Few-Shot Multi-Agent Perception with Ranking-Based Feature Learning

Chenyou Fan, *Member, IEEE,* Junjie Hu, *Member, IEEE,* and Jianwei Huang, *Fellow, IEEE*

Abstract—In this article, we focus on performing few-shot learning (FSL) under multi-agent scenarios in which participating agents only have scarce labeled data and need to collaborate to predict labels of query observations. We aim at designing a coordination and learning framework in which multiple agents, such as drones and robots, can collectively perceive the environment accurately and efficiently under limited communication and computation conditions. We propose a 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. Furthermore, we propose a specially designed ranking-based feature learning module, which can fully utilize the order information of training data by maximizing the inter-class distance, while minimizing the intra-class distance explicitly. We perform extensive numerical studies and demonstrate that our approach can achieve significantly improved accuracy in visual and acoustic perception tasks such as face identification, semantic segmentation, and sound genre recognition, consistently outperforming the state-of-the-art baselines by 5%-20%.

Index Terms—few-shot learning, multi-agent perception, semantic segmentation, optimal transport, image and audio classification

1 Introduction

In recent years, researchers have achieved remarkable progresses in single-agent visual perception tasks such as image classification [2], [3], object detection [4], [5], semantic segmentation [6], action recognition [7], [8], visual question answering [9], etc. However, in many realistic scenarios, multiple agents are deployed to observe the environment from different perspectives simultaneously. Comparing to the single-agent case, multi-agent perception (**MAP**) makes it possible to share useful information among the participating agents through inter-agent communications, augment the observation of a same scene from different perspectives, and expand the total scope with multiple scenes. Therefore, one critical research problem in MAP is how to establish an effective communication mechanism to represent and share multi-view observations among participating agents.

Existing studies of multi-agent learning [10], [11], [12], [13] chose to use data-hungry deep neural networks (DNNs) as base models. They proposed to learn shared DNNs to encode scenes to features on single agents first, then aggregate the features from all agents based on attention mechanism [14], [15], [16], and finally decode the fused features for downstream tasks such as perception or controlling. This data-driven process requires plenty of training examples from excessive sensory observations (e.g., point clouds, semantic labels) of the environment.

However, it can be highly labor-intensive and costly to collect and label a large amount of training data. Also, the scenes can be highly dynamic so 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?"*.

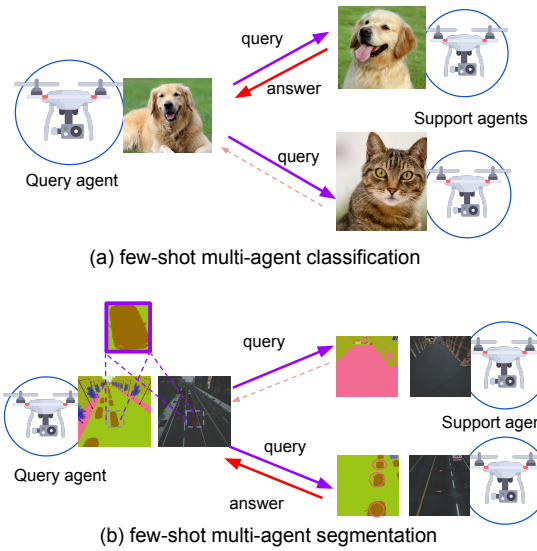


Fig. 1: Demo of few-shot multi-agent perception Tasks.

We formulate this question as a practical few-shot multi-agent perception (**FS-MAP**) task with a general setting: each agent owns just a few labeled examples as **support** data, while it also observes incoming unlabeled **query** examples in runtime. 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 to consider this practical yet under-explored

Chenyou Fan is with the School of Artificial Intelligence, South China Normal University, China. (email: fanchenyou@scnu.edu.cn).
Junjie Hu is with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), China. (e-mail: hujunjie@cuhk.edu.cn).
Jianwei Huang is with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China, and the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), China. (e-mail: jianweihuang@cuhk.edu.cn, corresponding author).
A preliminary version of this work that contains a part of the methodologies and a subset of the results was presented at 2021 ACM International Conference on Multimedia (ACMMM 2021) [1].

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 the observed images at different regions with varying sizes and contexts. For example, agents such as UAVs (Unmanned Aerial Vehicles) and UGVs (Unmanned Ground Vehicles) can take images from different heights and distances with various camera angles. Thus, it is critical to propose a robust distance metric to measure the similarity between query and support data with decent translation and orientation invariance. To achieve this, we first extract the fine-grained 3-D feature maps for both query and support data which preserve the spatial information. Then we broadcast the query features to support agents and evaluate the relevance between query and support data. We formulate the feature matching as a Regularized Optimal Transport (RegOT) [17] task and solve it efficiently. The most relevant support data can thus assign their labels to the corresponding queries.

We further utilize the order information implied in training batches to regularize model training. For a query sample, its corresponding few-shot support data should have higher rankings than irrelevant data in terms of their similarity scores. Inspired by this, we design two novel learning objectives. The first is to maximize the inter-class distance of support data from different categories, and the second is to minimize the intra-class distance of support data of the same category. We formulate these training objectives with differential ranking tasks and optimize them efficiently in an end-to-end manner.

Finally, as physical conditions often limit inter-agent bandwidth, balancing perception accuracy and communication costs is practical and critical. 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. We will demonstrate through extensive experiments that our framework can achieve superior performance on various perception tasks.

In conclusion, our contributions include:

- We consider a critical but under-explored task of learning visual perception tasks with very 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.
- To improve feature space and regularize model training with few-shot data, we formulate two learning-to-rank objectives with efficient solutions.
- Our approaches significantly outperform the state-of-the-art methods by 10%-15% on segmentation and classification tasks upon multimedia data, including images and sounds.

2 Related Work

We briefly review recent related work in categories: 1) multi-agent learning, 2) few-shot learning (FSL) , 3) optimal transport, and 4) learning-to-rank techniques. We will highlight the difference of our work with existing works.

Multi-agent learning is a broad research field, and our work is closely related to its topics of learning communication protocols [10], [11], [12], [13], [18], [19] to improve the effectiveness of collaboration, as well as learning perception tasks [19], [20], [21], [22]. VAIN [12] proposed to use kernel-based attention to measure the weight of each agent's message. TarMAC [13] used signature-based attention [16] to decouple query and key features to provide more flexibility of the communication such as selecting which other agents to communicate with. When2Com [22] further considered reducing bandwidth usage by using asymmetric query and key sizes. However, existing works utilized coarse-grained feature vectors. Our work improves performance by extracting fine-grained image feature maps and utilizes asymmetric query and key feature dimensions to balance performance and cost.

Few-shot learning (FSL) is learning new model capacities with very few labeled training samples. Our work is closely related to the metric-based few-shot learning approaches [23], [24], [25], [26], [27] that focus on learning a discriminative feature space that minimizes intra-class distances while maximizing inter-class distance. Recent FSL studies followed the nearest neighbor idea in performing metric learning, which can be further categorized as follows. Firstly, MatchingNet [23] and RelationNet [28] propose to minimize feature distance between unlabeled data and labeled data of the same class. Secondly, Prototypical networks [24] minimize feature distance between unlabeled data and class centroids formed by mean class member features. Both techniques measure the feature similarity with Euclidean or Cosine distance.

The third technique is learning a more advanced distance metric, such as Optimal Transport (OT) distance in DeepEMD [26] and deep Brownian distance in DeepBDC [29]. The fourth technique is explicit modeling the intra-class variations in order to improve inter-class discrimination. Two recently proposed methods, CTM [30] and TOAN [31], focus on extracting the intra-class commonality feature and thus better building inter-class features. Conversely, VFD [32] aims to augment the data by utilizing a variational auto-encoder to sample additional intra-class samples. Notedly, such techniques often necessitate specially designed modules or additional sampling steps.

In this study, we propose a novel rank-learning technique that simultaneously establishes ordering over all instance-instance pairs for intra-class minimization and instance-class pairs for inter-class maximization. Our rank-learning scheme solely focuses on enhancing the model's feature space during the training phase, without any impact on the inference stage as it remains decoupled from it.

FSL has many successful applications such as image classification [28], [33], [34], [35], [36], [37], [38] and semantic segmentation [39], [40], [41]. However, existing methods are designed for centralized training and execution for single-agent tasks. Our work proposes a general multi-agent few-shot learning framework that is applicable for a broad scope of multimedia recognition tasks, e.g., 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 [26], [42], [43], [44], [45], [46]. However, the computation of OT is complex and existing studies formulated it as linear programming task [26], [42], [47] 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 $O(d^2 \log d)$.

Learning-to-rank (LTR) aims to measure the order of a list of similarity scores typically for information retrieval [48], [49] and representation learning [50], [51], [52], [53], [54], [55], [56]. Generally, LTR methods can be categorized as pairwise [57], [58] and listwise [48], [59], [60] to model the orders over the data list. **Learning-to-Sort (LTS)** is a recent popular listwise ranking methodology [61], [62], [63], which formulates the ranking problem as a differential sorting task. Cuturi *et al.* [61] proposed to formulate sorting as an OT task. Xie *et al.* [63] further proposed an efficient way of finding the top-k elements of a list. However, it is an under-explored topic to utilize the sorting technique in visual learning tasks. We will show that LTS can be seamlessly integrated into our FS-MAP framework.

3 FS-MAP Tasks and Definitions

In a general few-shot multi-agent perception (**FS-MAP**) setting, each support agent can have a few labeled support data instances of arbitrary classes, and some agents may support overlapping classes. We consider a simplified scenario in which each agent owns a few labeled examples as support data for ONE class, which is also non-overlapping with each other. An agent is said to *support* a class if it holds support data of that class. We adopt this assumption of **one support class per agent** to facilitate the discussion. Later, we will show that our approach can easily extend to the general case that one agent can support multiple classes.

We formulate the FS-MAP task formally now. Following the conventions of few-shot learning studies [23], [24], we define FS-MAP as a C-way K-shot N-agent learning task. Each agent i could support C_i distinct classes and each class has K labeled samples as support data. The total C classes are covered by all agents such that $\sum_{i \in \mathcal{N}} C_i = C$. With the *one support class 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 four typical ones considered in this paper.

- **Image classification** is to predict the label of the query data out of C classes. A toy example is shown in Fig. 1(a).
- **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).
- **Face identification** is to match one person's face images correctly out of C distinct identities.
- **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 tracking a suspicious vehicle in the crowded city streets. The police first provide one photo for the target vehicle and then send out drones, patrol robots, and human forces with dash cameras to different zones to identify whether their observations match it. This distributed execution with multiple agents can significantly improve the efficiency of car identification, as only one query image (of the target vehicle) and one support image (of each observed vehicle in the scenes) is needed to perform the few-shot matching.

Fig. 2: The technical overview of our FS-MAP framework.

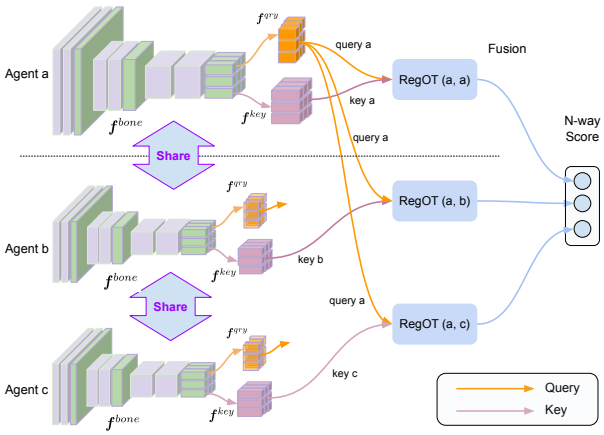


Fig. 3: Overview of FS-MAP architecture, including a shared backbone network $\boldsymbol{f}^{bone}$ for generating 3-D feature maps $H \times W \times C$, a key network $\boldsymbol{f}^{key}$ for generating key features, a query network $\boldsymbol{f}^{qry}$ for generating query features, and a RegOT module for measuring the distance between query and support data.

4 Our Approach

We introduce our FS-MAP framework with a pipeline of three subsequent logical steps as shown in Fig. 2. In Step (1), FS-MAP extracts query and support data features. Then, in Step (2), FS-MAP measures their similarity with a Regularized Optimal Transport (RegOT) distance. Finally, in Step (3), FS-MAP performs a novel rank-learning procedure by exploring the Few-Shot Learning (FSL) setting to improve training.

4.1 Model overview

We first illustrate our FS-MAP model in Fig. 3 which generates query and support features, measures their similarity and fulfills FSL tasks. The first main component is a backbone Convolutional Neural Network (CNN) $\boldsymbol{f}^{bone}$ as a feature extractor which encodes images to hidden feature maps with sizes $D \times H \times W$, in which D is channel size and $H \times W$ is spatial resolution. The second is a query sub-network $\boldsymbol{f}^{qry}$ which encodes hidden feature maps to compact query feature maps for query input. The third is a key sub-network $\boldsymbol{f}^{key}$ which encodes hidden feature maps to large-size key feature maps for support data. As we adopt the *centralized training and decentralized execution* strategy [10], [64], these modules are shared across all agents during inference.

We use "key" features to denote support data features and call this design as *signature-based communications*, by following TarMAC [13]. We 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 class per agent*, each support agent v also corresponds to v-th category. To simplify notations,

we will denote $v \in \mathcal{N}$ as abbreviation of $v = \{1, ..., N\}$. We assume column vectors by default.

4.2 Feature generation and broadcasting

We present our multi-agent communication scheme, which extracts feature maps for query and support images at each agent, broadcasts the query features to the support agents, and performs feature matching to fulfill few-shot learning tasks.

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 $\boldsymbol{f}^{bone}$ respectively, such that $\boldsymbol{h}_u = \boldsymbol{f}^{bone}(\boldsymbol{X}_u)$ and $\boldsymbol{h}_v = \boldsymbol{f}^{bone}(\boldsymbol{X}_v)$, in which D is channel size and $H \times W$ is spatial resolution.

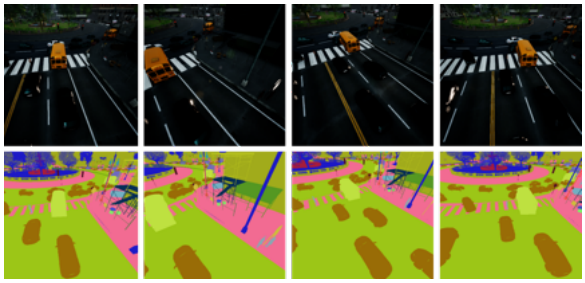


Fig. 4: Multi-view school bus images and segmentations with different camera viewpoints. Source from AirSim dataset [65].

For the query data of agent u, we generate its *query* feature $\boldsymbol{q}_u = \boldsymbol{f}^{qry}(\boldsymbol{h}_u)$ with the query sub-network. For the support data of agent v, we generate its *key* (a.k.a. support) feature $\boldsymbol{k}_v = \boldsymbol{f}^{key}(\boldsymbol{h}_v)$ with the key sub-network. We will use *key* feature and *support* feature interchangeably to denote $\boldsymbol{k}_v$.

We now discuss how to choose the feature dimensions $\boldsymbol{q}_u \in \mathcal{R}^{D_q \times H \times W}$ and $\boldsymbol{k}_v \in \mathcal{R}^{D_k \times H \times W}$. During communication, the query features will be broadcasted from query agents to all support agents to perform feature matching with a distance metric that we will discuss in next section. The support agents serve as the receiving ends and listen to incoming queries, and perform feature matching with local support data, then return the matching scores.

As only query features are transmitted, a compact $\boldsymbol{q}_u$ with a small channel size D_q will save bandwidth usage while preserving spatial resolution. The key (a.k.a. support) features are kept locally at the corresponding support agents without communicating to other agents. Therefore, we choose a large channel size D_k for key features to compensate for the accuracy loss caused by using small query features. 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 $D_q = 32$ and $D_k = 1024$. The cost of transmitting *query* feature maps of size $32 \times 8 \times 8$ is equal to a 1-D feature vector of length 2048 floats, a.k.a, the bandwidth usage 8 KiloByte per frame (KBpf). We will discuss the trade-off of channel size and resolution in the ablation study (6.6).

Previous multi-agent learning studies [12], [13], [19] used coarse-grained feature vectors instead of rich feature maps to represent observations. This would lead to inferior performance under few-shot perception setting, as a same object may appear in different image regions with distinct sizes and positions due to different agents' viewpoints, e.g., the school bus in Fig. 4. In the next section, we design to utilize the rich spatial information in the 3-D feature maps to perform fine-grained feature matching.

4.3 Structured matching of two feature maps

In our FS-MAP task setting, one key research topic is to measure the similarity of query and support images in a fine-grained manner in order to identify objects from different viewpoints. In the previous step, the query agent u has broadcasted its query feature $\boldsymbol{q}_u$ to the support agents at 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) [17] 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 weights 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 as 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 ($D_q \neq D_k$), we apply the general dot product [15] to calculate the cosine similarity:

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

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) of query and support feature map can be conveniently defined in matrix form as:

$$\boldsymbol{C}_{uv} = \{c_{uv,ij} = 1 - a_{uv,ij}, \ \forall i, j \in HW\}, \tag{2}$$

in which $\boldsymbol{C}_{uv} \in \mathcal{R}^{HW \times HW}$.

Step 2: Node weight assignment. The next step is to determine the total weight of each node as the importance of each 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 contain school buses. Thus, we determine the reciprocal src. and dst. node weight $s_{u,i}$ and $d_{v,j}$ as the average of total matching score such that

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

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 define the distance of two feature maps as the minimum cost of transporting the src. node weights of query data to the dst. nodes. We define the

regularized optimal transport distance $\text{regOT}(u, v)$ between query data u and support data v as

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

in which $\langle\cdot,\cdot\rangle$ is element-wise product, $\boldsymbol{P} \in \mathcal{R}^{HW\times HW}$ is the transport 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.

Lemma 1. *[Lemma 2, Cuturi [17]] For any cost matrix $\boldsymbol{C}$, the minimization in Eq.(4) has a unique minimum $\boldsymbol{P}_\lambda$ in the form of $\boldsymbol{P}_\lambda = \boldsymbol{X}\boldsymbol{A}\boldsymbol{Y}$, where $\boldsymbol{A} = exp(-\lambda\boldsymbol{C})$ and $X, Y \in \mathcal{R}_+^{n\times n}$ are both diagonal matrices. The matrices $(\boldsymbol{X}, \boldsymbol{Y})$ are unique up to a multiplicative factor.*

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 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 [66], to approximate the OT 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 transport cost while satisfying the constraints.

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

Output: Approximated optimal transport plan $\boldsymbol{X}$.

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

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

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

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 $\|C\|_\infty \le 2$, also both $\boldsymbol{s}, \boldsymbol{d}$ given by (3) sum to 1. By applying [67, 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. The operations in Algorithm 1 are 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 [26], [42] that solves the original OT

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

without the entropy term as opposed to our objective (4). The original OT task is linear programming which is usually solved by interior-point methods with a time complexity of $O(n^3\log n)$ [42]; while our approach shaves a factor of n in time complexity. Furthermore, it costs an enormous $O(n^4)$ memory usage in order to make it differentiable [68]. Our approach costs only $O(n^2)$ memory usage and is fully differentiable to be optimized by SGD with other DNN components.

4.4 1-shot multi-agent learning

With the proper query-support distance measure *regOT*, we fulfill the learning of 1-shot perception tasks with our framework. 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 OT 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,

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

which sums up the OT plan $\hat{\boldsymbol{P}}$ weighted by the inverse costs $\mathbf{1}-\boldsymbol{C}$ in a region-wise manner.

We have N pairwise similarity scores between query image u and 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 for the query image to match with all support images. Based on this, we compute the cross-entropy loss such that

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

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

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 of a support image v as

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

Thus $\boldsymbol{\varphi}_{uv} \in \mathcal{R}^{HW}$ implies the similarity scores of all regions with label v. Since the query agent will broadcast to all support agents, we will have $\boldsymbol{\varphi}_u = \{\boldsymbol{\varphi}_{uv}, v \in \mathcal{N}\} \in \mathcal{R}^{N\times HW}$ 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 [69] followed by a standard bi-linear upsampling upon $\boldsymbol{\varphi}_u$ such that $\boldsymbol{o}_u = \text{Upsample}(\boldsymbol{\varphi}_u)$ of size $\boldsymbol{o}_u \in \mathcal{R}^{N\times H_0\times W_0}$. Let $Y = \{y_i, i \in H_0W_0\}$ be ground-truth segmentation mask, we compute the pixel-wise cross-entropy loss to optimize the model in end-to-end fashion, such that

$$\ell^{seg}(Z_u, Y) = -\frac{1}{H_0W_0}\sum_{i=0}^{H_0W_0} \log\frac{\exp((\boldsymbol{o}_{uy_i})_i)}{\sum_{v=1}^N \exp((\boldsymbol{o}_{uv})_i)}\ , \tag{10}$$

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.

Algorithm 2: Execution of 1-shot Multi-Agent Perception for classification task.

Input: Query agent u with query data X_u, N support agents $\boldsymbol{V} = \{v_1, ..., v_N\}$ with $\boldsymbol{X}_V = \{X_{v_1}, ..., X_{v_N}\}$ as support data.
Output: Estimated label $\hat{y}_u$ for query data X_u.

Compute hidden feature maps for query data $\boldsymbol{h}_u \leftarrow \boldsymbol{f}^{bone}(X_u)$, then compute query feature $\boldsymbol{q}_u \leftarrow \boldsymbol{f}^{qry}(\boldsymbol{h}_u)$ at query agent.
Compute hidden feature maps for support data $\boldsymbol{h}_v \leftarrow \boldsymbol{f}^{bone}(X_v), \forall v \in \boldsymbol{V}$, then compute key feature $\boldsymbol{k}_v \leftarrow \boldsymbol{f}^{key}(\boldsymbol{h}_v), \forall v \in \boldsymbol{V}$ at support agents in parallel.
The query agent u broadcasts its generated query feature $\boldsymbol{q}_u$ to all support agents $\boldsymbol{V}$.
for each support agent $v \in \boldsymbol{V}$ **in parallel do**
    Compute node weights $(\boldsymbol{s}_u, \boldsymbol{d}_v)$ as in (3).
    Optimize $regOT(u, v)$ in (4) with Algorithm 1, and return estimated optimal transport plan $\hat{\boldsymbol{P}}$.
    Compute final similarity score ψ_{uv} between query data X_u and support data X_v with (7).
    Return ψ_{uv} to query agent u.
end
The query agent collects $\Psi_u = \{\psi_{uv}, \forall v \in \boldsymbol{V}\}$ from all support agents.
Assign the predicted image label $\hat{y}_u \leftarrow \arg\max_v \ \Psi_u$.
Return $\hat{y}_u$.

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 [24].

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. We take classification task as an example and show the details of this preprocess stage for preparing $\bar{X}_v$ in Algorithm 3. We found that setting $T = 10$ achieves a good balance of time and accuracy. The communication cost is thus sending the updated query feature $\boldsymbol{q}_u$ to support agents for T iterations.

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. Our framework can be easily extended to the case where each agent has data of multiple classes.

Algorithm 3: Pre-process of K-shot learning.

Input: Learning rate γ, max iteration T, K support data for each of N class $\{\boldsymbol{X}_v = \{X_v^i, i \in \mathcal{K}\}, v \in \mathcal{N}\}$
Output: N-class mean support images $\{\bar{X}_v, v \in \mathcal{N}\}$.

Initialize $\{\bar{X}_v \leftarrow \mathcal{N}(0,1), v \in \mathcal{N}\}$
$t \leftarrow 1$
while $t \leq T$ **do**
    for each query agent $u \in \mathcal{N}$ **in parallel do**
        // At sender's end
        $\boldsymbol{q}_u \leftarrow f^{qry}(\bar{X}_u)$
        Broadcast query feature to all support agents
        for each support agent $v \in \mathcal{N}$ **in parallel do**
            // At receiver's end
            $\psi_{uv} \leftarrow$ ComputeScore($\boldsymbol{q}_u$, $\boldsymbol{X}_v$)
        end
        // Compute the gradient of loss (8)
        $\delta_{\bar{X}_u} \leftarrow \frac{\partial \ell^{cls}}{\partial \bar{X}_u} = \frac{\partial \ell^{cls}}{\partial \boldsymbol{q}_u} \frac{\partial \boldsymbol{q}_u}{\partial \bar{X}_u}$
        // update mean support data
        $\bar{X}_u \leftarrow \bar{X}_u - \gamma \delta_{\bar{X}_u}$
    end
    $t \leftarrow t + 1$
end
Return $\{\bar{X}_v, v \in \mathcal{N}\}$.

ComputeScore($\boldsymbol{q}_u, \boldsymbol{X}_v$):
Input: query feature $\boldsymbol{q}_u$, support data $\boldsymbol{X}_v$ of K samples
Output: similarity score ψ_{uv} between $\boldsymbol{q}_u$ and $\boldsymbol{X}_v$
Randomly pick a support data example X_v^i out of $\boldsymbol{X}_v$
$\boldsymbol{k}_v \leftarrow f^{key}(X_v^i)$
Compute ψ_{uv} as in (7) with $\boldsymbol{q}_u$ and $\boldsymbol{k}_v$.
Return ψ_{uv}.

Let us consider a general case that each agent v supports a set of $|C_v|$ classes with K data samples per class. 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 label can be found by searching for the highest score in the combined score lists from all support agents.

5 Ranking-based Feature Learning

We propose two new training objectives for further regularizing the model training to learn a better distance metric.

Given a query sample $\boldsymbol{u}$ and a support sample $\boldsymbol{v}_i$, we call $(\boldsymbol{u}, \boldsymbol{v}_i)$ a **relevant pair** if they have the same label. For a support sample $\boldsymbol{v}_j$ of different label with $\boldsymbol{u}$, we call $(\boldsymbol{u}, \boldsymbol{v}_j)$ an **irrelevant pair**. Given a proper distance metric in the feature space, we can measure the distance of pairs of samples, as well as the distance of two class centers.

Intuitively, a *relevant pair should have a smaller distance (a larger similarity) than an irrelevant pair*. Also, data pairs within a same class should have smaller **intra-class distance**, while data pairs of different classes should have larger **inter-class distance**. These intuitions elicit two learning objectives:

- i) *maximize the inter-class distance of irrelevant data pairs*;
- ii) *minimize the intra-class distance of relevant data pairs*.

An overview of this section is shown in Fig. 7. We formulate objective i) as Problem (17) in Sec.5.1, and objective ii) as Problem (21) in Sec.5.2, with efficient solutions. We provide the final training algorithm in Appendix Algorithm 4.

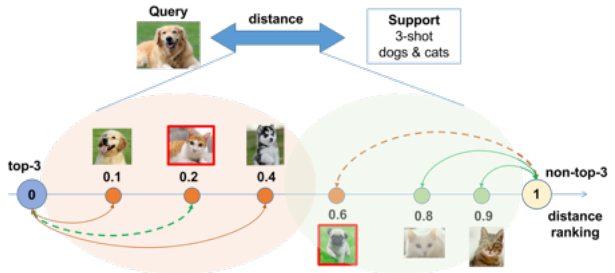


Fig. 5: Demo of ranking the top-K ($K = 3$) distance measures of support images (3-shot cats and dogs) with a query dog image. Ideally the support dog images are relevant to the query thus should be ranked higher (close to origin on the left).

Notations. Given one query and n support data samples, $\mathcal{A} = \{a_i\}_{i=1}^n$ denotes n distances from the query sample to n support samples. A smaller a_i indicates a higher similarity. Following [63], we let $\mathcal{B} = \{0, 1\}$ denote two categories: 0 for being among the top-k elements of a list and 1 for being non-top-k. Let $\mathbf{1}_n$ be an all one-vector of dimension n, $\boldsymbol{w} = \mathbf{1}_n/n$ be a uniform distribution probability vector of each distance score, and $\boldsymbol{z} = \left[\frac{k}{n}, \frac{n-k}{n}\right]^\top$ be the probabilities of being top-k and non-top-k.

5.1 Maximize inter-class distance

Our first optimization goal is to *maximize the inter-class distance between relevant support class and irrelevant classes*. The purpose is to optimize the model to produce a feature space to be discriminative for different classes.

Fig. 5 illustrates an example of our method. Given a query dog image, we have a collection of dog (relevant) and cat (irrelevant) images as support data. We compute the distances from the query image to all support images, as shown around the axis. Ideally the dogs should be ranked higher (close to origin point) than cats. Misranking happens for an orange cat and a bulldog (highlighted with red outlines), which will be penalized by our learning objective.

Given a C-way K-shot N-agent learning task (Section 3) with $C = N$, let $T = K \cdot N$ denote the total number of support data samples. We aim to create a ranking algorithm such that the relevant K support data samples as top-K while the rest as non-top-K. The learning objective is to search for the optimal top-K relevant support data out of the entire support set.

By (7), we can obtain the image-level similarity score $\psi_{ut} \in [-1, 1]$ between a query $\boldsymbol{u}$ and *a support data* $\boldsymbol{t}$. We take the distance metric $\phi_{ut} \in [0, 1]$ as the inverse of similarity score

$$\phi_{ut} = \frac{1}{2}(1 - \psi_{ut}),\ \forall t \in \{1, ..., T\}\ . \tag{11}$$

For T support data, we build a cost matrix $\boldsymbol{C}_u \in \mathcal{R}^{T\times 2}$ as

$$\boldsymbol{C}_u = \begin{bmatrix} \phi_{u,1}^2 & (1-\phi_{u,1})^2 \\ \vdots & \vdots \\ \phi_{u,t}^2 & (1-\phi_{u,t})^2 \\ \vdots & \vdots \\ \phi_{u,T}^2 & (1-\phi_{u,T})^2 \end{bmatrix} \in \mathcal{R}^{T\times 2}\ , \tag{12}$$

where the first column indicates the cost of being top-K, while the second column indicates the cost of being non-top-K.

To find the top-K (out of T) most similar support data of the query data $\boldsymbol{u}$, we formulate it as an OT task of transporting distance measures implied by $\boldsymbol{C}_u$ to top-K indicator $\mathcal{B}$ as

$$\begin{aligned} \boldsymbol{S}_u^* &= \underset{\boldsymbol{S}_u \geq 0}{\arg\min}\ \langle \boldsymbol{S}_u, \boldsymbol{C}_u \rangle\ , \\ s.t.\ \boldsymbol{S}_u \mathbf{1}_2 &= \frac{1}{T}\mathbf{1}_T,\ \boldsymbol{S}_u^\top \mathbf{1}_T = \left[\frac{K}{T}, \frac{T-K}{T}\right]^\top\ , \end{aligned} \tag{13}$$

in which the plan $\boldsymbol{S}_u^* \in \mathcal{R}^{T\times 2}$ indicates the optimal probability of assigning each of T support data to be top-K or not.

Lemma 2. *The optimal OT plan $\boldsymbol{S}_u^*$ of Problem (13) provides the top-K most similar support data.*

Proof. Given the cost matrix $\boldsymbol{C}_u$ defined in (12), a higher similarity of a query-support pair ψ_{ut} yields a lower cost $\phi_{u,t}^2$ of assigning it as *top-K*, while a lower similarity ψ_{ut} yields a lower cost $(1-\phi_{u,t})^2$ of assigning *non-top-K*. By [63, Proposition 1], solving Problem (13) yields the optimal solution $\boldsymbol{S}_u^*$ such that each of its rows satisfies

$$S_{u,t,\cdot}^* = \begin{cases} (\frac{1}{T}, 0), & \text{if } \phi_{u,t} \text{ is top-}K \text{ smaller cost,} \\ (0, \frac{1}{T}), & \text{if } \phi_{u,t} \text{ is non-top-}K \text{ smaller cost.} \end{cases} \tag{14}$$

Thus, the indices of the K rows which equate $(\frac{1}{T}, 0)$ correspond to the top-K most similar support data to query $\boldsymbol{u}$. □

We can further derive the indices of the top-K and non-top-K elements from OT plan $\boldsymbol{S}^*$, respectively.

Definition 1. *Let $\boldsymbol{A} = [A_1, \ldots, A_T]^\top$ be the top-K result vector which satisfies*

$$A_t = \begin{cases} 1, & \textit{if } x_t \textit{ is a top-K element,} \\ 0, & \textit{if } x_t \textit{ is a non-top-K element.} \end{cases} \tag{15}$$

Definition 2. *Let $\boldsymbol{A}^c = [A_1^c, \ldots, A_T^c]^\top$ be the counterpart top-K result vector to indicate non-top-K indices such that*

$$A_t^c = \begin{cases} 0, & \textit{if } x_t \textit{ is a top-K element,} \\ 1, & \textit{if } x_t \textit{ is a non-top-K element.} \end{cases} \tag{16}$$

Corollary 1. *Based on (14), we have $\boldsymbol{A} = T \cdot \boldsymbol{S}_u^* \cdot [1, 0]^\top$ and $\boldsymbol{A}^c = T \cdot \boldsymbol{S}_u^* \cdot [0, 1]^\top$.*

Problem (13) is also a linear programming problem, which has high computational cost with large dimensional data. Inspired by Theorem 1, we optimize the convex relaxation of Problem (13) such as

$$\begin{aligned} \hat{\boldsymbol{S}}_u^* &= \underset{\boldsymbol{S}_u \geq 0}{\arg\min}\ \langle \boldsymbol{S}_u, \boldsymbol{C}_u \rangle - \lambda H(\boldsymbol{S}_u)\ , \\ s.t.\ \boldsymbol{S}_u \mathbf{1}_2 &= \frac{1}{T}\mathbf{1}_T,\ \boldsymbol{S}_u^\top \mathbf{1}_T = \left[\frac{K}{T}, \frac{T-K}{T}\right]^\top\ , \end{aligned} \tag{17}$$

in which $H(\boldsymbol{S}_u) = -\sum_{i,j} s_{u,ij} \log s_{u,ij}$ is the entropy term, and the optimal estimated plan $\hat{\boldsymbol{S}}_u^*$ of Problem (17) is a smoothed estimation of $\boldsymbol{S}_u^*$ of Problem (13). Then we can estimate the non-top-K indicator $\hat{\boldsymbol{A}}^c$ with (16), indicating the non-top-K probability of each support sample.

Note that Problem (17) depends on the cost matrix $\boldsymbol{C}_u$, where its elements depend on the model to produce the distance metric ψ_{ut} as (7). However, a randomly initialized model has to learn the proper distance metric to produce the correct ranking. Fortunately, during training, we have the true indices I of the top-K support

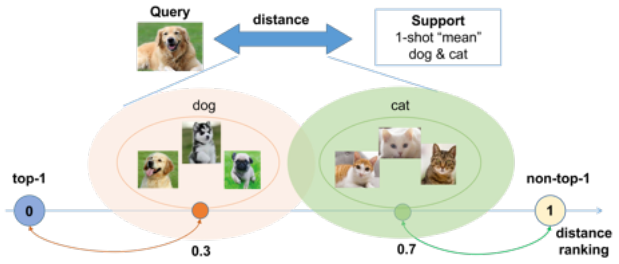


Fig. 6: Demo of ranking for the top-1 relevant support class, given a query dog image. We form a class center for each class (dog and cat) by its 3-shot support images. We aim to rank the most relevant class center (i.e., dog) to be top-1.

data samples for a given query sample. We can treat their non-top-K probabilities as a proper loss function and minimize it to train the model, such as

$$\ell^{inter}(\hat{\boldsymbol{S}}_u, Y) = \frac{1}{K}\sum_{k=1}^{K} \hat{A}^c(I(k)),\ I = \{i|Y_i = y_u\}\ . \quad (18)$$

The objective (18) represents the **inter-class loss**, as it separates relevant data samples from irrelevant ones. We can minimize ℓ^{inter} with SGD as Algorithm 1 in an end-to-end manner. This trains the model to produce better ranking results of Problem (17) and optimizes the image-level similarity $\boldsymbol{C}_u$ which (17) depends on. These two steps of solving Problem (17) and minimizing ℓ^{inter} are shown as yellow boxes in Appendix Fig. 7.

5.2 Minimize intra-class distance

Our second goal is to *minimize the intra-class distance of samples within each same class*. The benefit is to enforce each class in tight feature space so that different classes have separable boundaries that can be better classified. To achieve this, we create a center for each support class and optimize each center to be close to its corresponding query data.

If a query data $\boldsymbol{u}$ belongs to class i, we call the class i as the **relevant class** of $\boldsymbol{u}$; otherwise i is an **irrelevant class**. We construct the *mean* support feature of each class i as the class center $\boldsymbol{c}_i$, and define the distance from query $\boldsymbol{u}$ to class i as the distance between $\boldsymbol{u}$ and $\boldsymbol{c}_i$.

Given a query example, we aim to rank its relevant support class to be top-1 of all support classes, in order to optimize the data features within the class to be close to the center.

Fig. 6 illustrates the intuition through an example. Given a query dog image, we have a collection of dog (relevant) and cat (irrelevant) images as support data. We compute the centers for dog and cat respectively. Then we compute the distances from the query dog image to both centers, as shown around the axis. Ideally, the dog center should be ranked top-1 (closest to origin point) as it's relevant to the query.

We briefly explain how to compute the class center. For a general N-way K-shot ($K > 1$) learning task, Algorithm 3 of Section 4.5 constructs an optimal *mean of K-shot data* $\{\bar{X}_v, v \in \mathcal{N}\}$ for each data category v. Let $\psi'_{uv} \in [-1, 1]$ be the similarity score between the query $\boldsymbol{u}$ and the mean data $\bar{X}_v$ for class v with (7). Similar to (11), the distance metric $\phi'_{uv} \in [0, 1]$ is as follows:

$$\phi'_{uv} = \frac{1}{2}(1 - \psi'_{uv}),\ \forall v \in \{1, ..., N\}\ . \quad (19)$$

The cost matrix of a query $\boldsymbol{u}$ with each support class $\boldsymbol{v}$ is

$$\boldsymbol{D}_u = \begin{bmatrix} \phi'^2_{u,1} & (1-\phi'_{u,1})^2 \\ \vdots & \vdots \\ \phi'^2_{u,v} & (1-\phi'_{u,v})^2 \\ \vdots & \vdots \\ \phi'^2_{u,N} & (1-\phi'_{u,N})^2 \end{bmatrix} \in \mathcal{R}^{N\times 2}\ , \quad (20)$$

where the first and second column indicates the cost of being the top-1 or a non-top-1 respectively.

To find the top-1 (out of N) most similar support class of the query data $\boldsymbol{u}$, we formulate it as an OT task of transporting distance measures implied by $\boldsymbol{D}_u$ to top-1 indicator $\mathcal{B}$ as

$$\begin{aligned} \hat{\boldsymbol{R}}^*_u &= \arg\min_{\boldsymbol{R}_u \geq 0} \langle \boldsymbol{R}_u, \boldsymbol{D}_u \rangle - \lambda H(\boldsymbol{R}_u)\ , \\ s.t.\ \boldsymbol{R}_u \mathbf{1}_2 &= \frac{1}{N}\mathbf{1}_N,\ \boldsymbol{R}_u^\top \mathbf{1}_N = \left[\frac{1}{N}, \frac{N-1}{N}\right]^\top\ , \end{aligned} \quad (21)$$

in which $H(\boldsymbol{R}_u) = -\sum_{i,j} r_{u,ij} \log r_{u,ij}$ is the entropy term, and the output is the estimated OT plan $\hat{\boldsymbol{R}}_u \in \mathcal{R}^{N\times 2}$. We can estimate the non-top-1 indicator $\hat{A}^c = N \cdot \hat{\boldsymbol{R}}_u \cdot [0, 1]^\top$ as (16), indicating the non-top-1 probability of each support class. During training, as the true support class y_u of query $\boldsymbol{u}$ is known, we can minimize the non-top-1 probability of y_u estimated in $\hat{A}^c$ as:

$$\ell^{intra}(\hat{\boldsymbol{R}}_u, Y) = \hat{A}^c(y_u)\ . \quad (22)$$

The objective (22) represents the **intra-class loss**, as it optimizes data features to be tightened with their class center. We can minimize ℓ^{intra} with Algorithm 1 in an end-to-end manner. This trains the model to produce better ranking results of Problem (21) and optimizes the image-center similarity $\boldsymbol{D}_u$ which (21) depends on. These two steps of solving Problem (21) and minimizing ℓ^{intra} are shown as green boxes in Appendix Fig. 7.

5.3 Final training objective and pipeline

In summary, we can optimize the model by jointly solving the ranking Problems (17) and (21), and then minimize the regular task loss ℓ^{task} (e.g., classification), ℓ^{inter}, and ℓ^{intra} together with SGD. The final joint loss function is as follows:

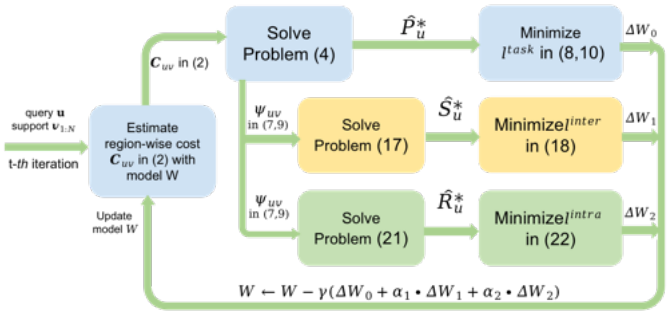
$$\ell^{final} = \ell^{task} + \alpha_1 \ell^{inter} + \alpha_2 \ell^{intra}\ , \quad (23)$$

with scaling factors α_1 and α_2 defaulted to be 0.2 for both in our experiments. We call our method of performing FS-MAP with ranking-based learning target (23) as **MAP-RegOT-Rank**.

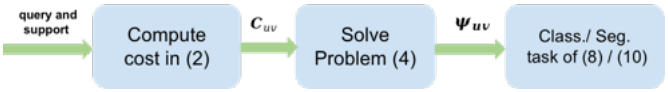
We summarize the whole training and testing processes in Fig. 7, with the algorithm details shown in Appendix Algorithm 4. We highlight the inter- and intra-rank learning modules in yellow and green, respectively, which integrate into the pipeline by joint optimization objective ℓ^{final} in (23).

6 Experiment

We evaluate our framework on distinct perception tasks and compare 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, as well as a self-collected human face dataset to verify face recognition performance with distinct viewpoints.



(a) Training process.



(b) Testing process.

Fig. 7: Flow chart of ranking-based feature learning approach.

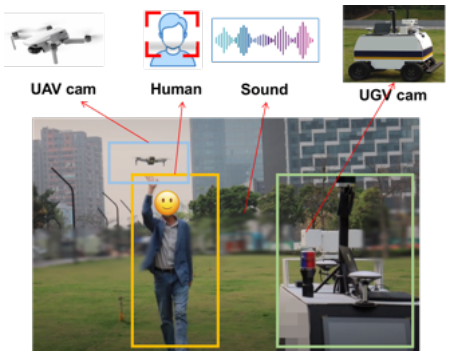


Fig. 8: Data collection with air-ground collaboration.

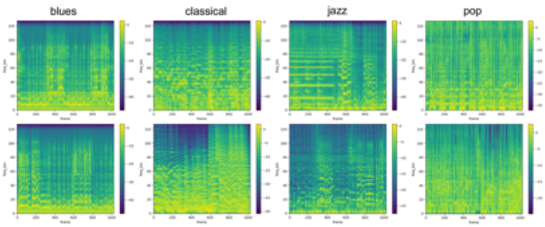


Fig. 9: Mel spectrograms of 2 samples of 4 genres.

6.1 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* [19] which simulates flying multiple drones over a series of landmarks in the AirSim "CityEnviron" environment [65]. 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. Appendix Table 7 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 Fig.8, 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. Appendix Table 8 shows the statistics of the dataset. We also use the large-scale CelebA [70] face dataset to pre-train our backbone models instead of directly training from FS-AirFace from scratch.

GTZAN [71] is a music genre dataset with soundtracks of 10 genres such as blues, classical, pop, rock. Each genre has 100 16-bit Mono sound waves of 30 seconds. We split the genres into 5 for training/validation and the rest 5 for testing. We convert the sound waves to the time-frequency domain by FFT and extract the Mel spectrograms as the 2D acoustic features. 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. Each column of Fig.9 shows the spectrograms of 2 sampled soundtracks for each of four genres in different columns.

6.2 Our approaches and baselines

We compare several variants 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-RegOT-Rank** further integrates ranking-based feature learning objectives (5) as our best method. **MAP-OT** is a baseline of MAP-RegOT which solves the original OT with LP solver as [26] 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** [13] and **When2Com** [19], and the current SOTA FSL approaches including MAML [72] and MTL [73] as representatives for optimization-based learners. In addition, we compare with state-of-the-art metric-based learners including ProtoNet [24] and RelationNet [28] for classification, as well as PANet [39] and MPNet [40] for segmentation. Note that MPNet [40] can also extend to distributed scenarios with its original attention design.

6.3 Implementation Details

For our approaches, we choose the ResNet-12 [3] as the backbone network $\boldsymbol{f}^{bone}$, for fair comparison with previous FSL studies as MAML [72] and MTL [73]. 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 ($\boldsymbol{f}^{qry}$, $\boldsymbol{f}^{key}$), 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 [19] feature vectors to be the same with ours, and set the query size of TarMAC [13] 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. For the ranking-based learning objective (23), we choose $\alpha_1 = \alpha_2 = 0.2$.

6.4 Results of few-shot segmentation

In Table 1, we compare different methods with 3-way 1-shot and 5-shot semantic segmentation tasks on FS-AirSim. In our setting,

TABLE 1: Segmentation results on FS-AirSim dataset.

Method	3-Way 1-Shot		3-Way 5-Shot	
	Acc	IoU	Acc	IoU
When2Com+MAML [19], [72]	0.593	0.203	0.733	0.310
When2Com+MTL [19], [73]	0.652	0.259	0.735	0.321
TarMAC+MTL [13], [73]	0.660	0.310	0.752	0.328
TarMAC+PANet [13], [39]	0.661	0.292	0.762	0.335
MPNet [40]	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
MAP-RegOT-Rank (ours)	**0.738**	0.322	**0.813**	**0.379**

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.12. 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-Rank performs the best among all approaches, leading MAP-RegOT by 1.5% and 3.8% in 1- and 5-shot tasks, relatively, thanks for the ranking-based feature learning.
- MAP-RegOT outperforms all other approaches except MAP-RegOT-Rank in both Acc and IoU. It outperforms the MAP-OT by 5% and 2.6% relatively in 1- and 5-shot tasks respectively, and larger for other baselines.
- MAP-RegOT outperforms MPNet by 3% (0.727 v.s. 0.705) due to the better image-level similarity measures provided by RegOT, while both significantly outperform other baselines which do not consider fine-grained feature matching.

TABLE 2: Classification results on FS-AirSim.

Method	5-Way 1-Shot		5-Way 5-Shot	
	Acc	mAP	Acc	mAP
When2Com+MAML [19], [72]	0.458	0.413	0.482	0.443
When2Com+MTL [19], [73]	0.516	0.480	0.530	0.591
TarMAC+MTL [13], [73]	0.503	0.485	0.601	0.602
TarMAC+ProtoNet [13], [24]	0.531	0.424	0.684	0.607
TarMAC+RelationNet [13], [28]	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
MAP-RegOT-Rank (ours)	**0.729**	**0.769**	**0.791**	**0.841**

TABLE 3: Face recognition results on FS-AirFace.

Method	5-Way 1-Shot		5-Way 5-Shot	
	Acc	mAP	Acc	mAP
When2Com+MTL [19], [73]	0.283	0.301	0.309	0.322
TarMAC+MTL [13], [73]	0.310	0.312	0.315	0.345
TarMAC+ProtoNet [13], [24]	0.596	0.642	0.602	0.643
TarMAC+RelationNet [13], [28]	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
MAP-RegOT-Rank (ours)	**0.700**	**0.775**	**0.716**	**0.820**

TABLE 4: Music genre classification results on GTZAN.

Method	3-Way 1-Shot		3-Way 5-Shot	
	Acc	mAP	Acc	mAP
When2Com+MTL [19], [73]	0.355	0.361	0.325	0.329
TarMAC+MTL [13], [73]	0.341	0.349	0.376	0.389
TarMAC+ProtoNet [13], [24]	0.498	0.512	0.541	0.558
TarMAC+RelationNet [13], [28]	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
MAP-RegOT-Rank (ours)	**0.603**	**0.635**	**0.744**	**0.816**

6.5 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 2 on FS-AirSim that

- MAP-RegOT-Rank performs the best among all approaches, outperforming MAP-RegOT by 9.6% and 9.9% relatively in 1- and 5-shot tasks, thanks for the improved feature learning.
- MAP-RegOT outperforms all other approaches except MAP-RegOT-Rank 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 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- and 5-shot tasks respectively, indicating the effectiveness of our fine-grained metric-learning approach.

We consider the few-shot face identification tasks on the FS-AirFace dataset in Table 3. We observe that MAP-RegOT-Rank performs the best among all approaches, outperforming MAP-RegOT by 4.3% (0.700 v.s. 0.671) and 3.3% (0.716 v.s. 0.693) relatively respectively in 1-shot and 5-shot tasks. Besides, 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.11. As our approach better considers the difference in query and support data's perspectives, it outperforms the baseline approaches naturally. We also show the precision-recall curve of MAP-RegOT and MAP-RegOT-Rank for each person in Appendix Fig.10 (a) and (b) respectively. Second column of Fig.10 shows that MAP-RegOT-Rank improves mAP especially in 5-shot case (0.820 v.s. 0.751), thanks for the additional ranking-based feature learning.

A similar trend for the few-shot music genre recognition tasks on GTZAN dataset is shown in Table 4. We observe that MAP-RegOT-Rank and 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.9. 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.

6.6 Discussions and ablation study

6.6.1 *The cost the signature-based communication*

We develop the asymmetric signature-based attention to balance comm. (communication) costs and performance. In particular, we extract query features of size $D_q \times H \times W$, where D_q is the feature dim and $H \times W$ is the spatial resolution. Since we broadcast only query feature, the comm. cost will be $D_q \times H \times W$, regardless of the support feature size. Previous works extracted feature vectors of spatial resolution $H = W = 1$. They either chose a large

vector size (e.g., 1024) to guarantee performance (TarMAC), or chose a smaller size (e.g., 32) to reduce comm. cost (When2Com) by sacrificing the performance. Instead of reducing performance or increasing comm. cost, we extract fine-grained feature maps of size $32 \times 8 \times 8$ to perform metric learning, by measuring the RegOT distance between query and support feature maps. Our approach outperforms the baselines by 5%-15% (Table 1-4) while using exact the same comm. costs in various tasks. We also 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, 0.647, and 0.643, respectively, on face recognition task. Compared with our MAP-RegOT (acc 0.671, Table 3), the best performance of TarMAC with dim 3072 (acc 0.647) costs 3 times of the comm. cost, while still underperforms our approach by 3.7%. This shows the superiority of our method in metric-learning design.

6.6.2 The efficiency of our methods

The inference speed of TarMAC is 1100 FPS (frame per second) while our MAP-RegOT and MAP-RegOT-Rank is 180 FPS, all with one Nvidia Titan V GPU. Our approach is slower because of the finer-grained metric-learning with RegOT. But still, the speed of our approach is practical and real-time. On the other hand, our approach significantly outperforms the baseline by 24% (0.700 v.s. 0.564 in Table 3). For a critical task such as searching for lost children, it's worthwhile trading speed for accuracy.

6.6.3 The benefit of inter- and intra-class regularization

We evaluate our methods on FS-AirSim with four ablation settings: MAP-RegOT, MAP-RegOT with inter-class loss (col. $+\ell^{inter}$), MAP-RegOT with intra-class loss (col. $+\ell^{intra}$), and with both losses (col. +both), a.k.a, MAP-RegOT-Rank. We take MAP-RegOT as the benchmark, and show the increased relative accuracy in Table 5.

TABLE 5: Ablation study on FS-AirSim.

setting	MAP-RegOT	+ ℓ^{inter}	+ ℓ^{intra}	+ both
1-shot	0.665	0.729 (+9.6%)		
5-shot	0.720	0.756 (+5.0%)	0.770 (+6.9%)	0.791 (+9.9%)

In 1-shot case, ℓ^{intra} is equivalent to ℓ^{inter} as the *mean* K-shot data is the 1-shot data itself. The ℓ^{inter} can improve accuracy with 9.6% upon MAP-RegOT with the optimal scaler $\alpha_1 = 0.4$.

In 5-shot case, both ℓ^{inter} and ℓ^{intra} are critical to the performance boost. Our MAP-RegOT-Rank (+both) yields 9.9% improvement of accuracy, while each of ℓ^{intra} and ℓ^{inter} yields performance boost less than 7% if used separately.

6.6.4 Comparison of different metrics

TABLE 6: Metrics on FS-AirSim classification task.

Method	5-Way 1-Shot		5-Way 5-Shot	
	Acc	mAP	Acc	mAP
MAP-L1	0.589	0.588	0.637	0.647
MAP-L2	0.573	0.565	0.621	0.635
MAP-Cosine	0.601	0.620	0.631	0.671
MAP-DeepBDC [29]	0.549	0.540	0.662	0.726
MAP-OT (DeepEMD) [26]	0.617	0.643	0.702	0.754
MAP-RegOT (ours)	0.665	0.697	0.720	0.793

We measure the query-support similarity with $L1$, $L2$, Cosine, DeepBDC [29], OT (a.k.a. DeepEMD [26]) and RegOT for comparison. The image-level $L1$ and $L2$ similarities are the sum of inverse patch-wise distance $d^l_{uv,ij}$ such that $\psi_{uv} = \sum_i \sum_j \exp(-d^l_{uv,ij})$ in which $l \in \{1, 2\}$. The image-level Cosine similarity is $\psi_{uv} = \sum_i \sum_j a_{uv,ij}$ with $a_{uv,ij}$ in (3). The DeepBDC [29] similarity uses the BDC pooling layer ρ_{bdc} to produce Brownian Distance Covariance matrices $\rho_{bdc}(\boldsymbol{q}_u)$ and $\rho_{bdc}(\boldsymbol{k}_v)$ as query and support image features and applies the cosine similarity to obtain ψ_{uv}.

We observe in Table 6 that *OT (a.k.a., DeepEMD) and DeepBDC substantially outperform $L1$, $L2$ and Cosine metrics* as better distance measures, consistent with previous studies [26], [29]. OT outperforms DeepBDC by 12% (0.617 vs. 0.549) and 6% (0.702 vs. 0.662) in 1-shot and 5-shot FS-MAP tasks, respectively. RegOT leads all metrics with virtues of OT and the additional differentiable formulation.

In our unique FS-MAP tasks, the agents often have different viewpoints and capture the object of interest in various parts of the image, as shown in Fig. 4. The OT and RegOT explicitly perform fine-grained patch-wise image matching, allowing them to precisely find aligned objects. In contrast, DeepBDC formulates the BDC matrix via spatial pooling operations, resulting in the loss of local spatial information. For this reason, we find that OT and RegOT are better metrics for dealing with FS-MAP tasks.

7 Conclusion

In this paper, we proposed to tackle multi-agent perception tasks in data-scarce scenarios. We designed a query-support communication mechanism to coordinate multiple support agents for perception tasks. We proposed a fine-grained metric-learning approach to robustly measure query-support similarities as an OT task. We further developed two ranking-based metric learning objectives to shape a better feature space. Extensive studies proved that our approaches can significantly improve FS-MAP results on various tasks, including face identification, semantic segmentation, and sound genre recognition.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Projects 62106156 and 62271434), Shenzhen Science and Technology Program (Project JCYJ20210324120011032), Guangdong Basic and Applied Basic Research Foundation (Project 2021B1515120008), Shenzhen Key Lab of Crowd Intelligence Empowered Low-Carbon Energy Network (No. ZDSYS20220606100601002), and the Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

[1] C. Fan, J. Hu, and J. Huang, "Few-shot multi-agent perception," in *ACM MultiMedia*, 2021, pp. 1712–1720.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1106–1114.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[5] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2015.

[7] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatio-temporal features with 3d convolutional networks," in *Intl. Conf. on Computer Vision*, 2015, pp. 4489–4497.

[9] C. Fan, X. Zhang, S. Zhang, W. Wang, C. Zhang, and H. Huang, "Heterogeneous memory enhanced multimodal attention model for video question answering," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 1999–2007.

[10] S. Sukhbaatar, R. Fergus *et al.*, "Learning multiagent communication with backpropagation," in *Advances in Neural Information Processing Systems*, 2016, pp. 2244–2252.

[11] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," in *Advances in Neural Information Processing Systems*, 2018, pp. 7265–7275.

[12] Y. Hoshen, "Vain: Attentional multi-agent predictive modeling," in *Advances in Neural Information Processing Systems*, 2017, pp. 2701–2711.

[13] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau, "Tarmac: Targeted multi-agent communication," in *Intl. Conf. on Machine Learning*, vol. 97, 2019, pp. 1538–1546.

[14] C. Hori *et al.*, "Attention-based multimodal fusion for video description," in *IEEE Intl. Conf. on Computer Vision*, 2017, pp. 4203–4212.

[15] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Conf. on Empirical Methods in NLP*, 2015.

[16] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[17] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems*, 2013, pp. 2292–2300.

[18] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Intl. Conf. on Machine Learning*, 1993, pp. 330–337.

[19] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 4105–4114.

[20] B. Song, A. T. Kamal, C. Soto, C. Ding, J. A. Farrell, and A. K. Roy-Chowdhury, "Tracking and activity recognition through consensus in distributed camera networks," *IEEE Trans. on Image Processing*, vol. 19, no. 10, pp. 2564–2579, 2010.

[21] U. Jain *et al.*, "Two body problem: Collaborative visual task completion," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 6689–6699.

[22] Y.-C. Liu, J. Tian, C.-Y. Ma, N. Glaser, C.-W. Kuo, and Z. Kira, "Who2com: Collaborative perception via learnable handshake communication," in *IEEE Intl. Conf. on Robotics and Automation*, 2020, pp. 6876–6883.

[23] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.

[24] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[25] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 4367–4375.

[26] C. Zhang, Y. Cai, G. Lin, and C. Shen, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 12 200–12 210.

[27] J. He, R. Hong, X. Liu, M. Xu, Z.-J. Zha, and M. Wang, "Memory-augmented relation network for few-shot learning," in *ACM Multimedia*, 2020, pp. 1236–1244.

[28] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[29] J. Xie, F. Long, J. Lv, Q. Wang, and P. Li, "Joint distribution matters: Deep brownian distance covariance for few-shot classification," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2022.

[30] H. Li, D. Eigen, S. Dodge, M. Zeiler, and X. Wang, "Finding task-relevant features for few-shot learning by category traversal," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1–10.

[31] H. Huang, J. Zhang, L. Yu, J. Zhang, Q. Wu, and C. Xu, "Toan: Target-oriented alignment network for fine-grained image categorization with few labeled samples," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 853–866, 2021.

[32] J. Xu, H. Le, M. Huang, S. Athar, and D. Samaras, "Variational feature disentangling for fine-grained few-shot classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8812–8821.

[33] B. N. Oreshkin, P. R. López, and A. Lacoste, "TADAM: task dependent adaptive metric for improved few-shot learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 719–729.

[34] Z. Peng, Z. Li, J. Zhang, Y. Li, G.-J. Qi, and J. Tang, "Few-shot image recognition with knowledge transfer," in *IEEE Intl. Conf. on Computer Vision*, 2019, pp. 441–449.

[35] D. Das and C. S. G. Lee, "A two-stage approach to few-shot learning for image recognition," *IEEE Trans. on Image Processing*, vol. 29, pp. 3336–3350, 2020.

[36] Z. Ji, X. Liu, Y. Pang, W. Ouyang, and X. Li, "Few-shot human-object interaction recognition with semantic-guided attentive prototypes network," *IEEE Trans. on Image Processing*, vol. 30, pp. 1648–1661, 2021.

[37] C. Fan and J. Huang, "Federated few-shot learning with adversarial learning," in *Intl. Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks*, 2021, pp. 264–271.

[38] L. Wu, Y. Wang, H. Yin, M. Wang, and L. Shao, "Few-shot deep adversarial learning for video-based person re-identification," *IEEE Trans. on Image Processing*, vol. 29, pp. 1233–1245, 2020.

[39] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *IEEE Intl. Conf. on Computer Vision*, 2019, pp. 9196–9205.

[40] P. Li, Y. Wei, and Y. Yang, "Meta parsing networks: Towards generalized few-shot scene parsing with adaptive metric learning," in *ACM Multimedia*, 2020, pp. 64–72.

[41] B. Liu, J. Jiao, and Q. Ye, "Harmonic feature activation for few-shot semantic segmentation," *IEEE Trans. on Image Processing*, vol. 30, pp. 3142–3153, 2021.

[42] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *Intl. Conf. on Computer Vision*, 2009, pp. 460–467.

[43] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Intl. Conf. on Machine Learning*, 2017, pp. 214–223.

[44] Q. Zhao, Z. Yang, and H. Tao, "Differential earth mover's distance with its applications to visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 274–287, 2010.

[45] T. Lin, C. Fan, N. Ho, M. Cuturi, and M. I. Jordan, "Projection robust wasserstein distance and riemannian optimization," in *Advances in Neural Information Processing Systems*, 2020.

[46] R. Geng, Y. Hu, Z. Lu, C. Yu, H. Li, H. Zhang, and Y. Chen, "Passive non-line-of-sight imaging using optimal transport," *IEEE Trans. on Image Processing*, vol. 31, pp. 110–124, 2022.

[47] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Intl. Conf. on Computer Vision*, 1998, pp. 59–66.

[48] J. Xu and H. Li, "AdaRank: a boosting algorithm for information retrieval," in *ACM SIGIR*, 2007, pp. 391–398.

[49] T.-Y. Liu, *Learning to Rank for Information Retrieval*, 2009.

[50] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 98:1–98:10, 2015.

[51] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Conf. on Computer Vision and Pattern Recognition*, 2006.

[52] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," in *Intl. Conf. on Learning Representations*, 2014.

[53] J. Wang *et al.*, "Learning fine-grained image similarity with deep ranking," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2014.

[54] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Intl. Conf. on Computer Vision*, 2015.

[55] C. Fan, J. Lee, M. Xu, K. K. Singh, Y. J. Lee, D. J. Crandall, and M. S. Ryoo, "Identifying first-person camera wearers in third-person videos," in *Conf. on Computer Vision and Pattern Recognition*, 2017.

[56] M. Xu, C. Fan, Y. Wang, M. S. Ryoo, and D. J. Crandall, "Joint person segmentation and identification in synchronized first- and third-person videos," in *European Conf. on Computer Vision*, 2018.
[57] T. Joachims, "Optimizing search engines using clickthrough data," in *ACM Conf. on Knowledge Discovery and Data Mining*, 2002.
[58] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Intl. Conf. on Machine Learning*, 2005.
[59] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Intl. Conf. on Machine Learning*, 2007.
[60] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, "Listwise approach to learning to rank: theory and algorithm," in *Intl. Conf. on Machine Learning*, 2008.
[61] M. Cuturi, O. Teboul, and J.-P. Vert, "Differentiable ranking and sorting using optimal transport," in *Advances in Neural Information Processing Systems*, 2019.
[62] M. Blondel, O. Teboul, Q. Berthet, and J. Djolonga, "Fast differentiable sorting and ranking," in *Intl. Conf. on Machine Learning*, 2020.
[63] Y. Xie, H. Dai, M. Chen, B. Dai, T. Zhao, H. Zha, W. Wei, and T. Pfister, "Differentiable top-k with optimal transport," in *Advances in Neural Information Processing Systems*, 2020.
[64] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 2137–2145.
[65] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, vol. 5, 2018, pp. 621–635.
[66] P. A. Knight, "The sinkhorn–knopp algorithm: convergence and applications," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 1, pp. 261–275, 2008.
[67] J. Altschuler, J. Niles-Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration," in *Advances in Neural Information Processing Systems*, 2017, pp. 1964–1974.
[68] S. Barratt, "On the differentiability of the solution to convex optimization problems," *arXiv preprint arXiv:1804.05098*, 2018.
[69] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010, pp. 2528–2535.
[70] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *IEEE Intl. Conf. on Computer Vision*, 2015, pp. 3730–3738.
[71] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals"," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, 2002.
[72] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Intl. Conf. on Machine Learning*, vol. 70, 2017, pp. 1126–1135.
[73] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.

Chenyou Fan (Member, IEEE) received the B.S. degree in computer science from the Nanjing University, China, in 2011, and the M.S. and Ph.D. degrees from Indiana University, USA, in 2014 and 2019, respectively. He serves as the Associate Professor in South China Normal University. His research interests include machine learning and computer vision. He served in the program committee of CVPR, NeurIPS, ACM MM and top AI journals for more than 20 times.

Junjie Hu (Member, IEEE) received the B.S. degree in computer science and technology from the Tianjin University of Science and Technology, Tianjin, China, in 2014, and the M.S. and Ph.D. degrees from the Graduate School of Information Science, Tohoku University, Sendai, Japan, in 2017 and 2020, respectively. He serves as a research scientist with the Shenzhen Institute of Artificial Intelligence and Robotics for Society, China. His research interests include machine learning, computer vision, and robotics.

Jianwei Huang is a Presidential Chair Professor and Associate Vice President of the Chinese University of Hong Kong, Shenzhen, and the Associate Director of Shenzhen Institute of Artificial Intelligence and Robotics for Society. He received the Ph.D. degree in ECE from Northwestern University in 2005, and worked as a Postdoc Research Associate in Princeton University during 2005-2007. From 2007 until 2018, he was on the faculty of Department of Information Engineering, The Chinese University of Hong Kong. His research interests are in the area of network optimization, network economics, and network science, with applications in communication networks, energy networks, data markets, and crowd intelligence. He has published 320+ papers in leading international venues, with a Google Scholar citation of 15000+ and an H-index of 62. He has co-authored 11 Best Paper Awards, including the 2011 IEEE Marconi Prize Paper Award in Wireless Communications. He has co-authored seven books, including the textbook on "Wireless Network Pricing." He has been an IEEE Fellow, an IEEE ComSoc Distinguished Lecturer, a Clarivate Web of Science Highly Cited Researcher, and an Elsevier Most Cited Chinese Researcher. He is the Editor-in-Chief of IEEE Transactions on Network Science and Engineering (TNSE), and was an Associate Editor-in-Chief of IEEE Open Journal of the Communications Society (OJ-COM).