channel selection can be complemented by feature maps from other nearby AVs. The number of participating AVs determines the performance of cooperative perception. For an open detection region like our Slim-FCP ($80 \times 70.4\ m^2$), five participating AVs can achieve satisfactory detection results for the whole area. More AVs may need to participate to guarantee the performance when more occlusion occurs on the region, e.g., heavy traffic roads.

Acknowledgement

The work is supported by National Science Foundation grants CNS-1852134, OAC-2017564, ECCS-2010332, and the research grant from Toyota Motor North America, R&D InfoTech Labs.

V. Conclusion

In this paper, we propose a lightweight feature based cooperative perception idea on CAVs, which significantly reduces the size of the transmission data required by feature based cooperative perception solutions. Specifically, we design a semantic feature encoder to further slim the size of feature maps and remove irrelevant features using the irrelevant feature remover. To encounter extreme cases such as possible bandwidth limitation, we propose our channel selection strategy to select representative channels on feature maps for transmission. Compared with previous state-of-the-art approaches, the size of compressed feature maps of Slim-FCP is only $1/4$ of them, with a slight sacrifice on recall. Our experimental results show that Slim-FCP works well for detecting objects on various road environments, enables more automated vehicles and road infrastructures to participate in the cooperative perception with a limited amount of computational and network resources. We believe our Slim-FCP is very computational efficient with an acceptable trade-off on performance and will help the cooperative perception tasks on automated vehicles for better driving safety.

References

[1] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transportation research part C: emerging technologies*, vol. 89, pp. 384–406, 2018.

[2] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 514–524.

[3] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019, pp. 88–100.

[4] S. S. Ogden and T. Guo, "{MODI}: Mobile deep inference made efficient by edge computing," in *{USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[5] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[6] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[8] S.-W. Kim, Z. J. Chong, B. Qin, X. Shen, Z. Cheng, W. Liu, and M. H. Ang, "Cooperative perception for autonomous vehicle control on the road: Motivation and experimental results," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5059–5066.

[9] S.-W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle cooperative driving using cooperative perception: Design and experimental validation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 663–680, 2014.

[10] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, J. Tu, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," *arXiv preprint arXiv:2008.07519*, 2020.

[11] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *computer*, vol. 50, no. 10, pp. 58–67, 2017.

[12] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 2. IEEE, 2003, pp. 1521–1531.

[13] J. Flinn, S. Park, and M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing," in *Proceedings 22nd International Conference on Distributed Computing Systems*. IEEE, 2002, pp. 217–226.

[14] J. Xu, M. Zhao, and J. A. Fortes, "Cooperative autonomic management in dynamic distributed systems," in *Symposium on Self-Stabilizing Systems*. Springer, 2009, pp. 756–770.

[15] M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, "Energy-delay tradeoffs in smartphone applications," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010, pp. 255–270.

[16] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," in *Advances in neural information processing systems*, 1994, pp. 3–10.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[20] J. Guo, D. Carrillo, S. Tang, Q. Chen, Q. Yang, S. Fu, X. Wang, N. Wang, and P. Palacharla, "Coff: Cooperative spatial feature fusion for 3d object detection on autonomous vehicles," *arXiv preprint arXiv:2009.11975*, 2020.

[21] Y. Li, S. Lin, B. Zhang, J. Liu, D. Doermann, Y. Wu, F. Huang, and R. Ji, "Exploiting kernel sparsity and entropy for interpretable cnn compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2800–2809.

[22] A. B. Sharma, L. Golubchik, R. Govindan, and M. J. Neely, "Dynamic data compression in multi-hop wireless networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1, pp. 145–156, 2009.

[23] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3429–3437.

[24] J. Wagner, J. M. Kohler, T. Gindele, L. Hetzel, J. T. Wiedemer, and S. Behnke, "Interpretable and fine-grained visual explanations for convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9097–9107.

[25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[26] D. Salomon, *Data compression: the complete reference*. Springer Science & Business Media, 2004.

[27] D. Salomon and G. Motta, *Handbook of data compression*. Springer Science & Business Media, 2010.

[28] A. Chowdhery, P. Bahl, and T. Zhang, "Bandwidth efficient video surveillance system," Apr. 7 2020, uS Patent 10,616,465.

[29] S.-J. Lee, S. Banerjee, P. Sharma, P. Yalagandula, and S. Basu, "Bandwidth-aware routing in overlay networks," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 1732–1740.

[30] Shailesh Patil, "How nr based sidelink expands 5g c-v2x to support new advanced use cases," 2020, https://www.qualcomm.com/media/documents/files/nr-c-v2x-webinar-march-2020-presentation.pdf, Last accessed on 2021-06-23.