When2com: Multi-Agent Perception via Communication Graph Grouping

Yen-Cheng Liu, Junjiao Tian,* Nathaniel Glaser,* Zsolt Kira
Georgia Institute of Technology
{ycliu, jtian73, nglaser3, zkira}@gatech.edu

Abstract

While significant advances have been made for single-agent perception, many applications require multiple sensing agents and cross-agent communication due to benefits such as coverage and robustness. It is therefore critical to develop frameworks which support multi-agent collaborative perception in a distributed and bandwidth-efficient manner. In this paper, we address the collaborative perception problem, where one agent is required to perform a perception task and can communicate and share information with other agents on the same task. Specifically, we propose a communication framework by learning both to construct communication groups and decide when to communicate. We demonstrate the generalizability of our framework on two different perception tasks and show that it significantly reduces communication bandwidth while maintaining superior performance.

1. Introduction

Remarkable progress has been achieved for single-agent perception and recognition, where one or more sensor modalities are used to perform object detection [30, 31, 22] and segmentation [3, 12, 19], depth estimation [10, 38, 11], and various other scene understanding tasks. However, in many applications, such as robotics, there may be multiple agents distributed in the environment, each of which has local sensors. Such multi-agent systems are advantageous in many cases, for example, to increase coverage across the environment or to improve robustness to failures.

Thus, we tackle the problem of *multi-agent collaborative perception*, an under-studied topic in the literature, where multiple agents are able to exchange information to improve overall accuracy towards perception tasks (*e.g.*, semantic segmentation or object recognition). One major challenge for multi-agent collaborative perception is the transmission bandwidth, as high bandwidth results in network congestion and latency in the agent network. We therefore investigate the scenario where information across all agents (and hence sensors) is not available in a centralized manner, and agents can only communicate through bandwidth-limited channels. We also consider several challenging scenarios where some sensor data may be uninformative or degraded.

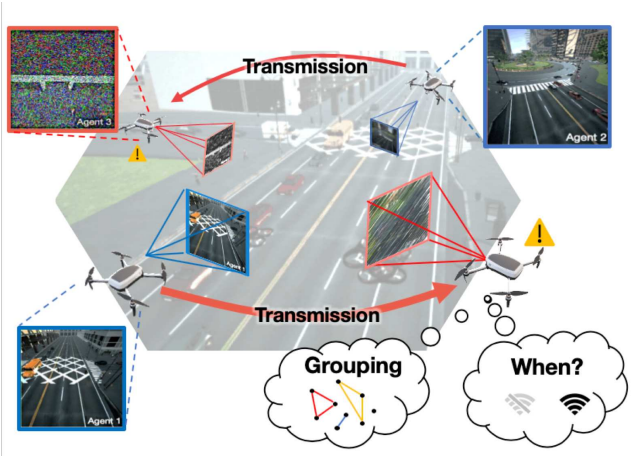


Figure 1: **Illustration of multi-agent collaborative perception.** We construct a multi-agent perception system to improve the agent-wise perception accuracy and reduce the transmission bandwidth. Each agent learns to construct *communication groups* and decide *when to communication*.

Prior works on learning to communicate [34, 8] mainly address decision-making tasks (rather than for improving perception) under simple perceptual environments. In addition, these methods also do not consider bandwidth limitations: They learn to communicate across a fully-connected graph (i.e. all agents communicate with each other through broadcasts). Such methods cannot scale as the number of agents increases. Similarly, since all information is broadcast there is no decision of *when* to communicate conditioned on the need. An agent does not need to consume bandwidth when the local observation is sufficient for the prediction. When messages sent by other agents are degraded or irrelevant, communication thus could be detrimental to the perception task.

In this paper, we propose a learning-based communi-

*equal contribution

cation model for collaborative perception. We specifically view the problem as learning to construct the communication group (i.e. each agent decides what to transmit and which agent(s) to communicate with) and to decide when to communicate without explicit supervision for such decisions during training. In contrast to broadcast-based methods (e.g. TarMac [6]) and inspired by the general attention mechanisms, our method decouples the stages of communication and this allows for *asymmetric message and key sizes*, reducing the amount of transmitted data.

Our method can be generalized to several downstream vision tasks, including multi-agent collaborative semantic segmentation (dense prediction) and multi-agent 3D shape recognition (global prediction). Our model is able to be trained in an end-to-end manner with only supervision from downstream tasks (e.g. ground-truth masks for segmentation and class labels for image recognition) and without the need for explicit ground-truth communication labels.

We demonstrate across different tasks that our method can perform favorably against previous works on learning to communicate while using less bandwidth. We provide extensive analyses, including trade-offs between message and query sizes, the correlation between ground-truth key and predicted message, and visualization of the learned communication groups.

Our contributions are listed as follows:

- We address the under-explored area of collaborative perception, which is at the intersection of perception, multi-agent systems, and communication.
- We propose a unified framework that learns both how to construct communication groups and when to communicate. It does not require ground truth communication labels during training, and it can dynamically reduce bandwidth during inference.
- Our model can be generalized to several down-stream tasks, and we show through rigorous experimentation that it can perform better when compared with previous works investigating learned communication.
- We provide a collaborative multi-agent semantic segmentation dataset, AirSim-MAP, where each agent has its own depth, pose, RBG images, and semantic segmentation masks. This dataset allows researchers to further investigate solutions to multi-agent perception.

2. Related works

Learning to communicate. Communication is an essential component for effective collaboration, especially for multi-agent decision-making and perception tasks. Early works [35, 27] facilitated information flow and collaboration through pre-defined communication protocols. Similarly, auction-based methods [21, 29] for camera grouping use several assumptions (e.g., static cameras) and heuristic rules to decide the agents' communication. However, such rigid protocols do not evolve with dynamic environments and do not easily generalize to complex environments. Thus, in recent years, several multi-agent reinforcement learning (MARL) works have explored learn-able interactions between agents. For example, assuming full cooperation across agents, each agent in CommNet [34] broadcasts its hidden state to a shared communication channel so that other agents can decide their actions based on this integrated information. A similar scheme was proposed by Foerster *et al.* [8], where agents instead communicate via learned, discrete signals. To further leverage the interactions between agents, Battaglia *et al.* [2] and Hoshen [15] integrate kernel functions into CommNet. Additionally, several works have addressed communication through recurrent neural networks (RNN). For example, DIAL [7] uses an RNN to derive the individual Q-value of each agent based on its observation and the messages from other agents. BiCNet [28] connects all agents with a Bi-directional LSTM to integrate agent-specific information, and ATOC [18] additionally applies an attention unit to determine what to broadcast to the shared channel. Although substantial progress has been made by several MARL works, most experimental tasks are built on simplistic 2D-grid environments where each agent observes low-dimensional 2D images. As mentioned in Jain *et al.* [16], studying agents' interactions in simplistic environments does not permit study of the interplay between perception and communication.

Recent works have proposed to construct communication groups based on pre-defined rules [18, 17] or a unified communication network [33, 34, 15, 28, 33, 6]. With these techniques, bandwidth usage during communication increases as the number of agents scales up. While Who2com [23] uses a handshake communication to reduce the bandwidth usage, this model assumes all agents *always*

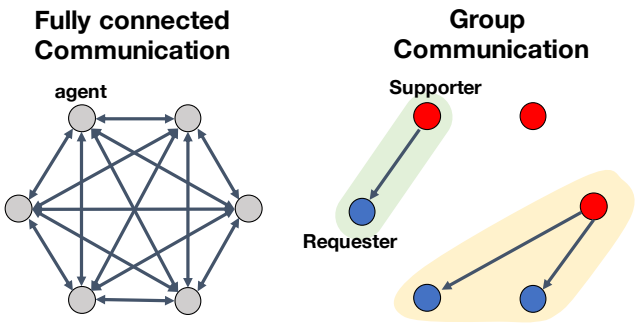
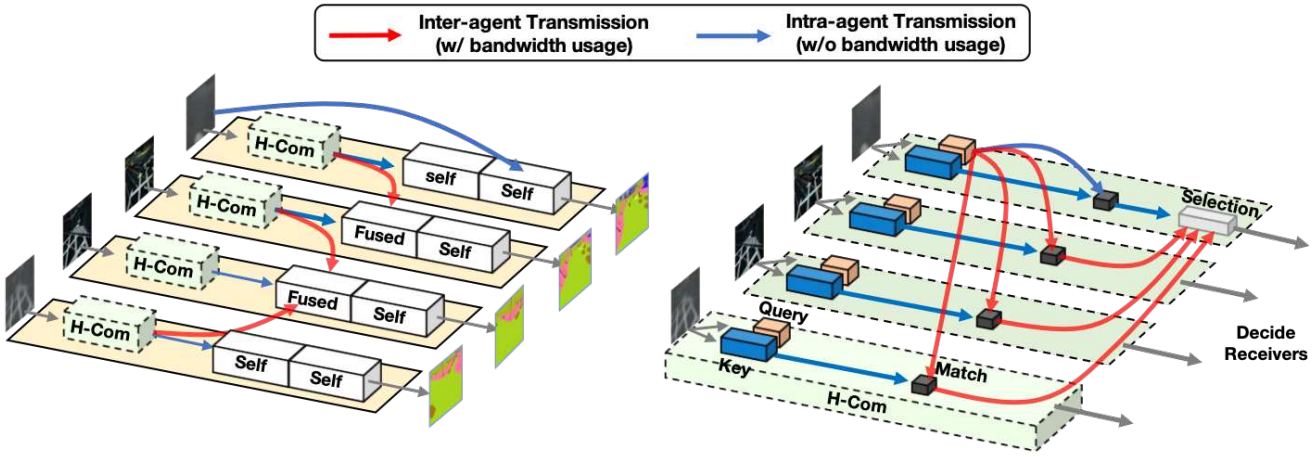


Figure 2: **Fully connected versus group communication.** Fully connected communication results in a large amount of bandwidth usage, growing on the order of $\mathcal{O}(N^2)$, where N represents the number of agents in a network. Group communication is able to prune irrelevant connections and can substantially reduce the overall network complexity.



(a) (b)

Figure 3: Our (a) **Multiple-Request Multiple-Support Model** and its (b) **Handshake-Communication (H-Com) Module**.

need to communicate with one of the other agents. This results in the waste of bandwidth consumption and cannot prevent the issue of detrimental messages. In contrast, our proposed framework alleviates these problems by learning to decide when to communicate and to create communication groups.

Attention mechanisms. Attention mechanisms have been widely used in recent learning-based models. In a nutshell, the attention mechanism can be viewed as a soft fusion scheme to weight different values based on a similarity between query and keys. A few noticeable and widely used attention mechanisms are *additive* [1], *scale dot-product* [36], and *general* [25]. One key finding of our work is that the *general* mechanism allows for asymmetric queries and keys, which makes it especially suitable for tasks with bandwidth considerations: An agent's transmitted query message can be smaller than its retained key, and hence its overall bandwidth consumption can be reduced.

3. Method

The goal of our proposed model is to address the multi-agent collaborative perception problem, where an agent is able to improve its perception by receiving information from other agents. In this paper, we are specifically interested in *learning to construct communication groups* and *learning when to communicate* in a bandwidth-limited way.

3.1. Problem Definition and Notation

We assume an environment consisting $\boldsymbol{N}$ agents with their own observations $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1,\ldots,\boldsymbol{N}}$. Among those agents, some of them are degraded $\tilde{\boldsymbol{X}} = \{\tilde{\boldsymbol{x}}_l\}_{l=1,\ldots,\boldsymbol{L}}$, and the set of degraded agents is a subset of all agents $\tilde{\boldsymbol{X}} \subset \boldsymbol{X}$. Each agent outputs the prediction of perception task $\tilde{\boldsymbol{Y}} = \{\tilde{\boldsymbol{y}}_n\}_{n=1,\ldots,\boldsymbol{N}}$ with the proposed communication mechanism. Note that each agent is a ***requester*** and ***supporter*** simultaneously. However, which agents are degraded is **unknown** in our problem setting.

3.2. Communication Groups Construction

As demonstrated in Figure 2, previous works on learning to communicate applied fully-connected communication for information exchange across agents. This framework results in a large amount of bandwidth usage and is difficult to scale up when the number of agents increases.

To reduce the network complexity and bandwidth usage, inspired by communication network protocol [20], we propose a two-step group construction procedure: we first apply the handshake communication [23] to determine the weights of connections, and we further prune the less important connections with an activation function.

To start constructing a communication group, we apply a three-stage handshake communication mechanism [23], which consists of three stages: request, match, and select. Agent i first compresses its local observations $\boldsymbol{x}_i$ into a compact query vector $\boldsymbol{\mu}_i$ and a key vector $\boldsymbol{\kappa}_i$:

$$\boldsymbol{\mu}_i = G_q(\boldsymbol{x}_i;\theta_q) \in \mathbb{R}^Q, \quad \boldsymbol{\kappa}_i = G_k(\boldsymbol{x}_i;\theta_k) \in \mathbb{R}^K, \tag{1}$$

where G_q is a query generator parameterized by θ_q and G_k is a key generator parameterized by θ_k. We further broadcast the query to all of other agents, and note that this only causes limited amount of bandwidth transmission as all queries are compact compared to the high-resolution images. To decide who to communicate with, we compute a matching score $\boldsymbol{m}_{i,j}$ between an agent i (as a requester) and an agent j (as a supporter),

$$\boldsymbol{m}_{i,j} = \Phi(\boldsymbol{\mu}_i, \boldsymbol{\kappa}_j), \quad \forall i \neq j, \tag{2}$$

where $\Phi(\cdot,\cdot)$ is a learned similarity function which measures the correlation between two vectors. The matching

score $m_{i,j}$ implies correlation between agent i and j, and intuitively this value also represents the amount of information the supporting agent j can provide for the requesting agent i.

However, the above method does not learn "when" to communicate, and it results in wasted bandwidth when an agent has sufficient information and the communication is not necessary. An ideal communication mechanism is to switch on transmission when the agent requires information from other agents to improve its perception ability, while it should also switch off the transmission when it has sufficient information for its own perception task. Toward this end, inspired by self-attention mechanism [4], we use the correlation between the key and query from the same agent to determine if the agent potentially requires more information and thus learn when to communicate,

$$m_{i,i} = \Phi(\mu_i, \kappa_i). \tag{3}$$

Note that $\hat{m}_{i,i} \approx 1$ represents that the agent has sufficient information and does not need communication for perception tasks.

In order to minimize bandwidth usage during transmission, we further propose an asymmetric message method, which compresses the query into an extremely low-dimensional vector (which is transmitted) while keeping a larger size for the key vector (which is not transmitted). Once extremely compact queries are passed to receiver agents, we use a scaled general attention [25, 36] to compute the correlation between agent i and agent j:

$$\Phi(\mu_i, \kappa_j) = \frac{\mu_i^T W_g \kappa_j}{\sqrt{K}}, \tag{4}$$

where $W_g \in \mathbb{R}^{Q \times K}$ is a learnable parameter to match the size of query and key, and Q and K are dimension of query and key respectively.

Based on the above self-attention and cross-attention mechanism across all queries and keys, we thus derive the matching matrix M:

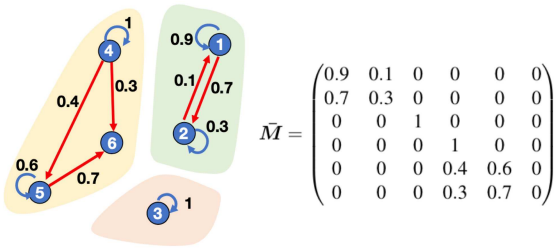
$$M = \sigma\left(\begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,N} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ m_{N,1} & m_{N,2} & \cdots & m_{N,N} \end{pmatrix}\right) \in \mathbb{R}^{N \times N}, \tag{5}$$

where $\sigma(\cdot)$ is a row-wise softmax function.

To construct the communication groups, we prune the less connections with an activation function:

$$\bar{M} = \Gamma(M; \delta), \tag{6}$$

where $\Gamma(\cdot; \delta)$ is an element-wise function, which zeros out the elements smaller than δ. (We set $\delta = \frac{1}{N}$ in our experiment.)



$$\bar{M} = \begin{pmatrix} 0.9 & 0.1 & 0 & 0 & 0 & 0 \\ 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 0 & 0 & 0.3 & 0.7 & 0 \end{pmatrix}$$

Figure 4: **An example of our constructed communication groups and the corresponding adjacency matrix.** Blue arrow indicates the intra-agent transmission without bandwidth consumption, and red arrow represents the inter-agent transmission with bandwidth consumption.

The derived matrix $\bar{M}$ can be regarded as the adjacency matrix of a directed graph, where the entries of the matrix indicate when to communicate and non-entries represent who to communicate with as demonstrated in Figure 4. Each row of the matrix represents how a receiver agent collects information from different supporting agents, and each column of the matrix is how one supporter sends its own information to different requesting agents.

As shown in Figure 3, once a requesting agent collects the information from its linked supporting agents, the requesting agent i integrates its local observation and the compressed visual feature maps from supporters based on the matching scores:

$$\hat{y}_i = D([f_i; f_i^{inf}]; \theta_d), \quad f_i^{int} = \sum_{\substack{j=1 \\ \bar{m}_{i,j} \neq 0}}^{N} \bar{m}_{i,j} f_j, \tag{7}$$

where D is an perception task decoder parameterized by θ_d, $\bar{m}_{i,j}$ is the element located in i-th row and j-th of the matrix $\bar{M}$, $f_i = E(x_i; \theta_e)$ is an feature map of agent i encoded by an image encoder E, $[\cdot;\cdot]$ is concatenation operation along channel dimension. It is worth noting that our perception task decoder is not limited for specific vision tasks, and we demonstrate our communication framework can be generalized to different visual tasks in our experiments.

3.3. Learning of Communication

Our learning strategy follows the centralized training and decentralized inference procedure [24]. Precisely, all agents are able to access all local observations of agents in the training stage, while each agent can only observe its own local observation in the inference stage. Our goal is to learn a bandwidth-efficient communication mechanism, so that in the inference stage, our proposed model is able to perform multi-agent collaborative perceptions in a bandwidth-limited and distributed manner.

We follow the aforementioned handshake communication to compute the matching matrix M, and we weight the agents' feature maps based on the matching matrix M and further integrate them as:

$$\boldsymbol{f}_i^{all} = \sum_{j=1}^{N} \tilde{\boldsymbol{m}}_{i,j} \boldsymbol{f}_j, \tag{8}$$

where $\tilde{\boldsymbol{m}}$ the element located in i-th row and j-th of the matrix $\boldsymbol{M}$. Note that in the above equation $\boldsymbol{m}_{i,j}\boldsymbol{f}_j$ represents who to communicate with, and $\boldsymbol{m}_{i,i}\boldsymbol{f}_i$ indicates when to communicate. Then, a client agent i combines its own feature map f_i and the integrated feature $\boldsymbol{f}_i^{all}$ to compute the prediction for downstream visual tasks,

$$\tilde{\boldsymbol{y}}_i = D([\boldsymbol{f}_i; \boldsymbol{f}_i^{all}]; \theta_d), \tag{9}$$

In order to train our model, we use the label for downstream tasks (*e.g.,* segmentation masks) as supervision, we compute the loss as:

$$\mathcal{L} = \mathcal{H}(\boldsymbol{y}_j, \tilde{\boldsymbol{y}}_j), \tag{10}$$

where $\mathcal{H}(\cdot,\cdot)$ can be the objective function for any downstream visual tasks (*e.g.* pixel-wise cross-entropy for segmentation tasks or cross-entropy for recognition tasks). We later update the weights of our model $\Theta = (\theta_k, \theta_q, \theta_e, \theta_d)$ using the above loss in an end-to-end manner.

4. Experiment

We evaluate the performance of our proposed framework on two distinct perception tasks: collaborative semantic segmentation and multi-view 3D shape recognition.

4.1. Experimental Cases and Datasets

4.1.1 Collaborative Semantic Segmentation

Our first task is collaborative 2D semantic segmentation of a 3D scene. Given observations (an RGB image, aligned dense depth map, and pose) from several mobile agents, the objective of this task is to produce an accurate 2D semantic segmentation mask for each agent.

Since current semantic segmentation datasets [9, 5, 26, 14] only provide RGB images and labels captured from the perspective of single agent, we thus use AirSim simulator [32] to collect our AirSim-MAP (Multi-Agent Perception) dataset. For this dataset, we fly a swarm of five to six drones with different yaw rates through a series of waypoints in the AirSim "CityEnviron" environment. We record pose, depth maps, and RGB images for each agent. Note that we also provide semantic segmentation masks for **all** drones.

We consider three experimental cases within this task. We refer to the agent attempting segmentation as the **requesting** agent, and all other agents as the **supporting** agents. Details for each case are listed as follows:

Single-Request Multiple-Support (SRMS) This first case examines the effectiveness of communication for a single requesting agent under the assumption that if an agent is degraded, then its original, non-degraded information will be present in one of the supporting agents. We include a total of five agents, of which only one is selected for possible degradation. We add noise to a random selection of 50% of this agent's frames, and we randomly replace one of the remaining agents with the *non-degraded* frame of the original agent. Note that only the segmentation mask of the original agent is used as supervision.

Multiple-Request Multiple-Support (MRMS) The second case considers a more challenging problem, where multiple agents can suffer degradation. Instead of requiring a single segmentation output, this case requires segmentation outputs for all agents, degraded and non-degraded. We follow the setup of the previous case, and we ensure that each of the several degraded requesting agents has a corresponding non-degraded image among its supporting agents.

Multiple-Request Multiple-Partial-Support (MRMPS) The third case removes the assumption that there exists a clean version of the degraded view among the supporting agents. Instead, the degraded agent must select the most informative view(s) from the other agents, and these views might have a variable degree of relevance. Specifically, as the drone group moves through the environment, the images from each drone periodically and partially overlap with those of other drones. Intuitively, the segmentation output of the requesting drone is only aided from the supporting drones that have overlapping views.

4.1.2 Multi-Agent 3D Shape Classification

In addition to the semantic segmentation task, we also consider a multi-agent 3D shape classification task. For this experimental case, we construct a multi-agent variant of the **ModelNet 40** dataset [37]. The original dataset contains 40 common object categories from ModelNet with 100 unique CAD models per category and 12 different views of each model. However, our variant adds a communication group structure to the original dataset. Specifically, we sample three sets of class-based image triplets. Each triplet corresponds to a randomly selected 3D object model and each triplet contains three randomly selected 2D views of its corresponding object model. To make this problem setting more challenging, we further degrade one image from each triplet. The objective of this task is to predict the corresponding object class for each agent by leveraging the information from all agents. Figure 6 shows an example of the dataset in one trial with 9 agents. This modified task is essentially a distributed version of the multi-view classifi-

Table 1: **Experimental results on Multiple-Request Multiple-Support and Multiple-Request Multiple-Partial-Support.** Note that we evaluate these models with the metric of mean intersection of union (mIoU) and use MBytes per frame (Mbpf) and the averaged number of links per agent for measuring bandwidth.

		Multiple-Request Multiple-Support				Multiple-Request Multiple-Partial-Support			
	Models	Bandwidth (Mbpf / # of links)	Noisy	Normal	Avg.	Bandwidth (Mbpf / # of links)	Noisy	Normal	Avg.
	AllNorm	-	57.85	57.74	57.80	-	47.9	48.37	48.14
Fully-Connect.	CatAll	2.5 / 5	29.07	51.83	40.45	2.0 / 4	26.86	45.27	36.07
	AuxAttend	2.5 / 5	33.69	56.27	44.98	2.0 / 4	26.97	51.03	39.00
	CommNet [34]	2.5 / 5	23.68	52.67	38.18	2.0 / 4	26.56	49.07	37.82
	TarMac [6]	2.5 / 5	51.09	56.74	53.92	2.0 / 4	29.78	**51.39**	40.59
Distri.	RandCom	0.5 / 1	21.22	52.74	36.98	0.5 / 1	24.13	45.19	34.66
	Who2com [23]	0.5 / 1	31.96	56.11	44.04	0.5 / 1	26.97	50.71	38.84
	Ours	**0.385 / 0.77**	**56.52**	**58.04**	**57.28**	**0.55** / 1.08	**30.38**	51.26	**40.82**
	OccDeg	-	30.06	56.31	43.19	-	25.2	46.74	35.97

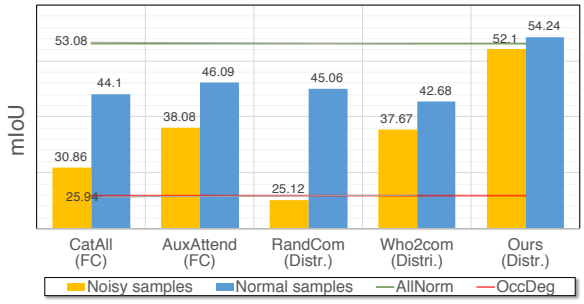


Figure 5: **Experimental results of Single-Request Multiple-Support.**

cation task [37].

4.2. Baselines and Evaluation Metrics

Here we consider several fully-connected (FC) and distributed communication (DistCom) models as our baselines. FC models fuse all of the agents' observations (either weighted or unweighted) whereas DistCom models only fuse a subset of those observations.

- *CatAll* **(FC)** is a naive FC model baseline which concatenates the encoded image features of all agents prior to subsequent network stages.
- *Auxiliary-View Attention (AuxAttend;**FC**)* uses an attention mechanism to weight auxiliary views from the supporting agents.
- *RandCom* **(DistCom)** is a naive distributed baseline which randomly selects one of other agents as a supporting agent.
- *Who2com* [23] **(DistCom)** excludes self-attention mechanism such that it always communicates with one of the supporting agents.
- *OccDeg and AllNorm* are baselines that employ no communication, i.e. each agent (view) independently computes the output for itself. For *OccDeg* the data is degraded similarly as before, while in *AllNorm* we use clean images for all views. These two serve as an upper and lower reference for comparison.

We also consider communication modules of *CommNet* [34], *VAIN* [15], and *TarMac* [6] as our baseline methods for all multiple-outputs tasks. For a fair comparison, we use ResNet18 [13] as the feature backbone for our and all mentioned baseline models. For the 3D recognition task, we also add MVCNN [37] as a baseline.

We evaluate the performance of all the models with mean IoU on the segmentation task and prediction accuracy on the 3D shape recognition task. In addition, we report Bandwidth of all FC and DistCom models in Megabyte per frame (MBpf). To obtain MBpf, We add the size of the feature vectors which need to be transmitted to the requesters and size of keys broadcast to all supporters and multiply by the number of bytes required for storage.

4.3. Quantitative results

Single-Request Multiple-Support (SRMS) The goal of this case is to examine if our model is able to learn when to communicate and learn who to communicate with for a single requesting agent. Figure 5 shows the performance of our proposed model and several baseline models. Although most fully-connected methods can improve the prediction mIoU compared with *NoCom*, they need to propagate all information in a fully-connected manner and thus require high bandwidth consumption. In contrast, our model reports higher prediction accuracy yet smaller bandwidth usage (Who2com [23]: 2 MBpf; ours: 0.98 MBpf). Another observation is that our model is able to further improve compared with *Who2com* [23]. This demonstrates the benefit of learning when to communicate, which reduces the waste of bandwidth and prevent detrimental message when the requesting agent has sufficient information and communication is not required.

Table 2: **Experimental results on Multi-agent 3D Shape recognition.** We report the accuracy of the degraded split, and all methods perform similar results for the normal split (≈ 83%).

	OccDeg	AllNorm	RandCom	CatAll	MVCNN [37]	CommNet [34]	VAIN [15]	TarMac [6]	Ours
Degraded Split Accuracy (%)	55.02	83.66	54.28	73.82	31.80	71.52	75.09	78.73	**80.72**
Bandwidth (links/MBpf)	-	-	0.11 / 0.89	1 / 8	1 / 8	1 / 8	1 / 8	1 / 8	**0.176 / 1.32**

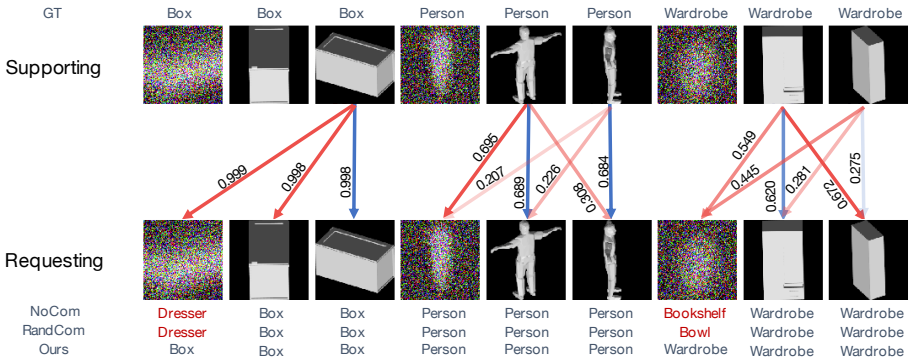


Figure 6: **Bipartite communication graph between supporting and requesting agents.** During the query phase, each requesting agent sends a low-dimensional query vector to all other agents (including itself) to establish communication. Then during the transmission phase, supporting agents transmit their high-dimensional feature representations. We visualize the flow of data during the transmission phase, where blue and red arrows refer to internal and external communication, respectively. More prominent colors and larger numerical values indicate stronger feature weightings, whereas missing arrows represent the pruned links in the communication graph. Note that these images are ordered for visualization purposes; the actual dataset is unordered, and each agent observes a random class with a random chance of degradation.

Multiple-Request Multiple-Support (MRMS) In this case, we further address a more challenging problem, where multiple agents suffer degradation. Each agent should (1) identify when it needs to communicate, (2) decide who to communicate with when it needs to, and (3) avoid the selection of noisy views from the supporting agents. We list the experiment results in the Table 1. It can be seen that, when the requesting agents cannot prevent the selection of noisy supporting agents, both *CatAll* and *RandCom* perform even worse than *NoCom*. This verifies our intuition that the information from the supporting agents is not always beneficial for the requesting agents, and selection of incorrect information may even hinder the prediction of the requesters.

With the use of attention mechanisms for weighting the feature maps from the supporting agents, both *AuxAttend* and *Who2com* [23] are able to prevent incorrect views from deteriorating performance and thus improve with respect to *NoCom*, *CatAll*, and *RandCom*. However, without learning when to communicate, those models are forced to always request information from at least one supporting agent resulting in both poorer performance and unnecessary bandwidth usage.

In addition to the above baseline methods, we also consider CommNet [34] and TarMac [6]. Even though CommNet integrates the information from other agents by using an average pooling mechanism, it does not improve the prediction of either degraded or non-degraded requesting agents because it indiscriminately incorporates all views.

On the other hand, TarMac [6] is able to provide better results compared with the baseline models. However, TarMac uses one-way communication and results in large bandwidth usage which presents difficulty in the real scenario. On the contrary, our model is not only able to outperform it on both degraded and non-degraded samples, but also consumes less bandwidth by using our asymmetric query mechanism and pruning redundant connections within the network with the activation function.

Multiple-Request Multiple-Partial-Support (MRMPS) In this case, there is less chance to have completely overlapped observations between any two agents. This presents an inherent difficulty in the perception task because only incomplete information is available for the prediction. As shown in the right part of Table 1, the performance improvement margin of all FC and DistCom models is smaller with respect to *NoCom*, in comparison to more significant improvement observed in the previous scenario.

Nonetheless, we observe that all methods exhibit a similar trend as the previous scenario. Our model is still able to maintain a similar prediction accuracy as fully-connected models, while we only use one-fourth bandwidth for com-

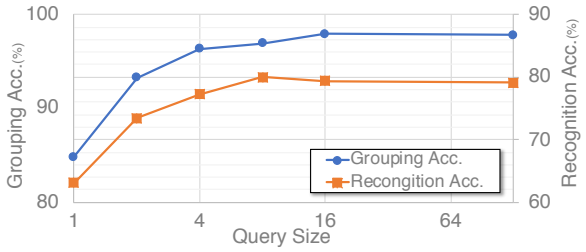


Figure 7: **Ablation study on varying query size.**

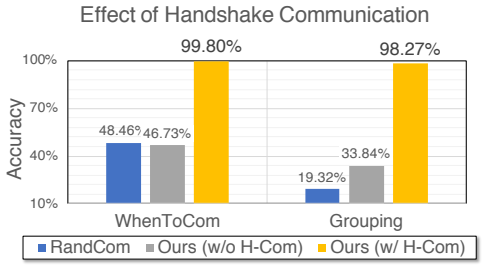


Figure 8: **Effect of our proposed H-Com.** Handshake communication significantly improves the communication accuracy.

munication across agents. This demonstrates the superior bandwidth-efficiency of our model.

Multi-agent 3D shape Recognition In order to demonstrate the generalization of our model, here we apply our model to the task of multi-agent 3D shape classification. Table 2 provides the quantitative evaluation on this task using our proposed model and other baselines, including *VAIN* [15], *CommNet* [34], and *TarMac* [6]. Our model is able to perform competitively compared with *TarMac* [6] with only approximately one-eighth bandwidth usage. We also provide qualitative results in Figure 6 to demonstrate the effectiveness of our model, which allows agents to communicate with the correct and informative agents.

4.4. Analyses

To investigate the source of our model's improvement over the baselines, we computed two selection accuracy metrics on the SRMS dataset, *WhenToCom* and *Grouping*. *WhenToCom* accuracy measures how often communication between a requester and a supporter(s) is established and when it is needed; and *Grouping* measures how often the correct group is created when there is indeed communication. We also comment on the trade-off between bandwidth and performance of communication by conducting a controlled experiment on the size of query and key on the 3D recognition dataset.

Effect of handshake communication As demonstrated in Figure 8, we conduct an ablation study on the proposed handshake communication. In the *Ours (w/o H-Com)* model, we remove the handshake communication module, so that each agent only uses its local observations to compute both (1) the communication score and (2) its communication group.

We additionally provide the result of *RandCom*. We observed that our model with the proposed handshake communication offers a significant improvement over both *RandCom* and our model without handshake communication. This finding demonstrates the necessity of communication for deciding when to communicate and who to communicate with. That is, an agent without communication cannot decide what information it needs and which supporter has the relevant information to help better perform on perception tasks.

Figure 6 visualizes three examples from the 3D shape recognition task. Each agent clearly knows when communication is needed based on information provided by the supporters and its own observation. For example, in the first three box examples, the degraded agent on the left knows to select an informative view from the other agents; the non-degraded agent in the middle decides to select a more informative view even though it possesses sufficient information; and the third agent decides that communication is not needed because it has the most informative view among all. It is worth mentioning that all 9 views are provided to every agent and the agent needs to identify informative views and detrimental views based on the matching scores.

Query and key size We further analyze the effect of query and key size on *Grouping* accuracy and classification accuracy on the 3D shape classification task. We vary the query size from 1 to 128 with a fixed key size of 16 as shown in Figure 7. We observe that both selection and classification accuracy improve as the message size increases. Our model can perform favorably with a message size of 4. The same trend is also observed for key sizes. Most noticeably, we find that there exists asymmetry in query-key size. While the selection accuracy saturates at 16-dimensional query, selection accuracy consistently improves with increasing key size until 1024-dimensional key. Our model exploits this asymmetry to save bandwidth in communication while maintaining high performance.

4.5. Conclusion

In this paper, we proposed a general bandwidth-efficient communication framework for collaborative perception. Our framework learns both how to construct communication groups and when to communicate. This framework can be generalized to several down-stream tasks including (but not limited to) multi-agent semantic segmentation and multi-agent 3D shape recognition. We demonstrated superior performance with lower bandwidth requirements across all compared methods.

5. Acknowledgement

This work was supported by ONR grant N00014-18-1-2829.

References

[1] Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 3

[2] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018. 1

[4] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016. 4

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 5

[6] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Michael Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019. 2, 6, 7, 8

[7] Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[8] Jakob Foerster, Yannis M Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 1, 2

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5

[10] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 1

[11] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3838, 2019. 1

[12] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[14] Simon Hecker, Dengxin Dai, and Luc Van Gool. End-to-end learning of driving models with surround-view cameras and route planners. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–453, 2018. 5

[15] Yedid Hoshen. Vain: Attentional multi-agent predictive modeling. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2, 6, 7, 8

[16] Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari, Svetlana Lazebnik, Ali Farhadi, Alexander G Schwing, and Aniruddha Kembhavi. Two body problem: Collaborative visual task completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[17] Jiechuan Jiang, Chen Dun, and Zongqing Lu. Graph convolutional reinforcement learning for multi-agent cooperation. *arXiv preprint arXiv:1810.09202*, 2018. 2

[18] Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2

[19] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9404–9413, 2019. 1

[20] James F Kurose. *Computer networking: A top-down approach featuring the internet, 3/E*. Pearson Education India, 2005. 3

[21] Yiming Li, Bir Bhanu, and Wei Lin. Auction protocol for camera active control. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 4325–4328. IEEE, 2010. 2

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1

[23] Yen-Cheng Liu, Junjiao Tian, Chih-Yao Ma, Nathaniel Glaser, Chia-Wen Kuo, and Zsolt Kira. Who2com: Collaborative perception via learnable handshake communication. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 2, 3, 6, 7

[24] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 4

[25] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015. 3, 4

[26] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 5

[27] Francisco S Melo, Matthijs TJ Spaan, and Stefan J Witwicki. Querypomdp: Pomdp-based communication in multiagent systems. In *Proceedings of the 9th European conference on Multi-Agent Systems*, 2011. 2

[28] Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017. 2

[29] Faisal Qureshi and Demetri Terzopoulos. Smart camera networks in virtual reality. *Proceedings of the IEEE*, 96(10):1640–1656, 2008. 2

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 1

[31] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[32] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017. 5

[33] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*, 2018. 2

[34] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 1, 2, 6, 7, 8

[35] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1993. 2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 3, 4

[37] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. 5, 6, 7

[38] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1