Few-Shot Multi-Agent Perception

Chenyou Fan[1], Junjie Hu[1], Jianwei Huang[1,2]
{fanchenyou, hujunjie, jianweihuang}@cuhk.edu.cn
[1]Shenzhen Institute of Artificial Intelligence and Robotics for Society, China
[2]The Chinese University of Hong Kong, Shenzhen, China

ABSTRACT

We study few-shot learning (FSL) under multi-agent scenarios, in which participating agents only have local scarce labeled data and need to collaborate to predict query data labels. Though each of the agents, such as drones and robots, has minimal communication and computation capability, we aim at designing coordination schemes such that they can collectively perceive the environment accurately and efficiently. We propose a novel metric-based multi-agent FSL framework which has three main components: an efficient communication mechanism that propagates compact and fine-grained query feature maps from query agents to support agents; an asymmetric attention mechanism that computes region-level attention weights between query and support feature maps; and a metric-learning module which calculates the image-level relevance between query and support data fast and accurately. Through analysis and extensive numerical studies, we demonstrate that our approach can save communication and computation costs and significantly improve performance in both visual and acoustic perception tasks such as face identification, semantic segmentation, and sound genre recognition.

CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**; *Music retrieval*; • **Computing methodologies** → **Multi-agent systems**; **Computer vision tasks**; *Image segmentation*.

KEYWORDS

few-shot learning, multi-agent perception, semantic segmentation, image and audio classification

ACM Reference Format:
Chenyou Fan[1], Junjie Hu[1], Jianwei Huang[1,2], {fanchenyou, hujunjie, jianweihuang}@cuhk.edu.cn, [1]Shenzhen Institute of Artificial Intelligence and Robotics for Society, China, [2]The Chinese University of Hong Kong, Shenzhen, China, . 2021. Few-Shot Multi-Agent Perception. In *Chengdu '21: ACM Multimedia, Oct 20–24, 2021, Chengdu, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recent research has achieved great advances in single-agent visual perception tasks such as image classification [14, 20], object detection [35, 36], semantic segmentation [4, 29] and action recognition [39, 46]. However, in many circumstances, multiple agents are observing the environment from different perspectives simultaneously. Comparing to the single-agent case, multi-agent perception (MAP) has the advantage of being able to share useful information among the participating agents through inter-agent communications and augment the observation of a same scene from different perspectives, as well as expanding the total scope with multiple scenes. Hereby, one key research topic in multi-agent learning is to establish an effective communication mechanism to share observations and coordinate actions among participating agents.

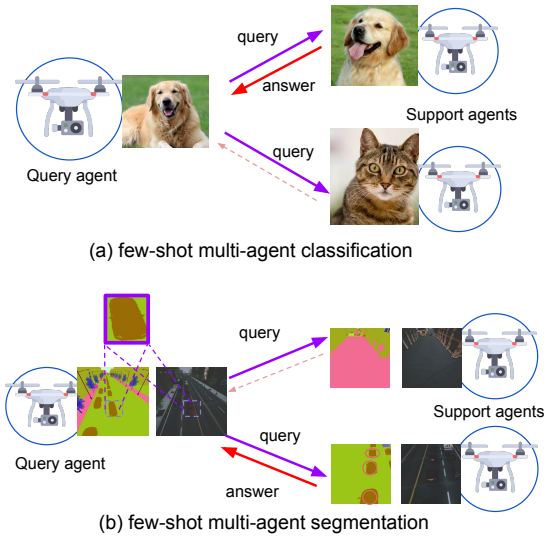


Figure 1: Demo of few-shot multi-agent perception Tasks.

Existing studies of multi-agent learning [7, 16, 18, 41] have achieved good progress in improving the effectiveness of communications with deep-learning based approaches. The key idea is to learn a shared deep neural network (DNN) for agents to encode scenes to features, then aggregate features from all agents based on attention-mechanism [15, 30, 48], and finally decode the fused feature for downstream tasks such as perception or controlling. However, this process is data-driven and requires plenty of training examples.

In reality, it can be very costly to label sufficient training data, e.g., identify and label objects in the sensory data (e.g., point clouds) collected with self-driving cars. Also, the scenes can be highly dynamic in which a single agent may encounter distinct objects just one or a few times. These observations motivate us to consider the following question: *"How to make multi-agent perception effective in the data-scarce scenario?"*.

We formulate this question as a novel few-shot multi-agent perception (**FS-MAP**) task and consider a general and practical setting:

Chengdu '21, Oct 20–24, 2021, Chengdu, China

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

each agent owns just a few labeled examples as support data, and it also observes unlabeled query examples. We define FS-MAP as a task for the agents to predict labels for query examples by learning to collaborate and search for the most relevant support data through inter-agent communications. To our best knowledge, we are the first of considering this practically important yet under-explored research topic, and we will provide a general framework of solving learning tasks under the studied scenarios.

In multi-agent scenarios, the same object of interest may appear in different regions with different sizes in images taken from different perspectives, for example, images taken by multiple heterogeneous agents such as UAVs (Unmanned Aerial Vehicles) and UGVs (Unmanned Ground Vehicles) from different heights and distances with various camera angles. Thus, it is critical to propose a robust distance metric that can measure the similarity between query and support data with certain translation and orientation invariance. To achieve this, we design a novel multi-agent metric-learning framework to tackle the FS-MAP tasks. We first extract the fine-grained 3-D visual or audio feature maps which preserve the spatial information, then we broadcast the query features to support agents to evaluate the relevance between query and support data with a fine-grained distance metric. Specifically, we propose to formulate the feature matching as a Regularized Optimal Transportation (RegOT) [5] task and solve it efficiently. The most relevant support data can thus assign their labels to the corresponding query examples.

However, transmitting feature maps across agents would bring in high communication costs and delays, especially when the cross-agent bandwidth is limited. To tackle this issue, we design to extract and transmit compact feature maps for query data and extract large feature maps for local support data to compensate for information loss. We can flexibly set the feature sizes to reach optimal performance with constrained communication resources.

In this paper, we will demonstrate that our framework is not only communication and computation efficient, but also significantly outperforms existing methods.

In conclusion, our contributions include:

- We are the first to consider a critical but under-explored task of how to effectively learn visual and acoustic perception tasks from only a few training examples in multi-agent scenarios.
- We solve the challenge of collaborating distributed agents for learning few-shot tasks by proposing a unified framework that integrates multi-agent communication and metric learning.
- To reduce cross-agent communication costs, we propose to generate asymmetric query and support feature maps to balance perception accuracy and bandwidth usage.
- To robustly measure the relevance of structured query and support data, we propose a novel distance metric with invariance to translation and viewpoints.
- Our approach significantly outperforms state-of-the-art methods by 10%-15% on segmentation and classifications tasks upon multimedia data including images and sounds.

2 RELATED WORK

Multi-agent learning has a broad scope of topics. Our work is closely related to the topics of learning communication protocols [7, 12, 16, 18, 26, 33, 41, 44] to improve the effectiveness of collaboration, as well as learning perception tasks [17, 26, 27]. VAIN [16] proposed to use kernel-based attention to measure the weight of each agent's message. TarMAC [7] used signature-based attention [48] to decouple query and key features to provide more flexibility of the communication such as selecting which other agents to communicate with. When2Com [27] further considered reducing bandwidth usage by using asymmetric query and key sizes. Our communication design builds upon the existing signature-based attention mechanism but further considers improving the performance by communicating fine-grained image feature maps. We choose asymmetric query and key sizes and cross-validate the feature map resolution to strike the balance of performance and communication cost. Also, our work adopts the widely used centralized training and decentralized execution paradigm [10, 41].

Few-shot learning (FSL) is a task of learning new skills with very few labelled training samples. Our work is closely related to the metric-based few-shot learning approaches [11, 13, 40, 45, 49, 51, 53] which focus on learning a good metric in feature space, such that data samples of different classes can be distinguished. FSL has been successfully applied in image classification [8, 21, 23, 31, 34, 43] and semantic segmentation [22, 25, 50] tasks. MPNet [22] considers a centralized few-shot segmentation approach which computes the dot products between query and support image segments to provide the attention of each spatial location. Instead of learning similarity of image patches, we consider the data of a same class as a discrete probability measure, and explicitly learn to represent data of same labels to have close empirical measures which can be calculated by the Wasserstein distance. By optimizing the representation of each class, our approach better establishes the feature space to have tight intra-class representations, which is critical in learning with few-shot examples. In addition, our work focuses on building a general multi-agent few-shot learning framework that is applicable for a broad scope of multimedia recognition tasks, such as face identification, semantic segmentation, audio recognition, etc.

Optimal Transport (OT) theory and Wasserstein distance define a family of advanced distance metrics that have recently been used to compare similarity between two structured data samples such as images [2, 24, 32, 54**? ?** , 55]. However, the computation of OT is complex and existing studies formulated it as linear programming task [32, 37, 53] which has a high time complexity $O(d^3 \log d)$ with d as the dimension of the feature. Our approach approximates the distance with an entropic regularization term, which turns it into a strictly convex problem that can be solved efficiently with a time complexity of only $O(d^2 \log d)$.

3 FS-MAP TASKS AND DEFINITIONS

In a general FS-MAP setting, each agent can have a few labeled support data instances as well as unlabeled query instances of arbitrary classes. We consider a simplified scenario in which each agent owns a few labeled examples as support data for ONE category, which is non-overlapping with each other. An agent is said to support a category if it holds support data of that category. We adopt this assumption of *one support category per agent* to facilitate the discussion. Later, we will show that our approach can extend to the general case that one agent can support multiple classes.

We formulate the few-shot multi-agent perception (FS-MAP) task formally now. Following the conventions of few-shot learning studies, we define FS-MAP as a C-way K-shot N-agent learning task. Each agent i could observe C_i distinct categories and each category has K labeled samples. The total C categories are covered by all agents such that $\sum_{i\in\mathcal{N}} C_i = C$. With the *one support category per agent* assumption, we simply have $C_i = 1$ and $C = N$. We show that this definition of FS-MAP can generalize to various perception tasks, among which we describe three typical multimedia perception tasks considered in this paper.

- **Image classification** is to predict the label of the query data out of C classes, e.g., the toy example in Fig. 1(a). Typically, we consider the face identification task to match face images to the correct one out of C identities.
- **Image segmentation** is to predict each pixel's class label out of C classes in the query image, e.g., assigning the "car" label for pixels in the highlighted area as shown in Fig. 1(b).
- **Musical genre classification** is to predict a soundtrack's genre out of C total genre categories. Specifically, we convert sound waves to spectrograms and consider the acoustic perception task as a special image classification task.

As a real-world example, we consider the task of searching for lost children in crowded scenes with police equipment such as drones and patrol robots. The parents first provide one photo for their child as the query instance. Then the police send out multiple drones, robots, and humans to different zones to recognize human faces and identify whether their observations could match the lost child's query photo. This distributed execution with multiple agents can significantly improve the efficiency of face identification. Note that we allow using as few as one query image (of the lost child) and one support image (of each observed scene participant) to perform the few-shot identity matching. We illustrate this in Fig. 1(a), with cartoon images instead of human faces for privacy concerns.

4 OUR APPROACH

In this section, we introduce a unified framework to tackle our proposed FS-MAP in detail. We first give an overview of our framework and elaborate on the design details in the 1-shot learning paradigm. Then we extend it to the general multi-shot learning paradigm and multi-class support data setting.

4.1 Model overview

We show the overview of our framework in Fig.2. Our framework has three main components: 1) a backbone Convolutional Neural Network (CNN) f^{bone} as a feature extractor which encodes images to hidden feature maps; 2) a query sub-network f^{qry} which encodes hidden feature maps to compact query feature maps, and 3) a key sub-network f^{key} which encodes hidden feature maps to large-size key feature maps. As we adopt the *centralized training and decentralized execution* strategy [10, 41], these modules are shared across all agents during query execution time.

We adopt the names "query" and "key" features by following TarMAC [7], and we call this design as *signature-based communications*. We also denote the unlabeled query images of a query agent u as $\boldsymbol{X}_u$, and support images of each support agent v as $\boldsymbol{X}_v$. With the assumption of *one support category per agent*, each support agent

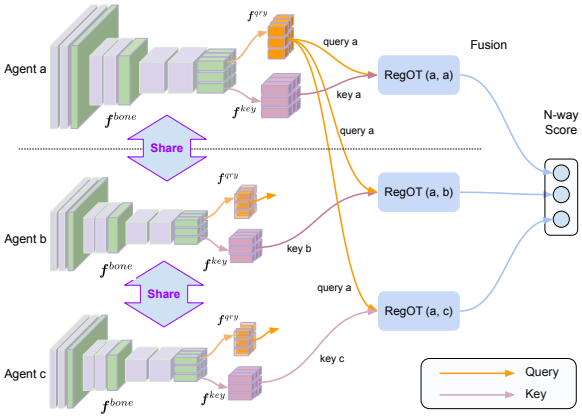


Figure 2: Overview of FSL-MAP architecture, including a shared backbone network f^{bone} for generating $3-D$ feature maps, a key network f^{key} for generating key features, a query network f^{qry} for generating query features, and a RegOT module for measuring the distance between query and support data.

Figure 3: School bus images of various camera viewpoints.

v also corresponds to v-th category. To simplify notations, we will denote $v \in N$ as abbreviation of $v = \{1, ..., N\}$.

4.2 Feature generation and broadcasting

To process a query or support image, we firstly extract their 3-D hidden feature maps $\boldsymbol{h}_u, \boldsymbol{h}_v \in \mathcal{R}^{D_h \times H_h \times W_h}$ with backbone network f^{bone} respectively, such that $\boldsymbol{h}_u = f^{bone}(\boldsymbol{X}_u)$ and $\boldsymbol{h}_v = f^{bone}(\boldsymbol{X}_v)$, in which D is channel size and $H \times W$ is spatial resolution.

Then for the query data of agent u, we generate its *query* feature $\boldsymbol{q}_u = f^{qry}(\boldsymbol{h}_u)$ with the query sub-network. For the support data of agent v, we generate its *key* feature $\boldsymbol{k}_v = f^{key}(\boldsymbol{h}_v)$ with the key sub-network. We will use *key* feature and *support* feature interchangeably to denote $\boldsymbol{k}_v$. The feature dimensions are $\boldsymbol{q}_u \in \mathcal{R}^{D_q \times H \times W}$ and $\boldsymbol{k}_v \in \mathcal{R}^{D_k \times H \times W}$. We choose their spatial resolution $H \times W$ to be the same (e.g., 8×8), while let their channel sizes D_q and D_k be asymmetric such that $D_q \ll D_k$, which in default are 32 and 1024.

In our design, the query features will be broadcasted from query agents to all support agents so that a compact $\boldsymbol{q}_u$ with a small channel size D_q will save bandwidth usage while preserving spatial resolution. Also, a large channel size D_k for key features will compensate for the accuracy loss caused by using small query features. For example, the cost of transmitting feature maps of size $32 \times 8 \times 8$ will be equal to sending a 1-D feature vector of length 2048 floats, a.k.a, the bandwidth usage 8 MegaByte per frame (MBpf). We can flexibly set the channel sizes in real applications for different channel capacities. We will discuss the trade-off of channel size and resolution in the ablation study.

Using coarse-grained 1-D image features to represent data as in previous work [7, 16, 26] would bring inferior performance in FSMAP as real-world multi-agent data is inherently heterogeneous and misaligned. For example, in Figure 3, the same yellow school bus appears in different image regions with distinct sizes and positions due to different camera viewpoints. In the next section, we design to utilize the rich spatial information in the 3-D feature maps to perform fine-grained feature matching between the query and support data.

4.3 Structured matching of two feature maps

In the previous step, the query agent u has broadcasted its query feature $\boldsymbol{q}_u$ to the support agents as the receiving ends. In this section, we explain how to measure the similarity between query feature $\boldsymbol{q}_u$ and support feature $\boldsymbol{k}_v$, under multi-agent scenarios. We propose a novel fine-grained metric-learning approach based on the Optimal Transport (OT) [6] which considers the similarity between two structured data representations as the minimum cost of transporting all units from one data distribution to the other.

We use region i to denote the i-th spatial location in a feature map of resolution $H \times W$, and use $\boldsymbol{q}_{u,i} \in \mathcal{R}^{D_q}$ to denote i-th feature vector in feature maps $\boldsymbol{q}_u$. We call the i-th region in a query feature map as source (**src.**) node i, and j-th region in support feature map as destination (**dst.**) node j. We propose a 3-step procedure to calculate the minimum cost of moving the total units from all src. nodes in query features to dst. nodes in support features.

Step 1: Region-wise similarity measure. In first step, we calculate the similarity between every pair of src. and dst. nodes, indicating the region-wise similarity between query and support data. Specifically, for every region pair (i, j), we compute the dot-product of query feature $\boldsymbol{q}_{u,i}$ from src. node i, with the key feature $\boldsymbol{k}_{v,j}$ from dst. node j. Since we use asymmetric query and key features, they could be of different dimensionality ($D_q \neq D_k$). We utilize the general dot product [30] to calculate the cosine similarity

$$a_{uv,ij} = \frac{\boldsymbol{q}_{u,i}^T \boldsymbol{W}_g \boldsymbol{k}_{v,j}}{\|\boldsymbol{W}_g^T \boldsymbol{q}_{u,i}\| \|\boldsymbol{k}_{v,j}\|}, \quad i, j \in HW \tag{1}$$

in which $\boldsymbol{W}_g \in \mathcal{R}^{D_q \times D_k}$ is a learnable parameter for matching dimensionality of query and key vectors; $i \in HW$ is abbreviation of $i \in \{1, ..., HW\}$. The cost of matching each region pair (i, j) can be conveniently defined as

$$c_{uv,ij} = 1 - a_{uv,ij} \ . \tag{2}$$

We use $\boldsymbol{C}_{uv} \in \mathcal{R}^{HW \times HW}$ to denote the costs between every pair of src. and dst. nodes, i.e., the costs of moving one unit from each region of query feature map to each region of support feature map.

Step 2: Node weight assignment. The next step is to determine the total weight of each src. and dst. node, which stands for the importance of a spatial region. The intuition is that a dst. node's weight is highly associated with its relevant src. nodes, e.g., a dst. node with school bus representation should have high importance if one or multiple src. nodes also have school buses. Thus, we determine the reciprocal src. and dst. node weight $s_{u,i}, d_{v,j}$ as the average of total matching score such that

$$\begin{aligned}
\hat{s}_{u,i} &= \max\left(\boldsymbol{q}_{u,i}^T \boldsymbol{W}_g \frac{\sum_{j=1}^{HW} \boldsymbol{k}_{v,j}}{HW}, \eta\right), \ \ s_{u,i} = \frac{\hat{s}_{u,i}}{\sum_{i=1}^{HW} \hat{s}_{u,i}} \\
\hat{d}_{v,j} &= \max\left(\left(\frac{\sum_{i=1}^{HW} \boldsymbol{q}_{u,i}}{HW}\right)^T \boldsymbol{W}_g \boldsymbol{k}_{v,j}, \eta\right), \ \ d_{v,j} = \frac{\hat{d}_{v,j}}{\sum_{j=1}^{HW} \hat{d}_{v,j}} \\
\boldsymbol{s}_u &= \{s_{u,i}, i \in HW\}, \ \boldsymbol{d}_v = \{d_{v,j}, j \in HW\}
\end{aligned} \tag{3}$$

in which η is a small number (e.g., $1e^{-3}$) to keep the weights positive. The $\boldsymbol{s}_u, \boldsymbol{d}_v$ denote the weights over the entire spatial regions for query and support feature maps respectively.

Step 3: Distance of two feature maps. We now define the distance of two feature maps as the minimum cost of transporting the src. node weights of query data to the dst. nodes. With the concept of *regularized optimal transportation distance* $\text{regOT}(\cdot, \cdot)$, we define the distance $\text{regOT}(u, v)$ between query data u and support data v as

$$\begin{aligned}
\text{regOT}(u, v) &= \min_{\boldsymbol{P} \in \mathcal{U}_{\boldsymbol{s},\boldsymbol{d}}} \langle \boldsymbol{P}, \boldsymbol{C}_{u,v} \rangle - \frac{1}{\lambda} H(\boldsymbol{P}) \\
\mathcal{U}_{\boldsymbol{s},\boldsymbol{d}} &:= \{\boldsymbol{P} \in \mathcal{R}_+^{n \times n} : \boldsymbol{P}\mathbf{1} = \boldsymbol{s}_u, \ \boldsymbol{P}^\intercal \mathbf{1} = \boldsymbol{d}_v\}
\end{aligned} \tag{4}$$

in which $\langle \cdot, \cdot \rangle$ is element-wise product, $\boldsymbol{P} \in \mathcal{R}^{HW \times HW}$ is the transportation plan and $H(\boldsymbol{P}) = -\sum_{i,j} p_{ij} \log p_{ij}$ is its entropy. The terms $\boldsymbol{s}_u$ and $\boldsymbol{d}_v$ are the node weights defined in (3). The feasible set $\mathcal{U}_{\boldsymbol{s}_u,\boldsymbol{d}_v}$ contains all possible plans that move src. node weights to dst. nodes.

The objective is to search for an optimal plan $\boldsymbol{P}^*$ which minimizes the total cost given by $\langle \boldsymbol{P}, \boldsymbol{C}_{u,v} \rangle$ as well as an entropy term that encourages the smoothness of the plan. Lemma 1 [5] in supplementary material shows that the search for $\boldsymbol{P}^*$ is a convex optimization problem with a global minimizer that can be decomposed to certain diagonal forms. We show that there exists an efficient and bounded iterative algorithm, called Sinkhorn-Knopp approach [19], to approximate the optimal transportation plan $\hat{\boldsymbol{P}}$ as in Algorithm 1. The intuition is to alternatively refine two diagonal matrices $\boldsymbol{X}, \boldsymbol{Y}$ implied by Lemma 1 to minimize the total transportation cost while satisfying the constraints.

Algorithm 1: RegOT $(\boldsymbol{C}, \ \boldsymbol{s}, \ \boldsymbol{d}, \ \lambda, \ n, \varepsilon)$

Output: Approximated optimal transport plan $\boldsymbol{X}$.
$\boldsymbol{A} \leftarrow \exp(-\lambda \boldsymbol{C}), \quad \boldsymbol{P} \leftarrow \mathcal{N}(0, 1)$
$\boldsymbol{u}^0 \leftarrow \mathbf{0}, \ \ \boldsymbol{v}^0 \leftarrow \mathbf{0}, \ \ \boldsymbol{P}^{(\mathbf{0})} \leftarrow \boldsymbol{P} / \|\boldsymbol{P}\|_1$
while $||\boldsymbol{P}^{(\boldsymbol{k})}\mathbf{1} - \boldsymbol{s}||_1 + ||(\boldsymbol{P}^{(\boldsymbol{k})})^T \mathbf{1} - \boldsymbol{d}||_1 > \varepsilon$ **do**
  $k \leftarrow k + 1$
  $\boldsymbol{u} \leftarrow \log(\frac{\boldsymbol{s}}{\boldsymbol{P}^{(k-1)}\mathbf{1}}), \ \ \boldsymbol{u}^k \leftarrow \boldsymbol{u} + \boldsymbol{u}^{k-1}$
  $\boldsymbol{v} \leftarrow \log(\frac{\boldsymbol{d}}{\boldsymbol{P}^{(k-1)\intercal}\mathbf{1}}), \ \ \boldsymbol{v}^k \leftarrow \boldsymbol{v} + \boldsymbol{v}^{k-1}$
  $\boldsymbol{P}^{(k)} \leftarrow \text{diag}(\exp(\boldsymbol{u}^k)) \ \boldsymbol{A} \ \text{diag}(\exp(\boldsymbol{v}^k))$
end
Return $\hat{\boldsymbol{P}} \leftarrow \boldsymbol{P}^{(k)}$.

THEOREM 1. *Algorithm 1 produces an approximated $\hat{\boldsymbol{P}}$ s.t.*

$$\langle \hat{\boldsymbol{P}}, \boldsymbol{C} \rangle \leq \min_{\boldsymbol{P} \in \mathcal{U}_{\boldsymbol{s},\boldsymbol{d}}} \langle \boldsymbol{P}, \boldsymbol{C} \rangle + \epsilon \ , \tag{5}$$

in $O(n^2(\log n)(\epsilon^{-3}))$ where $n = HW$, the cost matrix $\boldsymbol{C}$ is defined by (2), and node weights $\boldsymbol{s}, \boldsymbol{d}$ are defined by (3).

PROOF. The cost matrix $\boldsymbol{C}$ given by (2) has $\|C\|_\infty \leq 2$, also both $\boldsymbol{s}, \boldsymbol{d}$ given by (3) sum to 1. By applying [1, Theorem 1], Algorithm 1 has a bounded time complexity of $O(n^2(\log n)(\epsilon^{-3}))$. □

In practice, we choose a reasonably large stopping criterion in Algorithm 1 such as $\varepsilon = 0.1$ so that it computes the plan fast. Also, the operations in Algorithm 1 are fully differentiable thus the gradients can be back-propagated to update the network parameters.

We also compare our metric with recent studied Earth Mover's Distance [32, 53] that solves the original OT

$$OT(u, v) = \min_{\boldsymbol{P} \in \mathcal{U}_{\boldsymbol{s},\boldsymbol{d}}} \langle \boldsymbol{P}, C_{u,v} \rangle \tag{6}$$

without the entropy term as in our Problem 4. The original OT task is a linear programming, which is usually solved by interior-point methods with a time complexity of $O(n^3 \log n)$ [32]; while our approach shaves a factor of n in time complexity. Furthermore, their method costs an enormous $O(n^4)$ memory usage in order to make it differentiable [3]. Our approach costs only $O(n^2)$ memory usage and is fully differentiable.

In summary, we have introduced the procedures for estimating the optimal transportation plan between two feature maps of image or sound data. We will illustrate that the estimated plan can be seamlessly integrated into multi-agent few-shot learning framework to measure the relevance of a query example with the support data owned by the participating agents.

4.4 1-shot multi-agent learning

In this section, we summarize the procedures of learning 1-shot perception tasks with our framework, including classification and segmentation. We will extend our approach to multi-shot learning in section 4.5.

We denote Z_u as a query image from query agent u, and X_v as the 1-shot support image of agent v. We generate their query and support features $\boldsymbol{q}_u$ and $\boldsymbol{k}_v$ respectively, and estimate their regularized optimal transportation plan $\hat{\boldsymbol{P}} = \{\hat{p}_{ij},\ i, j \in HW\}$ with Algorithm 1.

1-shot classification task. For a classification task, we compute the fine-grain structured similarity between query image Z_u and support image X_v as follows,

$$\psi_{uv} = \langle \hat{\boldsymbol{P}}, \boldsymbol{1} - \boldsymbol{C} \rangle = \sum_{i=1}^{HW} \sum_{j=1}^{HW} \hat{p}_{ij}(1 - c_{ij}) \ . \tag{7}$$

Thus we will have N pairwise similarity scores between query image u with every support agent $v \in \mathcal{N}$, which we denote as $\{\psi_{uv}, v \in \mathcal{N}\}$. We interpret these values as N-way probability scores, based on which we compute the cross-entropy loss such that

$$\ell^{cls}(Z_u, y) = -\log \frac{\exp(\psi_{uy})}{\sum_{v=1}^{N} \exp(\psi_{uv})} \ . \tag{8}$$

Here y is the ground truth label indicating the true category of the query data. In the multi-agent setting, y is equivalent to the corresponding agent that has the support data point. The result of the inference is to compute the predicted image label $\hat{y} = \arg\max_v \ \psi_{uv}$.

1-shot segmentation task. For segmentation task, we need to produce a class label for each region of the query image, and expand its resolution to original image size. The first step is to define the averaged similarity of each region i in query feature $\boldsymbol{q}_u$ to all regions

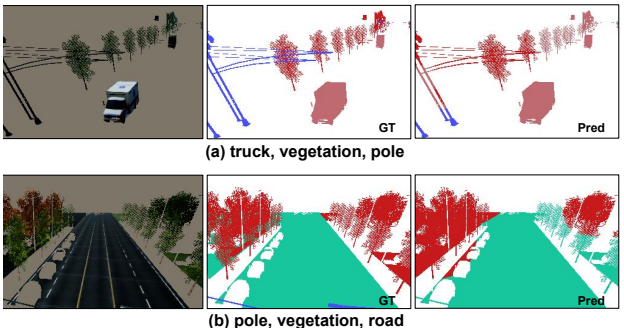


Figure 4: Sample images of FS-AirSim containing truck, vegetation, pole and road, with ground truth masks (mid) and predicted masks (right) with MAP-RegOT.

of a support image v as

$$(\boldsymbol{\varphi}_{uv})_i = \sum_{j=1}^{HW} \hat{p}_{ij}(1 - c_{ij}), \ \text{ for } i \in HW \ . \tag{9}$$

Thus $(\boldsymbol{\varphi}_{\boldsymbol{uv}}) \in \mathcal{R}^{HW}$ implies the similarity scores of all regions with label v. Since the query agent will broadcast to all support agents, we will have $\boldsymbol{\varphi}_u = \{\boldsymbol{\varphi}_{ui}, i \in \mathcal{N}\} \in \mathcal{R}^{N \times HW}$ which forms the N-way segmentation mask with resolution $H \times W$. To expand it to the original image's size $H_0 \times W_0$, we apply multiple transposed convolution layers [52] followed by a simple bi-linear upsampling upon $\boldsymbol{\varphi}_u$ such that $\boldsymbol{o}_u = \text{Upsample}(\boldsymbol{\varphi}_u)$ of size $\boldsymbol{o}_u \in \mathcal{R}^{N \times H_0 \times W_0}$. We can compute the pixel-wise cross-entropy loss to optimize the model in end-to-end fashion, such that

$$\ell^{seg}(Z_u, Y) = -\frac{1}{H_0 W_0} \sum_{i=0}^{H_0 W_0} \log \frac{\exp((\boldsymbol{o}_{uY_i})_i)}{\sum_{v=1}^{N} \exp((\boldsymbol{o}_{uv})_i)} \ , \tag{10}$$

in which Y_i is the ground truth label of i-th pixel of the query image, and $(\boldsymbol{o}_{uY_i})_i$ is the corresponding region's predicted score of the true label. The result of the inference is to compute pixel-wise label $\hat{Y}_i = \arg\max_v \ (\boldsymbol{o}_{uv})_i, i \in H_0 W_0$.

We summarize the complete 1-shot multi-agent perception procedures at execution time in Algorithm 2 in supplementary material.

4.5 K-shot multi-agent learning

We now extend our framework to K-shot multi-agent learning tasks, where each support agent owns multiple support images for each class. One naive way is to perform 1-shot learning K times to measure the relevance of all support images per class and take the highest score. However, this may lead to severe overfitting [40].

We adopt an early fusion strategy, which guides each support agent to learn one synthetic support image $\bar{X}_v$ for its class v based on all K support images. We randomly initialize $\bar{X}_v$ and iteratively update it with $\min_{\bar{X}_v} \ell(\bar{X}_v, v)$ for a fixed number of iterations (e.g., 10) to query for its true label v, with ℓ defined as (8) or (10). Specifically, $\bar{X}_v$ is sent from agent v to all support agents as a "query" image and gets updated as a normal 1-shot learning task. The purpose is to search for an optimal representative image for each class to distinguish its class from others best. During inference, we first synthesize $\bar{X}_v$ for each class locally on each agent, then we take it as a single versatile support image to answer queries so that the K-shot task converts to 1-shot.

4.6 General multiple support classes per agent

In Sec. 3, we assumed *one support category per agent* to facilitate discussion, i.e., each support agent v corresponds to the v-th class. We now generalize our framework to support multiple classes for each support agent.

In this case, each agent v supports a set of $|C_v|$ classes with K data samples per class. Support agent v can generate support features $\boldsymbol{k}_v^j, j \in C_v$. Once it receives a query feature $\boldsymbol{q}_u$, agent v will compute the similarity score with each support feature $\boldsymbol{k}_v^j$ individually and return the list with tuples $\{(s_j, j, v), j \in C_v\}$ of score, class index j, and agent index v back to the query agent u. The query agent u can determine the query data label by searching for the highest score in the combined score lists from all support agents.

5 EXPERIMENT

We evaluate our FS-MAP framework on distinct perception tasks and compare it with various state-of-the-art baselines to show its effectiveness. We report the results on two benchmark datasets for image segmentation and music genre classification tasks. We also collect a human face dataset with heterogeneous devices to verify face recognition task performance with distinct multi-agents. We perform ablation studies on parameter choices.

5.1 Datasets

We first briefly describe three datasets to be used for the evaluation of FS-MAP models.

FS-AirSim. We build the *FS-AirSim* dataset upon *AirSim-MAP* [26] which simulates flying multiple drones over a series of landmarks in the AirSim "CityEnviron" environment [38]. Our FS-AirSim contains 12K RGB frames of resolution 512×512 accompanied by semantic segmentation masks over 10 classes. They were recorded by 5-6 virtual drones from different perspectives in 118 scenes. We split the classes to 5 for training/validation, and the rest 5 for testing in a non-overlapping manner for few-shot learning purpose. Table 1 shows the class names in each split and the total number of frames of each class. We will evaluate both classification and segmentation tasks on this dataset.

FS-AirFace. We collect a few-shot face recognition dataset of 16 persons with UAVs and UGVs in four different scenes. As shown in Figure 5, we use a video camera mounted on a DJI Mavic to capture the videos from views in the air, and a camera on an automated patrol vehicle to capture the videos from views on the ground. We manually labeled 354 and 307 human faces from air and ground perspectives, respectively, and resize them to a resolution of 84×84. Table 2 shows the statistics of each split and the total number of frames of each class. We will evaluate the face identification task on this dataset. We also use the large-scale CelebA [28] face dataset to pre-train our backbone models instead of directly training from FS-AirFace from scratch.

GTZAN [47] is a widely used music genre dataset with soundtracks collected from diverse sources, including CDs, radio, microphones, etc. This dataset contains 10 genres such as blues, classical, pop, rock, and each genre has 100 16-bit Mono sound waves of 30 seconds. We split the genres into 8 for training/validation and the rest 8 for testing. We convert the sound waves to the time-frequency domain by FFT and extract the Mel spectrograms as the 2D acoustic features for the few-shot genre recognition task. Concretely, we set the FFT size 1024, the number of Mel scales 128 and split to multiple 128-sample chunks in the temporal dimension. Thus, each soundtrack is represented by a series of 2D acoustic features of resolution 128×128. In each column of Fig. 6, we show the spectrograms of 2 sampled soundtracks for each of four genres in different columns.

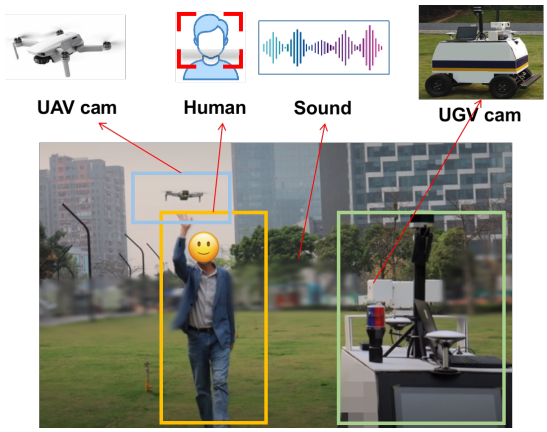


Figure 5: Data collection with air-ground collaboration. We build our FS-AirFace dataset upon the collected data.

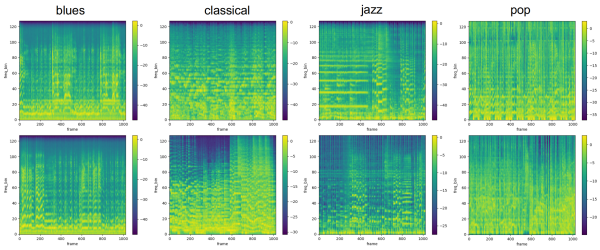


Figure 6: Mel spectrograms of 2 samples of 4 genres.

split \ labels	person	sidewalk	sky	building	car
train	91	4285	784	6180	5763
val	165	1558	568	2167	1928
split \ labels	road	pole	bus	truck	vegetation
test	1510	398	109	329	1162

Table 1: Statistics of FS-AirSim dataset.

split	Person Num	UAV Frames	UGV Frames	Total
train/val	8	199	161	360
test	8	155	146	301

Table 2: Statistics of FS-AirFace dataset.

5.2 Our approaches and baselines

We compare two approaches of our proposed FS-MAP framework. **MAP-RegOT** integrates our signature-based communication mechanism (4.2) and fine-grained metric-learning module RegOT(4.3) with smoothed matching results. **MAP-OT** is a baseline of MAP-RegOT which solves the original OT with LP solver as [53] with a much higher computational cost and non-smoothed matching results.

We compare our approaches with baselines that utilize different combinations of multi-agent communication mechanisms with FSL approaches to tackle the FS-MAP task. We choose the current SOTA communication designs **TarMAC** [7] and **When2Com** [26], and the current SOTA FSL approaches including MAML [9] and MTL [42] as representatives for optimization-based learners. In addition, we compare with state-of-the-art metric-based learners including ProtoNet [40] and RelationNet [43] for classification, as well as PANet [50] and MPNet [22] for segmentation. Note that MPNet [22] can also extend to distributed scenarios with its original attention design.

5.3 Implementation Details

For our approaches, we choose the ResNet-12 [14] as the backbone network f^{bone}, for fair comparison with previous FSL studies as MAML [9] and MTL [42]. The resolutions of input UAV images of FS-MAP, Mel spectrograms of GTZAN dataset, and face images of FS-AirFace dataset are 512×512, 128×128 and 84×84 respectively, and their extracted feature maps $\boldsymbol{h}$ are of sizes 8×8, 8×8 and 6×6 respectively, with a same channel size 512. For query and key sub-networks (f^{qry} , f^{key}), we use two 3-layer CNNs to project $\boldsymbol{h}$ to channel sizes $D_q = 32$ and $D_k = 1024$ with same resolutions as $\boldsymbol{h}$. We set the dimensions of When2Com [26] feature vectors to be the same with ours, and set the query size of TarMAC [7] to be same as key size ($D_q = D_k = 1024$) according to its model design. Otherwise, we follow their original settings for all baselines methods.

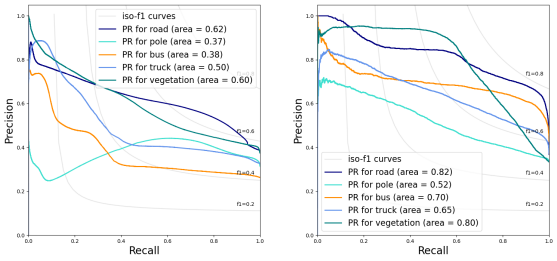
5.4 Results of few-shot segmentation

Method	3-Way 1-Shot		3-Way 5-Shot	
	Acc	IoU	Acc	IoU
When2Com+MAML [9, 26]	0.593	0.203	0.733	0.310
When2Com+MTL [26, 42]	0.652	0.259	0.735	0.321
TarMAC+MTL [7, 42]	0.660	0.310	0.752	0.328
TarMAC+PANet [7, 50]	0.661	0.292	0.762	0.335
MPNet [22]	0.705	0.287	0.770	0.346
MAP-OT (ours)	0.692	0.261	0.764	0.318
MAP-RegOT (ours)	**0.727**	**0.334**	**0.783**	**0.366**

Table 3: Segmentation results on FS-AirSim dataset.

In Table 3, we compare different methods with 3-way 1-shot and 5-shot semantic segmentation tasks on FS-AirSim. In our setting, each support agent is aware of one exclusive semantic label so that 3 agents together are aware of 3 classes. For a query image, the areas of interest are the unions of pixels of the 3 class labels. An example of a pair of support image and mask is shown in Fig. 4. We train all models to learn to predict correct labels for pixels of interest, and evaluate the segmentation performance with two metrics: the per-pixel accuracy (Acc) and the intersection-over-union (IoU) with true masks. We can observe that

- MAP-RegOT outperforms all other approaches in both Acc and IoU. It outperforms the MAP-OT by 5% and 2.6% in 1-shot and 5-shot tasks respectively, and even larger for other baselines.
- MAP-RegOT outperforms MPNet by 3% (0.727 v.s. 0.705) due to the better metric provided by RegOT, while both significantly outperform other baselines which do not consider fine-grained feature matching.



(a) Segmentation. **(b) Classification.**

Figure 7: PR curves for 1-shot tasks with MAP-RegOT.

- We show the precision-recall curve for each semantic class in Fig.7 (a). The APs for classes of more pixels in images such as *sky, road* are high, while APs for classes of *pole, bus* are relatively low. Indeed, small and rare objects are harder to localize [35] and thus decrease the overall IoUs.

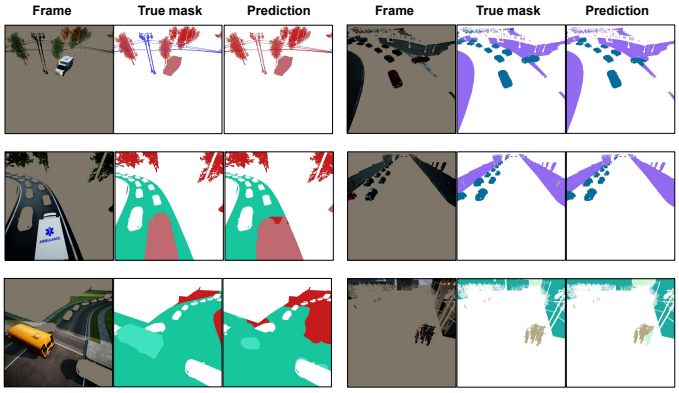


Figure 8: More segmentation results of FS-AirSim. The left group shows results on testing set, while the right group shows results on validation set.

5.5 Results of few-shot classification

We perform 5-way 1-shot and 5-shot classification tasks on FS-AirSim and FS-AirFace to evaluate image classification and face identification performance. We show the results with two metrics: the image classification accuracy (Acc) and the mean average precision (mAP) over all classes. We observe in Table 4 on FS-AirSim that

- MAP-RegOT outperforms other approaches in both metrics. It outperforms the second best MAP-OT by 7.7% (0.665 v.s. 0.617) and 2.5% (0.720 v.s. 0.702) for 1-shot and 5-shot tasks respectively.
- MAP-RegOT outperforms the SOTA combination of TarMAC and RelationNet by 8.3% (0.665 v.s. 0.614) and 10.8% (0.720 v.s. 0.650) for 1-shot and 5-shot tasks respectively, indicating the effectiveness of our fine-grained metric-learning approach.
- We show the PR curves in Fig.7 (b) and observe that the overall APs of classification are significantly higher than segmentation. This is because the label prediction is in image-level so that all regions of the image can provide hints to deduce the true label, which makes the task much easier.

Method	5-Way 1-Shot		5-Way 5-Shot	
	Acc	mAP	Acc	mAP
When2Com+MAML [9, 26]	0.458	0.413	0.482	0.443
When2Com+MTL [26, 42]	0.516	0.480	0.530	0.591
TarMAC+MTL [7, 42]	0.503	0.485	0.601	0.602
TarMAC+ProtoNet [7, 40]	0.531	0.424	0.684	0.607
TarMAC+RelationNet [7, 43]	0.614	0.623	0.650	0.657
MAP-OT (ours)	0.617	0.643	0.702	0.754
MAP-RegOT (ours)	**0.665**	**0.697**	**0.720**	**0.793**

Table 4: Classification results on FS-AirSim.

Method	5-Way 1-Shot		5-Way 5-Shot	
	Acc	mAP	Acc	mAP
When2Com+MTL [26, 42]	0.283	0.301	0.309	0.322
TarMAC+MTL [7, 42]	0.310	0.312	0.315	0.345
TarMAC+ProtoNet [7, 40]	0.596	0.642	0.602	0.643
TarMAC+RelationNet [7, 43]	0.564	0.665	0.627	0.687
MAP-OT (ours)	0.636	0.690	0.670	0.737
MAP-RegOT (ours)	**0.671**	**0.740**	**0.693**	**0.751**

Table 5: Face recognition results on FS-AirFace.

Method	3-Way 1-Shot		3-Way 5-Shot	
	Acc	mAP	Acc	mAP
When2Com+MTL [26, 42]	0.355	0.361	0.325	0.329
TarMAC+MTL [7, 42]	0.341	0.349	0.376	0.389
TarMAC+ProtoNet [7, 40]	0.498	0.512	0.541	0.558
TarMAC+RelationNet [7, 43]	0.503	0.521	0.566	0.624
MAP-OT (ours)	**0.579**	**0.612**	0.704	0.773
MAP-RegOT (ours)	**0.581**	**0.615**	**0.722**	**0.786**

Table 6: Music genre classification results on GTZAN.

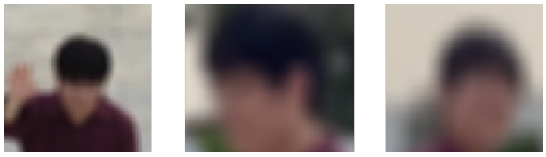


Figure 9: Face images taken from air (first) and ground (mid and right) viewpoints. Images are blurred for anonymity.

We consider the few-shot face identification tasks on the FS-AirFace dataset in Table 5. We observe that MAP-RegOT consistently outperforms MAP-OT (0.671 v.s. 0.636) and significantly outperforms the best coarse-grained baselines by more than 12.6% (0.671 v.s. 0.596) and 10.5% (0.693 v.s. 0.627) in 1-shot and 5-shot tasks. Note that the query face images and support face images are taken by UAVs and UGVs from different angles and perspectives, as shown in Fig.9. As our approach better considers the difference in query and support data's perspectives, it outperforms the baseline approaches naturally.

We find a similar trend for the few-shot music genre recognition tasks on GTZAN dataset, as shown in Table 6. We observe that MAP-RegOT consistently outperforms the baselines by more than 15% in both 1-shot and 5-shot tasks relatively. For two soundtracks of the same genre, their Mel spectrograms could capture similar time-frequency patterns but at different timestamps. A typical example is shown in column 1 of Fig.6. Our approach can better align the acoustic patterns such as crests and troughs in the frequency domain, thus it outperforms in matching soundtracks of same genres.

5.6 Discussion and visualization

Query size D_q	Resolution $H \times W$	Accuracy	Speed (fps)
8	16×16	0.969	160
32	8×8	**0.976**	**176**
128	4×4	**0.974**	172
512	2×2	0.965	**179**

Table 7: Ablation study of query feature resolutions.

Query vector size. In Table 7, we study the trade-off between channel size D_q and spatial resolution $H \times W$ of query feature maps. Given a fixed channel bandwidth 8MBpf (4.2), i.e., $D_q \times H \times W = 2048$, we report the 1-shot validation Acc and mAP, and inference speed (fps) of MAP-RegOT. Except for 16×16, all combinations have roughly similar inference speed and validation accuracy, showing the stability of our model design in terms of parameter choice. We found a resolution of $32 \times 8 \times 8$ achieves the best balance of time and accuracy. Meanwhile, our framework is flexible in choosing the parameters to suit preferences of speed or accuracy.

Visualization. We show more few-shot segmentation samples in Fig. 8. Overall, our approach can successfully identify the correct pixel labels for moderate-size objects and smooth scenes. Objects lying across multiple regions (e.g., school bus in the left column, last row) could be inconsistently segmented though, as it is hard to learn semantic information about the unseen testing classes with such few labeled data. We want to raise the topic of refining inter-region segmentation consistency to future work.

Enhanced Baselines. Previous works extracted feature vectors of spatial resolution H=W=1, and chose either a large dim (e.g., 1024) to guarantee performance (TarMAC [7]), or a small dim(e.g., 32) to reduce comm. cost (When2Com [26]) by sacrificing the performance. We find that even by increasing comm. costs, the baselines cannot achieve comparable results with our methods. We evaluate TarMAC with an increased feature dim from 1024 to 2048, 3072 and 4096, respectively, but get saturated accuracies of 0.564, 0.618, 0.647and 0.643, respectively, on face recognition task. Compared with our MAP-RegOT (acc 0.671, Tab.5), the best performing TarMAC (dim 3072, acc 0.647) has 3 times of the comm. cost, while still underperforms our approach by 3.7%.

6 CONCLUSION

In this paper, we proposed a unified framework that tackles multi-agent perception tasks in data-scarce scenarios. Our design of a signature-based communication mechanism integrated with a fine-grained metric-learning approach achieved significantly improved FS-MAP results on various tasks, including face identification, semantic segmentation, and sound genre recognition. Future work can focus on improving the inter-region segmentation consistency for few-shot segmentation tasks and explore more scenarios and forms of multimedia data where FS-MAP can apply.

REFERENCES

[1] J. Altschuler, J. Niles-Weed, and P. Rigollet. 2017. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *NeurIPS*. 1964–1974.
[2] M. Arjovsky, S. Chintala, and L. Bottou. 2017. Wasserstein generative adversarial networks. In *ICML*.

[3] Shane Barratt. 2018. On the differentiability of the solution to convex optimization problems. *arXiv preprint arXiv:1804.05098* (2018).

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

[5] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*. 2292–2300.

[6] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*.

[7] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. 2019. Tarmac: Targeted multi-agent communication. In *ICML*.

[8] Chenyou Fan and Jianwei Huang. 2021. Federated Few-Shot Learning with Adversarial Learning. *arXiv preprint arXiv:2104.00365* (2021).

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML* (2017).

[10] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *NeurIPS*.

[11] Spyros Gidaris and Nikos Komodakis. 2018. Dynamic Few-Shot Visual Learning Without Forgetting. In *CVPR*.

[12] Xiyue Guo, Junjie Hu, Junfeng Chen, Fuqin Deng, and Tin Lun Lam. 2021. Semantic Histogram Based Graph Matching for Real-Time Multi-Robot Global Localization in Large Scale Environment. In *IEEE Robotics and Automation Letters*.

[13] Jun He, Richang Hong, Xueliang Liu, Mingliang Xu, Zheng-Jun Zha, and Meng Wang. 2020. Memory-Augmented Relation Network for Few-Shot Learning. In *ACM Multimedia*.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

[15] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. 2017. Attention-based multimodal fusion for video description. In *ICCV*.

[16] Yedid Hoshen. 2017. Vain: Attentional multi-agent predictive modeling. In *NeurIPS*.

[17] Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari, Svetlana Lazebnik, Ali Farhadi, Alexander G Schwing, and Aniruddha Kembhavi. 2019. Two body problem: Collaborative visual task completion. In *CVPR*.

[18] Jiechuan Jiang and Zongqing Lu. 2018. Learning attentional communication for multi-agent cooperation. In *NeurIPS*.

[19] Philip A Knight. 2008. The Sinkhorn–Knopp algorithm: convergence and applications. *SIAM J. Matrix Anal. Appl.* (2008).

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

[21] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. 2020. Boosting Few-Shot Learning With Adaptive Margin Loss. In *CVPR*.

[22] Peike Li, Yunchao Wei, and Yi Yang. 2020. Meta parsing networks: Towards generalized few-shot scene parsing with adaptive metric learning. In *ACM Multimedia*.

[23] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. 2019. Revisiting local descriptor based image-to-class measure for few-shot learning. In *CVPR*.

[24] T. Lin, N. Ho, and M. Jordan. 2019. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In *ICML*. 3982–3991.

[25] Lizhao Liu, Junyi Cao, Minqian Liu, Yong Guo, Qi Chen, and Mingkui Tan. 2020. Dynamic Extension Nets for Few-shot Semantic Segmentation. In *ACM Multimedia*.

[26] Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. 2020. When2com: Multi-Agent Perception via Communication Graph Grouping. In *CVPR*.

[27] Yen-Cheng Liu, Junjiao Tian, Chih-Yao Ma, Nathan Glaser, Chia-Wen Kuo, and Zsolt Kira. 2020. Who2com: Collaborative perception via learnable handshake communication. (2020).

[28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *ICCV*.

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

[30] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. (2015).

[31] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NIPS*.

[32] O. Pele and M. Werman. 2009. Fast and robust earth mover's distances. In *ICCV*. IEEE.

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

[34] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. 2019. Few-Shot Image Recognition With Knowledge Transfer. In *ICCV*.

[35] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *CVPR*.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

[37] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 1998. A metric for distributions with applications to image databases. In *ICCV*.

[38] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*.

[39] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *NIPS*.

[40] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NIPS*.

[41] Sainbayar Sukhbaatar, Rob Fergus, et al. 2016. Learning multiagent communication with backpropagation. In *NeurIPS*.

[42] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. 2019. Meta-transfer learning for few-shot learning. *CVPR* (2019).

[43] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. *CVPR* (2018).

[44] Ming Tan. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *ICML*.

[45] Hao Tang, Zechao Li, Zhimao Peng, and Jinhui Tang. 2020. BlockMix: Meta Regularization and Self-Calibrated Inference for Metric-Based Meta-Learning. In *ACM Multimedia*.

[46] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatio-temporal Features with 3d Convolutional Networks. In *ICCV*.

[47] G. Tzanetakis and P. Cook. 2002. Musical genre classification of audio signals". *IEEE Transactions on Speech and Audio Processing"* (2002).

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

[49] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *NIPS*.

[50] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. 2019. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*.

[51] Zeyuan Wang, Yifan Zhao, Jia Li, and Yonghong Tian. 2020. Cooperative Bi-Path Metric for Few-Shot Learning. In *ACM Multimedia*.

[52] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. 2010. Deconvolutional networks. In *CVPR*.

[53] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. 2020. DeepEMD: Few-Shot Image Classification With Differentiable Earth Mover's Distance and Structured Classifiers. In *CVPR*.

[54] Peng Zhao and Zhi-Hua Zhou. 2018. Label distribution learning by optimal transport. In *AAAI*.

[55] Qi Zhao, Zhi Yang, and Hai Tao. 2008. Differential earth mover's distance with its applications to visual tracking. *PAMI* (2008).