Over-the-Air Fusion of Sparse Spatial Features for Integrated Sensing and Edge AI over Broadband Channels

Zhiyan Liu, Qiao Lan, and Kaibin Huang

Abstract*—The *sixth-generation* (6G) mobile networks are differentiated from 5G by two new usage scenarios – distributed sensing and edge *artificial intelligence* (AI). Their natural integration, termed *integrated sensing and edge AI* (ISEA), promised to create a platform for enabling environment perception to make intelligent decisions and take real-time actions, which empowers wide-ranging applications, e.g., autonomous driving and collaborative robotics. A basic operation in ISEA is for a fusion center to acquire and fuse features of spatial sensing data distributed at many edge devices (known as agents). Its implementation is confronted by a communication bottleneck due to multiple access by numerous agents over hostile wireless channels. To overcome the bottleneck, we propose a novel framework, called *Spatial Over-the-Air Fusion* (Spatial AirFusion), which exploits radio waveform superposition to aggregate spatially sparse features over the air and thereby enables simultaneous access. The technology, which targets environment perception, is more sophisticated than conventional *Over-the-Air Computing* (AirComp) as it supports simultaneous aggregation over multiple voxels, which partition the 3D sensing region, and across multiple subcarriers. The efficiency and robustness of Spatial AirFusion are derived from exploitation of both spatial feature sparsity and multiuser channel diversity to intelligently pair voxel-level aggregation tasks and subcarriers with the objective of maximizing the minimum receive signal-to-noise ratio among voxels subject to instantaneous power constraints. Optimally solving the resultant mixed-integer problem of *Voxel-Carrier Pairing and Power Allocation* (VoCa-PPA) is a focus of this work. The proposed approach hinges on two useful results: (1) deriving the optimal power allocation as a closed-form function of voxel-carrier pairing and (2) discovering a useful property of VoCa-PPA that allows dramatic dimensionality reduction of the solution space. Building on the two results, both a low-complexity greedy algorithm and an optimal tree-search based approach are designed for VoCa-PPA. The latter features a customised design of a compact search tree and a fast tree search enabled by intelligent node pruning and agent ordering. Extensive simulations using real datasets demonstrate that Spatial AirFusion achieves significant reduction in computation errors and improvement in sensing accuracy as opposed to conventional AirComp without awareness of spatial sparsity.

***Index Terms*—Edge AI, distributed sensing, multiple access, over-the-air computation.**

I. Introduction

The *sixth-generation* (6G) mobile network warrants two essential capabilities, sensing and *artificial intelligence* (AI) [1]. The first capability involves the integration of diversified sensing modalities such as camera, mmWave, and LiDAR sensors to collect information from sensory data. The second capability is envisioned to support AI model deployments in 6G edge networks, enabling the delivery of intelligent services. Integrating these two essentials for advanced 6G applications, ranging from high-precision perception to human-machine symbiosis, leads to an emerging paradigm called *Integrated Sensing and Edge AI* (ISEA) [2]. In such a system, an edge device equipped with sensors, termed an *agent*, in a distributed sensing system acquires sensory data from its surroundings and sends features extracted using its local perception model to the edge server (i.e., fusion center) for aggregation and then inference to support a downstream AI application [3], [4]. However, ISEA faces a communication bottleneck due to the aggregation of high-dimensional sensing features over resource-constrained wireless channels [5], [6]. One promising solution for overcoming the bottleneck is called *Over-the-Air Computation* (AirComp), which exploits waveform superposition in simultaneous access to realize over-the-air data aggregation [7]–[10]. Based on AirComp, we develop a novel framework, termed *Spatial Over-the-Air Fusion* (Spatial AirFusion), for communication-efficient multi-sensor fusion in environment perception over a broadband channel. Its distinctive feature is to exploit spatial feature sparsity and channel frequency selectivity to intelligently map voxels, which divide the sensing region, to subcarriers for performing voxel-level AirComp tasks. Thereby, the sensing performance is improved while computation complexity reduced.

Precise environment perception underpins a set of killer application scenarios of 6G, e.g., autonomous driving and collaborative robots. State-of-the-art perception models [11] leverage LiDAR, mmWave, and camera data to generate spatial feature vectors associated with certain locations in the physical world, as opposed to location-agnostic features in conventional classification and object detection. This type of feature is known as *voxel features*, where one voxel represents a spatial region in an evenly spaced 3D grid of the sensing range [12], [13]. To support low-latency and large-scale environment perception in 6G networks requires task-oriented air-interface design targeting ISEA. As a specific use case of edge inference, ISEA can be implemented on the well-known split inference architecture [14]–[17]. In this architecture, a global inference model is split into a device and a server sub-model with the former used for local feature extraction and the latter for remote inference [16]. It can be generalized to distributed split inference (for which ISEA is a special case) by deploying models at multiple devices for local feature extraction (or inference) and performing local-feature (or label) aggregation at the server to attain a high inference accuracy [15], [18], [19]. Most recently, Researchers have made an attempt on designing AirComp techniques to realize different feature-aggregation

Z. Liu, Q. Lan and K. Huang are with Department of Electrical and Electronic Engineering at The University of Hong Kong, Hong Kong. Contact: K. Huang (Email: huangkb@eee.hku.hk).

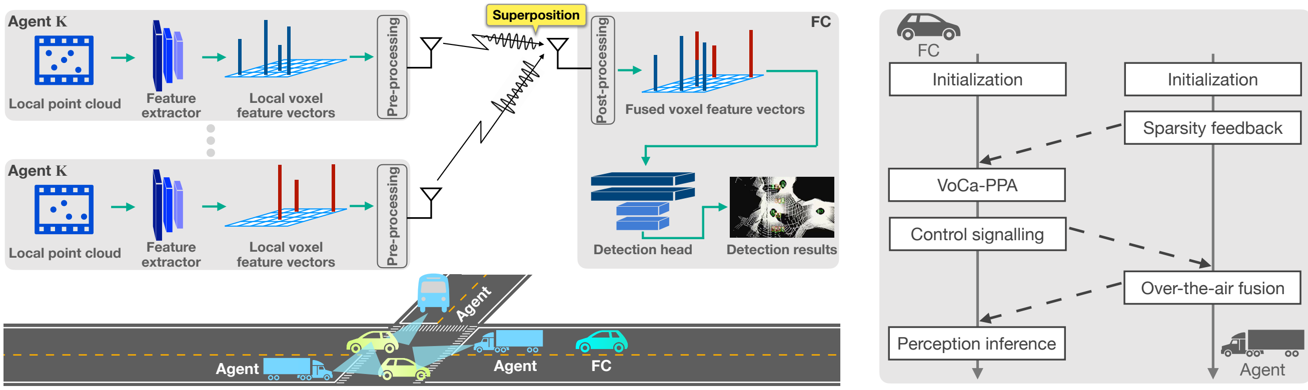
functions, which include maximization, in an ISEA system based on an end-to-end sensing performance metric [3].

AirComp in its own right is a fast-growing area [7]. The principle of AirComp is to exploit the superposition of signals simultaneously transmitted by multiple agents such that the desired aggregation functions, e.g., averaging, multiplication, and maximization, can be realized over the air [20], [21]. To materialize accurate functional computation via AirComp requires coping with channel fading and noise. For this purpose, a line of techniques has been designed to minimize AirComp errors including optimal power control [22], *multiple-input-multiple-output* (MIMO) beamforming [23] and interference management [24]. Broadband transmission is prevalent in modern high-rate mobile systems, which is assumed in the current system model. This motivates researchers to study broadband AirComp by addressing issues such as power allocation among subcarriers [19], [25], subcarrier truncation to avoid deep fading [8] and exploitation of channel frequency diversity [26]. The co-existing information-transfer users and AirComp devices participating in federated learning are also studied where the rate-maximizing subcarrier allocation for the former is designed subject to a guarantee on the learning performance of the latter [27]. Latest research focuses on the application of AirComp to increasing the communication efficiency of *federated learning* (FL), creating an area called *over-the-air FL* (AirFL) [8]–[10]. In this paradigm, AirComp realizes over-the-air aggregation of local gradients or models uploaded by devices, from which the result is used to update a global model at an edge server [8], [9]. While traditional AirComp techniques aim at computation error minimization, the design objective of AirFL techniques is to accelerate learning and account for the specific characteristics of transmitted data (i.e., local gradients/models). This results in a rich set of task-oriented wireless techniques such as power control based on gradient statistics [28], data- and channel-aware sensor scheduling [29], adaptive precoding [30], etc.

Existing studies on AirComp as discussed above all assume single-stream data sources without considering data spatial distributions. Nevertheless, spatial feature variation is a key characteristic of environment perception as reflected in two aspects. On one hand, spatial feature sparsity is due to the fact that features are dense only in voxels containing detectable objects (e.g., vehicles and pedestrians) but not others. On the other hand, spatial feature distributions as observed by different agents are heterogeneous because of their non-identical fields of perception and view angles. Another aspect of heterogeneity is multiuser channel diversity. One key effect of spatial feature variation is the spatial variation of AirComp error as elaborated in the sequel. Let the task of spatial feature aggregation be divided into voxel-level sub-tasks. Due to the sparsity and heterogeneity of spatial feature distributions, the subset of agents participating in aggregation varies from voxel to voxel, resulting in differences in corresponding AirComp errors as they depend on the numbers of participating agents (see, e.g., [31]) and qualities of the associated channels. The errors can be manipulated using a mechanism called *Voxel-Carrier (VoCa) Pairing* that maps voxels to subcarriers for executing their sub-tasks. Via this mechanism, a large number of degrees-of-freedom due to numerous voxels and subcarriers can be exploited to improve the performance of Spatial AirFusion. Furthermore, VoCa Pairing can be integrated with power allocation over subcarriers to obtain additional performance gain, giving rise to the problem of optimal *VoCa Pairing and Power Allocation* (VoCa-PPA).

Let the performance of Spatial AirFusion be measured using the metric of the minimum receive SNR among all voxels, which serves as an indicator of the largest AirComp error. Given the objective of maximizing the metric, a subcarrier under favourable channel conditions should be ideally paired with a voxel with many participating agents. However, given the heterogeneity in multiple voxels and sub-channels of multiple agents, the optimal VoCa-PPA problem becomes a sophisticated mixed integer program. In this work, we present the framework that consists a set of algorithms for efficiently solving the problem via exploiting the unique features of Spatial AirFusion. The key contributions are summarized as follows.

- **AirFusion Protocol.** A communication protocol is presented to realize spatial AirFusion in a multi-agent system, comprising the following three phases. First, each agent sends binary sparsity indicators of all voxels in the sensing region to the server. In the second phase of radio resource allocation, the server performs VoCa-PPA using one of the proposed algorithms based on the sparsity indicators and broadband channel states. Last, in the over-the-air fusion phase, the agents' feature vectors on voxels are transmitted simultaneously and aggregated over the air using the assigned subcarriers and power.
- **Greedy VoCa-PPA Algorithm.** A low-complexity algorithm is designed to compute a sub-optimal solution for the VoCa-PPA problem by sequentially solving the problems of optimal power allocation and VoCa Pairing. First, given VoCa Pairing, the optimal allocated power for subcarriers is derived in closed-form. As revealed by the result, the minimum receive SNR depends solely on a bottleneck agent charaterized by poorest associated channels. Second, given the derived power allocation, the VoCa-PPA problem is reduced to the problem of optimal VoCa Pairing, which is combinatorial and NP-hard [32]. It is solved using a low-complexity greedy algorithm that iteratively matches each voxel with the best-matched subcarrier under the criterion of minimizing the maximum channel-inversion power over all participating agents. In this regard, voxels with relatively high feature densities tend to involve more agents participating in AirComp, which degrades receive SNRs. For this reason, they are given higher priorities so as to be matched to better sub-channels.
- **Optimal VoCa-PPA Algorithms.** Leveraging the optimal power allocation derived previously simplifies the optimal VoCa-PPA problem to optimal VoCa Pairing without sacrificing the solution's optimality. Despite a simpler form, the latter is a max-linear assignment problem that does not admit polynomial-time solutions. To address the issue, a solution approach is designed to



(a) (b)

Fig. 1. (a) An ISEA system for environment perception in the context of autonomous driving. (b) Spatial AirFusion protocol.

significantly reduce the computation complexity. The approach is comprised of two designs - a compact search tree and a *depth-first search* (DFS) algorithm that are both customised for VoCa Pairing. Underpinning these algorithms is a useful property of the problem that two voxels with identical sparsity indicators are equivalent from the perspective of minimizing the objective. The property is exploited to convert the VoCa Pairing problem from the original *one-to-one mapping* to *subset-to-subset mapping*. As a result, orders-of-magnitude reduction in computation complexity is achievable. The complexity of tree search is further reduced using two proposed schemes. The first is intelligent early stopping and node pruning based on criteria developed by comparing the current best global objective and local objectives in each step. The other is agent ordering in DFS based on a designed priority indicator combining each agent's channel states and sparsity pattern.

- **Experiments.** The performance of Spatial AirFusion is evaluated by extensive experiments using both synthetic and real datasets (i.e., OPV2V [33]). The proposed framework is demonstrated to outperform traditional AirComp without awareness of spatial sparsity by a large margin, e.g., 10 dB gain in AirComp error suppression and significantly improved end-to-end inference accuracy.

II. System Models

We consider an ISEA system targeting environment perception, where K agents are distributed in the space and cooperate to complete a sensing task as coordinated by a *fusion center* (FC). The system is illustrated in Fig. 1(a) for the context of autonomous-driving perception where agents are helper vehicles and the fusion center is an ego vehicle. For each perception instance, each agent acquires a view (e.g., a LiDAR frame) of the surrounding environment via its sensor and extracts its local features. The fusion center then employs an AirFusion technique as proposed in subsequent sections to wirelessly aggregate local features and perform inference for the global perception results. Relevant models and the performance metric are described in the following subsections.

A. Agent Perception Model

Each agent is equipped with a LiDAR or camera sensor that has specific perception ranges in depth, height, and width. The resultant three-dimensional perception space is partitioned into a regular grid with each cell referred to as a *voxel*. The numbers of partitions along the depth, height, and width directions are denoted as V_{d}, V_{h}, and V_{w}, respectively. Then the total number of voxels is given as $V = V_{\mathsf{d}} V_{\mathsf{h}} V_{\mathsf{w}}$, which is assumed identical for all agents for ease of notation. Then the voxels for each agent can be indexed by $v = 1, 2, ..., V$. As illustrated in Fig. 1(a), each agent utilizes its voxel-perception model to generate an L-dimensional feature vector for every voxel to capture the spatial object information contained within the voxel, termed *voxel feature vector* [11], [12]. For voxel v, its feature vector on agent k is denoted as $\mathbf{f}_{k,v} \in \mathbb{R}^L$. It can be a zero vector (i.e., $\mathbf{f}_{k,v} = \mathbf{0}$) if agent k detects no objects in voxel v, which may be a true or false negative result due to occlusion or limitations of the agent's sensor.

B. Cooperative Sensing Model

The agents upload their voxel feature vectors, $\{\mathbf{f}_{k,v}\}_{1 \leq k \leq K, 1 \leq v \leq V}$, to the fusion center over wireless links. Considering an arbitrary voxel, say voxel v, the result from fusing the associated vectors is denoted as $\mathbf{g}_v$. For two representative fusion functions, namely average-pooling and max-pooling[1], the ℓ-th element of $\mathbf{g}_v$ is given as

$$g_v[\ell] = \begin{cases} \frac{1}{K} \sum_{k=1}^{K} f_{k,v}[\ell], & \text{average pooling}, \\ \max_{1 \leq k \leq K} f_{k,v}[\ell], & \text{max-pooling}. \end{cases} \tag{1}$$

Finally, the fusion center feeds the fused feature vectors, $\{\mathbf{g}_v\}_{v=1}^{V}$, into its perception model to obtain the perception results (e.g., object label).

C. Communication Model

Spatial AirFusion wirelessly implements the above feature-fusion process over a broadband channel, which is modeled as

[1]The proposed method can accommodate other fusion functions, e.g., weighted-sum pooling and square-root pooling, via linearly scaling each vector or employing the generalized AirPooling pre- and post-processing functions [3].

follows. The total bandwidth B is partitioned into M subcarriers using *orthogonal frequency division multiplexing* (OFDM). Without loss of generality, it is assumed that $M \geq V$, as otherwise transmission for all V voxels can be carried out over multiple channel coherence blocks. The channel follows block fading where a subcarrier remains constant within a channel coherence block. It is assumed that the channel coherence time is not shorter than the duration of L symbol slots and that *channel state information* (CSI) is available at the receiver and transmitters. This allows a voxel feature vector to be uploaded within a single channel coherence block. Assuming symbol-level synchronization (see [8] for synchronization techniques), all agents simultaneously transmit their feature vectors on assigned subcarriers. The ℓ-th symbol received by the fusion center on the m-th subcarrier, $y_m[\ell]$, is given by

$$y_m[\ell] = \sum_{k=1}^{K} h_{k,m} p_{k,m}[\ell] x_{k,m}[\ell] + z_m[\ell], \tag{2}$$

where $x_{k,m}[\ell]$ denotes the ℓ-th symbol transmitted by the k-th agent on the m-th subcarrier, $h_{k,m}$ the complex channel coefficient of subcarrier m from agent k to the fusion center, $p_{k,m}[\ell]$ the precoding coefficient, and $z_m[\ell] \sim \mathcal{CN}(0, N_0)$ the i.i.d. Gaussian noise with power N_0. Using training data, the symbols $\{x_{k,m}[\ell]\}$ can be normalized to be zero-mean and unit-variance on a long-term basis [3]. Channel inversion precoding is adopted for magnitude alignment between received signals [9], [34]. The transmit power of agent k on subcarrier m is then given by $|p_{k,m}[\ell]|^2 = \frac{P_{\mathsf{rx},m}[\ell]}{|h_{k,m}|^2}, \forall l$, where $P_{\mathsf{rx},m}[\ell] \geq 0$ denotes the receive SNR coordinated by the fusion center for the ℓ-th symbol transmitted on subcarrier m. As the channel remains constant for all $\ell = 1, 2, \ldots, L$, we set $P_{\mathsf{rx},m}[\ell] \triangleq P_{\mathsf{rx},m}, \forall \ell$, and consequently $p_{k,m}[\ell] \triangleq p_{k,m}, \forall \ell$. Each agent limits the total transmission power per OFDM symbol to P_{max}, which is given as

$$\sum_{m=1}^{M} |p_{k,m}|^2 \leq P_{\mathsf{max}}, \ \forall k. \tag{3}$$

D. Performance Metric

The presence of channel distortion in Spatial AirFusion results in AirComp error, defined as the mean square error between the over-the-air aggregated data and the ground-truth fusion result [7]. Under per-agent power constraints, AirComp error, known to be inversely proportional to the receive SNR, is dominated by the worst channel due to the required magnitude alignment of received signals via channel inversion [22]. In the sensing context, the end-to-end sensing accuracy, prone to distortion in the aggregated intermediate features, has been shown to improve with the receive SNR in [3]. In AirFusion, we denote the receive SNR for the sub-task of aggregating voxel v's features as γ_v. It is determined by the coordinated SNR level for its assigned subcarrier, i.e., $\gamma_v = P_{\mathsf{rx},m(v)}$ if subcarrier $m(v)$ is assigned for voxel v. As the voxel v's aggregated features encode information of sensed spatial objects, a low γ_v implies high feature distortion and hence low object detection accuracy in voxel v. This can lead to catastrophic results in mission-critical tasks, such as missing a pedestrian or vehicle in autonomous driving. Hence, to guarantee satisfactory object detection accuracies over the entire sensing region, we propose to define the performance metric for Spatial AirFusion, denoted by U, as the minimum receive SNR across all voxels: $U = \min_{v \in \{1,\ldots,V\}} \gamma_v$.

III. Spatial AirFusion Protocol and Operations

The proposed Spatial AirFusion framework aims at efficiently aggregating multi-agent voxel features over a broadband channel, where the feature vectors on different agents but attributed to the same voxel are aggregated over a particular subcarrier. Targeting environment perception, Spatial AirFusion is differentiated from generic AirComp in that features exhibit heterogeneous sparsity across voxels due to diversified occlusion and finite detection ranges of agents, which is exploited for optimized resource allocation by VoCa Pairing and power control. The steps of the Spatial AirFusion protocol are illustrated in Fig. 1(b) and detailed below.

A. Sparsity Feedback

Assume that agents are synchronized in indexing voxels of the sensing region due to coordination by the fusion center (see Section II-A). Voxel v is called *sparse* on agent k if and only if the corresponding feature vector $\mathbf{f}_{k,v}$ is a zero vector. Each agent calculates a binary sparsity vector $\mathbf{s}_k \in \{0,1\}^V$, $k = 1, \ldots, K$, indicating the observed sparsity pattern of its voxels. Specifically, $\mathbf{s}_k[v] = 0$ if voxel v on agent k is sparse and $\mathbf{s}_k[v] = 1$ otherwise, i.e.,

$$\mathbf{s}_k[v] = \begin{cases} 1, & \|\mathbf{f}_{k,v}\|_0 \geq 1, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $\|\mathbf{f}\|_0$ is the vector zero-norm defined as the number of non-zero elements in $\mathbf{f}$. All agents report their sparsity vectors to the fusion center via a reliable control channel. The server assembles them into a sparsity pattern: $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_K]^T$. The entry on the k-th row and v-th column of matrix $\mathbf{S}$ is denoted as $S_{k,v} = \mathbf{s}_k[v]$.

B. Radio Resource Allocation

Given the sparsity pattern, $\mathbf{S}$, and transmit CSI, $\{h_{k,m}\}$, the server allocates subcarriers and transmit power for each agent. We denote $\mathbf{A} \in \{0,1\}^{V \times M}$ as the VoCa pairing matrix, where the (v,m)-th entry is given as

$$A_{v,m} = \begin{cases} 1, & \text{subcarrier } m \text{ paired with voxel } v, \\ 0, & \text{otherwise}. \end{cases} \tag{5}$$

To assure orthogonality between aggregations of all voxels, the following constraints are applied on assigning subcarriers:

$$\sum_{v=1}^{V} A_{v,m} \leq 1, \ \forall m = 1, 2, \cdots, M. \tag{6}$$

On the other hand, each voxel occupies exactly one subcarrier:

$$\sum_{m=1}^{M} A_{v,m} = 1, \ \forall v = 1, 2, \cdots, V. \tag{7}$$

Let $|p_{k,m}|^2$ denote the transmit power invested to subcarrier m by agent k. Then, $\{|p_{k,m}|^2\}$ depend on the sparsity of paired voxels channel gains, and receive SNRs (after aggregation). To be specific, the agents containing zero voxel feature vectors do not participate in transmission on the designated subcarrier, i.e., $p_{k,m} = 0$ if $\sum_{v=1}^{V} S_{k,v}A_{v,m} = 0$. All the agents participating in the transmission on the designated subcarrier shall set transmit power to align their signal magnitude as required for AirComp [8]. It follows that the receive SNR, denoted as γ_v for voxel v, is given as

$$\gamma_v = \sum_{m=1}^{M} A_{v,m}\frac{|p_{k,m}h_{k,m}|^2}{N_0},\ \forall k \in \{k'|S_{k',v}=1\}. \tag{8}$$

The above resource allocation decisions, $\mathbf{A}$ and $\{\gamma_v\}_{v=1}^{V}$, are broadcast to all agents such that each onboard agent sets its precoding coefficients accordingly as follows:

$$p_{k,m} = \frac{\sqrt{N_0}}{h_{k,m}}\sum_{v=1}^{V}\sqrt{\gamma_v}S_{k,v}A_{v,m}. \tag{9}$$

The control of resource allocation, i.e., VoCa-PPA, is optimized in the subsequent sections.

C. Over-the-Air Fusion

All agents simultaneously transmit their voxel features using assigned subcarriers and power levels as specified in $\mathbf{A}$ and $\{p_{k,m}\}$. Consider an arbitrary agent k and an arbitrary symbol ℓ. Assume that average pooling is the desired fusion function. Then, the feature pre-processing is implemented by normalizing the ℓ-th feature coefficient of voxel v on agent k, $f_{k,v}[\ell]$, yielding the pre-processed feature $\tilde{x}_{k,v}[\ell]$ as given by

$$\tilde{x}_{k,v}[\ell] = \frac{1}{\sigma}\left(f_{k,v}[\ell]-\mu\right), \tag{10}$$

where the normalization parameters σ and μ in (10) are shared by all agents and set such that the distribution of pre-processed features is zero-mean and unit-variance. The extension to other fusions functions (e.g., max-pooling [3]) is straightforward by applying additional post- and/or pre-processing functions. The pairing matrix $\mathbf{A}$ maps the pre-processed features, $\{\tilde{x}_{k,v}[\ell]\}_{v=1}^{V}$ to the ℓ-th symbol of each subcarrier. Then the symbol transmitted by agent k over subcarrier m can be written as

$$x_{k,m}[\ell] = \sum_{v=1}^{V} A_{v,m}\tilde{x}_{k,v}[\ell]. \tag{11}$$

Combining (9) and (11) with the AirComp operation in (2) yields the ℓ-th symbol received at the fusion center:

$$\begin{aligned} y_m[\ell] &= \sum_{k=1}^{K}\left[\left(\sum_{v=1}^{V}\sqrt{N_0\gamma_v}S_{k,v}A_{v,m}\right)x_{k,m}[\ell]\right] + z_m[\ell], \\ &= \sum_{v=1}^{V}\left[A_{v,m}\sqrt{N_0\gamma_v}\left(\sum_{k=1}^{K}S_{k,v}\tilde{x}_{k,v}[\ell]\right)\right] + z_m[\ell]. \end{aligned} \tag{12}$$

The post-processing operation (i.e., denormalization and real-part extraction) is then designed to compute the estimated fused feature vector for voxel v, $\tilde{\mathbf{g}}_v \in \mathbb{R}^L$, such that its ℓ-th element is given as

$$\begin{aligned} \tilde{g}_v[\ell] &= \Re\left[\sum_{m=1}^{M}\left[\frac{A_{v,m}}{K\sqrt{N_0\gamma_v}}\left(\sigma y_m[\ell] + \sum_{k=1}^{K}\mu S_{k,v}\right)\right]\right], \\ &= g_v[\ell] + \Re\left[\frac{\sigma\sum_{m=1}^{M}A_{v,m}z_m[\ell]}{K\sqrt{N_0\gamma_v}}\right]. \end{aligned} \tag{13}$$

It follows that the vector can be expressed in terms of its ground-truth given in (1) as

$$\tilde{\mathbf{g}}_v = \mathbf{g}_v + \mathbf{w}_v, \tag{14}$$

where $\mathbf{w}_v$ is a vector of i.i.d. Gaussian noise variables following $\mathcal{N}\left(0, \frac{1}{2}K^{-2}\gamma_v^{-1}N_0^{-1}\sigma^2\right)$. Last, the fusion center assembles all the fused voxel feature vectors, $\{\tilde{\mathbf{g}}_v\}$, and feeds them into the downstream perception head to obtain the final inference results.

IV. VoCa-PPA: Problem Formulation

Recall that the VoCa-PPA problem of Spatial AirFusion aims at allocating subcarriers and transmit power to agents/voxels so as to maximize the minimum receive SNR among voxels, which is formulated as follows. Given the pairing constraints (6) and (7) and by substituting the channel inversion (9) into the instantaneous power constraints in (3), the optimization problem can be formulated as

$$\text{(P1)}\quad \begin{aligned} \max_{\mathbf{A},\{\gamma_v\}_{v=1}^{V}}\ &\min_{v}\quad \gamma_v \\ \text{s.t.}\quad & A_{v,m}\in\{0,1\},\ \forall v,m, \\ & \sum_{v=1}^{V}A_{v,m}\le 1,\ \forall m, \quad \sum_{m=1}^{M}A_{v,m}=1,\ \forall v, \\ & \sum_{m=1}^{M}\frac{N_0}{|h_{k,m}|^2}\sum_{v=1}^{V}S_{k,v}A_{v,m}\gamma_v \le P_{\mathsf{max}},\ \forall k. \end{aligned}$$

Problem P1 is a mixed-integer programming problem. To simplify it, we derive the optimal receive SNRs as functions of the pairing matrix $\mathbf{A}$, shown in the following lemma. Its proof is by a standard transformation of the power allocation problem given $\mathbf{A}$ into a linear program and solving it via Lagrange duality and thus omitted for brevity.

Lemma 1 (*Optimal Power Allocation*)**.** Given the VoCa pairing matrix $\mathbf{A}$, setting an equal SNR level across all voxels, i.e., $\gamma_v = \gamma^*(\mathbf{A})$ for all v, is optimal for Problem P1, where $\gamma^*(\mathbf{A})$ is given as

$$\gamma^*(\mathbf{A}) = P_{\mathsf{max}}\left(\max_k N_0\sum_{v=1}^{V}\sum_{m=1}^{M}\frac{S_{k,v}A_{v,m}}{|h_{k,m}|^2}\right)^{-1}. \tag{15}$$

Substituting $\gamma^*(\mathbf{A})$ into (9) yields the optimal transmit power of each agent over a subcarrier,

$$p_{k,m}^*(\mathbf{A}) = \frac{\sqrt{N_0\gamma^*(\mathbf{A})}}{h_{k,m}}\sum_{v=1}^{V}S_{k,v}A_{v,m}. \tag{16}$$

It can be observed from (15) that the achievable SNR levels depend on a bottleneck agent characterized by weakest overall channel conditions by considering all voxels and subcarriers. Without compromising its optimality, Problem P1 can be simplified by substituting (15) into the objective. This leads to the following equivalent VoCa Pairing problem:

$$
\begin{aligned}
\text{(P2)} \quad & \min_{\mathbf{A}} \quad \max_{k} \quad \sum_{v=1}^{V}\sum_{m=1}^{M} c_{k,m} S_{k,v} A_{v,m} \triangleq F(\mathbf{A}) \\
& \text{s.t.} \quad A_{v,m} \in \{0,1\},\ \forall v, m, \\
& \qquad \sum_{v=1}^{V} A_{v,m} \le 1,\ \forall m, \quad \sum_{m=1}^{M} A_{v,m} = 1,\ \forall v,
\end{aligned}
$$

where the constant $c_{k,m} \triangleq \frac{N_0}{|h_{k,m}|^{-2}}$. This is a combinatorial optimization problem with a max-linear objective, which is known to be NP-hard in general [32]. A set of algorithms are designed in the following sections to overcome this challenge.

V. Greedy VoCa-PPA Algorithm

In this section, we first develop a low-complexity solution to Problem P2 for VoCa Pairing based on a greedy heuristic. Then, combining the greedy algorithm and the optimal power allocation scheme yields the greedy VoCa-PPA algorithm for Spatial AirFusion control.

A. Greedy VoCa Pairing

The proposed greedy pairing algorithm in principle sequentially pairs a single voxel with the locally optimal subcarrier. The specific algorithm is designed as follows.

- **Initialization.** The pairing matrix $\mathbf{A}$ is initialized as $\mathbf{A} \leftarrow \mathbf{0}^{V\times M}$.
- **Iteration.** In each iteration, say the v-th one, $\mathbf{A}$ is updated in a greedy manner, i.e., upon solving an optimization problem that seeks the best subcarrier for the v-th voxel. Specifically, only the v-th voxel is addressed in this iteration. Dropping other voxels in Problem P2 yields the greedy optimization problem for voxel v as

$$
\begin{aligned}
\text{(P3)} \quad & \min_{\{A_{v,m}\}_{m=1}^{M}} \quad \max_{k} \quad \sum_{m=1}^{M} c_{k,m} S_{k,v} A_{v,m} \\
& \text{s.t.} \quad A_{v,m} \in \{0,1\},\ \forall m, \\
& \qquad \sum_{n=1}^{v} A_{n,m} \le 1,\ \forall m, \quad \sum_{m=1}^{M} A_{v,m} = 1.
\end{aligned}
$$

  In Problem P3, only the pairing parameters for voxel v are optimized while the others are fixed. The optimal solution to Problem P3, $\{A^*_{v,m}\}_{m=1}^{M}$, can be easily obtained for any given v as,

$$
A^*_{v,m} = \begin{cases} 1, & m = \underset{m\in\{m' \mid \sum_{n=1}^{v-1} A_{n,m'}=0\}}{\arg\min} \ \max_{k} \ c_{k,m} S_{k,v}, \\ 0, & \text{otherwise.} \end{cases} \tag{17}
$$

  To end the v-th iteration, the entries specifying pairing of voxel v in $\mathbf{A}$ are updated as $A_{v,m} \leftarrow A^*_{v,m}$ for all m.

Sequence optimization. An optimized sequence of voxels in greedy pairing can boost the performance, i.e., improve the achieved voxel-level receive SNRs. To this end, we first propose a metric for sorting the voxels. One voxel can differ from another in the level of sparsity, i.e., the number of agents participating in aggregation. We refer to the voxels involving a small number of agents as *high-sparsity* voxels, in comparison against the *low-sparsity* voxels involving a large number of agents. Intuitively, the latter should be assigned subcarriers with favorable channel conditions as it is well-known in the AirComp literature that the receive SNR decreases as more agents participate [31]. Based on this principle, we propose a sparsity-aware permutation strategy that prioritizes low-sparsity voxels in greedy pairing. The permutation function $\pi(\cdot)$ maps an arbitrary entry v in the set $\{1, 2, ..., V\}$ to its image $\pi(v)$, which determines the index of iteration in the greedy pairing algorithm. Specifically, $\pi(\cdot)$ is constructed via sorting the sequence $1, 2, ..., V$ in descending order of their sparsity levels $\sum_{k=1}^{K} S_{k,v}$. This yields a sorted sequence $\pi(1), \pi(2), ..., \pi(V)$, where we place v_1 before v_2 in the case of $\sum_{k=1}^{K} S_{k,v_1} = \sum_{k=1}^{K} S_{k,v_2}$ if $v_1 < v_2$. It can be easily verified that the constructed $\pi(\cdot)$ is an *bijective* function and prioritizes low-sparsity voxels.

Algorithm 1: Greedy VoCa-PPA

Input: Sparsity matrix $\mathbf{S}$ and channel matrix $\mathbf{H}$;
Prioritization: Determine $\pi(\cdot)$ as given in Section V;
Initialization: $\mathbf{A}^\dagger = \mathbf{0}$;
for $v = 1, 2, \cdots, V$ **do (greedy pairing)**
    Evaluate $A^\dagger_{\pi(v),m}$ for $m = 1, 2, ..., M$ by (17) ;
Setting SNR: Substitute $\mathbf{A}^\dagger$ into (15) for $\gamma^*\left(\mathbf{A}^\dagger\right)$;
Signalling: Broadcast the control parameters $\mathbf{A}^\dagger$, $\gamma^*\left(\mathbf{A}^\dagger\right)$ to agents, which then set their transmit power by (16);

B. Greedy VoCa-PPA

The control algorithm, named greedy VoCa-PPA, combines the above greedy pairing with an optimized sequence and the power allocation scheme in Lemma 1, which is summarized in Algorithm 1. Its input $\mathbf{H}$ is a K-by-M matrix of channel gain with $h_{k,m}$ being its entry in the k-th row and m-th column. As a remark, the control signalling in Algorithm 1 involves broadcasting a *sparse* and *binary* matrix $\mathbf{A}^\dagger$ and a scalar $\gamma^*\left(\mathbf{A}^\dagger\right)$. The former of the two control parameters can be easily encoded into $\log_2\left(\frac{M!}{(M-V)!}\right) \le V\log_2(M)$ bits while the latter can be quantized into 32 bits following the floating-point precision convention. The signalling thus can be implemented over a downlink feedback channel with its overhead neglected.

VI. Optimal VoCa-PPA: Compact Tree Design

The greedy VoCa-PPA algorithm in the preceding section is computation-efficient but sub-optimal. In this and the next sections, we present an optimal and efficient approach for solving the VoCa-PPA Problem in P1 or equivalently Problem P2. The tree-search based approach consists of two components – compact tree design in this section and fast tree search in the next section. In general, Problem P2 can be viewed as a special case of the max-linear assignment problem, and its optimal solution can be searched for using the well-known ranking method (see, e.g., [35]). The novelty of our design, which yields a higher efficiency than the existing method, lies in exploiting the special structure of Problem P2. In particular, a derived useful property of its objective leads to a dramatic reduction of the dimensionality of the search space. As a common practice in solving bipartite matching problems, assume equal numbers of voxels and subcarriers $M = V$ in the sequel without loss of generality as the case of $M > V$ can be augmented with $(M - V)$ dummy voxels with all-zero sparsity indicators for all agents.

A. A Useful Property of Objective Function

Consider the objective function of Problem P2, $F(\mathbf{A}) = \max_k f_k(\mathbf{A})$, where $f_k(\mathbf{A}) \triangleq \sum_{v=1}^{V}\sum_{m=1}^{M} c_{k,m}S_{k,v}A_{v,m}$. To facilitate exposition, let $m(v)$ denote the index of the unique non-zero entry in the v-th row of $\mathbf{A}$, which indicates that the v-th voxel is mapped to the $m(v)$-th subcarrier. We thus have $m(v_i) \neq m(v_j)$ when $v_i \neq v_j$. Then, $f_k(\mathbf{A})$ can be rewritten as

$$f_k(\mathbf{A}) = \sum_{v=1}^{V} S_{k,v}c_{k,m(v)} = \sum_{v\in\mathcal{V}_k} c_{k,m(v)} = \sum_{m\in\mathcal{M}_k} c_{k,m}, \tag{18}$$

where $\mathcal{V}_k = \{v|S_{k,v} = 1\}$, the index set of all non-sparse voxels for agent k, and $\mathcal{M}_k = \{m(v)\}_{v\in\mathcal{V}_k}$, the set of subcarriers selected for non-sparse voxels, is the image of $\mathcal{V}_k$ under the mapping $m(\cdot)$ with $|\mathcal{M}_k| = |\mathcal{V}_k|$. An important observation is that f_k depends only on the set $\mathcal{M}_k$ but not the specific one-to-one mappings. As a result, for two voxels v_1 and v_2 both in $\mathcal{V}_k$ or $\mathcal{V}\setminus\mathcal{V}_k$ (or equivalently having identical sparsity indicators $S_{k,v_1} = S_{k,v_2}$ on agent k) swapping their associated subcarriers does not alter the value of f_k as $\mathcal{M}_k$ remains unchanged. This argument can be extended from f_k to the objective $F(\mathbf{A})$ since it is a function of $\{f_k(\mathbf{A})\}$. Specifically, consider the case that two voxels v_1 and v_2 have identical sparsity indicators for all K agents, or in other words, the two voxels have exactly the same sparsity vector, i.e., $\mathbf{t}_{v_1} = \mathbf{t}_{v_2}$, where $\mathbf{t}_v$ is the v-th column of the sparsity pattern matrix $\mathbf{S}$. Then exchanging their assigned subcarriers does not change the objective value. In such cases, we call the two voxels *homogeneous* due to their equivalence in subcarrier assignment. Aggregating all voxels which are homogeneous to each other results in the concept of a *homogeneous subset*, denoted by $\mathcal{H}(\mathbf{r}^q) \triangleq \mathcal{H}^q$, where $\mathbf{r}^q \in \{0,1\}^K$ indicates the sparsity vector shared by all voxels in $\mathcal{H}^q$. Mathematically, for all $v \in \mathcal{H}^q$, $\mathbf{t}_v = \mathbf{r}^q$. As $\mathbf{r}^q$ is a binary vector with length K, it has at most 2^K possibilities, as indexed by $q = 1, \ldots, 2^K$. The above property is stated formally in the following lemma.

Lemma 2. Consider a VoCa Pairing $m(\cdot) : \mathcal{V} \to \mathcal{M}$ and two voxels in the same *homogeneous subset* $v_1, v_2 \in \mathcal{H}^q$. A new pairing $m'(\cdot)$ with $m'(v_1) = m(v_2)$, $m'(v_2) = m(v_1)$ while $m(v) = m'(v)$ for all $v \neq v_1, v_2$ yields the same objective value as $m(\cdot)$.

The above lemma suggests that once the mapping between a homogeneous subset of voxels to an equal-size subcarrier subset is determined, the element-wise mapping can be arbitrary without altering the objective value. The property is the fundamental reason for the efficiency of the proposed solution approach.

B. Compact Solution Space

The property in Lemma 2 is exploited in the sequel to define a compact solution space comprised of subset-to-subset mappings, which features much lower dimensionality as opposed to the original space of all possible one-to-one mappings $m(\cdot) : \mathcal{M} \to \mathcal{V}$.

To begin with, relevant terminologies are introduced as follows. Let $\{\mathcal{P}^j\}_{j=1}^{N(\varphi)}$ be a non-overlapping partition of the voxel set $\mathcal{V}$ with $\bigcup_{j=1}^{N(\varphi)} \mathcal{P}^j = \mathcal{V}$ and $\mathcal{P}^i \cap \mathcal{P}^j = \emptyset$ for any $i \neq j$, where $1 \leq N(\varphi) \leq V$ is the number of disjoint subsets. A subset-to-subset mapping φ with $\mathrm{dom}(\varphi) = \{\mathcal{P}^j\}_{j=1}^{N(\varphi)}$ pairs $\mathcal{P}^j$ with $\varphi(\mathcal{P}^j)$ for $j = 1, \ldots, N(\varphi)$ where $\{\varphi(\mathcal{P}^j)\}_{j=1}^{N(\varphi)}$ is required to be a non-overlapping partition of $\mathcal{M}$, i.e., $\bigcup_{j=1}^{N(\varphi)} \varphi(\mathcal{P}^j) = \mathcal{M}$ and $\varphi(\mathcal{P}^i) \cap \varphi(\mathcal{P}^j) = \emptyset$ for any $i \neq j$. In addition, equal sizes are set for a voxel subset and its paired subcarrier subset, as given by $|\mathcal{P}^j| = |\varphi(\mathcal{P}^j)|$ for all j. A bijective mapping, $m(v)$, *satisfies* φ if and only if for any v, $v \in \mathcal{P}^j$ leads to $m(v) \in \varphi(\mathcal{P}^j)$. In this sense, φ encompasses all bijective mappings between $\mathcal{V}$ and $\mathcal{M}$ that maps $\mathcal{P}^j$ exactly to $\varphi(\mathcal{P}^j)$.

To completely determine the objective function of Problem P2 requires a subset-to-subset mapping φ_{sol} with $\mathrm{dom}(\varphi_{\mathrm{sol}}) = \{\mathcal{H}^j\}_{j=1}^{2^K}$, which specifies the mapped subcarrier subset for any homogeneous voxel subsets, say, $\mathcal{H}^j$, as $\varphi_{\mathrm{sol}}(\mathcal{H}^j)$. Denote the set of all bijective mappings that satisfy φ_{sol} as $\mathcal{C}(\varphi_{\mathrm{sol}})$, which by Lemma 2 yield the same objective value. Note that the union of $\mathcal{C}(\varphi_{\mathrm{sol}})$ for all possible φ_{sol} covers exactly the original solution space. It is therefore equivalent to consider the reduced-dimension space of φ_{sol} as the solution space of Problem P2. The dimensions of the new solution space are determined by the number of possibilities of disjoint set partitions $\{\varphi_{\mathrm{sol}}(\mathcal{H}^1), \varphi_{\mathrm{sol}}(\mathcal{H}^2), \ldots, \varphi_{\mathrm{sol}}(\mathcal{H}^{2^K})\}$ with $\bigcup_{j=1}^{2^K} \varphi_{\mathrm{sol}}(\mathcal{H}^j) = \mathcal{M}$ and the size of each subset fixed as $|\varphi_{\mathrm{sol}}(\mathcal{H}^j)| = |\mathcal{H}^j|$, which is calculated as $\frac{M!}{|\mathcal{H}_1|!|\mathcal{H}_2|!\cdots|\mathcal{H}_{2^K}|!}$. Thereby, we can achieve complexity reduction by orders of magnitude as compared with the original solution space, which encompasses all possible mappings between $\mathcal{M}$ and $\mathcal{V}$ and therefore has a size of $M!$.

C. Tree Construction

Finding the optimal solution to Problem P2 can be achieved by an enumeration of the compact solution space defined in the preceding subsection, which is still exponential in M due to the suggested NP-completeness of Problem P2. We propose to organize the solution enumeration into a tree search. A naive approach to tree construction would be to sequentially branch on the selection of subsets $\varphi_{\mathrm{sol}}(\mathcal{H}^j)$, but this method is unlikely to benefit from complexity reduction by node pruning. Instead, our approach is to sequentially branch on the local objective $f_k(\mathbf{A})$ by assigning subcarriers to certain groups of homogeneous subsets identified by the sparsity indicators of

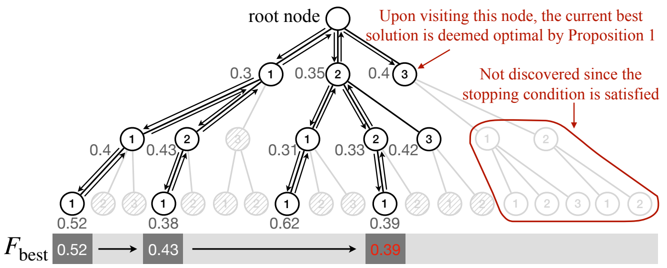


Fig. 2. An example of a search tree for the optimal solution of Problem P2, with maximum depth, i.e., the number of agents, $K = 3$. Nodes pruned by Proposition 2 are marked with strides.

the currently considered agent, which underpins the efficient tree-search algorithm with node pruning in Section VI-A. In the sequel, we index the K agents sequentially from 1 to K. However, such an agent ordering can be arbitrary, which affects not the optimality but the empirical complexity. In this aspect, an agent-ordering algorithm is presented in the next section. The search tree is illustrated in Fig. 2. For a general node w, let $d(w)$ denote its depth, i.e., the length of the (shortest) path connecting it to the root node. The maximum depth of the search tree equals K, and a node with depth K is defined as a *leaf node*.

1) Branching on Local Objectives: To begin with, we discuss the branches of the root node w_0 with depth 0, i.e., its set of child nodes with depth 1, by analyzing possible local objectives for agent 1. Recall that for agent 1, its associated local objective, $f_1(\mathbf{A})$ in (18), is fully determined by the subcarrier set assigned to agent 1's non-sparse voxels $\mathcal{V}_1$. Let r_k^q be the k-th element of $\mathbf{r}^q$, indicating whether the homogeneous subset $\mathcal{H}^q$ is sparse on agent k. We can express $\mathcal{V}_1$ as $\mathcal{V}_1 = \bigcup\{\mathcal{H}^q | r_1^q = 1\} \triangleq \mathcal{U}_1$, i.e., the union set of homogeneous subsets which are non-sparse on agent 1, and similarly $\mathcal{V} \setminus \mathcal{V}_1 = \bigcup\{\mathcal{H}^q | r_1^q = 0\} \triangleq \mathcal{U}_0$. This can be interpreted as dividing all homogeneous sets into two groups according to the sparsity on agent 1. The local objective values f_1 are then determined by φ_1 with $\mathrm{dom}(\varphi_1) = \{\mathcal{U}_1, \mathcal{U}_0\}$, which characterizes the assignment of subcarriers between non-sparse and sparse voxels of agent 1, and mathematically given by

$$f_1(\varphi_1) = \sum_{m \in \varphi_1(\mathcal{U}_1)} c_{1,m}. \tag{19}$$

The number of all possible φ_1, which generate (generally) distinct local objective values f_1, is equal to the number of size-$|\mathcal{U}_0|$ subsets of $\mathcal{M}$, i.e., $N(w_0) = \frac{M!}{|\mathcal{U}_0|!(M-|\mathcal{U}_0|)!}$. Each of the possible φ_1 is represented by one child node of the root node, w_j, where $j = 1, \ldots, N(w_0)$. The node set $\{w_1, \ldots, w_{N(w_0)}\}$ constitute all branches, or child nodes of the root node w_0.

Consider an arbitrary node, say w_j, and its associated subset-to-subset mapping is denoted as $\varphi_1^{w_j}$ with $\mathrm{dom}(\varphi_1^{w_j}) = \{\mathcal{U}_1, \mathcal{U}_0\}$. While $\varphi_1^{w_j}$ fixes agent 1's local objective value to $f_1(\varphi_1^{w_j})$ by (19), it only specifies the image of $\mathcal{U}_1, \mathcal{U}_0$, which are unions of homogeneous subsets, rather than each of $\{\mathcal{H}_j\}$, resulting in under-determined values for f_j, $j > 1$. We thus aim to further subdivide the current mapping, $\varphi_1^{w_j}$, to a finer granularity by considering the sparsity pattern of the next agent 2 such that f_2 is determined while f_1 fixed as $f_1(\varphi_1^{w_j})$. Since f_2 depends on the image of $\bigcup\{\mathcal{H}^q | r_2^q = 1\}$ and $\bigcup\{\mathcal{H}^q | r_2^q = 0\}$, to determine both f_1 and f_2 requires mapping each of $\{\mathcal{U}_{11}, \mathcal{U}_{10}, \mathcal{U}_{01}, \mathcal{U}_{00}\}$ to a subcarrier subset, where $\mathcal{U}_{b_1 b_2} \triangleq \bigcup\{\mathcal{H}^q | r_1^q = b_1, r_2^q = b_2\}$. Such a mapping is denoted as φ_2 with $\mathrm{dom}(\varphi_2) = \{\mathcal{U}_{11}, \mathcal{U}_{10}, \mathcal{U}_{01}, \mathcal{U}_{00}\}$. On the other hand, conditioning on $f_1 = f_1(\varphi_1^{w_j})$ requires $\varphi_2(\mathcal{U}_{11}) \cup \varphi_2(\mathcal{U}_{10}) = \varphi_1^{w_j}(\mathcal{U}_1)$ and $\varphi_2(\mathcal{U}_{01}) \cup \varphi_2(\mathcal{U}_{00}) = \varphi_1^{w_j}(\mathcal{U}_0)$. Under the above condition, the possible outcomes of f_2 while fixing f_1 constitute all possible branches of w_j. This branching procedure can be recursively applied until reaching a leaf node, which determines all $\{f_j\}_{j=1}^K$. In the sequel, the branching procedure for a general node is presented.

2) General Nodes: Consider a general node w. The steps to discover its child nodes are as follows. The node w with depth $d(w)$ represents a partial solution to Problem P2 characterized by a subset-to-subset mapping $\varphi_{d(w)}^w$. It domain is given as $\mathrm{dom}(\varphi_{d(w)}^w) = \{\mathcal{U}_{b_1 b_2 \cdots b_{d(w)}}\}_{b_i \in \{0,1\}}$, where $\mathcal{U}_{b_1 b_2 \cdots b_{d(w)}} \triangleq \bigcup\{\mathcal{H}^q | r_1^q = b_1, \ldots, r_{d(w)}^q = b_{d(w)}\}$. The local objectives $f_1, \ldots, f_{d(w)}$ are determined by $\varphi_{d(w)}^w$, as given by

$$f_j(\varphi_{d(w)}^w) = \sum_{m \in \bigcup_{b_j=1} \varphi_{d(w)}^w \left(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)}}\right)} c_{j,m}, \quad 1 \le j \le d(w). \tag{20}$$

Each node is recorded with its latest local objective value $f_{d(w)}(\varphi_{d(w)}^w)$. If w is a leaf node, i.e., $d(w) = K$, then φ_K^w yields a *solution* to Problem P2 as it determines the local objective for all K agents and thus the global objective, which is the maximum of single-agent objective values recorded with nodes on the path from the root node to node w. Mathematically, the resultant global objective is

$$F(\varphi_K^w) = \max_{j=1,\ldots,K} f_j(\varphi_K^w). \tag{21}$$

One can also verify that $\mathrm{dom}(\varphi_K^w) = \{\mathcal{H}^j\}_{j=1}^{2^K}$, implying that φ_K^w specifies the mapped subset of all homogeneous voxel subsets. If $d(w) < K$, then $\varphi(w)$ is a partial solution to Problem P2, suggesting that node w needs further subdivision for determining the next local objective value $f_{d(w)+1}$, of which the different possibilities constitute the set of child nodes of w. Each of these child nodes, say $\tilde{w}$, defines a subset-to-subset mapping with a finer granularity, $\varphi_{d(w)+1}^{\tilde{w}}$, with domain $\mathrm{dom}(\varphi_{d(w)+1}^{\tilde{w}}) = \{\mathcal{U}_{b_1 b_2 \cdots b_{d(w)+1}}\}_{b_i \in \{0,1\}}$. To keep the previous local objectives unchanged, we require $\varphi_{d(w)+1}^{\tilde{w}}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 1}) \cup \varphi_{d(w)+1}^{\tilde{w}}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 0}) = \varphi_{d(w)}^w(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)}})$ for all $b_1, \ldots, b_{d(w)} \in \{0,1\}$. In other words, constructing $\varphi_{d(w)+1}^{\tilde{w}}$ is equivalent to selecting a subset $\varphi_{d(w)+1}^{\tilde{w}}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 1}) \subset \varphi_{d(w)}^w(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)}})$ with size $|\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 1}|$ and assigning the de-selected ones as $\varphi_{d(w)+1}^{\tilde{w}}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 0})$ for all $b_1, \ldots, b_{d(w)} \in \{0,1\}$.

Furthermore, we can incrementally rank the child nodes of w in the order of ascending $f_{d(w)+1}(\varphi_{d(w)+1}^w)$ in an online manner, i.e., without listing and sorting all possible child nodes. This, as shown later, in most cases avoids enumerating all possible branches when combined with the depth-first search procedure and the derived pruning criteria. To achieve this is equivalent to finding the j-best solution for the following subcarrier selection problem:

$$
\begin{aligned}
\text{(P4}(w)\text{)} \quad & \min_{\{\varphi_{d(w)+1}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 1})\}} \sum_{m \in \bigcup_{b_j} \varphi_{d(w)+1}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 1})} c_{j,m} \\
& \text{s.t.} \quad |\varphi_{d(w)+1}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 1})| = |\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 1}|, \\
& \qquad \forall b_1, \ldots, b_{d(w)} \in \{0,1\}, \\
& \qquad \varphi_{d(w)+1}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 1}) \subset \varphi_{d(w)}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)}}), \\
& \qquad \forall b_1, \ldots, b_{d(w)} \in \{0,1\}.
\end{aligned}
$$

Since the selection of each $\varphi_{d(w)+1}(\mathcal{U}_{b_1 b_2 \cdots b_{d(w)} 1})$ is decoupled with each other, this can be achieved by standard algorithms such as priority queues.

The example of a search tree with number of agents $K = 3$ is illustrated in Fig. 2, where each node with depth d is marked with its corresponding local objective value f_d.

VII. Optimal VoCa-PPA: Fast Tree-Search

Given the compact search tree constructed in the preceding section, we present in this section two novel algorithms to accelerate the tree search via node pruning and agent ordering by exploiting the properties of VoCa-PPA.

A. Tree-Pruning Algorithm

The search tree constructed in the preceding section systematically organizes all possible solutions to Problem P2, represented by all its leaf nodes. However, in practice, it is computationally prohibitive to store all tree nodes and then perform an exhaustive search for the optimal solution. To address this issue, we hereby introduce an efficient tree search method combining DFS and problem-specific pruning criteria.

1) Depth-First Search with Priority: The DFS starts with visiting the root node w_0, and repeats visiting an unvisited child node of the last visited node, thereby increasing the search depth, until reaching a leaf node with the maximum depth. When a leaf node is visited, or the node visited has no unvisited child node, the algorithm backtracks to visit its parent node. In particular, when visiting a node with multiple child nodes, the one with the minimum local objective value is always prioritized. Not only is it a greedy heuristic which minimizes the cost for the current agent considered, but such a priority order can facilitate node pruning discussed in the sequel to reduce the number of nodes visited. Moreover, using the said method of incrementally ranking the nodes, the unvisited child nodes with lower priorities need not be explicitly defined and stored but are instantiated per request.

2) Stopping and Pruning Conditions: The optimal solution of the search tree minimizes the global objective among solutions associated with all of the tree's leaf nodes. The stopping and pruning conditions in the process of DFS build on the observation that every local objective constitutes a lower bound of the original objective, i.e., $F \geq f_j$ for all $j = 1, \ldots, K$. Thus, the objective value achieved by all descendants of node w is lower bounded by the single-objective value achieved by node w itself, i.e., $f_{d(w)}(\varphi^w_{d(w)})$. By applying this argument to the child nodes of the root node w_0, i.e., nodes with depth 1, we argue that any depth 1 node, say w_j, cannot develop into a better solution than φ_{best} if $f_1(\varphi_1^{w_j}) \geq F(\varphi_{\text{best}})$, and neither can any sibling nodes with a larger index than w_j as all child nodes are ranked in ascending order of the local objective. This results in the global optimal condition stated as follows.

Proposition 1 (*Stopping Condition*)**.** During a DFS over the defined search tree in Section VI-C, the current best solution to Problem P2, denoted as φ_{best}, is optimal if

$$f_1(\varphi_1^{w_j}) \geq F(\varphi_{\text{best}}), \tag{22}$$

where w_j is the last visited depth-1 node.

Proof: According to the sequence of node visiting in DFS, any unvisited leaf node, say node w', must be a child node of either w_j or a sibling node of w_j, say $w_{j'}$, with $j' > j$. In the former case, we have its solution value $F(\varphi_K^{w'}) \geq f_1(\varphi_1^{w_j}) \geq F(\varphi_{\text{best}})$. In the latter case, we have $F(\varphi_K^{w'}) \geq f_1(\varphi_1^{w_{j'}}) \geq f_1(\varphi_1^{w_j}) \geq F(\varphi_{\text{best}})$ due to the ascending order of local objectives. This completes the proof. ☐

In the searching process, the updating of φ_{best} is triggered if a newly found leaf node outperforms the current best solution, and the optimality condition (22) is checked if φ_{best} is updated or a new depth 1 node is visited.

The stopping condition for the global optimum is derived by bounding the global objective with the local ones achieved by child nodes of the root node w_0. On the other hand, each node, say node w, is associated with a *sub-tree* with itself being the root node. Similar to the original tree, define an objective function $F_w(\varphi_K^w) \triangleq \max_{d=d(w)+1\ldots,K} f_d(\varphi_K^w)$ of the sub-tree for a mapping φ_K^w associated with a leaf node, which considers only a subset of agents instead of all K agents. The optimal solution of the said sub-tree is defined to minimize $F_w(\varphi_K^w)$. Thus, a natural question is: *can we generalize Proposition 1 to the sub-trees to enable further node pruning?* This is justified by the intuition that in the search for the global optimum, it suffices to look at the optimum of a sub-tree instead of all solutions of the sub-tree since a global optimum must also be a local optimum. This is formalized in the following lemma, with its proof omitted for brevity.

Lemma 3. Let φ_K^w be a solution associated with a leaf node w of a sub-tree. Then, the optimal solution of the sub-tree $\varphi_K^{w^*}$ is at least at good as φ_K^w in terms of the global objective function F, i.e., $F(\varphi_K^{w^*}) \leq F(\varphi_K^w)$.

Thus, the enumeration of the sub-tree's nodes can be stopped if its optimal solution is already found using a condition similar to (22). The following proposition follows for pruning nodes which are unable to yield better solutions than visited nodes, with its proof omitted due to its similarity to that of Proposition 1.

Proposition 2. *(Pruning Criteria)* During a DFS over the defined search tree in Section VI-C, for a sub-tree associated with any node, say node w, its unvisited nodes can be pruned, i.e., need not be visited if

$$f_{d(w)+1}(\varphi_{d(w)+1}^{\tilde{w}_j}) \geq F_w(\varphi_{\text{best},w}), \tag{23}$$

where $\varphi_{\text{best},w}$ is the current best solution of the said sub-tree, and $\tilde{w}_j$ is the last visited child node of w.

A direct result follows: for a node w with depth $K-1$, i.e., whose child nodes are leaf nodes of the search tree, upon visiting its leftmost child node, the remaining child node can be immediately pruned as the objective function of the sub-tree, F_w is exactly f_K, giving $F_w(\varphi_{\text{best},w}) = f_K(\varphi_K^{w_1})$ which triggers Proposition 2. An example of tree search with pruning is illustrated in Fig. 2, where nodes pruned according to Proposition 2 are marked with strides and Proposition 1 is used for the optimality test.

B. Agent-Ordering Algorithm

Determining the agent ordering can significantly affect the number of nodes visited before the algorithm finds the optimal solution. Selecting the agent order can be translated to determining the agent priority. The agents can be arranged in the descending order of their priorities. To this end, agent 1 is given the highest priority because in the DFS process, the first child node we visit minimizes the objective f_1. Conditioned on f_1, we proceed to minimize the objective for agent 2 with the second highest priority, and so forth. Following the intuition that the bottleneck agent should be given high priority, we propose an ordering heuristic based on a priority indicator, which is defined for each agent, say agent i, as

$$\psi(i) = f_i^* = \min_{|\mathcal{M}|=|\mathcal{V}_i|} \sum_{m\in\mathcal{M}} c_{i,m}. \tag{24}$$

This indicator can be interpreted as the cost of the locally optimal mapping between non-sparse voxels and subcarriers for agent i without considering other agents. A larger $\psi(i)$ indicates poorer channel states or more non-sparse voxels that need to be transmitted for agent i. From the tree-searching perspective, $\psi(i)$ is the objective lower bound obtained by visiting the very first child node of the root node if agent i is visited first (see Proposition 1). As a result, letting agent 1 be the one with the highest priority indicator yields the tightest initial lower bound. Thus we propose to arrange the agent index in descending order of the priority indicator, i.e., assigning index such that $\psi(i) \geq \psi(i')$ for any $1 \leq i \leq i' \leq K$.

C. Fast Tree-Search Algorithm

The fast tree search for optimal VoCa-PPA (i.e., solving Problem P2), which incorporates the two algorithms in the preceding subsections, is summarized in Algorithm 2.

Algorithm 2: Fast Tree Search for Optimal VoCa-PPA

Input: Sparsity matrix $\mathbf{S}$ and channel matrix $\mathbf{H}$;
Prioritization: Determine the agent indexing as elaborated in Section VII-B;
Initialization: Root node w_0 with $d(w_0) = 0$;
φ^* = DFS(w_0);
Designate the optimal mapping $m^*(v)$ as an arbitrary one that satisfies φ^*;
Recover $\mathbf{A}^*$ from $m^*(v)$;

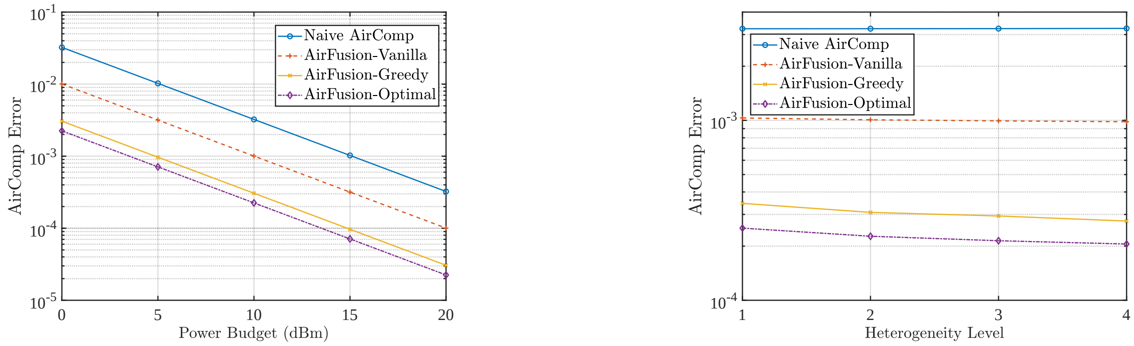
function DFS(w)
  for node $\hat{w}$ **in** all non-root parent nodes of w **do**
    Invoke Proposition 2 to prune all unvisited nodes of w if possible;
  if optimality test passes via Proposition 1 **then**
    $\varphi^* \leftarrow$ current best solution of the full tree;
    return optimal solution φ^*
  if $d(w) < K$ **then**
    while w has unvisited child nodes **do**
      Create child node $\tilde{w}$ with $d(\tilde{w}) = d(w) + 1$;
      $\varphi_{d(w)+1}^{\tilde{w}} \leftarrow$ the next best solution to P4(w);
      Call DFS($\tilde{w}$);
      if DFS($\tilde{w}$) returns optimal solution φ^* **then**
        return optimal solution φ^*;
  return continue search
end function

VIII. Experimental Results

A. Experimental Settings

We evaluate the performance of Spatial AirFusion on an ISEA system as illustrated in Fig. 1(a). The channel between the fusion center and K agents is assumed to follow i.i.d. Rician fading with the ratio between the power of line-of-sight (LoS) and non-LoS paths set as 3 dB and the path loss set as -15 dB. Following the Wi-Fi 6E standard, the total number of subcarriers in each resource block is $M = 26$, each spanning a bandwidth of $B_{\text{sub}} = 120$ kHz. The receive noise power per subcarrier is set as -40 dBm. Average pooling is adopted as the fusion function. The performance of Spatial AirFusion and baseline schemes is evaluated on the following two datasets.

- **Synthetic dataset.** The synthetic dataset involves $K = 4$ agents, each with a randomly generated feature map. The feature sparsity pattern $\mathbf{S}$ is a random binary matrix with $1/3$ probability for each of its elements to be non-zero, while it is ensured that each column has at least one non-zero element, i.e., at least one agent has non-zero observations on each voxel. The simulated performance is averaged over 1000 realizations with i.i.d. randomly generated channel matrices and sparsity patterns.
- **OPV2V dataset.** The OPV2V dataset [33] considers a vehicle-to-vehicle communication scenario where an ego vehicle fuses sensory features from helping vehicles detect other vehicles in a traffic scene. A data frame involves two to five vehicles, one of which is selected as the ego vehicle. Each vehicle captures a LiDAR point cloud of the surrounding environment and objects, which is projected onto the ego vehicle's coordinates and processed by a PointPillar backbone into a two-dimensional local spatial feature map with $V_{\mathsf{h}} = 256$ and $V_{\mathsf{w}} = 352$ being the number of voxels along the height and width of the perception region, respectively. Each voxel is associated with a feature vector with dimension $L = 128$, and thus the size of the local feature map is $128 \times 256 \times 352$. Therein, 50591 out of $V_{\mathsf{h}}V_{\mathsf{w}} = 90112$ voxels are empty over all samples in the dataset, regarded as dummy, and waived of transmission for all evaluated methods. Following [36], the ego vehicle wirelessly aggregates the feature map from all other vehicles and inputs the fused feature map into a region proposal network to obtain the vehicle detection result. The detection performance is



(a) (b)

Fig. 3. The performance of variants of Spatial AirFusion and naive AirComp on the synthetic dataset.

measured by the average precision (AP) at an Intersection over Union (IoU) threshold of 0.7. It is defined as the area under the precision-recall curve resulting from the said detection model, where a detected bounding box is considered true-positive if it overlaps with a ground-truth bounding box with an IoU higher than 0.7 [11].

We compare the performance of Spatial AirFusion controlled by VoCa-PPA with three benchmarking schemes, called *naive AirComp*, *digital air interface*, and *AirFusion-Vanilla*.

- **Naive AirComp.** Naive AirComp aggregates each voxel over the air on an assigned subcarrier similar to Spatial AirFusion, but does not involve the feedback of the feature sparsity matrix. Thus, all agents participate in AirComp over all subcarriers regardless of sparsity [8]. The subcarriers are allocated in sequential order and the receive SNR, which is fixed for all subcarriers in each coherence block, is chosen such that all agents' power constraints are satisfied.
- **Digital air interface.** The scheme corresponds to the conventional digital broadband orthogonal-access approach, where each agent is assigned a subset of subcarriers for feature uploading. On the agent side, each feature coefficient is encoded into 2 to 5 bits, depending on the desired latency-precision tradeoff, via uniform quantization. The radio resource management scheme with max-marginal-rate subcarrier assignment and equal power allocation, proposed and shown to be near-optimal in [37], is adopted. Then the communication latency is calculated using Shannon capacity given the assigned subcarrier and power. After receiving data from all agents, the server decodes the bits stream to reconstruct features.
- **AirFusion-Vanilla.** This scheme implements the system architecture and operations of AirFusion as in Section II and Section III, but pairs voxels with subcarriers in a sequential order without optimization. Given the default pairing, the power is optimally allocated using Lemma 1.

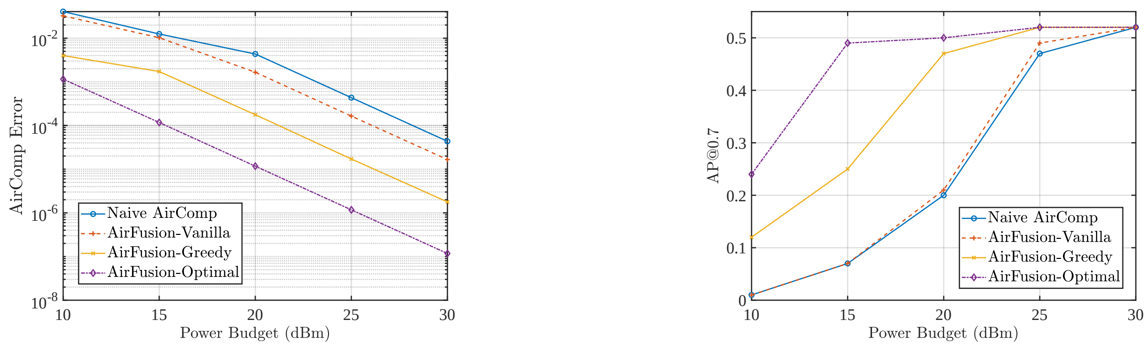
B. Performance Evaluation on Synthetic Datasets

First, the performance of Spatial AirFusion and naive AirComp is evaluated on the synthetic dataset. We test Spatial AirFusion controlled by Greedy VoCa-PPA in Algorithm 1 and Optimal VoCa-PPA in Algorithm 2, termed "AirFusion-Greedy" and "AirFusion-Optimal", respectively. The performance is measured by AirComp error, defined as the mean square error of feature aggregation results compared with the ideal ground-truth case, i.e., (1). The curves of AirComp error versus transmit power budget on each agent are plotted in Fig. 3(a). We observe that the sparsity-aware Spatial AirFusion protocol design can roughly reduce the AirComp error by 70% with AirFusion-Vanilla which does not optimize subcarrier allocation. This can be attributed to the reduction in communication overhead combined with smarter power allocation by exploiting sparsity of spatial features. On top of vanilla Spatial AirFusion, incorporating optimal VoCa pairing further improves the Spatial AirFusion performance as observed from the greedy and optimal cases. The small optimality gap between the algorithms renders greedy VoCa-PPA a close-to-optimal heuristic with low computational complexity.

Fixing the transmit power budget at 10 dBm, we vary the sparsity heterogeneity measured by the entropy of the empirical distribution of homogeneous subsets, as given by $-\sum_{q=1}^{2^K} \frac{|\mathcal{H}^q|}{V} \log \frac{|\mathcal{H}^q|}{V}$. It reaches the maximum when voxels are uniformly distributed to all homogenous subsets and zero when all voxels belong to the same homogeneous subset. The AirComp error performance against heterogeneity level is plotted in Fig. 3(b). We find a reduction in AirComp error when the heterogeneity level increases for Spatial AirFusion but not for naive AirComp that does not exploit spatial sparsity. The reason is that with a more heterogeneous voxel distribution, the proposed framework is provisioned with more degrees-of-freedom for VoCa pairing. This aligns with the intuition that in the extreme case where all voxels belong to the same homogeneous subset, the gain of the proposed approach diminishes since the homogeneity of voxels renders arbitrary VoCa allocation optimal.

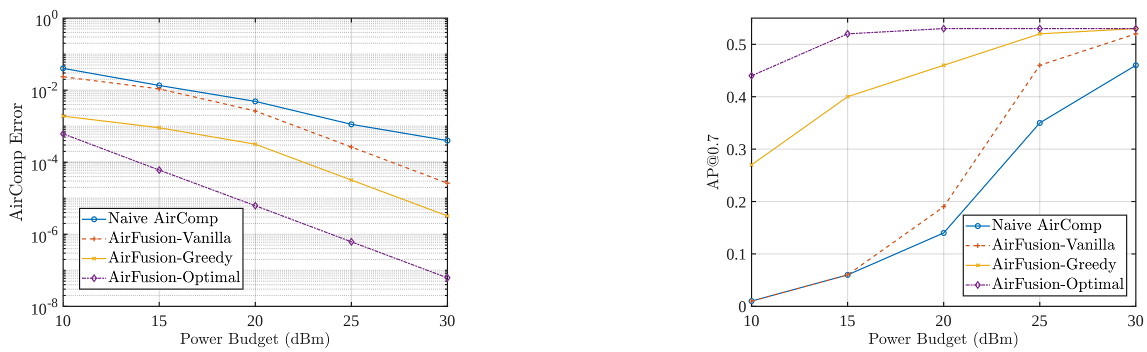
C. Performance Evaluation on the OPV2V dataset

The experimental results of Spatial AirFusion and naive AirComp obtained on the OPV2V dataset are presented in Fig. 4. The curves of AirComp error versus power budget for 3 and 4 participating vehicles, as plotted in Figs. 4(a) and 5(a), respectively, show a trend similar to that on the synthetic dataset where Spatial AirFusion significantly outperforms



(a) (b)

Fig. 4. The performance of variants of Spatial AirFusion and naive AirComp on the OPV2V dataset with number of CAVs $K = 3$.



(a) (b)

Fig. 5. The performance of variants of Spatial AirFusion and naive AirComp on the OPV2V dataset with number of CAVs $K = 4$.

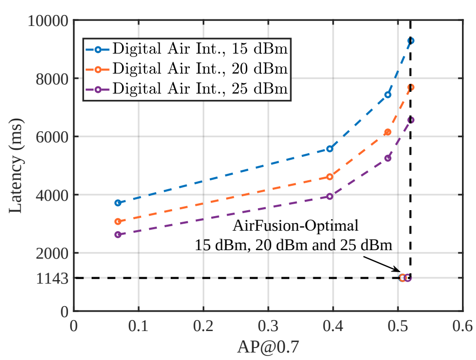


Fig. 6. The tradeoff between communication latency (in *millisecond* (ms)) and sensing performance measured in AP@0.7 for Spatial AirFusion and digital air interface different transmit power budgets on the OPV2V dataset.

naive AirComp. In terms of inference accuracy shown in Figs. 4(b) and 5(b), which is measured by the average precision at an intersection over union (IoU) threshold of 0.7, Spatial AirFusion delivers substantially better performance than naive AirComp. As the transmit power budget reaches 20 dBm, the accuracy of Spatial AirFusion saturates at about 50%, which is due to the inherent robustness of the perception model that tolerates a certain amount of distortion in the aggregated features without losing accuracy.

Finally, we compare the Pareto fronts of latency-precision tradeoff for digital air interface and Spatial AirFusion. The communication latency is defined as the average transmission time required to aggregate all features of a single perception instance. For AirComp, the said latency is independent of the transmit power and given by $L_{\rm H} = LN_{\rm v}/B_{\rm sub}$, where $\tilde{N}_{\rm v}$ is the average number of non-sparse voxels in each LiDAR frame. For digital air interface, the latency $L_{\rm D}$ depends on the total number of OFDM rounds required to transmit all features. Therein, a lower transmit power budget or poorer channels lead to lower communication rates and thus more required rounds. Given a certain transmit power budget, the latency-precision tradeoff in digital air interface is regulated by feature quantization resolution varying from 2 bits to 5 bits. The results are plotted in Fig. 6. We observe that under the same precision requirement and transmit power budget, Spatial AirFusion can reduce the latency by up to an order of magnitude. For example, digital air interface requires 5-bit quantization to achieve a target precision of 51% at 25 dBm power budget, where the resultant latency is $6,565$ ms. In contrast, Spatial AirFusion completes transmission in only $1,143$ ms, reducing the latency by 5.74 times. Two factors contribute to the latency reduction. The first is the exploitation of waveform superposition to avoid orthogonal transmission of each agent's feature. Second, through the sparsity pattern feedback, a substantial number of sparse voxels need not be transmitted in the case of Spatial AirFusion.

IX. Concluding Remarks

In this paper, we have presented the framework of Spatial AirFusion, a broadband task-oriented air interface targeting multi-agent environment perception tasks. The Spatial AirFusion protocol is developed to exploit spatial feature sparsity, a critical property of perception models, for enhancing communication efficiency. A mixed-integer programming problem, i.e., the VoCa-PPA problem, is formulated for joint allocation of power and subcarriers to maximize the minimum received SNR among all voxels. We solve this problem by designing a low-complexity greedy VoCa pairing algorithm and also an optimal tree search approach via exploiting useful properties of the problem structure. Experimental results show significant improvement in error suppression, sensing performance, and latency reduction compared with conventional approaches.

This work opens up several research directions on task-oriented communication schemes for ISEA. For example, digital Spatial AirFusion can be developed for better compatibility with existing digital systems, enabling incorporation of digital transmission techniques such as modulation and coding schemes. Another interesting topic is the interplay between Spatial AirFusion and more sophisticated physical layer techniques such as MIMO. In addition, integrating Spatial AirFusion with semantic data sourcing, which broadcasts low-dimensional queries to trigger transmission on semantically relevant agents, can further reduce communication cost [38].

References

[1] ITU-R, "Framework and overall objectives of the future development of IMT for 2030 and beyond," [Online] https://www.itu.int/rec/R-REC-M.2160-0-202311-I/en, 2023.

[2] X. Chen, K. B. Letaief, and K. Huang, "On the view-and-channel aggregation gain in integrated sensing and edge AI," to appear in *IEEE Trans. Wireless Commun.*, 2024.

[3] Z. Liu, Q. Lan, A. E. Kalør, P. Popovski, and K. Huang, "Over-the-air multi-view pooling for distributed sensing," *IEEE Trans. Wireless Commun.*, early access, Dec. 25, 2023, doi: 10.1109/TWC.2023.3343086.

[4] D. Wen, P. Liu, G. Zhu, Y. Shi, J. Xu, Y. C. Eldar, and S. Cui, "Task-oriented sensing, computation, and communication integration for multi-device edge AI," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 2486–2502, Mar. 2024.

[5] H. Xing, G. Zhu, D. Liu, H. Wen, K. Huang, and K. Wu, "Task-oriented integrated sensing, computation and communication for wireless edge AI," *IEEE Netw.*, vol. 37, no. 4, pp. 135–144, July/August 2023.

[6] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.

[7] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive IoT," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 57–65, Aug. 2021.

[8] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[9] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.

[10] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021.

[11] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recogn. (CVPR)*, Salt Lake City, UT, USA, Jun. 18–23, 2018.

[12] D. Rukhovich, A. Vorontsova, and A. Konushin, "ImVoxelNet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection," in *Proc. IEEE/CVF Winter Conf. Appl. Computer Vision (WACV)*, Waikoloa, HI, USA, Jan. 3–8, 2022.

[13] E. Xie, Z. Yu, D. Zhou, J. Philion, A. Anandkumar, S. Fidler, P. Luo, and J. M. Alvarez, "M^{2}BEV: Multi-camera joint 3D detection and segmentation with unified birds-eye view representation," [Online] https://arxiv.org/pdf/2204.05088.pdf, 2022.

[14] Q. Lan, Q. Zeng, P. Popovski, D. Gündüz, and K. Huang, "Progressive feature transmission for split classification at the wireless edge," *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 3837–3852, Jun. 2023.

[15] J. Shao, Y. Mao, and J. Zhang, "Task-oriented communication for multidevice cooperative edge inference," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 73–87, Jan. 2023.

[16] J. Shao and J. Zhang, "Communication-computation trade-off in resource-constrained edge inference," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 20–26, Dec. 2020.

[17] X. Huang and S. Zhou, "Dynamic compression ratio selection for edge inference systems with hard deadlines," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8800–8810, Sep. 2020.

[18] S. F. Yilmaz, B. Hasırcıoğlu, and D. Gündüz, "Over-the-air ensemble inference with model privacy," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Espoo, Finland, Jun. 26 – Jul. 1, 2022.

[19] Z. Zhuang, D. Wen, Y. Shi, G. Zhu, S. Wu, and D. Niyato, "Integrated sensing-communication-computation for over-the-air edge AI inference," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3205–3220, Apr. 2024.

[20] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

[21] O. Abari, H. Rahul, and D. Katabi, "Over-the-air function computation in sensor networks," [Online] https://arxiv.org/pdf/1612.02307.pdf, 2016.

[22] X. Cao, G. Zhu, J. Xu, and K. Huang, "Optimized power control for over-the-air computation in fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7498–7513, Nov. 2020.

[23] G. Zhu and K. Huang, "MIMO over-the-air computation for high-mobility multimodal sensing," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6089–6103, Aug. 2019.

[24] X. Cao, G. Zhu, J. Xu, and K. Huang, "Cooperative interference management for over-the-air computation networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2634–2651, Apr. 2021.

[25] M. Krouka, A. Elgabli, C. ben Issaid, and M. Bennis, "Communication-efficient split learning based on analog communication and over the air aggregation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 7-11, 2021.

[26] T. Qin, W. Liu, B. Vucetic, and Y. Li, "Over-the-air computation via broadband channels," *IEEE Wireless Commun. Lett.*, vol. 10, no. 10, pp. 2150–2154, Oct. 2021.

[27] Z. Lin, H. Liu, and Y.-J. A. Zhang, "CFLIT: Coexisting federated learning and information transfer," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8436–8453, Nov. 2023.

[28] N. Zhang and M. Tao, "Gradient statistics aware power control for over-the-air federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5115–5128, Aug. 2021.

[29] Y. Sun, Z. Lin, Y. Mao, S. Jin, and J. Zhang, "Channel and gradient-importance aware device scheduling for over-the-air federated learning," *IEEE Trans. Wireless Commun.*, early access, Dec. 4, 2023, doi: 10.1109/TWC.2023.3336277.

[30] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, "Over-the-air federated learning from heterogeneous data," *IEEE Trans. Signal Process.*, vol. 69, pp. 3796–3811, Jun. 2021.

[31] W. Liu, X. Zang, Y. Li, and B. Vucetic, "Over-the-air computation systems: Optimization, analysis and scaling laws," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5488–5502, Aug. 2020.

[32] M. Ehrgott, "A discussion of scalarization techniques for multiple objective integer programming," *Ann. Oper. Res.*, vol. 147, no. 1, pp. 343–360, 2006.

[33] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Philadelphia, PA, USA, May 23–27, 2022.

[34] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[35] L. Belhoul, L. Galand, and D. Vanderpooten, "An efficient procedure for finding best compromise solutions to the multi-objective assignment problem," *Comput. Oper. Res.*, vol. 49, pp. 97–106, Sep. 2014.

[36] 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 *Proc. ACM/IEEE Symp. Edge Comput.*, Washington, DC, USA, Nov. 7–9, 2019.

[37] K. Kim, Y. Han, and S.-L. Kim, "Joint subcarrier and power allocation in uplink OFDMA systems," *IEEE Commun. Lett.*, vol. 9, no. 6, pp. 526–528, Jun. 2005.

[38] K. Huang, Q. Lan, Z. Liu, and L. Yang, "Semantic data sourcing for 6G edge intelligence," *IEEE Commun. Mag.*, vol. 61, no. 12, pp. 70–76, Dec. 2023.