Self-Supervised Collaborative Scene Completion: Towards Task-Agnostic Multi-Robot Perception

Anonymous Author(s)
Affiliation
Address
email

Abstract: Collaborative perception learns how to share information among multiple robots to perceive the environment better than individually done. Past research on this has been task-specific, such as object detection and semantic segmentation. However, this may lead to different information sharing for different tasks, which could hinder the large-scale deployment of collaborative perception. We propose the first task-agnostic collaborative perception paradigm that learns a single collaboration module in a self-supervised manner for different downstream tasks. This is done by a novel task termed *collaborative scene completion*, where each individual robot learns to effectively share information for reconstructing a complete scene viewed by all robots. Moreover, we propose a spatial-temporal-aware autoencoder that amortizes over time the communication cost by spatial sub-sampling and temporal mixing when sharing information. We conduct extensive experiments with various baselines to validate our method's effectiveness on scene completion and collaborative perception tasks in autonomous driving.

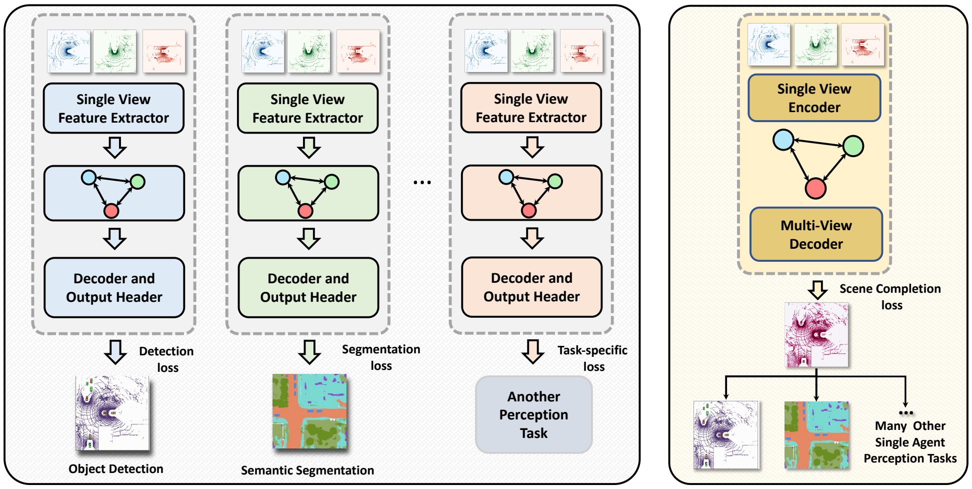
Keywords: Multi-Robot Perception, Scene Completion, Representation Learning

1 Introduction

Single robot perception has been widely studied on different tasks, such as object detection [1] and semantic segmentation [2]. However, it suffers from various challenges such as occlusion and sparsity in raw observations. Collaborative perception is promising to alleviate those issues. It provides more environment observations from different perspectives by information sharing to improve perception performance and robustness. Amongst different collaboration strategies, feature-level collaboration [3, 4, 5] transmits the intermediate representations generated by deep neural networks (DNNs) of each robot. Since these intermediate features are easy to compress and can preserve contextual information of the scene, feature-level collaboration demonstrates better performance-bandwidth trade-off compared to raw-data-level and output-level collaboration [6, 7].

However, existing feature-level collaboration methods [8, 4, 3] are fully supervised by task-specific losses to learn the entire model including a feature extractor, a collaboration module, and a decoder, as shown in Fig. 1 (a). Such a task-specific framework requires re-training the whole model for different perception tasks. Besides, existing collaborative perception requires training data recordings to be synchronized among all robots in time, which is more demanding than data collection in single-robot perception. How can we design a collaborative perception framework that is (1) independent from downstream tasks and (2) trainable from asynchronous dataset?

To answer this question, we propose a novel self-supervised learning task termed *collaborative scene completion* (CSC). It enables multiple robots to collaboratively use an autoencoder to reconstruct a complete scene based on latent features shared between each other. The completed scene could then be fed into various downstream tasks *without additional training*, as shown in Fig. 1 (b). This allows us to decouple the collaboration training from downstream task learning. Moreover, it seamlessly



(a) Task-specific framework

(b) Task-agnostic framework

Figure 1: **Task-specific vs Task-agnostic collaboration.** Previous methods are all task-specific. As shown in (a), such a paradigm learns different models with different losses for each task. However, in our paradigm in (b), the model learns to directly reconstruct the complete multi-robot scene based on each robot's message, which is independent from yet still usable by all downstream tasks.

supports both synchronous and asynchronous training datasets by different learning objectives: complete scene reconstruction if synchronous, and individual view reconstruction if asynchronous.

Yet naive autoencoders are not designed to balance between scene reconstruction performance and communication volume, which is an established criteria to evaluate collaborative perception. To address this challenge, we further design a spatio-temporal-aware autoencoder (STAR), inspired by the recent masked autoencoders (MAE) [9]. It reconstructs a scene using a spatial-temporal mixture of patch tokens: some tokens encoded from randomly sub-sampled patches in the current frame, others cached from the past. The sampling ensures that all patches in the mixture could jointly cover the whole spatial region while being self-disjoint. This allows each robot to only transmit the sub-sampled tokens in the current frame instead of the entire latent feature maps, leading to orders of magnitude lower of communication bandwidth than prior works. *Our key insight behind such an amortized communication cost* is that features of many patches (e.g., static or nearly static) do not need to be shared in every frame.

In summary, our main contributions are threefold:

- We propose a brand-new task-agnostic collaborative perception framework based on scene completion, which decouples the collaboration learning from downstream tasks. Also, our method does not need synchronous multi-robot perception training data.
- We develop a novel spatio-temporal-aware autoencoder (STAR) that reconstructs scenes based on temporally mixed information. It amortizes the spatial communication volume over time to improve the performance-bandwidth trade-off.
- We conduct extensive experiments to verify our method's effectiveness in autonomous driving settings, in terms of scene completion and downstream perception tasks.

2 Related Works

Collaborative perception. Collaborative perception has been proposed to improve the flexibility, resilience, and efficiency of the individual perception with limited field of view. With recent advances in deep learning, researchers have developed feature-level collaborative perception in which intermediate representations produced by deep neural networks (DNNs) from multiple viewpoints are propagated in a team of robots, *e.g.*, a swarm of drones [8, 3] or a group of vehicles [4, 6]. Existing works commonly consider a specific downstream task, and use the corresponding loss function to learn a collaboration module, *e.g.*, graph neural network (GNN) [4, 3], attention network [8, 10], and convolutional neural network [5]. Several downstream tasks have been investigated in collaborative scenarios, such as object detection [4], semantic segmentation [8], and depth estimation [3]. In this work, we develop a task-agnostic collaborative perception paradigm based on a geometrical task of collaborative scene completion, where multiple robots need to reconstruct full view based on the shared intermediate features from the collaborators.

Scene completion. Autonomous navigation [11] requires robots to understand geometry and semantics of 3D scenes. However, vision sensors solely capture partial observations because of limited field of view as well as sparse sensing, leading to an incomplete spatial representation. To solve this, scene completion (SC) has been proposed to infer the complete 3D scene geometry given sparse 2D/3D observations [12, 13, 14]. Following scene completion, semantic scene completion (SSC) has been introduced to jointly estimate both geometry and semantic information based on partial observation [2, 15, 16]. On one hand, single robot scene completion is able to rely on semantic prior knowledge to complete the partially-observed objects. On the other hand, it is unrealistic to hallucinate the totally invisible objects. Different from single robot scene completion, the proposed CSC task does not consider invisible structure for the robot team to avoid unreasonable hallucination.

Masked autoencoders. Self-supervised representation learning aims to provide powerful features without the need for massive annotated datasets, thereby receiving extensive attention [17]. Masked autoencoder (MAE) achieves state-of-the-art self-supervised representation learning performance with a simple reconstruction objective [9]. Specifically, MAE employs an asymmetric architecture with a large encoder that only process unmasked patches and a lightweight decoder that reconstructs the masked patches from the latent representation and mask tokens, which speeds up pre-training. Recent works extend MAE into multimodal representation learning [18, 19], and video representation learning [20, 21]. Meanwhile, there are several attempts to utilize MAE on downstream tasks such as 2D image completion [22].

3 Collaborative Scene Completion: Formulation and Evaluation

We provide an overview of collaborative scene completion (CSC) in this section: first, we define the multi-robot scene completion problem, which is to reconstruction the full view based on each robot's incomplete observation. Then, we introduce how each individual observation is encoded and communicated between robots; our method only communicates intermediate features. Next, we will present how features are decoded in each robot and how the loss is computed. Finally, we will talk about evaluation metrics for the proposed task.

Problem definition. We consider that N robots present in the same geographical location are simultaneously perceiving the 3D environment such as a fleet of autonomous vehicles located at a certain crossroad. In order to understand the surrounding environment better, these robots communicate with each other about their observations. Each robot is equipped with a 3D sensor such as a LiDAR to generate a binary occupany grid map $\mathbf{M}_i \in \{0,1\}^{H \times W \times C}$ defined in its local coordinate, where H, W, and C respectively denote the length, width, and height resolution.

Feature extraction. We aim to achieve intermediate collaboration with better performance-bandwidth trade-off [5]. Each robot encodes its individual observation into a feature map denoted by $\mathbf{F}_i = \mathbf{\Theta}(\mathbf{M}_i)$, where $\mathbf{\Theta}$ denotes a feature extractor. Now $\mathbf{F}_i \in \mathbb{R}^{\bar{H} \times \bar{W} \times \bar{C}}$ has lower spatial resolution $\bar{H} \times \bar{W}$, while keeping a higher feature dimension $\bar{C}$ compared to the original feature map $H \times W \times C$. Then, each robot will broadcast $\mathbf{F}_i$ to its peers as well as its pose $\boldsymbol{\xi}_i \in \mathfrak{se}(3)$ defined in the global coordinate.

Feature decoding. The robot i receives the messages from the neighboring robots $\{\mathbf{F}_j, \boldsymbol{\xi}_j\}_{j \neq i}$, and then uses a decoder and a pose-aware aggregator (collectively denoted by $\mathbf{\Phi}$ for simplicity) to aggregate the messages, and output a completed occupancy grid map $\hat{\mathbf{Y}}_i = \mathbf{\Phi}(\mathbf{F}_i, \boldsymbol{\xi}_i, \{\mathbf{F}_j, \boldsymbol{\xi}_j\}_{j \neq i})$, where $\hat{\mathbf{Y}}_i$ has the same dimension and describe the same spatial range as $\mathbf{M}_i$ yet is a more comprehensive spatial representation for the scene.

Loss function. We treat such kind of scene completion task as a binary classification problem and use cross entropy loss to train a neural network composed of $\mathbf{\Theta}$ and $\mathbf{\Phi}$. Specifically, the ground-truth $\mathbf{Y}_i \in \{0,1\}^{H\times W\times C}$ defined in the coordinate of robot i represents a multi-view occupancy voxel grid with two classes, *i.e.,* free and occupied. Therefore, the loss can be computed by:

$$\mathcal{L} = -\sum_{i=0}^{N-1}\sum_{k=0}^{L-1}\sum_{c=0}^{1} y_{i,k,c} log(\frac{e^{\hat{y}_{i,k,c}}}{\sum_c e^{\hat{y}_{i,k,c}}}), \tag{1}$$

where i is the robot index, k is the voxel index, L is the total number of the voxel ($L = H \times W \times C$), c is number of class (2 in our case), $\hat{y}_{i,k,c}$ is the predicted logits for the k-th voxel belonging to class c, $y_{i,k,c}$ is the k-th element of $\mathbf{Y}_i$ and is a one-hot vector ($y_{i,k,c} = 1$ if voxel k of robot i belongs to class c). Here we show the training using synchronous multi-robot data yet this task can also be supervised by individual view reconstruction on asynchronous data (will be shown in Section 4).

Evaluation metrics. We follow the evaluation protocol in single-robot scene completion [14, 23] which uses the voxel-level intersection over union (IoU) between predicted voxel labels $\hat{\mathbf{Y}}_i$ and ground truth labels $\mathbf{Y}_i$ for each robot. Note that only non-empty voxels are evaluated.

4 STAR: Spatio-Temporal-Aware Autoencoder

In addition to the collaborative scene completion task, we also propose a novel architecture called **S**aptio-**T**emporal-**A**ware autoencode**R** (STAR) to tackle this problem. We will present our key design motivation, detailed modules, training and inference procedures, respectively.

4.1 Design desiderata

We build our brand-new architecture based on a few high-level desiderata explained as follows.

Partially broadcasting. Inspired by the idea of ”masking” in MAE [9], we employ a similar asymmetric design as MAE yet with different purposes: MAE is to design a nontrivial self-supervisory task for pre-training via randomly masking, while the goal of STAR is to reduce the communication volume in multi-robot systems via partial broadcasting. More specifically, STAR deploys an encoder at the *sender robot* to map the entire observation to an intermediate feature representation which is only selectively transmitted to lower the bandwidth. Meanwhile, STAR deploys a decoder at the *receiver robot* that reconstructs the original observation from the received partial representation.

Temporal amortization. Directly applying random masking of partial observations does not work for our case. Unlike MAE which is mainly for object-level image recognition, we aim at large-scale dynamic scene modeling. Once objects are completely masked during encoding, it is not possible for the decoder to hallucinate the corresponding objects without such kind of knowledge. To solve this problem, we propose to exploit *historical tokens* to replace *mask tokens* during decoding. Essentially, we amortize the communication cost over the temporal domain by spatial sub-sampling and temporal mixing, and such operation ensures that all patches in the mixture could jointly cover the whole spatial region.

Synchronization-free training. Traditional collaborative perception approaches consider a synchronization training strategy which requires synchronous (potentially with a small temporal latency) multi-robot recordings, in order to train a feature-space collaboration strategy with task-specific loss functions [4, 5]. In contrast, we try to realize synchronization-free training which doesn’t require perception data being simultaneously captured by multiple robots. Specifically, we use single-view observation as the supervision, and aggregate the reconstructed results of each view for the final output.

4.2 Architecture

We consider a homogeneous set of robots deploying the same neural network following [4, 5]. Each robot serves as both message sender and receiver during collaboration, and each robot is equipped

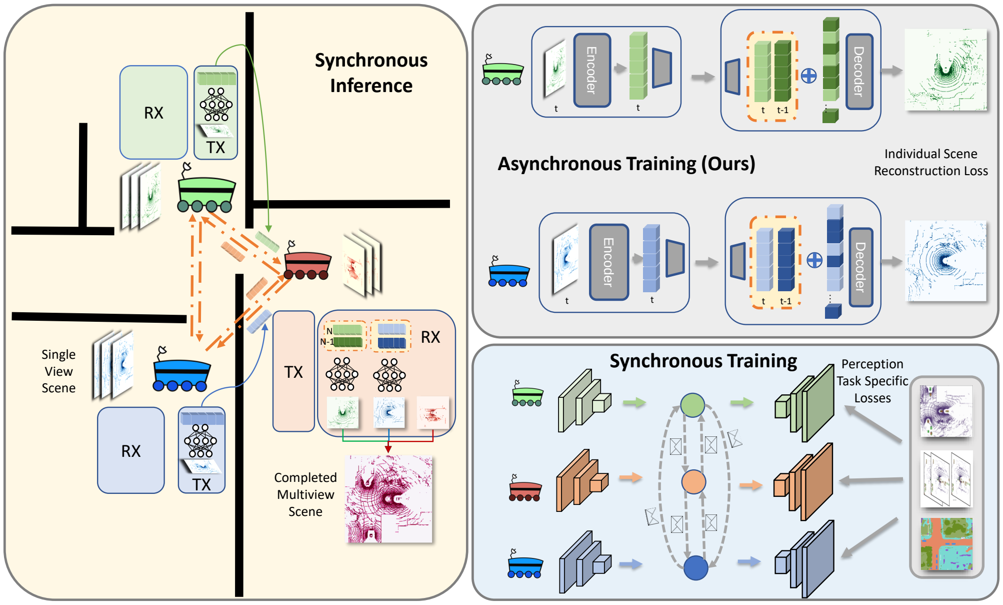

Synchronous Inference
RX
TX
RX
TX
N
N-1
RX
TX
Single View Scene
Completed Multiview Scene
Encoder
t
t
t-1
Decoder
Asynchronous Training (Ours)
Individual Scene Reconstruction Loss
Synchronous Training
Perception Task Specific Losses

Figure 2: **Asynchronous training.** Illustrated in the top right, the asynchronous training trains the model to reconstruct the scene observations and does not require communication between robots, as opposed to the **synchronous training** shown in the bottom right where robots communicate intermediate representations and optimized with respect to each specific task loss. The **synchronous inference** is illustrated on the left. The *sender robots* transmit encoded representations from their encoders (TX) to the *receiver robot*'s decoders (RX). The *receiver robots* use a mixture of spatio-temporal tokens to complete the multi-view scene observation.

with our model composed of an encoder for observation abstraction, and a decoder for view reconstruction. Since our model processes a spatial-temporal mixture of patch tokens, we call it spatio-temporal-aware autoencoder (STAR).

STAR encoder. Different from MAE [9], the STAR encoder uses a vision transformer (ViT) [24] backbone which operates on all patches yet only sends out a subset. Specifically, the entire grid map for robot i at time t denoted by $\mathbf{M}_{i,t}$ is divided into multiple patches, and each patch is encoded with a linear projection with additional positional embedding (following ViT [24]), and then processed using a series of Transformer blocks to generate the final message $\mathbf{F}_{i,t}$. Note that we adopt a complementary transmission strategy in the temporal domain regarding the patch index (*i.e.*, the observed spatial locations), in order to avoid the loss of information for the dynamic scene.

STAR decoder. Different from MAE using mask tokens to replace the missed patch embeddings, robot i as a receiver aggregates the historic tokens $\mathbf{F}_{j,t-1}$ together with the current tokens $\mathbf{F}_{j,t}$ from robot j, which form a full observation towards the entire spatial range. Temporal embeddings are added to the tokens from the respective timestamps to enhance the temporal awareness, before feeding these tokens into a series of Transformer blocks to obtain the robot j' reconstruction $\hat{\mathbf{M}}_{j,t}$. Note that here we use a two-timestamp case as an example for simplicity reason. The STAR decoder is also able to process more historical timestamps. After decoding all robots' views denoted by $\{\hat{\mathbf{M}}_{j,t}\}_{j\neq i}$, the ultimate prediction of the complete view could be created by $\hat{\mathbf{Y}}_i = \Gamma(\mathbf{M}_{i,t}, \{\hat{\mathbf{M}}_{j,t}\}_{j\neq i})$, where Γ indicates aggregation with coordinate synchronization.

4.3 Asynchronous training

Training phase. The model is trained with single view ground-truth $\mathbf{M}_i$, and adopt cross-entropy loss during training:

$$\mathcal{L} = -\sum_{i=0}^{N-1}\sum_{k=0}^{L-1}\sum_{c=0}^{1} m_{i,k,c} log(\frac{e^{\hat{m}_{i,k,c}}}{\sum_c e^{\hat{m}_{i,k,c}}}), \tag{2}$$

where i is the robot index, k is the voxel index, L is the total number of the voxels, $c = 2$ is number of class, $m_{i,k,c}$ denotes the k-th element of $\mathbf{M}_i$ and is a one-hot vector same as $y_{i,k,c}$ in Eq. 1, $\hat{m}_{i,k,c}$ is the prediction for the k-th voxel belonging to class c. Note that the training loss is calculated in a voxel-wise way, with respect to the self-supervision signal from each robot's own single view observation. This design decouples our training phase from communication with other robots: the model on each robot does not require synchronous observations from neighbor robots in the training phase, making the training asynchronous (asynchronous training in Fig. 2). This is greatly different from the training framework in previous collaborative perception works such as [5] where robots will communicate and aggregate the features broadcasted by their neighbors , through which the collaborative perception is achieved (synchronous training in Fig. 2). Our training framework can relax the need of carefully-collected and hard-to-annotate multi-robot dataset, and can exploit the large amount of single-robot data to learn powerful as well as compact feature representations.

Inference phase. During inference, each robot uses the same model equipped with the STAR encoder and decoder. The sender robots' encoders will encode and broadcast a subset of their current timestamp's observation. Then, the decoders on the receiver side will leverage the transmitted intermediate representation along with the pose information to reconstruct the corresponding view, optionally with historical features as described above. Then, the receivers use corresponding pose information to transform the single observations into a multi-view completed scene. We illustrated the pipeline on the left side of Fig. 2.

5 Experimental Results

5.1 Experimental setup

Dataset. We conduct our experiments on the V2X-Sim Dataset [5]. It is a large-scale dataset that simulates urban multi-vehicle driving scenes with CARLA [25]. We use 8000 scenes as training set and 1000 scenes for testing. The dataset is sampled at 5 Hz. We preprocess the voxels grids with range $[-32m, 32m]$ in x and y axis, and $[-3m, 2m]$ in z axis. Finally we can get the voxel grids with a spatial resolution of $256 \times 256 \times 13$.

Implementation details. For scene completion, we use a modified FaFNet [26] as the convolutional neural network (CNN) baseline method, where we substitute the detection head with a classification head that outputs the logits for binary classification. A 12-block ViT encoder with hidden dimension 768 is used for the STAR encoder. Then an MLP is used to compress the intermediate representations to 32 dimension and feed them to the decoder, where they are projected back to 512 dimension and sent to a 8-layer transformer decoder. A FaFNet [26] is used for single-robot object detection. A UNet [27] serves the same purpose for the semantic segmentation task. Note that all the perception models take the three-dimensional voxel grid as input and output results in bird's eye view (BEV), *i.e.*, bounding boxes and semantic labels. Our models are all trained on the single-view data.

Evaluation metrics. For the scene completion task, we measure the completion quality using the intersection-over-union (IoU) at three different scales by down-sampling the voxels accordingly. For the perception task, we report the average precision (AP) at threshold 0.5 and 0.7 for vehicle detection, and IoU for vehicle category and the overall mIoU for semantic segmentation.

5.2 Quantitative results on scene completion

We present quantitative results of multi-robot scene completion task in Table 1, including the IoU at different scales for the CNN baseline and STAR with different timestamps and spatial resolutions, as well as the corresponding communication bandwidth.

Spatial resolution. We test three resolutions: 32×32, 16×16 and 8×8. We can see that in general a higher spatial resolution leads to a better completion quality: the spatial resolution 32×32 which has a patch size of 8 achieves the best performance.

Timestamp	IoU scale 1:1			IoU scale 1:2			IoU scale 1:4			Communication Bandwidth		
	32x32	16x16	8x8	32x32	16x16	8x8	32x32	16x16	8x8	32x32	16x16	8x8
STAR TS1	**52.20**	**49.61**	48.47	**70.79**	**65.19**	62.65	**77.98**	**72.27**	**69.38**	1.3MB/s	320.0KB/s	80.0KB/s
STAR TS2	50.66	49.54	**48.72**	67.38	64.67	**62.66**	74.01	71.20	69.08	640.0KB/s	160.0KB/s	40.0KB/s
STAR TS3	50.97	49.51	48.41	68.37	64.53	61.42	75.41	71.28	67.38	427.0KB/s	106.7KB/s	26.7KB/s
STAR TS4	49.36	48.83	48.03	64.87	63.32	61.16	71.45	69.87	67.25	**320.0KB/s**	**80.0 KB/s**	**20.0KB/s**
CNN backbone	55.37			77.17			83.51			155.0MB/s		

Table 1: **Quantitative results on scene completion and the communication bandwidth.** Results across different spatial resolutions and timestamps (TS) are presented.

Timestamp	All	Partial
1	**65.19**	-
2	**64.68**	61.36
3	**64.53**	63.77

(a) **Patches to encode.** All means encoding all the patches for each timestamp before masking and transmission. Encoding partial means masking is done before encoding the patches.

Timestamp	Decode Multi	Decode Single
1	**65.19**	-
2	**64.68**	52.07
3	**64.53**	51.97

(b) **Timestamps to decode.** Performance for timestamp 1, 2 and 3 are reported. For timestamp 1, decoding single timestamp is equivalent to decoding multi-timestamp.

Timestamp	Temporal Emb.	
	w/	w/o
2	**64.68**	64.29
3	**64.53**	61.83

(c) **Temporal embedding.** W/ means temporal embeddings are added to the patches in the decoder. W/o means not.

Strategy	Timestamp		
	2(50%)	3(66%)	4(75%)
random	50.88	52.36	52.14
complementary	**64.45**	**64.20**	**63.27**

(d) **Masking strategy.** Timestamp 2(50%) means that the random masking method removes 50% of the patches for each timestamp, which is equivalent to the ratio of complementary masking.

Table 2: **Ablation studies.** The performance is reported in IoU 1:2 for the spatial resolution 16x16. The observation under other settings are consistent.

Timestamps. Our method allows the multi-robot system to amortize the spatial communication bandwidth over the temporal domain. We can see from the Table 1 that from timestamps 1 to 4, the performance only varies slightly while largely reducing the bandwidth.

Bandwidth. Bandwidth is calculated to reflect the required data volume for communication per second. A trade-off between performance and communication is clear. We can see that though the CNN baseline can perform better than ViT baselines, it requires much higher bandwidth introduced by the skip connection in UNet. STAR requires much lower bandwidth, and a finer-grained spatial resolution with better performance requires a higher bandwidth.

5.3 Ablation studies on scene completion

We conduct several ablation studies to investigate the effectiveness of the key components in our method. Results are presented in the Table 2 and are discussed in details below.

Patches to encode. As shown in Table 2a, only encoding the patches that will be transmitted could result in a minor drop of performance, while it can reduce some computations, thereby beneficial for computation-restricted robotic systems.

Timestamps to decode. We investigate the effect of whether the decoder incorporates previous timestamps or just the current single timestamp combined with learnable mask tokens. Results in Table 2b indicates that historical information is essential.

Temporal embedding. In the STAR decoder, we add temporal embedding to the patches of different timestamps respectively similar to the approach in [20, 21]. Ablation study in Table 2c shows that adding temporal embedding is beneficial.

Masking strategy. We compared our complementary masking strategy with random masking strategy proposed in MAE [9] in Table 2d. Results show that switching from complementary to random masking leads to a degradation in the completion performance.

Paradigm	Method	Detection		Semantic Segmentation	
		AP@IoU=0.5	AP@IoU=0.7	Vehicle	mIoU
Single-robot perception	Lower-bound	**49.90**	**44.21**	**45.93**	**36.64**
Task-specific multi-robot perception	When2com [8]	44.02	39.89	47.87	34.49
	Who2com [10]	44.02	39.89	47.84	34.49
	V2VNet [4]	68.35	62.83	**58.35**	41.17
	DiscoNet [5]	**69.03**	**63.44**	55.84	**41.34**
Task-agnostic multi-robot perception	STAR	58.30	52.33	54.09	37.56
	CNN baseline	59.85	54.05	54.61	38.32
	Upper-bound	**65.09**	**60.26**	**60.34**	**40.45**

Table 3: **Quantitative results on downstream perception tasks.** Lower-bound is a single-robot perception model trained using individual observations. The task-specific multi-robot perception methods achieve excellent performance via the elaborate supervised learning with synchronous multi-robot recordings. The task-agnostic methods are built on single-robot perception models consuming multi-robot observations (either original or reconstructed). In task-agnostic methods, the upper-bound directly transmitting original point clouds requires a bandwidth of 32.5 MB/s. CNN requires a bandwidth of 155.0 MB/s introduced by multi-scale feature maps. STAR achieves comparable performance with CNN yet with much lower bandwidth. Better completion performance or applying a stronger single-robot backbone could further enhance the perception performance.

5.4 Quantitative results on downstream perception

We directly feed the completed scenes to the single-robot perception model termed *lower-bound* without any fine-tuning, and the results are shown in Table 3. Our best STAR method improves the lower-bound by 18.4% and 17.8% in object detection (AP@IoU=0.7) and semantic segmentation (IoU of vehicle) respectively. Achieved by simply combining the completion model with off-the-shelf single-robot perception models, these improvements are promising because our framework: (1) has no knowledge about downstream tasks (*task-agnostic*); (2) does not require synchronous data in the training phase (*synchronization-free*); (3) is learned without manual annotations (*self-supervised*). We also investigate the performance of the single-robot perception model directly consuming original multi-view measurements without additional training, termed *upper-bound*. We find that it can achieve nearly comparable performance with DiscoNet [5] and V2VNet [4], both trained with full supervision using synchronous data for specific tasks. This demonstrates the potential of CSC: when the completions approach the ground truth scenes, it can perform similarly to the upper-bound on many downstream tasks.

6 Limitation

The performance gap between our method and the upper-bound on the downstream perception is still significant, which could be mainly caused by the non-perfect scene completion (IoU=52.2 at scale 1:1). Further improving the scene completion may be achieved by training with synchronized datasets, which is left as a future work that will ultimately improve performances for all downstream tasks. We believe when trained with more single-robot recordings, our method will achieve stronger performance and outperform task-specific approaches while maintaining great flexibility. We also inherit the common limitation in most existing collaborative perception works that all experiments are on simulated dataset due to the lack of public real-world datasets. We further ignore the influence of pose noises, although previous works [5] already revealed reasonable robustness.

7 Conclusion

We propose the first task-agnostic collaborative perception paradigm, where a single collaboration module is learned and can be transferred to a wide range of downstream tasks. Our key observation is that we can move communication between robots to temporal domain, which achieves great performance-bandwidth trade-off. Also, our self-supervised learning method sheds new lights into collaborative perception that reduces the importance of human annotations.

References

[1] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2021.

[2] R. Cheng, C. Agia, Y. Ren, X. Li, and L. Bingbing. S3cnet: A sparse semantic scene completion network for lidar point clouds. In *Conference on Robot Learning*, 2020.

[3] Y. Zhou, J. Xiao, Y. Zhou, and G. Loianno. Multi-robot collaborative perception with graph neural networks. *IEEE Robotics and Automation Letters*, 2022.

[4] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 605–621, 2020.

[5] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang. Learning distilled collaboration graph for multi-agent perception. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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

[7] E. Arnold, M. Dianati, R. de Temple, and S. Fallah. Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[8] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira. When2com: multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2020.

[9] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[10] Y.-C. Liu, J. Tian, C.-Y. Ma, N. Glaser, C.-W. Kuo, and Z. Kira. Who2com: Collaborative perception via learnable handshake communication. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6876–6883, 2020.

[11] S. Garg, N. Sünderhauf, F. Dayoub, D. Morrison, A. Cosgun, G. Carneiro, Q. Wu, T.-J. Chin, I. Reid, S. Gould, et al. Semantics for robotic mapping, perception and interaction: A survey. *Foundations and Trends® in Robotics*, 8(1–2):1–224, 2020.

[12] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*, pages 428–441. IEEE, 2002.

[13] M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5431–5440, 2016.

[14] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017.

[15] L. Roldao, R. De Charette, and A. Verroust-Blondet. 3d semantic scene completion: a survey. *International Journal of Computer Vision*, 2021.

[16] A.-Q. Cao and R. de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022.

[17] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.

[18] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir. Multimae: Multi-modal multi-task masked autoencoders. *arXiv preprint arXiv:2204.01678*, 2022.

[19] X. Geng, H. Liu, L. Lee, D. Schuurams, S. Levine, and P. Abbeel. Multimodal masked autoencoders learn transferable representations. *arXiv preprint arXiv:2205.14204*, 2022.

[20] Z. Tong, Y. Song, J. Wang, and L. Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022.

[21] C. Feichtenhofer, H. Fan, Y. Li, and K. He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022.

[22] C. Zheng, T.-J. Cham, and J. Cai. Tfill: Image completion via a transformer-based architecture. *arXiv preprint arXiv:2104.00845*, 2021.

[23] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

[24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[25] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[26] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018.

[27] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.