



# Context-adaptable radar-based people counting via few-shot learning

Gianfranco Mauro<sup>1,2</sup> · Ignacio Martinez-Rodriguez<sup>1</sup> · Julius Ott<sup>1,3</sup> · Lorenzo Servadei<sup>1,3</sup> · Robert Wille<sup>3</sup> · Manuel P. Cuellar<sup>4</sup> · Diego P. Morales-Santos<sup>2</sup>

Accepted: 8 June 2023  
© The Author(s) 2023

## Abstract

In many industrial or healthcare contexts, keeping track of the number of people is essential. Radar systems, with their low overall cost and power consumption, enable privacy-friendly monitoring in many use cases. Yet, radar data are hard to interpret and incompatible with most computer vision strategies. Many current deep learning-based systems achieve high monitoring performance but are strongly context-dependent. In this work, we show how context generalization approaches can let the monitoring system fit unseen radar scenarios without adaptation steps. We collect data via a 60 GHz frequency-modulated continuous wave in three office rooms with up to three people and preprocess them in the frequency domain. Then, using meta learning, specifically the Weighting-Injection Net, we generate relationship scores between the few training datasets and query data. We further present an optimization-based approach coupled with weighting networks that can increase the training stability when only very few training examples are available. Finally, we use pool-based sampling active learning to fine-tune the model in new scenarios, labeling only the most uncertain data. Without adaptation needs, we achieve over 80% and 70% accuracy by testing the meta learning algorithms in new radar positions and a new office, respectively.

**Keywords** Active learning · Meta learning · Radar · Few shot learning · People counting · Weighting network

## 1 Introduction

Counting the number of people in an environment can be a crucial task not only in industrial settings but also in medical and safety scenarios. In difficult times, such as during a pandemic, keeping track of the occupancy of an environment can greatly reduce the risk of spreading a pathogen [1, 2]. Estimating the presence of people can lead to other advantages, such as enabling energy management plans in places with frequent turnover of people, such as hospitals, by smartly activating equipment and heating systems [3]. A non-automated measure may be challenging or impossible in many contexts, such as for pedestrian crowds in public areas [4]. The majority of solutions designed for people monitoring rely on images captured by cameras and thermal sensors [5]. Most camera-based solutions use RGB or time of flight (ToF) sensors, and occupancy information is estimated using computer vision [6, 7] or machine

learning [8–10]. Camera systems that use cross techniques for image segmentation and edge detection, such as convolutional neural networks (CNNs), achieve high performance even in crowded environments, but suffer from the inherent problem of a lack of privacy [11]. Thermal sensors, on the other hand, are much less privacy-invasive because of the usage of infrared frequencies and often lower image resolution [12]. Thermal sensors also have the advantage of being usable in the dark, but they can be affected by thermal noise, caused, for example, by heaters and sunlight. In addition, the lack of depth information generally does not allow distinguishing between people moving in the same direction. In contrast to visual solutions, many other systems exploit the measurement of environmental quantities. Radio-frequency (RF) and laser technologies are typically classified as non-image-based approaches [13]. The CO<sub>2</sub> sensors, for example, can be used to estimate the occupancy of a room by the concentration of carbon dioxide produced by individuals. Such systems are frequently low-power but must account for venting systems and are practically unusable in open spaces [14]. LiDARs represent often another privacy-friendly solution for people counting and tracking. Through the use of pulsed lasers and a scanner, a LiDAR yields the generation of 2-D or 3-D maps of the surrounding

✉ Gianfranco Mauro  
gianfranco.mauro@infineon.com

✉ Manuel P. Cuellar  
manupc@ugr.es

Extended author information available on the last page of the article

space [15, 16]. Such systems frequently have high spatial resolution and frame rates, but they can be costly and power-consuming. RF-based systems have the advantage of having almost no privacy concerns and little dependence on light and weather conditions. These characteristics make them appropriate for monitoring several people. Wi-Fi technology, for example, can enable the recognition and segmentation of people even through walls and obstructions [17, 18]. Wi-Fi modules, however, require the development of high output power in the RF range ( $\approx$  W) and a continuous working operation to exploit their functionalities. On the contrary, radar sensors are more versatile in many applications thanks to lower power consumption ( $\approx$  mW) and optimized system power management. Among radar modulations, frequency-modulated continuous wave (FMCW) is particularly suited to people monitoring, allowing accurate estimation of the range and velocity of both dynamic and static targets located within the device's field of view (FoV) [19, 20]. Specifically, 60 GHz technology is particularly suitable for short-range people monitoring applications [21]. Radars transmitting around this frequency are cost-effective and versatile compared to other solutions such as cameras or LiDAR. Further, the 60 GHz frequency is much less susceptible to interference with other radio-frequency signals or Bluetooth devices. Image-based or high-resolution RF systems often implement a vision-based pipeline to predict the number of people in a given context. This approach can lead to high classification performance even in the challenging task of tracking through image segmentation, edge detection, and skeleton-pose extraction [6]. On the other hand, radar data are hardly interpretable through classical computer vision approaches. In this case, deep learning (DL) techniques are commonly used to process the information [22].

DL is nowadays finding the most varied uses for solving tasks and speeding up processes. Over the years, classes of DL models have been developed to extract valuable information from the available data for given tasks. Examples are CNNs for feature map generation or recurrent neural networks (RNNs) for processing time series. Over the years, multiple neural network topologies, such as Inception [23] and VGGNet [24] have been designed to solve specific tasks with successful outcomes. Yet, such topologies have the inherent need to be trained on a large amount of data to achieve robust performance across new contexts. Commonly, these models are adaptable to new tasks by leveraging transfer learning [25], tailoring parameters to newly collected data. However, the limited availability of data and the need for rapid adaptation to new contexts make transfer learning hardly usable for defined types of tasks. To deal with these challenges, a specific branch of DL called few-shot learning has gained momentum in recent years [26]. The goal of few-shot learning is to exploit the little available information and data patterns, leveraging previous experience to adapt

to new contexts or solve tasks that have not been tackled before. Few-shot learning is approached from different perspectives by specific DL sub-branches such as meta learning and active learning [27, 28].

Meta learning, or *learning to learn*, accounts for the set of algorithms where the primary goal is to learn how to approach new tasks given some past experience, or meta-data [29, 30]. This process not only encourages context generalization but also accelerates the fine-tuning of already observed tasks when new data are available. If the meta learning is optimization-based, an iterative learning process called episodic learning based on available training data is generally used. For a task defined in  $N$ -way, i.e.,  $N$  classes, the few available samples are called shots. To assess generalization performance,  $C$  samples of *support* and  $J$  samples of *query* are fed to the defined model for each class. Algorithms commonly used for meta learning are model agnostic meta learning (MAML) [31] and Reptile [32] which, thanks to their very general conceptualization, enable the episodic adaptation of most of the common topologies defined in DL. Frameworks based on optimization-based meta learning are highly effective and perform well in several data-poor tasks [33, 34]. However, they have an inherent need for training on a set of representative data for each new, unseen task to learn to generalize. A specific kind of method, called relation network [35], was created to obviate this need by exploiting the ability of the model to compare the features of different examples and learn to distinguish them. The comparison is possible by properly shaping the model topology and regressing a relation score between 0 and 1, comparing individual support and query examples. The relation scores are unconventionally regressed by minimizing the mean squared error (MSE) to the ground truth of query instances. This approach assumes that all available support instances are mutually independent of each other. Intuitively, the model relies on a one-to-one comparison rather than comparing the new query examples with all the available support samples. Such issues are addressed by the weighting network [36]. In this adapted topology, the relation between support and query is propagated through two modules. A first comparison module for the extraction of the similarity between the samples and a second weighting module that compresses the information into a one-dimensional vector representing the relation scores. This method leverages all available support sample features for query prediction. Further, the weighting network endorses the use of traditional classification cost functions such as crossentropy during episodic optimization.

Active learning, on the other hand, aims to optimize the model's performance with as few labeled instances as possible [37, 38]. To accomplish this, the algorithm has control over the inputs on which it trains, labels, or requests additional information about the data it deems most useful for learning. A common strategy is to assign a *priority score* to

the unlabeled data pool, exploiting, for example, the probability distribution generated by the model. Only the instances identified as most uncertain are then labeled and used during training. This procedure, called pool-based sampling, is normally repeated multiple times, increasing the amount of labeled training data, until satisfactory performance for a given task is achieved.

In this paper, we exhibit how few-shot learning techniques can grant generalization of scenarios (environments and locations) for an FMCW radar-based algorithm designed for people counting. The application of this system is intended for uncrowded areas or rooms where there is a need to count the presence of a few people. For this work, a specific dataset was collected using a 60 GHz radar that was set up for the task of counting people. The information was gathered in three different offices with at least four different in-room locations. Per location, 0 to 3 people took part in the data recording for at least 60 seconds per session. The data were preprocessed in frequency to extract range and Doppler information from the people in the scene. Meta learning is then used for the monitoring use case, estimating the number of people from radar data. Instead of using all the available data in a single training, we propose a few-shot episodic approach to foster and speed up adaptation. To meet the learning needs, we introduce both a new relation topology, which we call the Weighting-Injection Net, and an algorithm, which we call model-agnostic meta-weighting (MAMW). The Weighting-Injection Net represents a modification to the traditional weighting network presented in [36]. Instead of an embedding module that reduces the dimensionality of the support samples for the next comparison step, the proposed one uses an injection module. This module increases the dimensionality of input data, generating a feature-enriched representation of support and query samples for the next relational phase. The overall network scheme is shown in Fig. 1. The MAMW, on the other hand, combines the query relation strategy of the weighted network with the two-step optimization-based approach of MAML. This is meant to improve the stability of the few-shot episodic training, especially when only very few instances are available as training. Experiments with 1-, 2-, 5-, and 10-shot have been performed and analyzed for the proposed methods. The achieved generalization results have been compared with those of other state-of-the-art approaches. State-of-the-art comparisons are also conducted up to five-person counting, to test the limitations of the radar-based episodic approach.

We also exhibit how pool-based sampling active learning can be efficiently employed to fine-tune the performance of a relational model by exploiting the most uncertain data. Showing how, for adaptations in new contexts, the use of generalization information learned from episodic adaptation leads to a better fit than starting from random initialization. The active learning strategy has been used to fit the 1-shot-

pre-trained model on data from an office room used as a test that is therefore unseen in the meta-training phase.

For the meta learning algorithms, we also conducted experiments on a publicly available dataset for few-shot learning in the Appendix A. The main contributions of this paper are as follows:

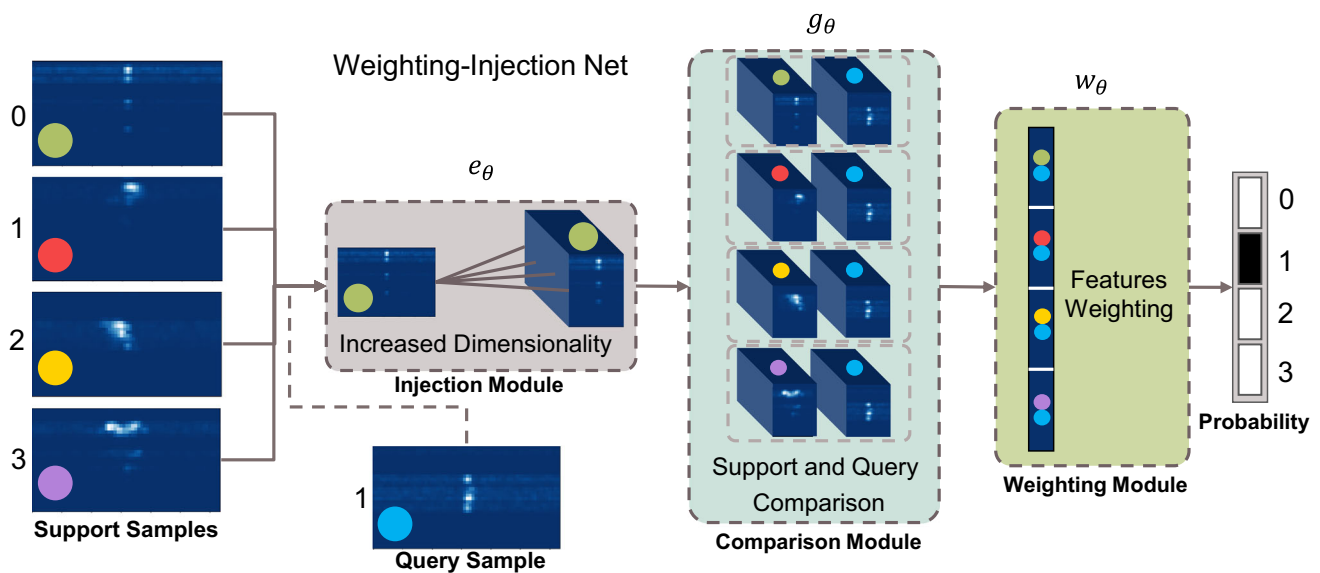
1. Implementation, to the best of our knowledge, of the first context-adaptable radar-based solution for counting people without a necessary adaptation training.
2. Design and implementation of the Weighting-Injection Net. This network represents a variation of the weighting network with an injection module. The injection operation increases the dimension of support and queries to ease feature matching in the subsequent comparison module.
3. Design of a cross-algorithm between MAML and the weighting network, called MAMW to increase the training stability of 1- and 2-shot experiments.
4. Development of a pool-based sampling active learning algorithm compatible with weighting network topologies.

## 2 Related works

In this section, we first investigate state-of-the-art solutions for people counting that offer similar features to radar-based systems, such as privacy preservation and low frame resolution. We then focus on the specific approaches aimed at context generalization and active learning.

When low frame resolution and privacy are system needs, traditional image segmentation and detection methods are often replaced or aided by deep learning. Neural networks can also be used to process time series or generate density maps for crowd monitoring.

Massa et al. [39] presented a recurrent neural network (RNN) architecture called LRCN-RetailNet (Long-term recurrent convolutional network) that takes as input sequences of low-resolution RGB frames and analyzes their spatiotemporal content for people counting. The strategy outperforms other state-of-the-art single-image-based approaches. The system based on temporal sequences may be unusable in low frame rate scenarios or with hardware implementation constraints. Gomez et al. [40] developed a system using long-wave infrared imaging and a CNN implementation on the NXP<sup>®</sup> LPC54102 microcontroller. The classification approach is binary, exploiting a small detection window on image sections to predict the presence or absence of heads. Because all weights fit in a 512 KB flash memory, the CNN can be easily deployed on the microcontroller. The counting algorithm using the embedded version of the model achieves an accuracy of 53.7% on test images and up



**Fig. 1** Weighting network with an injection module (Weighting-Injection Net). At least one instance per class, represented in the figure with a different marker color and a label, is used as support. A query example belonging to one of the classes is what is to be associated with a label by the classification algorithm. An injection module trained on the support images enables the concatenation of a query with an increased-

dimensionality representation of each support. A comparison module merges support and query information by mapping the relation into a one-dimensional vector. Finally, a weighting module composed of fully connected layers maps the relational information to the query label. The model parameters are represented by  $\theta$

to six people. This solution is very low-power and privacy-friendly, but the presence of heat sources in the environment could cause counting issues due to the low resolution of the thermal sensor.

The most common types of RF-based systems used for monitoring are Wi-Fi and radars that use impulse radio ultra-wide band (IR-UWB) or FMCW technology. Most of these solutions are inherently characterized by privacy preservation and low sensor resolution. Kianoush et al. [41] presented a people counting system via Wi-Fi radio infrastructure that uses an ensemble of models to leverage the space-frequency features of various transmission and reception channels. The ensemble exploits Bayesian techniques based on signal propagation statistics from RX to TX, a feed-forward neural network (FF-NN), and long-short-term memory (LSTM). Some of the constructed ensembles achieve an accuracy of over 95% in the test setup. However, a network of Wi-Fi terminals is employed for this purpose, which results in higher power consumption and challenges usability in other environments. Bao et al. [42] featured a CNN-based algorithm for people counting focusing on extracting multi-scale range-time maps from IR-UWB radar data. Sequences of radar frames are preprocessed to extract the peak information and remove the background. The single frames are then stacked together to form range-time maps. The method proved robust in counting up to 10 people in the selected environment. However, the time dependency and lack of velocity information may make the system unsuitable for

real-time applications where multiple people may be at the same distance. Stephan et al. [43] proposed a people counting solution via the *BGT60TR13* radar system (60 GHz FMCW) that makes use of knowledge distillation from synchronized camera data during the model generation. The suggested architecture first processes the camera RGB data, exploiting an OpenPose network that extracts the people's poses through pre-trained layers of the VGG-16 network and a multi-stage CNN. The extracted information is then fed to a triplet network with a 32-D embedding layer to generate clusters for each person count class. Radar information is first preprocessed in the form of range Doppler images (RDI) and fed to an encoder with fully connected final layers that learn through knowledge distillation from camera embeddings. Information transfer is possible by minimizing the Kullback-Leibler (KL) divergence between radar and camera embeddings. The method is robust and leads, in the test phase, to an accuracy of up to 71% for six people with another radar sensor with different positions and orientations. What is learned through knowledge distillation, however, could significantly affect the capabilities of the architecture in new environments where morphological and light conditions would directly influence the camera data.

A few cutting-edge works attempt to solve the people counting problem through active learning or aim at context generalization.

Vandoni et al. [44] featured a solution that uses active learning, coupled with SVMs, to improve training on subar-

eas of crowd images via head count. Samples that are more dissimilar than those already tagged are estimated in terms of their uncertainty via a metric that accounts for crowd density, called maximum excess over subarrays (MESA). Zhao et al. [45] also proposed an active learning solution for head counting in camera-based density maps. In this case, in the iterative process of instances sampling to be labeled, both crowd density information and dissimilarity from previous selections are employed. The sampling technique is a context-appropriate version of partition-based sample selection with weights (PSSW). The number of people is then regressed through mean absolute error (MAE) and MSE. Both methods presented in [44] and [45] result effective in improving the people count through uncertainty sampling in crowded scenes but are very dependent on the 2D RGB nature of the images. Zhang Yingying et al. [46] proposed a multi-column convolutional neural network (MCNN) to estimate crowd head counts from single images without temporal dependence. Even with a sparse number of people, the method outperforms other cutting-edge solutions on a variety of public datasets. The model, trained on a large dataset with various density map sizes, can be easily tuned for new datasets and contexts via transfer learning. The required resolution is nonetheless high and could create context-specific privacy issues. Reddy et al. [47] and Zan et al. [48] designed an adaptive algorithm to generate crowd density maps using MAML with episodic training. In [47] a backbone consisting of the first layers of VGG-16 and a density map estimator are trained on various RGB sequences collected in different environments. The pioneering approaches depict how meta learning can be effectively employed for people counting. Hou X. et al. [49] presented a cross-domain solution for the estimation of density maps by episodic learning. In this case, a domain-invariant feature representation module is exploited, where synthetic and real camera data are used as source and target domains, respectively. The density maps are then generated using a pre-trained CNN network and an algorithm called  $\beta$ -MAML, where  $\beta$  represents the generalization step's learning rate. The parameter  $\beta$  is dynamically adapted in the episodes by exploiting the gradient information of parts of the images. The number of people is finally estimated from the density maps. The meta learning approach presents more robust performance for the algorithm than other state-of-the-art methods for density map generation. However, the need for a sensor camera does not allow for low-resolution uses or where privacy is a requirement.

Some cutting-edge RF-based works also propose adaptive context generalization solutions. Hou H. et al. [50] illustrated a few-shot learning solution for indoor crowd counting using Wi-Fi technology. The solution consists of a two-stage framework called domain-agnostic and sample-efficient wireless indoor crowd (DaseCount). In a first stage of meta-training, two separate CNNs learn to extract human

activity information from wireless channel state information (CSI) measurements. Generalization performance is improved at this stage by knowledge distillation. In the meta-testing phase, the features extracted via CNNs from the CSI data are fed to a few-shot regression algorithm for the people counting task. The presented framework achieves, on average, over 96% accuracy for counting up to eight people in various domain setups. Yet, the solution is computationally expensive for classifier retraining and may not be suitable for frequent Wi-Fi transceiver location changes. Zhang Yong et al. [51] proposed a WI-FI-based few-shot learning solution for activity recognition that makes use of graph neural networks. The method uses a graph convolutional block attention module to extract activity-related information from CSI data. A final classification layer is used to classify the graph features and recognize the activity. The approach presents a robust 99.74% accuracy in the 5-way 5-shot experiment for new environments and activities. Yet, much computation and memory are required for model adaptations.

### 3 System setup and radar preprocessing

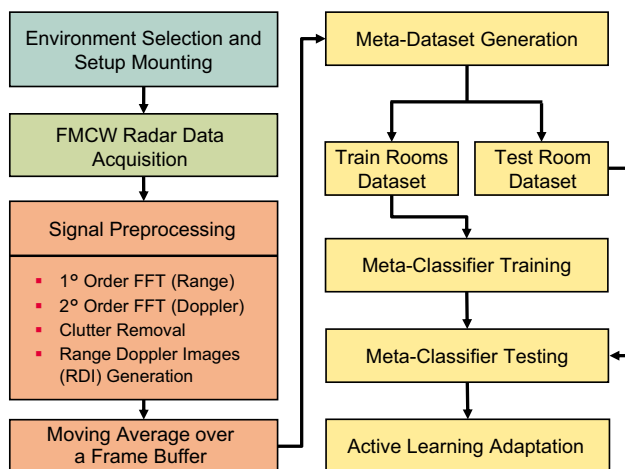
In this section, we propose a general overview of the system, discuss the data acquisition setup, and provide information about the employed radar board, its configuration, and the main preprocessing steps.

#### 3.1 General overview of the system

Figure 2 depicts the overall framework. First, rooms for data gathering are chosen for the few-shot learning approach. The radar data are then gathered from various in-room locations with varying numbers of people. Preprocessing is performed to extract range and Doppler information about the people in the FoV of the device. The sequences of preprocessed frames are averaged by moving average to generate the individual instances of the meta-dataset. The data are then saved and labeled in session-specific folders. The folder names denote the label encoding, from 0 to 3, of the number of people who attended the session. In most of the proposed experiments, the information recorded in two rooms is used as input data for the episodic training of the meta learning model. The third room is instead utilized for testing. Model fine-tuning can be performed via active learning on the test data, using the meta learning model as a baseline.

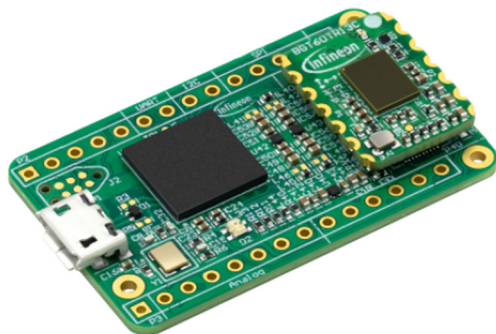
#### 3.2 Radar board

All radar data in this work were collected using the *BGT60TR13C* FMCW sensor [21] from Infineon Technologies AG. With a center frequency of  $f_0$  of 60 GHz



**Fig. 2** Proposed Framework. The setup is mounted in three rooms. Data sessions with a number of people from 0 to 3 in the scenario are collected and processed (orange). The frequency analysis is performed via the fast Fourier transform (FFT). Instances are generated via a moving average over frame sequences. A meta-dataset is then generated, and one room is used as the test dataset. A classifier is then episodically trained and tested. Active learning is used to fine-tune the model to a new environment (yellow)

and a bandwidth of about 6 GHz, this radar represents a miniaturized and low-power solution. This  $f_0$  and bandwidth are especially suitable in short-distance and indoor applications, resulting in low susceptibility to interference with other signals such as WiFi or Bluetooth. Thanks to an operation-optimized duty cycle, the power consumption for sensing within 5 m is minimized to only 5 mW. The *BGT60TR13C* has a transmit (TX) and three receive (RX) channels built into the package. The RX antennas are placed orthogonally to each other to enable the reconstruction of azimuth and elevation angles of arrival (AoA) for the targets placed in the FoV. The information collected from the RX channels is mixed with the TX and digitized with 12-bit resolution via the board connected to the radar sensor (Fig. 3).



**Fig. 3** *BGT60TR13C* Radar System. The board filters, mixes, and digitizes data from each RX channel, located on top of the radar sensor

### 3.3 Radar configuration

The *BGT60TR13C* transmits a series of linearly frequency-modulated signals called chirps in a defined bandwidth  $B_w$  around the central frequency  $f_0$ . Each chirp, of duration  $t_c$ , normally consists of a fixed number of samples  $n_s$ . During use, the information reflected in the RX channels is mixed with a transmitted signal reference and digitized, thus generating an output signal called intermediate frequency (IF). Normally, for further preprocessing, the radar information is packed into frames, each containing the IF relative to a sequence of chirps  $N_c$ . The theoretical maximum detection range  $R_{max}$  and range resolution  $\Delta r$  of an FMCW modulation are calculated using the following formulas:

$$\Delta r = \frac{c}{2B_w}, \quad (1)$$

$$R_{max} = \frac{\Delta r}{2} n_s, \quad (2)$$

where  $c$  stands for the speed of light in air. A narrow  $B_w$  of 0.48 GHz was chosen to achieve a  $R_{max}$  of about 10 m, which would cover the entire size of the chosen environments. A resolution  $\Delta r$  of at least 31 cm was chosen to let several targets placed in front of the radar be distinguished even at a considerable distance. A  $n_s$  per chirp of 64 has been specifically selected. The maximum discernible velocity of the targets  $V_{max}$  in one direction and the resolution  $\Delta v$  can instead be calculated with the following formulas:

$$V_{max} = \frac{c}{4f_0 t_c}, \quad (3)$$

$$\Delta v = \frac{2V_{max}}{N_c}. \quad (4)$$

The average human walking speed is about 1.42 m/s. To allow detecting even faster motions, we opted for a  $V_{max}$  of 3.5 m/s and a  $\Delta v$  of 1.1 cm/s. As a result, we set  $t_c$  to 351  $\mu$ s and  $N_c$  to 64. To collect approximately seven frames every half second, a frame repetition time  $f_{ps}$  of 75 ms was chosen. Furthermore, an analog-to-digital converter (ADC) sampling rate  $F_s$  of 2 MHz was chosen. The parameters used to configure the *BGT60TR13C* for the people counting recordings in all the selected rooms are listed in Table 1.

### 3.4 Recording setup

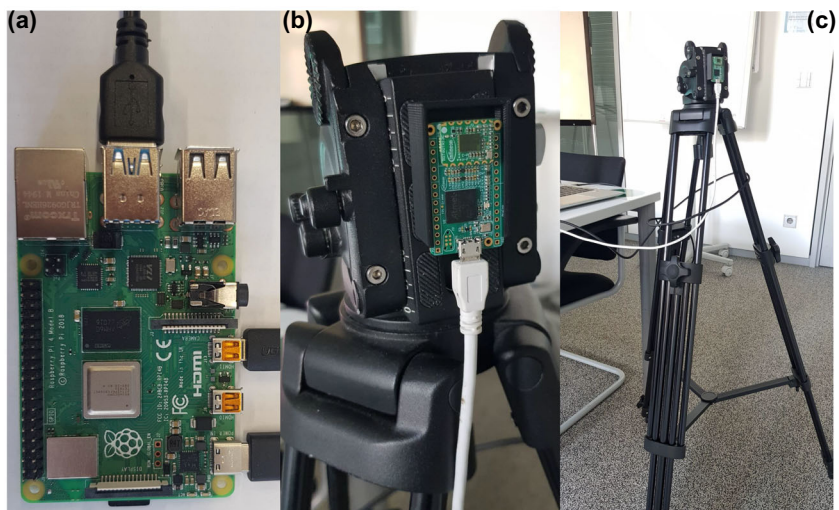
The *BGT60TR13C* radar system was mounted on a tripod for the people counting data, and the data were collected using a Raspberry<sup>®</sup> Pi 4. The raw radar data were then processed and labeled offline at a later time on an eight-generation Intel<sup>®</sup> Core<sup>™</sup> i5 processor (4 cores). Figure 4 depicts the used setup. Three different rooms of various sizes were chosen

**Table 1** Radar Sensor Parameters Configuration

Symbol	Quantity	Value
$f_0$	center frequency	60 GHz
$f_{ps}$	frames per second	13.33
$N_c$	number of chirps	64
$n_s$	samples per chirp	64
$t_c$	chirp time duration	351 $\mu$ s
$B_w$	bandwidth	[59.78 – 60.26] GHz
$F_s$	sampling frequency ADC	2 MHz

for data collection: an office of approximately 26 m<sup>2</sup> and two meeting rooms of about 20 and 39 m<sup>2</sup>, respectively. Only a portion of the office has been used, with walls separating the other two areas. Various types of furniture, such as cabinets, desks, tables, and chairs, were left in the rooms and were unremoved from their locations. The reflection of such objects represents the so-called clutter that characterizes the FMCW radar data. A graphical illustration of the three environments, indicated with the letters *S*, *M*, and *B*, standing for small, medium, and big, is provided in Fig. 5. Data were gathered in each room from at least the four corners. Data were also collected in three additional locations in the office room. At every location, the tripod was set up at a height ranging from 1.65 to 1.75 meters. Four sessions have been carried out per location, each lasting approximately 60 seconds for the meeting rooms and 90 seconds for the office. Each session contains data from 0 up to a maximum of 3 people in the room at the same time. Ten different people with heights ranging from 1.60 to 1.78 meters took part in the recordings. Some data up to 5 people have been gathered in the big room to further test the performance of the developed algorithm. Before collecting data, user consent was obtained, and as much privacy and data anonymization as possible were maintained during the

**Fig. 4** Data recording setup. A Raspberry<sup>®</sup> Pi4 (a) is used for data storage. For data collection, the BGT60TR13C radar system is mounted on the tripod (b). The tripod is moved between sessions in the various rooms and locations (c)



recordings. The collected data has not been made publicly available.

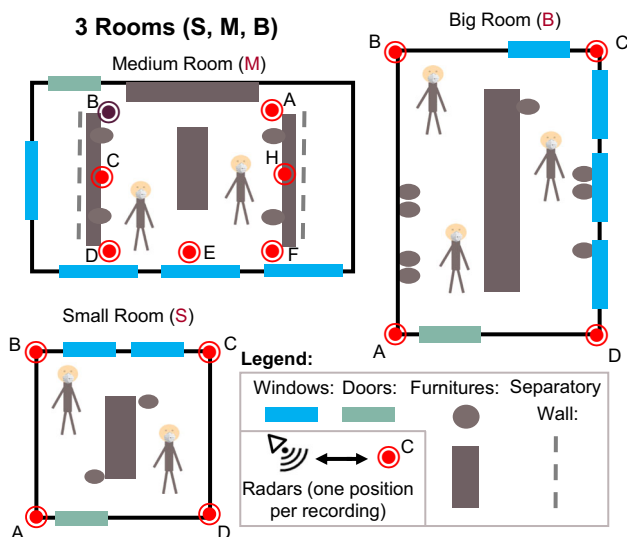
### 3.5 Radar preprocessing

Raw radar frames are difficult to interpret and label. The information to be fed to a DL model for learning purposes can be too noisy and highly context-dependent due to clutter. In this work, we propose to preprocess the raw data collected for people counting by removing the clutter and extracting the Doppler and range information of the targets through frequency analysis with the fast Fourier transform (FFT). We then perform two averages to reduce the noise in the data for the next model generation step. One for each frame, averaging the IF signal  $Ch_{IF}(i)$  generated for each of the three RX channels ( $i \in I_{RX}$ ), and another for each 7-frame recorded series. The whole process, given the  $f_{ps}$  of 75 ms, leads to the generation of about 2 RDI per second. The main preprocessing steps are shown in Fig. 6.

The preprocessing steps performed for each RX-generated IF signal are as follows:

1. For each chirp (slow time), the average value of the samples (fast time) is calculated and then subtracted.
2. The IF signal is then multiplied in fast time with a Hanning window to reduce the spectral leakage effects.
3. A 1-D FFT is performed on the samples to derive the range information of the targets.
4. A multiplication with a Hanning window is run also in the slow time.
5. A 1-D FFT is performed along the slow time to obtain the velocity information.
6. To drop the information of static objects, aka clutter, moving target indication (MTI) is utilized (5).

$$Ch_{IF}(i) = \mu Ch_{IF}(i) + (1 - \mu) \overline{Ch_{IF}}(i), \quad (5)$$



**Fig. 5** A graphic illustration of the environments chosen for data collection. Data from 0 to 3 people were collected from the four corners of the rooms. For the office *M*, data were also gathered at three other locations (C, E, and H, respectively). For *M*, data could not be collected from location B due to the presence of the front door

where  $\mu \in [0, 1]$  is set to 0.9, and weights the importance of the current frame against the average of the previous ones  $\overline{Ch_{IF}}(i)$ .

7. For each  $Ch_{IF}(i)$  a constant false alarm rate (CFAR) algorithm is used to locally select Range and Doppler peaks in frequency and discard the surrounding information, thus increasing the signal-to-noise ratio (SNR).
8. To further improve the SNR, the  $RDI_s(v)$  for each frame  $v \in V$  are computed as the absolute value of the

average of  $Ch_{IF}(i)$  (6).

$$RDI(v) = \left| \frac{1}{I_{RX}} \sum_{n=0}^{I_{RX}} Ch_{IF}(i) \right|. \tag{6}$$

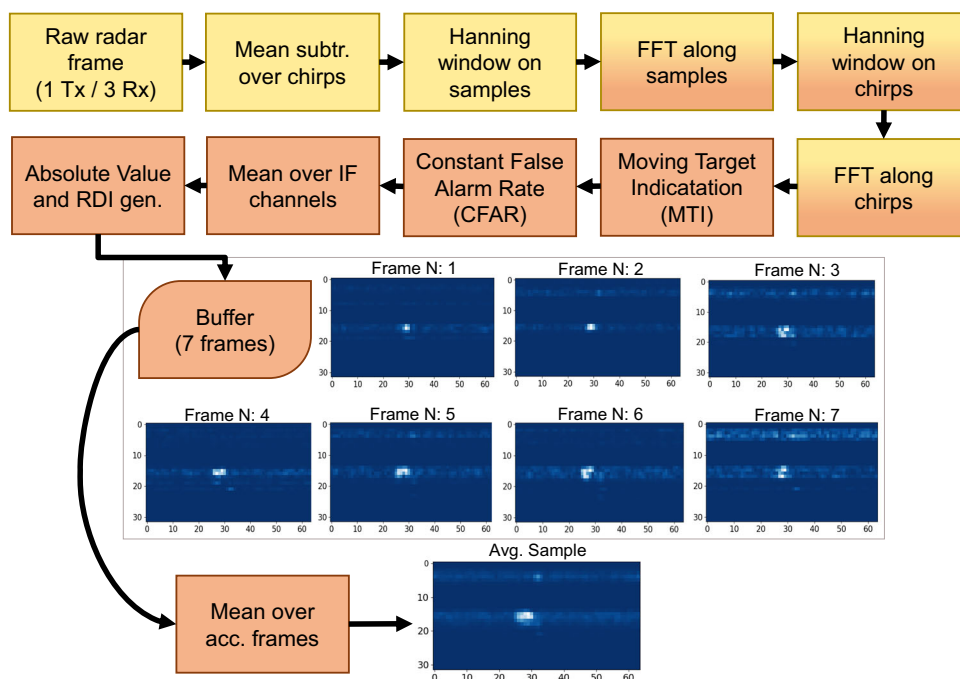
9. The  $RDI_s$  thus generated are stored in a seven frames buffer ( $N_v$ ), which corresponds to roughly half the frame rate. A moving average is performed on the buffer to further reduce the noise in the  $RDI_s$ . These  $RDI_s$  represent the individual instances of the people counting dataset that get labeled (7).

$$RDI = \left| \frac{1}{N_v} \sum_{v=0}^{N_v} RDI(v) \right|. \tag{7}$$

### 3.6 People counting dataset

For people counting, three different meta-datasets have been generated from the collected data of up to three people. Given a frame timing of 75 ms and the frames averaged performed on a seven frames buffer, a total of 7,669 labeled samples have been created. Each sample has a size of 32 times 64 pixels. The width of 64 pixels represents the velocity span, corresponding to the number of chirps per frame. The height of 32 pixels represents the range span, corresponding to half of the bin samples per frame. Independently of the recording room, labels represents the number of people  $P_m$  in the recording, with  $m \in [0, 3]$ . As shown in Fig. 5, the data has been divided into sub-folders of the tuple ( $R, P_m$ , and  $L$ ). The tuple components are the room's name  $R$ : *S*, *M*, or *B*,

**Fig. 6** Flow diagram representing the main preprocessing steps. The yellow blocks represent the main time-domain steps. The orange ones instead represent the frequency domain steps





the number of people ( $P_m$ ), and the location,  $L \in [A, H]$ . With an average duration of 60 seconds across all recordings in rooms  $S$  and  $B$ , a total of 1,677 and 1,702 examples were created, respectively. For  $M$ , a total of 4,290 examples were built with six available locations. With all the available instances, the following three meta-datasets have been generated:

- *Mixed-Dataset*: the data from the sub-folders ( $R, P_m, L$ ) were randomly split so that approximately 75% of the instances was training and 25% was testing. The number of training and test instances in this case are 5,803 and 1,866, respectively.
- *S-Test-Dataset*: in this case, all sub-folders ( $S, P_m, L$ ) were used as tests, while all others ( $[M, B], P_m, L$ ) were used as training. In total, for this meta-dataset, there are 5,922 training examples and 1,677 test examples.
- *B-Test-Dataset*: all the sub-folders ( $B, P_m, L$ ) were used as test, while all the others ( $[S, M], P_m, L$ ) were used as training. The number of training and test instances are 5,967 and 1,702, respectively.

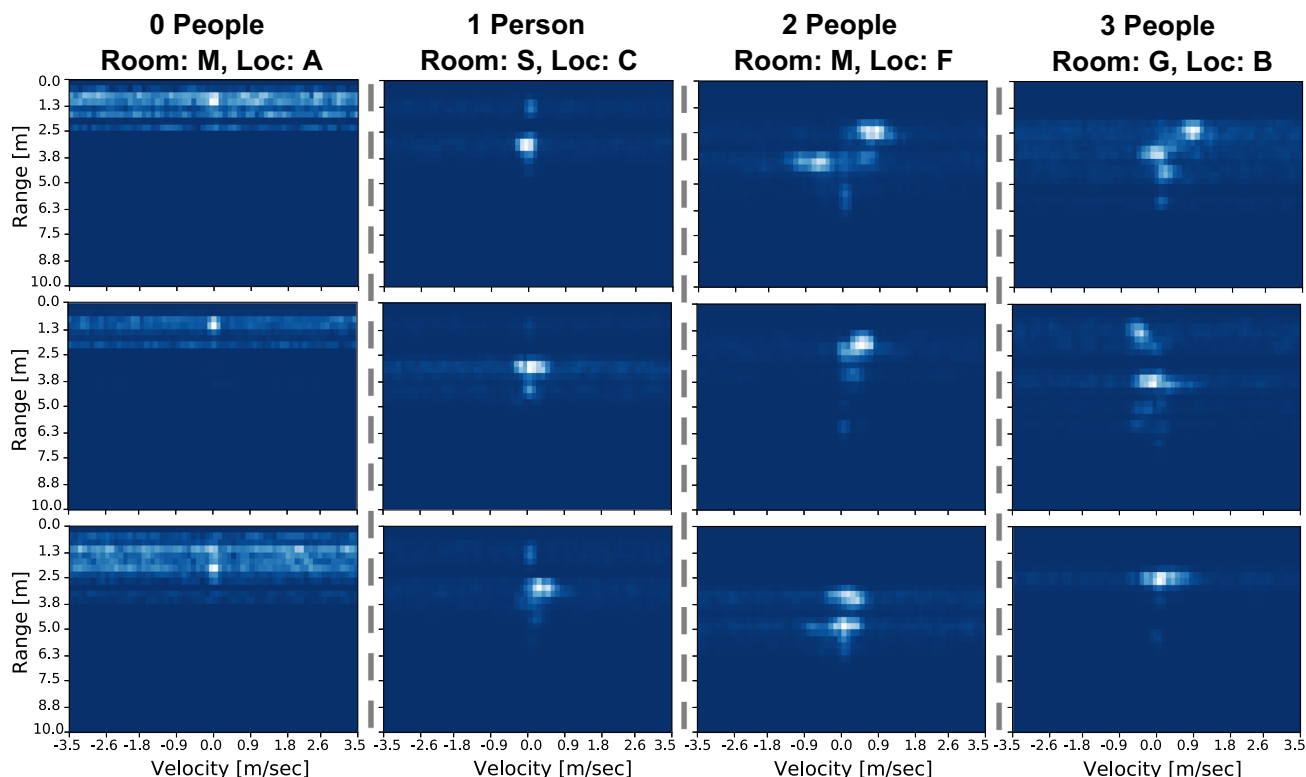
In general, for each of the three generated meta-datasets, the training and test instances are part of the respective training  $\mathcal{D}^{m-train}$  and test  $\mathcal{D}^{m-test}$  meta-dataset splits. Three different averaged RDI examples per class, sampled from the

different recordings in all rooms and locations, are shown in Fig. 7.

Even in the same environment, RDIs from classes 1 to 3 are difficult to distinguish from one another. Figure 8 shows a t-distributed stochastic neighbor embedding (t-SNE) with a 2-D component representation of all instances in the  $S$  room. The t-SNE succeeds in correctly clustering only data with zero people in the environment. A t-SNE representation of all collected data are shown in Fig. 9 according to the *B-Test-Dataset* split. Even with a larger amount of data, only the zero-person instances are easily clustered. In this case, it can also be observed that the test data, which represents the  $B$  room, have different features than the rest of the points. This is an important indication of the dependence of radar data on the location in which they are collected. Algorithms trained in a single location may be difficult to use in other environments and usually require adaptation. Euclidean distance was used as a metric, and Barnes-Hut was used as an optimization algorithm to generate the t-SNE representation.

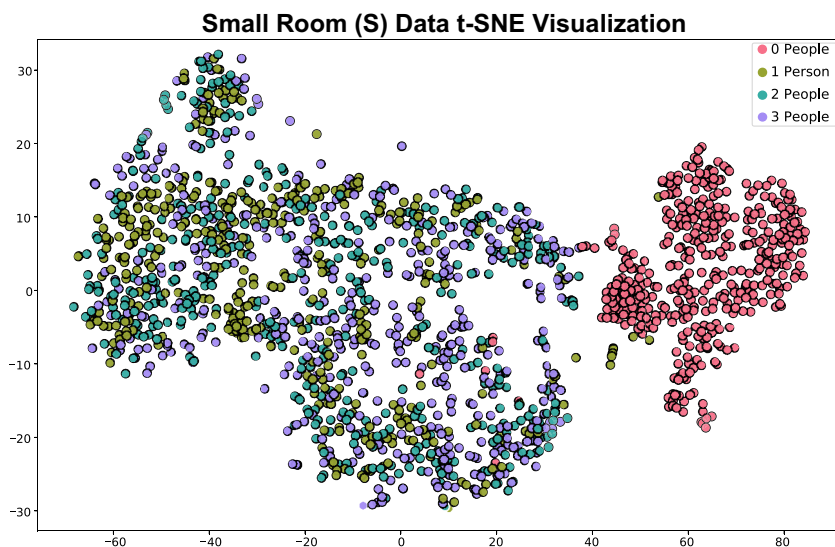
## 4 Proposed approach

In this section, we present our solutions for generalization learning. We begin by proposing a new network topology called the Weighting-Injection Net, which is inspired by the



**Fig. 7** Example RDI instances from the people counting dataset. Every row shows three examples per class, chosen from a random combination of rooms and locations. The axes indicate people relative motion velocity in m/sec and distance from the radar sensor in cm

**Fig. 8** 2-D t-SNE representation of all  $S$  room data. This t-SNE was obtained with a perplexity of 40 over 6,000 optimization iterations



weighting network [36]. We then propose an algorithm that makes use of optimization-based meta learning features from MAML [31], which we call MAMW. This modified version aims at increasing training stability when only a very limited number of shots per class are available. Then, we propose an active learning strategy tailored for weighting networks to allow fine-tuning in a new environment while minimizing the amount of required labeled data.

**4.1 Meta learning**

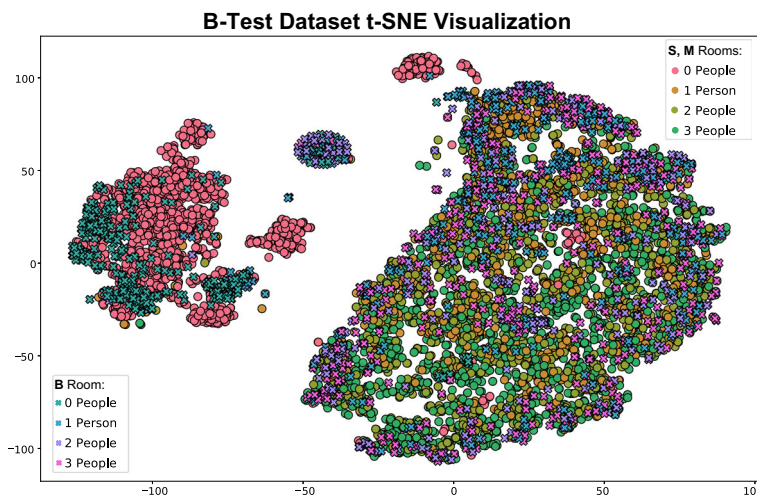
In episodic meta learning,  $K$  tasks are sampled from a distribution  $p(\mathcal{T}_r)$  defined over  $\mathcal{D}^{m-train}$ . As the episodes progress, the goal is to improve the performance of the model on tasks sampled from  $p(\mathcal{T}_s)$  defined on  $\mathcal{D}^{m-test}$ . In DL, task-based learning is often achieved via the gradient method, which involves training the parameters  $\theta'$  by minimizing a cost function  $\mathcal{L}_{\mathcal{T}_r}(f_{\theta'})$ , where  $f_{\theta'}$  represents the relation

between the input  $x$  and the predicted output  $\hat{y}$ . In the relation networks [35], generalization among tasks is directly achieved thanks to the intrinsic comparison of instances enabled by the topology. In optimization-based meta learning, such as in MAML [31], the information learned for tasks  $\mathcal{T}_r$  and encoded in the parameters  $\theta'$ , is transferred to a base model  $f_{\theta}$  with parameters  $\theta$ , minimizing an outer cost function  $\mathcal{L}_{\mathcal{T}_r}(f_{\theta})$ . In this case, the task-specific cost function depends on the parameters  $\theta$  of the base model  $\mathcal{L}_{\mathcal{T}_r}(f_{\theta})$ .

**4.1.1 Weighting-injection net**

The Weighting-Injection Net aims to compare the features of the arbitrary examples of query  $q$  with those of reference to the support  $s$  classes for each task  $k \in K$ . The Weighting-Injection Net, as shown in Fig. 1 is based on three main modules: injection, comparison, and weighting. During training, the gradient information is propagated through

**Fig. 9** 2-D t-SNE representation of the  $B$ -Test-Dataset, for all the recorded data. The  $B$  room data are represented by the “x” marker, while the rest of the data (rooms  $S$  and  $M$ ) are represented by the “o” marker. This representation was obtained with a perplexity of 30 over 7,000 optimization iterations



all modules in both forward and backpropagation steps. For a  $N$ -way 1-shot task, the idea is to map the relationship between support examples  $s_n$ , where  $n \in \mathbb{N}: [1, 2, \dots, N]$ , to each query example  $q_j$ , where  $j$  is the index of the  $j$ -th example of the set.

The injection module  $e_\theta$  generates a higher dimension representation of the input  $x$  to enhance the extraction and matching of features in the subsequent comparison step. Gradient information for the injection module is only propagated as  $e_{\theta'}(s_n)$  through the support instances. For the query, only the feature representation  $e_{\theta'}(q_j)$  is generated.

The comparison module  $c_\theta$ , takes as input the concatenation along  $N$  channels of  $e_{\theta'}(q_j)$ , with each of the  $n$  support samples. The number of channels  $N$  corresponds to the task number of ways. The features are extracted in the module using convolution layer sequences, yielding a comparison vector  $z$ . The vector  $z$  is generated in the following way:

$$z_{n,j} = g_{\theta'}(e_{\theta'}(s_n) \parallel e_{\theta'}(q_j)), \quad (8)$$

where  $\parallel$  denotes the operation of concatenation along the  $N$  channels.

Lastly, the weighting module  $w_\theta$  is designed to generate a probability density from the concatenated  $N$  channels in the  $z$  vector. Each  $z_{n,j}$  is the output of the comparison module, between the query  $q_j$  and a support  $s_n$ . The predicted output  $\hat{y}_j$  for the sample  $q_j$  can be expressed as follows:

$$\hat{y}_j = w_{\theta'}(\parallel_{n=1}^N z_{n,j}) = w_{\theta'}(z_{1,j} \parallel z_{2,j} \cdots \parallel z_{N,j}), \quad (9)$$

where  $\parallel$  represents the sequence of concatenations performed over the channels  $N$  of  $z$ .

In the case of a  $N$ -way  $C$ -shot task, where  $c \in \mathbb{N}: [1, 2, \dots, C]$ , the supports per class can be denoted as  $s_{n,c}$ . The Weighting-Injection Net can be leveraged in this case to create a more robust representation of the comparison vector  $z_{n,j}$ . This can be done by arithmetic averaging over  $C$  sets of  $N$ -channel concatenations, given by the embedded representations of  $q_j$  with each of the support sets  $s_{n,c}$ . Such a more robust representation yields the query class estimation with less bias than with the single support shot scenario. The mathematical expression for a single  $q_j$  is as follows:

$$z_{n,j} = \frac{1}{C} \sum_{c=1}^C g_{\theta'}(e_{\theta'}(s_{n,c}) \parallel e_{\theta'}(q_j)). \quad (10)$$

The Weighting-Injection Net, trained on  $p(\mathcal{T}_r)$ , can be tested, thanks to its inherent structure, on tasks from  $p(\mathcal{T}_s)$  without further training. Given a support set with elements  $s_{n,c}$  for a task  $\mathcal{T} \sim p(\mathcal{T}_s)$  a  $N$ -way  $C$ -shot, the class

probability density of the  $j$ -th query sample  $q_j$ , is directly estimated by inference.

#### 4.1.2 Model-agnostic meta-weighting

The weighting network [36] represents a robust episodic learning algorithm thanks to the inherent feature of instance comparison. Yet, this method can be characterized by learning instability when only a few-shot per class are available. Especially in 1-shot learning, this is due to the comparison of the query with the individual support instances, which may not be sufficiently descriptive of a class for a given task. Hence, we present a method called model-agnostic meta-weighting (MAMW), which tries to incorporate within the weighting network some features of optimization-based meta learning to enhance the stability and robustness of prediction in this setting. Specifically, in the MAMW, we propose to divide episodic learning into inner and outer steps. Given a  $N$ -way  $C$ -shot task:

1. In the *inner step*, the support instances are compared with a noisy version of themselves of Gaussian type via a function  $e_\theta(\phi((s_{n,c})))$ . This noise is generated at random from the  $\mathcal{N}(0, \sigma^2)$  distribution in the interval  $[-\sigma, \sigma]$ . Defined  $s_h$  as the  $h$ -th support example, where  $H = N \cdot C \implies h \in \mathbb{N}: [1, 2, \dots, H]$ , the computation of  $z_{n,h}$  can be expressed as follows:

$$z_{n,h} = \frac{1}{C} \sum_{c=1}^C g_\theta(e_\theta(s_{n,c}) \parallel e_\theta(\phi(s_h))), \quad (11)$$

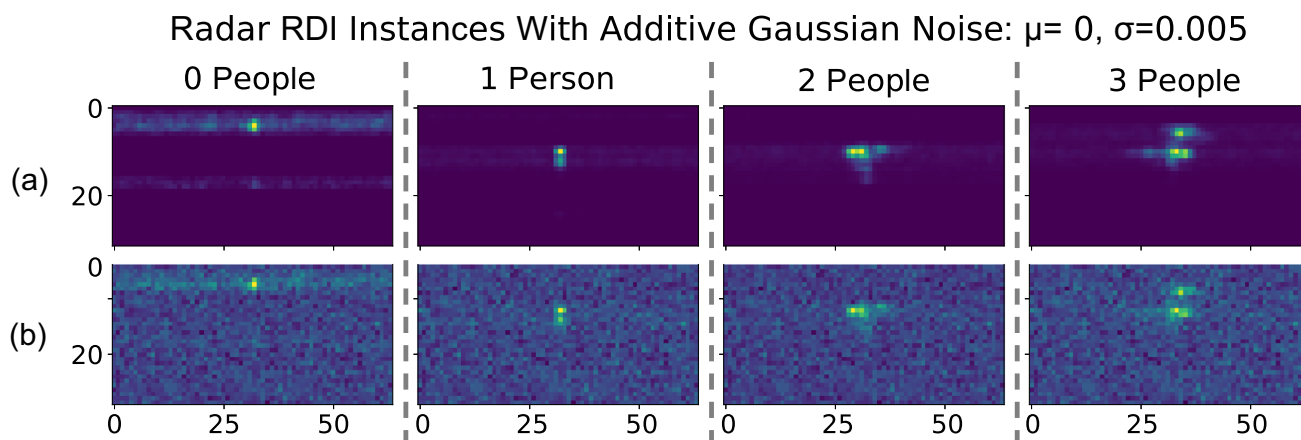
$$\hat{y}_h = w_\theta(\parallel_{n=1}^N z_{n,h}), \quad (12)$$

where  $\theta$  represent the parameters of the base model  $f_\theta$ . Such operations can also be carried out in batches. An example of people counting instances compared with their noisy version is shown in Fig. 10.

2. In the *outer step*, the comparison between the support examples  $s_{n,c}$  and each query  $q_j$  is performed, starting from the weights  $\theta'$  learned in the inner loop. In this case, the comparison vectors  $z$  are computed with the (10) and the predicted output  $\hat{y}_j$  with (9).

The main steps of the MAMW, in the case of few-shot, supervised learning with outer updates after every task, are defined in Algorithm 1.

The presented Weighting-Injection Net topology can be trained via the MAMW algorithm. Also with the MAMW episodic learning, the Weighting-Injection Net can tackle new test tasks without the necessary adaptation training.



**Fig. 10** Examples of RDI without (a) and with added Gaussian noise (b) used in the inner step training of the MAMW

MAMW does not need algorithmic modifications when an embedding module is used instead of the injection module.

---

**Algorithm 1** MAMW for  $N$ -way  $C$ -shot Supervised Learning

---

**Require:**  $N$ -way:  $n \in \mathbb{N}: [1, 2, \dots, N]$

**Require:**  $C$ -shot:  $c \in \mathbb{N}: [1, 2, \dots, C]$

**Ensure:**  $p(\mathcal{T})$ : distribution over tasks

**Ensure:**  $\alpha, \beta$ : step size hyperparameters

1: Randomly initialize  $\theta$

2: Random sample  $K$  tasks  $\mathcal{T}$  from  $p(\mathcal{T})$

3: **for**  $\mathcal{T}_k \in \mathcal{T}$  **do**

4:   Sample  $H = N \cdot C$  support instances  $s_h$  from  $\mathcal{T}_k$

5:   **for all**  $s_h$  **do**

6:     Compute  $z_{n,h}$  in (11)

7:     Compute  $\hat{y}_h$  in (12)

8:     Evaluate  $\nabla_{\theta} \mathcal{L}_{\mathcal{T}_k}(\hat{y}_h)$  by  $\mathcal{L}_{\mathcal{T}_k}$  for  $s_h$

9:     Compute adapted parameters with gradient descent:  $\theta' = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_k}(\hat{y}_h)$

10:   **end for**

11:   Sample  $J$  query instances  $q_j$  from  $\mathcal{T}_k$

12:   Compute  $z_{n,j}$  in (10)

13:   Compute  $\hat{y}_j$  in (9)

14:   Update  $\theta \leftarrow \theta - \beta \nabla_{\theta} \mathcal{L}_{\mathcal{T}_k}(\hat{y}_j)$  for  $q_j$

15: **end for**

---

## 4.2 Active learning

Active learning can also be used on top of a meta learning model to perform fine-tuning on a given task, leveraging the most uncertain queries during adaptation. We propose to use pool-based sampling active learning to fine-tune the Weighting-Injection Net on  $p(\mathcal{T}_s)$ , starting from what has been learned on  $p(\mathcal{T}_r)$ . We chose an uncertainty sampling strategy to let the algorithm decide at each training epoch which new examples to label. We test the approach with three different priority scores: least confidence (LC), margin sampling (MS), and entropy (E), respectively. For the instances

$q_j = \{x_j, y_j\}$  representing the input/output pairs on queries sampled by  $\mathcal{T}$ , the priority scores  $S_p$  can be defined as follows:

$$S_{LC} = \operatorname{argmax}_{x_j} (1 - P_{\theta}(\hat{y}_{max} | x_j)), \quad (13)$$

$$S_{MS} = \operatorname{argmin}_{x_j} (P_{\theta}(\hat{y}_{max} | x_j) - P_{\theta}(\hat{y}_{max-1} | x_j)), \quad (14)$$

$$S_E = \operatorname{argmax}_{x_j} \left( - \sum_{n=1}^N P_{\theta}(\hat{y}_n | x_j) \log P_{\theta}(\hat{y}_n | x_j) \right), \quad (15)$$

where  $P_{\theta}$  of  $\hat{y}_{max}$  is the highest posterior probability predicted by the model with  $\theta$  parameters for  $x_j$ , and  $N$  is the number of classes.

Algorithm 2 defines the main step of the proposed pool-based sampling on a task  $\mathcal{T}$ . In general, the Algorithm 2 represents a generalization of the pool-based sampling approach for relational models. For a given task, a set of class-related support examples is initially labeled. As the number of iterations increases, the uncertainty of the query examples is evaluated, and those with the highest priority score are added to the labeled dataset. A maximum number of support instances per class per iteration is also chosen. Instead of starting with random weights, parameters learned during episodic learning on training tasks can be used as the model initialization. The active learning procedure is therefore performed on unseen test tasks.

## 5 Experimental setup

In this section, we present all the results achieved on meta learning episodic experiments and active learning fine-tuning on the people counting meta-datasets (Section 3.6). The algorithms have been written in the Python programming

---

**Algorithm 2** Pool-based Sampling Active Learning for  $N$ -way  $C$ -shot Supervised Learning on Weighting-Injection Net.
 

---

**Require:**  $N$ -way:  $n \in \mathbb{N}: [1, 2, \dots, N]$ 
**Require:**  $C$ -shot:  $c \in \mathbb{N}: [1, 2, \dots, C]$ 
**Ensure:** Task  $\mathcal{T} \sim p(\mathcal{T})$ 
**Ensure:**  $J$ : Queries to sample per epoch

**Ensure:**  $A$ : Queries to label per epoch

 1: Initialize  $\theta$  with meta-learned weights

 2: Initialize  $\mathcal{D}_p = \{\}$  as labeled Pool.

 3: Sample in  $\mathcal{T}$  support instances:

$$s_{n,c} = \{x_{n,c}, y_{n,c}\}$$

 4: Add all  $s_{n,c}$  in  $\mathcal{D}_p$ 

 5: Sample in  $\mathcal{T}$ ,  $J$  query instances:

$$q_j = \{x_j, y_j\}$$

 6: **while** not done **do**

 7:   Compute  $z_{n,j}$  in (10)

 8:   Compute  $\hat{y}_j$  in (9)

 9:   Compute  $S_p$  of  $q_j$  with (13), (14) or (15)

 10:   With  $S_p$  of  $q_j$ , select  $A$  queries  $q_{j_a}$  and  $\hat{y}_{j_a}$ 

 11:   Add all  $q_{j_a}$  in  $\mathcal{D}_p$ 

 12:   Update  $\theta \leftarrow \theta - \beta \nabla_{\theta} \mathcal{L}_{\mathcal{T}_k}(\hat{y}_{j_a})$ 

 13:   Sample in  $\mathcal{D}_p$  support instances:

$$s_{n,c} = \{x_{n,c}, y_{n,c}\}$$

 14:   Sample in  $\mathcal{D}_p$ ,  $J$  query instances:

$$q_j = \{x_j, y_j\}$$

 15: **end while**


---

language, using the TensorFlow<sup>TM</sup> module to implement the DL models. Further experiments on a public dataset have been performed and discussed in the Appendix A. The codes related to the algorithms and topologies used for the meta learning experiments are available online<sup>1</sup>. As a process unit, we used an Nvidia<sup>®</sup> Tesla<sup>®</sup> P4 GPU and CUDA<sup>®</sup> Toolkit v11.1.0 for parallel computing.

## 5.1 Meta learning experiments

All the episodic experiments have been performed with the topology presented in Section 4.1.1 and Fig. 1. Specifically, 4-way experiments with 1-, 2-, 5-, and 10-shot have been performed. The topology has been trained with two different algorithms. First with the classical episodic few-shot training of weighting networks, as defined in [36], using the Weighting-Injection Net equations (Section 4.1.1). Further, the topology has been trained in episodic sequences of inner and outer steps, following the steps of the MAMW algorithm proposed in Section 4.1.2. All the results presented in this section refer to the two algorithms and are consistently called Weighting-Injection Net and MAMW. Comparison results of the two algorithms with the state-of-the-art are presented in the Section 5.1.1. The cutting-edge comparison also fea-

tures some application limit experiments for indoor people counting up to five individuals in a room.

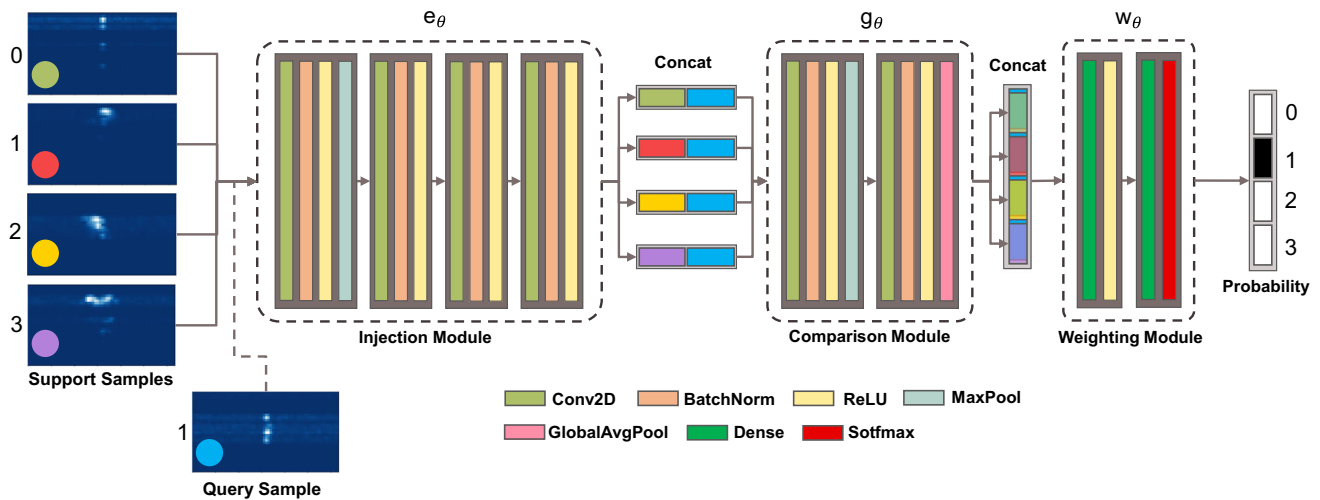
A graphical representation of the model modules and respective layers is shown in Fig. 11. The model consists of 283,379 trainable parameters in its entire module sequence. Of the total, the injection module consists of 239,680 parameters, the comparison module of 39,936, and the weighting module of the remaining 4,180. To rescale feature size, max pooling is used in cascade to the 2D convolution (Conv2D) for the two modules  $e_{\theta}$  and  $g_{\theta}$ . In addition, batch normalization is used to increase the stability of training. All batch normalization layers are followed by a rectified linear unit (ReLU) activation function. To map the output vector into a probability distribution over the classes, the softmax is used as an activation function for  $w_{\theta}$ . The cost function chosen for the query classification is categorical crossentropy, and the optimization algorithm is Adam.  $\beta_1$  and  $\beta_2$  for Adam have been set to 0 and 0.5, respectively. A learning rate of  $5e - 4$  has been chosen for the Weighting-Injection Net. A learning rate of  $5e - 4$  has also been chosen for both the inner and outer steps of MAMW. For the Gaussian noise statistic on the MAMW inner step, a value of  $\sigma^2$  equal to 0.005 has been chosen. This value represents an empirical choice, noting that larger values led to the loss of the main information in the support instances, while smaller values were less effective for the performance of the experiments.

Regardless of the number of shots, every meta-training experiment is performed over 22,000 episodes, each of a single training epoch. The episodic learning is carried out on  $\mathcal{D}^{m-train}$ . The validation and testing have been performed at the end of each episode on 10-shot per class (40 samples) on tasks sampled by  $\mathcal{D}^{m-train}$  and  $\mathcal{D}^{m-test}$  respectively.

All experiments have been carried out with an embedding size  $g$  of 64. Smaller embedding sizes resulted in non-convergent experiments, whereas larger sizes resulted in meta-overfitting on  $\mathcal{D}^{m-train}$ . For the injection module, an output representation of  $14 \cdot 14 \cdot g$  has been chosen (feature size). This led to a representation per image of 12,544 units (Table 2). On the Nvidia<sup>®</sup> Tesla<sup>®</sup> P4 GPU, the number of floating points operations per second (FLOPS) for the injection module with this configuration is 108 megaFLOPS. The size in bytes of the weights of the model when saved in ".h5" format, regardless of the chosen episodic training algorithm and the number of shots, is 1,148 KB. Some experiments at varying feature sizes are also presented later in this section to test the benefits of the injection module over the standard embedding module.

The obtained values of prediction accuracy, model size, and single-sample prediction latency are compared to state-of-the-art values obtained by training other algorithms on the people counting dataset employed in this work. The accuracy results for the Weighting-Injection Net are reported for varying numbers of shots. Each experiment by algorithm,

<sup>1</sup> The codes for the meta learning algorithms are available at: <https://github.com/GiancoMauro/TF-Meta-Learning>



**Fig. 11** Representation of the topology modules and respective layers used in the relational experiments. The injection module ( $e_\theta$ ) increases the data dimensionality via a sequence of convolutional layers. The query sample is compared with all the available support samples.

meta-dataset, and number of shots has been performed three times and tested on 10,000 final tasks sampled by  $\mathcal{D}^{m-test}$ . All presented results include the 95% confidence interval in addition to the average accuracy value.

The performance evaluation of each individual experiment is measured according to the validation and test accuracy values obtained by the model as the number of episodes increases. For every experiment, a box plot on the validation and testing accuracy statistics of tasks sampled by  $\mathcal{D}^{m-train}$

**Table 2** Network Layers Configuration - People Counting

Module	Type	Filter Shape <sup>1</sup>	Output Shape
Injection	Conv2D	$3 \times 7 \times 1 \times 64$	$j \times 30 \times 58 \times 64$
	MaxPool	$2 \times 2$	$j \times 15 \times 29 \times 64$
	Conv2D	$3 \times 7 \times 64 \times 64$	$j \times 13 \times 23 \times 64$
	Conv2D <sup>2</sup>	$3 \times 7 \times 64 \times 64$	$j \times 13 \times 19 \times 64$
	Conv2D <sup>2</sup>	$3 \times 7 \times 64 \times 64$	$j \times 14 \times 14 \times g$
Comparison	Conv2D <sup>2</sup>	$3 \times 1 \times 2g \times g$	$jc \times 44 \times 16 \times g$
	MaxPool	$3 \times 3$	$jc \times 14 \times 5 \times g$
	Conv2D	$3 \times 3 \times g \times g$	$jc \times 12 \times 3 \times g$
	AvgPool	$1 \times 1$	$jc \times g$
Weighting	Dense	$ng \times 16$	$j \times 16$
	Dense	$1 \times n$	$j \times n$

The indices  $n$  and  $c$  represent the class and shot number, respectively. The index of the  $j$ -th query shot is represented by  $j$ . The  $g$  represents the embedding size, which was set to 64 in the experiments

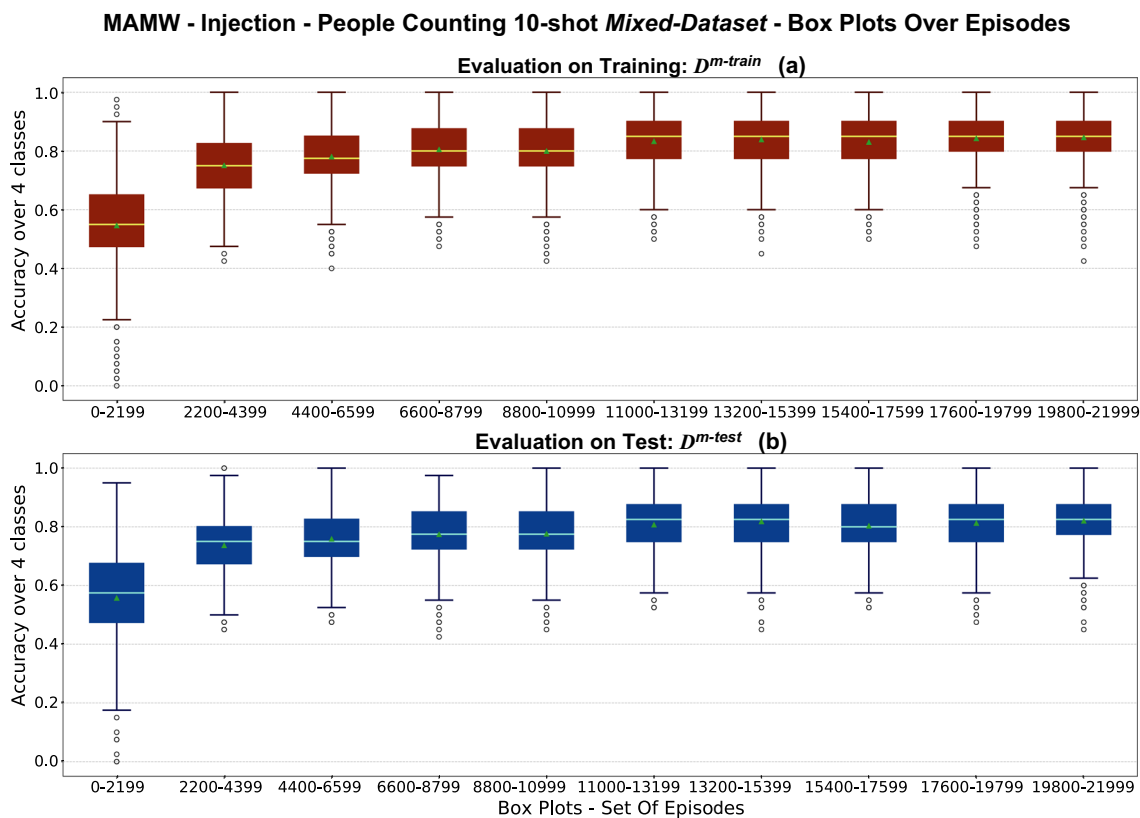
<sup>1</sup>For the Conv2D layers, the filter shape dimensions are, respectively, kernel height and width and input and output channels

<sup>2</sup>In these layers, a symmetric zero-padding of 1 is applied to both the width and height of the samples

To combine relevant features, the comparison module ( $g_\theta$ ) employs convolution and global average pooling. The weighting module ( $w_\theta$ ) generates a feature matching probability density using dense layers and softmax activation

and  $\mathcal{D}^{m-test}$  is constructed every 2,200 episodes. In the following plots and paragraphs, statistical insights from one of the experiments performed are analyzed. Specifically, a MAMW 10-shot experiment on *Mixed-Dataset* is chosen thanks to the good achieved generalization performance. Figure 12 shows the set of box plots generated as the training episodes advance for the considered experiment. As the episodes progress, the mean and median values of the distributions rise while the quartiles and whiskers narrow. With episodes progressing, even the outliers move closer to the upper limit of accuracy. The described behavior demonstrates how, thanks to previously acquired experience, the model can generalize better on new sampled tasks. This means that newly learned parameters  $\theta$  generalize better in new contexts, i.e., new locations and test rooms, resulting in higher performance under the same learning conditions.

Discrete accuracy density histograms can be used to represent the distribution underlying individual box plots. Graphical evidence of how the distribution tends to shift towards higher generalization accuracy can be observed by comparing the first and last histograms of the episodic optimization. Such density histograms can also be compared to a Gaussian probability distribution, thus showing what percentage of the achieved accuracy lies between the first and third quartiles. Figure 13 depicts a comparison of accuracy statistics for the examined experiment at the beginning and end of the episodic training. Even for tasks sampled only by  $\mathcal{D}^{m-test}$ , the probability density tends, as the episodes progress, to take on a negative skew towards the upper limit of accuracy. The actual distributions underlying the box plots are not Gaussian but multi-modal with density peaks due to the variable complexity of the sampled tasks.



**Fig. 12** Accuracy statistics box plots vs. episodes for a MAMW 10-shot *Mixed-Dataset* experiment. The red box plots are generated on the validation tasks (a), whereas the blue ones (b) are generated on test

tasks. The median and mean values are represented by a horizontal line and a green triangle in each box plot. The small circles represent the box plot outliers

The generalization capability can be addressed at the level of individual classes by constructing cumulative confusion matrices on task sequences. Labels 0 to 3 represent the real and predicted number of people for the two dataset splits. Figure 14 depicts the confusion matrices underlying the first and last box plots of Fig. 12 for both  $\mathcal{D}^{m-train}$  and  $\mathcal{D}^{m-test}$ .

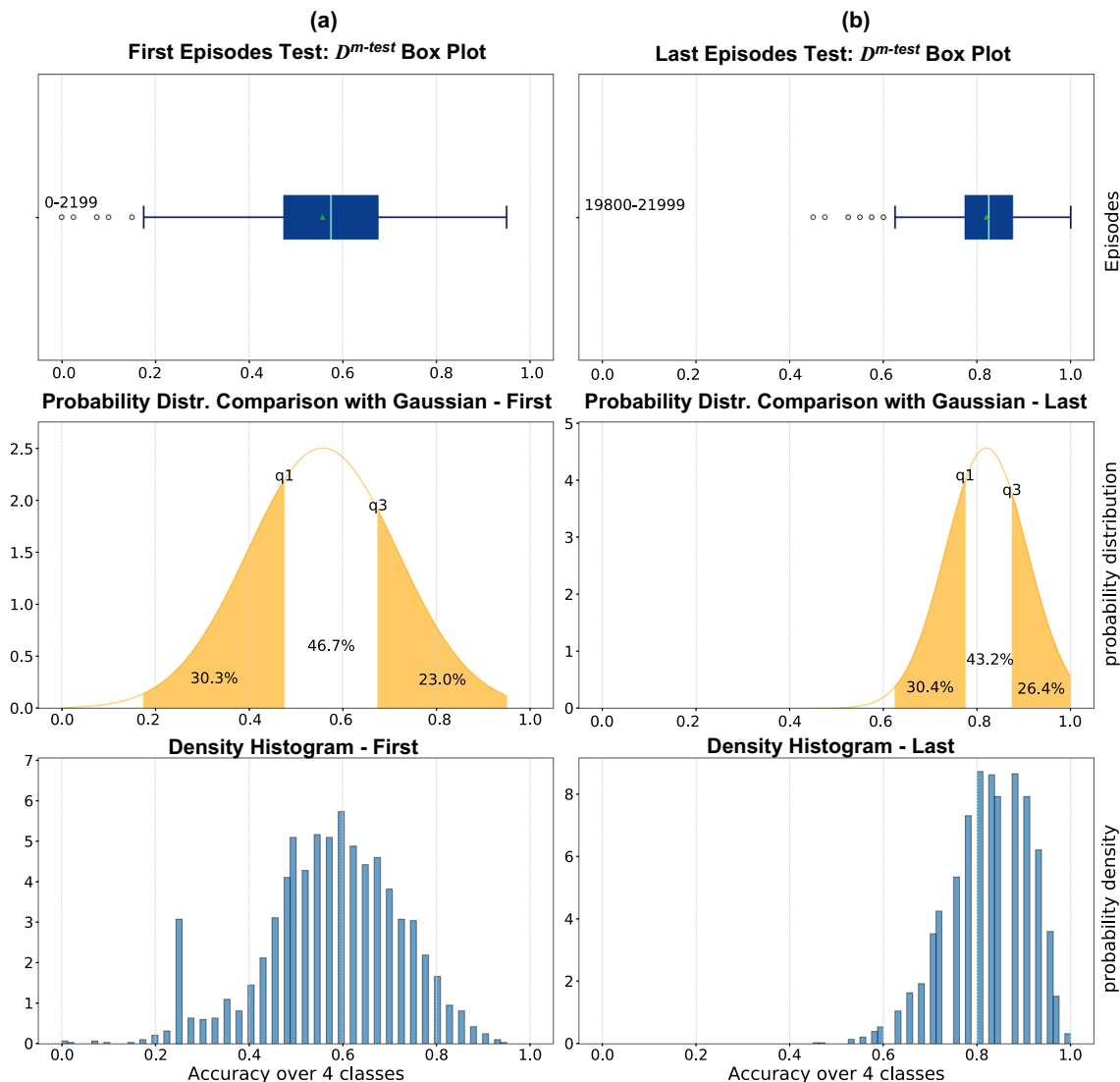
Figure 15 shows another example of cumulative confusion matrices for a Weighting-Injection Net 5-shot experiment on *S-Test-Dataset*. It is noticeable in both Figs. 14 and 15, that the model learns to generalize better as episodes progress for both unseen locations and rooms. Most miss-classifications, especially at the end of episodic learning, lie around the main diagonal. This means that the models, in most cases, count  $\pm 1$  person compared to the actual number of individuals in the environment. Moreover, the majority of the misclassifications happen for the classes of 1 to 3 persons, while the model easily succeeds in distinguishing the case 0 that corresponds to no people detected in the sensor's FoV. The per-class accuracy of the test confusion matrices in Fig. 15 turns out to be lower than that in Fig. 14. This is due not only to the use of 10-shot instead of 5-shot in the experiment but also to the higher complexity of the test tasks. In fact, the Fig. 15

experiment sampled all test tasks from a room not included in the training ( $S$ ).

The prediction accuracy values obtained as an average of the post-training tests for each experiment type are listed in Tables 3, 4, 5 for the three defined meta-datasets.

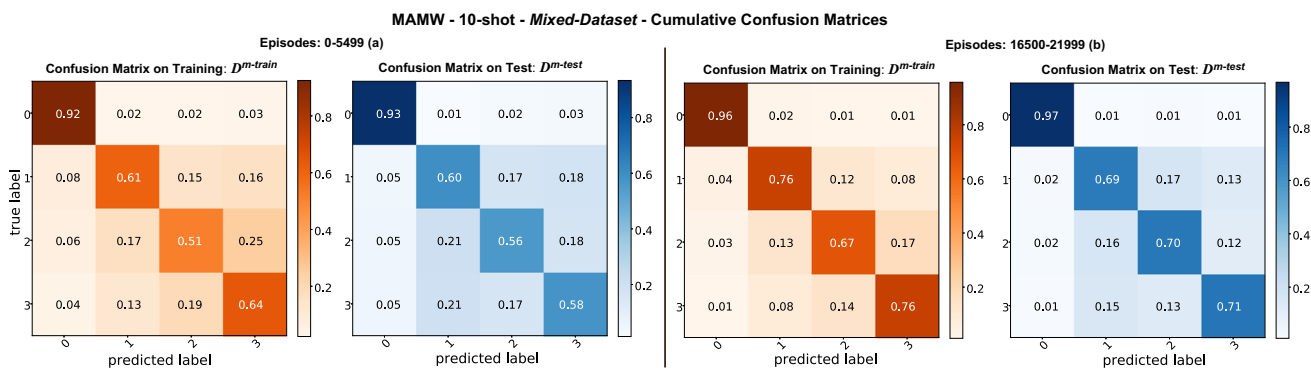
As can be observed from Tables 3, 4 and 5, regardless of the used meta-dataset, the 1- or 2-shot experiments performed with the MAMW lead to higher average accuracy values than the Weighting-Injection Net. In these specific cases, in episodic learning, the few supports per class make the prediction given by the Weighting-Injection Net less robust, where the learning depends solely and exclusively on the comparison with the query. MAMW instead supplies more information to the model thanks to the initial comparison with a noisy version of the support samples, thus emphasizing the potential intrinsic noise of the query data. For the 5- and 10-shot experiments, the two episodic approaches lead to different performances with respect to the used meta-dataset. The MAMW outperforms the Weighting-Injection Net on the *Mixed-Dataset*, regardless of the number of shots. The *Mixed-Dataset* contains, in fact, recordings from all rooms, but with different locations and numbers

**MAMW - Injection - 10-shot - Mixed-Dataset - First Vs Last Test Box Plot: Accuracy Distribution**



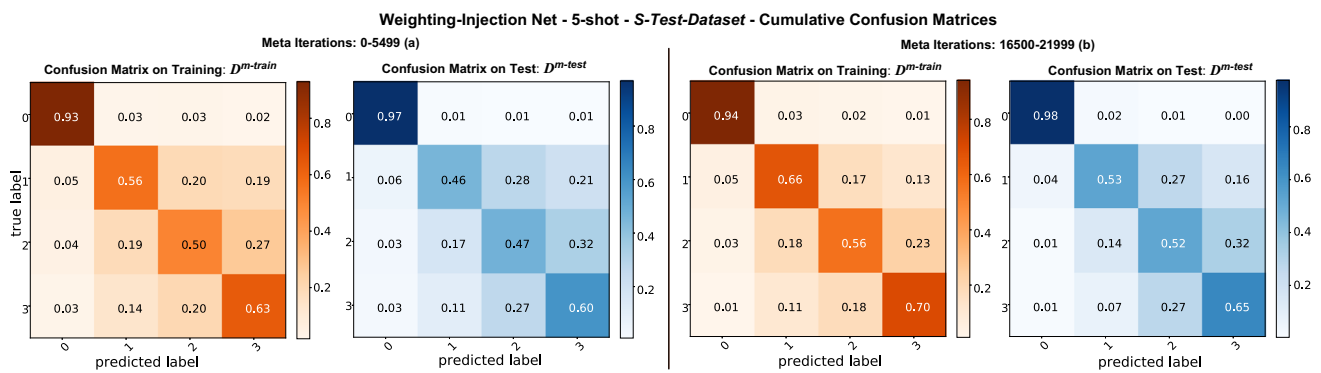
**Fig. 13** MAMW 10-shot experiment, first (a) and last (b) box plot underlying distributions, generated on test tasks sampled from *Mixed-Dataset*. The q1 and q2 values on the Gaussians indicate the first and third quartiles, respectively. The probability density histograms show

the actual non-Gaussian nature of the distribution. The accuracy probability density for the last box plot (b) exhibits a negative skew as a result of the generalization learning



**Fig. 14** Cumulative confusion matrices for a 10-shot MAMW *Mixed-Dataset* experiment. Confusion matrices are obtained on the first (a) and last (b) 5,550 meta-iterations in the validation phase for both  $D^{m-train}$  and  $D^{m-test}$  sampled tasks





**Fig. 15** Cumulative confusion matrices for a 5-shot Weighting-Injection Net  $S$ -Test-Dataset experiment. Confusion matrices are obtained on the first (a) and last (b) 5,550 meta-iterations in the validation phase for both  $D^{m-train}$  and  $D^{m-test}$  sampled tasks. In this case, the entire  $S$  room is utilized as the test set

of people. In this case, the MAMW goal of capturing noise similarity between support and query also aids query class recognition. This is thanks to the intrinsic features of the RDIs collected in the same room, which are thus influenced by the properties of that environment. On  $S$ -Test-Dataset and  $B$ -Test-Dataset instead, the Weighting-Injection Net outperforms MAMW in most 5- and 10-shot experiments. In these cases, given the relevant difference in context for the test room, the MAMW comparison with the noisy version of supports might shift the learning objective towards detecting noise rather than the class of query samples.

For relation-based topologies, there is no need to perform adaptation training for new tasks as a result of the direct comparison of features between the newly available support samples and the query. Therefore, the adaptation time to a new task is null. Instead, the inference time on a single sample (query) can be computed as a function of the number of shots. It corresponds to the time required by the model to predict the query class given the available supports. The time required to compute the  $z$  comparison vectors for all available supports is thus included in the inference time for single queries. As both the proposed algorithms share the same inference procedure, these values are independent of the employed approach. The single sample inference time is also independent of the selected counting meta-dataset,

**Table 3** Accuracy of the two meta learning approaches on people counting (4 classes):  $Mixed$ -Dataset

Accuracy [%] on Mixed-Dataset	Weighting-Injection Net	MAMW
1-shot	63.01 $\pm$ 0.21	66.95 $\pm$ 0.22
2-shot	71.79 $\pm$ 0.20	74.10 $\pm$ 0.20
5-shot	78.26 $\pm$ 0.18	78.63 $\pm$ 0.19
10-shot	81.40 $\pm$ 0.16	82.16 $\pm$ 0.16

given the same input size. Average inference values on a single query are listed in Table 6.

As can be seen from Table 6, the inference time for a single query increases as the number of shots increases. Multiple supports available per class enable a more robust prediction of the query class, as shown in (10). However, this requires the generation of multiple  $z$  comparison vectors, which, in proportion to the number of shots, lead to a progressive increase in inference time on a single query.

Classification accuracy is also dependent on the chosen feature representation dimension in the feature extraction module  $e_\theta$ . In specific experimental settings, the injection can counter episodic overfitting effects by increasing feature size as opposed to the standard embedding. The  $14 \cdot 14$  feature size chosen for all the other experiments is compared with two representations of  $4 \cdot 4$  and  $9 \cdot 9$  respectively. Given the size of an RDI example of  $32 \cdot 64 = 2,048$ , a feature representation of  $4 \cdot 4 \cdot 64 = 1,024$  converts the injection module into an embedding module. Compared with the 108 MegaFLOPS required by the feature size of  $14 \cdot 14$ , the size  $4 \cdot 4$  requires only 0.28 MegaFLOPS. Overall, the injection operation, compared to embedding, results in the GPU performing significantly more FLOPS. This is due to the larger size of the extracted features in the convolutional layers.

**Table 4** Accuracy of the two meta learning approaches on people counting (4 classes):  $S$ -Test-Dataset

Accuracy [%] on S-Test-Dataset	Weighting-Injection Net	MAMW
1-shot	59.85 $\pm$ 0.19	61.98 $\pm$ 0.19
2-shot	61.14 $\pm$ 0.16	64.48 $\pm$ 0.17
5-shot	71.77 $\pm$ 0.17	73.40 $\pm$ 0.18
0-shot	76.61 $\pm$ 0.16	73.53 $\pm$ 0.16

**Table 5** Accuracy of the two meta learning approaches on people counting (4 classes): *B-Test-Dataset*

Accuracy [%] on B-Test-Dataset	Weighting-Injection Net	MAMW
1-shot	54.26 ± 0.23	57.35 ± 0.23
2-shot	60.00 ± 0.22	60.83 ± 0.22
5-shot	69.98 ± 0.18	68.57 ± 0.18
10-shot	71.06 ± 0.18	70.74 ± 0.18

Table 7 features the results on the *S-Test-Dataset*, obtained with the Weighting-Injection Net as feature size, and the number of shots vary. The 1-shot experiment seems to benefit more from embedding than from an injection module. The squeezed representation of features in such experiments leads to a more compact representation. The entire weighting network can succeed in extracting key features from the few samples available per class in each episode bringing benefits of generalized learning. On the other hand, as the number of shots increases, a larger representation of features seems to lead to greater benefits in training. With 5- or 10-shot per class, a larger feature space upstream of the comparison module facilitates feature extraction from the available support samples and yields better generalization results. The effect of overfitting on individual tasks is clearly visible by comparing the accuracy obtained with the 4 · 4 feature size between the 5- and 10-shot experiments. Contrary to the common scenario, the performance of the model worsens as the number of shots doubles. Without tuning the other hyperparameters, the small feature size favors single-task adaptation rather than generalized learning, reducing so, the overall performance.

### 5.1.1 Comparison with the state-of-the-art and limitations

In this section, the results of Weighting-Injection Net and MAMW are compared to the results of other state-of-the-art meta learning methods for the task of people counting. Reptile [32] is used as a baseline algorithm. MAML 2<sup>nd</sup> [31] and a more stabilized and performant version of MAML presented in Antoniou et al. [52], are the other algorithms

**Table 6** Average single-sample inference time computed as the average of all MAMW and Weighting-Injection Net experiments on all defined meta-datasets, in function of the number of shots. Every experiment has been run over 10,000 final tasks on Nvidia<sup>®</sup> Tesla<sup>®</sup> P4 GPU

Number of Shots	Avg. Inference Time [ms]
1-shot	14.46
2-shot	16.21
5-shot	27.03
10-shot	43.73

used for comparison. The latter, labeled MAML<sup>+</sup>, leverages the contributions of multi-step loss optimization (MSL), derivative-order annealing (DA), and cosine annealing of meta-optimizer learning rate (CA). The model chosen for the state-of-the-art algorithms is a CNN suitable for the generalization goal, consisting of four main blocks. The first three blocks consist of a Conv2D with 64, 128, and 256 filters, followed by batch normalization and the ReLU activation function. The last block consists of a dense layer with 4 neurons, corresponding to the number of classes. This topology consists of 403,332 trainable parameters compared to the 283,379 of MAMW and the Weighting-Injection Net. The adaptation training was done with Adam as the optimizer, with learning rates of  $8e-3$  and  $7e-3$  in the inner and outer cycles, respectively. Likewise, in this case, the values of  $\beta_1$  and  $\beta_2$  for Adam have been set to 0 and 0.5, respectively. The model training was executed on 22,000 episodes with a batch size of 2 and a number of epochs per task of 4, respectively. The comparison was performed on 10,000 final tasks on *S-Test-Dataset* for 1-, 2-, 5- and 10-shot over 3 repetitions of each experiment. For each task, 10 test samples per class were randomly selected, resulting in 40 test instances in total. The computed mean classification accuracy values are listed in Table 8. As can be observed, the MAMW turns out to be the best-performing method in all experiments apart from the 10-shot experiment, where, as commented in Section 5.1, the Weighting-Injection Net achieves a higher average accuracy. The accuracy values obtained with the proposed methods are better despite using 30% fewer trainable parameters. As the number of shots increases, relation-based models show an even larger accuracy gap than optimization-based ones due to the more robust prediction given by averaging the comparison vectors computed for the available support samples.

Because of the direct mapping between sample and label in the learning process, the single-sample inference time for Reptile, MAML 2<sup>nd</sup> and MAML<sup>+</sup> is independent of the number of shots. Across all the experiments, on an average of 10,000 final tasks, the overall estimated inference time has been 33.47 ms. In comparison to the results in Table 6, only for the 10-shot experiments, the pure optimization-based methods turn out to be 25% faster for single inference, whereas they turn out to be slower in the other configurations.

The task adaptation time needed for the various algorithms is provided in Table 9. The considered state-of-the-art methods require an adaptation time per task that rises considerably as the number of shots increases. On the contrary, relation-based models, thanks to their comparison-based topology, do not require adaptation for new tasks and therefore lead to a null adaptation time. This results in a great advantage for relational topologies over traditional optimization-based topologies.

To test the application limits of the episodic learning approach for radar-based people counting, experiments were

**Table 7** Accuracy achieved for the Weighting-Injection Net with varying feature size on people counting (4 classes): *S-Test-Dataset*. The chosen embedding size  $g$  is 64

Accuracy [%] on <i>S-Test-Dataset</i>	1,024 ( $4 \cdot 4 \cdot g$ )	5,184 ( $9 \cdot 9 \cdot g$ )	12,544 ( $14 \cdot 14 \cdot g$ )
1-shot	61.63 $\pm$ 0.20	60.17 $\pm$ 0.21	59.85 $\pm$ 0.19
2-shot	63.84 $\pm$ 0.18	63.83 $\pm$ 0.17	61.14 $\pm$ 0.16
5-shot	68.82 $\pm$ 0.18	68.63 $\pm$ 0.16	71.77 $\pm$ 0.17
10-shot	67.94 $\pm$ 0.16	71.49 $\pm$ 0.17	76.61 $\pm$ 0.16

also conducted with up to five people per session in the big room  $B$  (Section 3.6). In this case, five sessions of one minute each per location and number of people were collected and used. Locations A and C were used to generate training tasks, and locations B and D were used for testing tasks. Table 10 presents the results obtained on test data for the average of three experiments and 10,000 final tasks. The results for this meta-dataset show similar characteristics to those where an entire room is used exclusively as a test. In general, the two proposed approaches outperform the state of the art regardless of the number of shots. The MAMW proves more stable and performs better in experiments with very few shots (1- and 2-). The Weighting-Injection Net, on the other hand, outperforms MAMW for the 5- and 10- shot approaches. The extension of the counting approach to up to five people and the limitation of radar resolution for close targets in this scenario make generalization more complex. The increased complexity is reflected in the RDIs input instances and features across the different recording locations. For this reason, with a larger number of shots, MAMW performs less well, favoring noise filtering in support samples rather than classification of query instances. Weighting-Injection Net, on the other hand, focuses directly on learning the query class and performs better in this scenario.

In general, although the proposed algorithms outperform the state of the art, they lead to an average accuracy of less than 60% over the six classes with 10-shots. This unfortunately shows that the purely episodic generalization approach with a few shots is limited to scenarios with a very small number of people. Adaptations to larger and more varied datasets or the use of radar sensors with higher resolution could obviate the current limitations. The weights of the counting model up to 5 people need an in-memory size of 1,156 KB. This value is slightly larger than the approach of up to 3 people.

**Table 8** Mean classification accuracy achieved by the various algorithms, for experiments on people counting (4 classes): *S-Test-Dataset*

Accuracy [%] on <i>S-Test-Dataset</i>	Reptile	MAML 2 <sup>nd</sup>	MAML <sup>+</sup>	Weighting-Injection Net	MAMW
1-shot	49.61 $\pm$ 0.16	49.92 $\pm$ 0.18	52.53 $\pm$ 0.17	59.85 $\pm$ 0.19	61.98 $\pm$ 0.19
2-shot	52.02 $\pm$ 0.15	53.79 $\pm$ 0.16	56.91 $\pm$ 0.16	61.14 $\pm$ 0.16	64.48 $\pm$ 0.17
5-shot	57.95 $\pm$ 0.15	60.26 $\pm$ 0.17	60.38 $\pm$ 0.16	71.77 $\pm$ 0.17	73.40 $\pm$ 0.18
10-shot	63.00 $\pm$ 0.16	65.49 $\pm$ 0.17	64.67 $\pm$ 0.16	76.61 $\pm$ 0.16	73.53 $\pm$ 0.16

More information on a single experiment for the adaptation of up to five people is provided in Appendix B.

## 5.2 Active learning experiments

Active learning experiments with the Algorithm 2 are intended to demonstrate how meta learning-driven model initialization benefits task fine-tuning. All the experiments have been carried out on the task of radar-based people counting, using 75% and 25% of the data collected in the  $S$  room as training and testing, respectively. This means that active learning aims to boost the estimation performance in counting people in the entire small room, given all the locations in which the RDIs were collected. Since all the in-room locations are considered at once, the adaptation in this case is more complex than during episodic training. The uncertainty-based experiments used priority scores  $S_p$  defined in (13), (14) and (15). As initialization, the parameters  $\theta$  obtained after the 1-shot episodic learning of Weighting-Injection Net and MAMW on the remaining two environments ( $M$  and  $B$ ) have been used. As  $D_p$  grows larger, the experiments are limited to a maximum of five supports per class. The selected number of epochs for the active learning training is 6,000. For each epoch, 4 queries ( $J$ ) are to be sampled, with  $A$  of them labeled using the uncertainty-based approach. Table 11 compares the average results from three experiments for each defined  $S_p$  score to the random initialization of  $\theta$ . As can be seen from the table, the results for initialization based on MAMW and Weighting-Injection Net vary very little as the chosen priority score differs. Such initialization, however, leads to a great performance gap compared to the random one, which also features training instability over repetitions. The Weighting-Injection Net also seems to achieve slightly better performance than the MAMW. This is most likely

**Table 9** Adaptation time per new task by algorithm and number of shots

Avg. Adaptation Time [ms]	Reptile	MAML 2 <sup>nd</sup>	MAML <sup>+</sup>	Weighting-Injection Net <sup>1</sup>	MAMW <sup>1</sup>
1-shot	130	135	135	–	–
2-shot	275	286	310	–	–
5-shot	606	660	667	–	–
10-shot	1,261	1,294	1,411	–	–

The values, computed on Nvidia<sup>®</sup> Tesla<sup>®</sup> P4 GPU, are averaged over three repetitions of each experiment for 10,000 tasks

<sup>1</sup>For MAMW and Weighting-Injection Net, considering only the need to compare the query with the available supports, the adaptation time is null (0 ms)

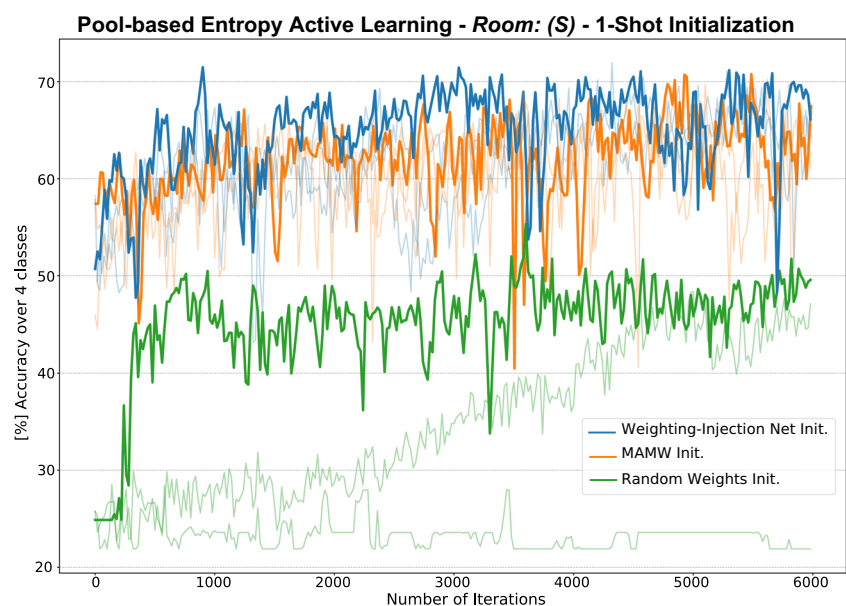
**Table 10** Mean classification accuracy achieved by the various algorithms, for people counting (6 classes): *B* room, B and D locations

Accuracy [%] on <i>B</i> room test	Reptile	MAML 2 <sup>nd</sup>	MAML <sup>+</sup>	Weighting-Injection Net	MAMW
1-shot	35.05 ± 0.11	36.82 ± 0.13	35.56 ± 0.13	36.00 ± 0.13	44.86 ± 0.16
2-shot	37.12 ± 0.12	41.01 ± 0.13	40.03 ± 0.13	46.60 ± 0.15	48.71 ± 0.13
5-shot	39.25 ± 0.12	44.74 ± 0.13	43.86 ± 0.13	55.69 ± 0.14	50.64 ± 0.14
10-shot	43.19 ± 0.12	48.56 ± 0.13	46.01 ± 0.12	57.83 ± 0.14	56.71 ± 0.14

**Table 11** Accuracy on people counting (4 classes), obtained through pool-based sampling active learning

Accuracy [%] on Small Room <i>S</i>	Weighting-Injection Net	MAMW	Random Init.
$S_{LC}$	63.09	60.81	31.14
$S_{MS}$	63.41	59.98	26.46
$S_E$	63.62	61.54	43.44

All the *S* room data have been used for the adaptation. The results are averaged over three experiment repetitions of 6,000 iterations each. The initialization consists of meta-learned weights for the *M* and *B* rooms

**Fig. 16** Entropy pool-based active learning accuracy across epochs. The thicker lines highlight the best experiments by type of initialization. Accuracy values are averaged per trial every 20 epochs. Random initialization (green) experiments are more unstable and collapse to 25% random learning on 4 classes

related to the large availability of labeled data, which for a test room setup, makes this method more performant than MAMW (Section 5.1). In the case of random initialization, however, the model succeeds in learning almost exclusively when entropy  $S_e$  is used as the scoring function. This may be due to the entropy formulation itself, which results in a more balanced query selection by taking into account the distribution over all classes for the score computation.

The accuracy learning curve for the entropy-based experiments is depicted in Fig. 16. Adaptation starting with Weighting-Injection Net and MAMW weights exhibits similar accuracy profiles as training epochs progress. Random initialization, on the other hand, not only leads to lower-performing learning but also to instability and experiment failure, collapsing to a 25% accuracy over the four classes. In this case, the algorithm encounters difficulties with only a few learning data at a time to generalize to all locations. Fluctuations in accuracy curves are due to adaptation to new labeled data sampled from different  $S$  room locations, which normally display different features. This behavior can be observed in the t-SNE representations of the data in Section 3.6.

## 6 Conclusion

This paper features how meta learning and active learning can be effectively employed for radar-based people counting using real-world data. For such a use case, multiple meta-datasets are generated based on different combinations of rooms and radar orientations. Episodic learning for few-shot adaptation is carried out through a comparative approach. The model learns task-wise to map features of query examples to representative support instances belonging to the same class. In this way, the belonging class of a radar instance is predicted by comparing it with representative support examples of classes zero to three people. With respect to the traditional weighting network, an injection module increases the input data dimensionality before the comparison step. This process facilitates the comparison of query and support features, reducing episodic task overfitting and aiding generalization. The overall topology with an injection module is called the Weighting-Injection Net.

An episodic adaptation algorithm called model-agnostic meta-weighting is then presented for specific adaptations to very few-shot per task. This two-step training algorithm combines the weighting network topology and the optimization-based meta learning approach to enhance the feature extraction capabilities of the model. The approach features an inner step task adaptation that compares support instances with a noisy version of themselves, leading to more stable generalization training, especially in

the 1-shot training. Finally, a pool-based active learning approach designed specifically for relation-based methods is presented. Using only the available samples with the highest prediction uncertainty, this algorithm seeks to minimize the number of examples needed for learning.

The presented meta learning achieves cutting-edge accuracy in people counting while also yielding other performance advantages. The relation-based topology grants no training time for adaptation at new radar test locations. Furthermore, the availability of multiple support examples per class allows for more robust averaged query estimation. Both the presented algorithms are up to 15% more accurate than the state-of-the-art for 1- and 10-shot. They are also found to be up to 50% faster for computing single-sample inference when the model is tested on a new task. The active learning algorithm performs better and is more stable when the initialization is set to the episodically learned weights rather than at random. Nonrandom initialization improves radar adaptation accuracy by 30% on test room radar instances.

Despite the great benefits shown, the work presented is only tested offline on previously collected data. In the future, it will be important to test such a system in a real-time setting. The monitoring approach with more than three people leads to accuracy performance which may be insufficient in several practical contexts. Future work will focus on using relation-based topologies and sensor fusion to counter the current limitations. The use of an unconventional injection module for the relational networks could bring additional benefits for feature representation in episodic learning. In-depth studies will therefore be conducted on the possible applications and limitations of such a module. Research on the injection module will also be carried out in the field of the interpretability of neural networks and training complexity. Also, further active learning and uncertainty sampling strategies that focus on episodic learning with relation-based approaches will be investigated.

## Appendix A: Experiments on public dataset

This section presents the results obtained with the Weighting-Injection Net (Section 4.1.1) and MAMW (Section 4.1.2) on Omniglot [53] public dataset.

### A.1 Omniglot dataset

Omniglot [53], is a dataset specifically created for few-shot learning. That dataset contains hand-written instances of as many as 1,623 characters taken from 50 alphabets. Each character was drawn by different people a total of 20 times each. The meta-dataset, divided between  $\mathcal{D}^{m-train}$  and  $\mathcal{D}^{m-test}$ ,

as defined in the Omniglot repository<sup>2</sup>, was used for training and testing the Weighting-Injection Net and MAMW.

## A.2 Experiments on omniglot

On Omniglot, the experiments have been performed with 1-shot for 2- and 5-way and 5-shot for 5- and 10-way. The topology used for these experiments is the same as for radar-based people counting (Figs. 1 and 11), but it has been adapted to the larger input size. Each handwritten sample has a resolution of  $105 \times 105$  pixels. The chosen embedding size  $g$  and feature size for the injection module have been 32 and 22, respectively. The configuration of the layers is presented in Table 12. In this case, task classification is also accomplished by minimizing categorical crossentropy with Adam as the optimization method, with  $\beta_1$  and  $\beta_2$  set to 0 and 0.5, respectively. The episodic learning rate used for the Weighting-Injection Net and the outer step learning rate used for MAMW experiments have been set to  $3e - 4$ . For the MAMW, an inner step learning rate of  $5e - 5$  has been utilized. Regardless of the number of shots, one query sample  $q_j$  per class per episode is used for tasks sampled on  $p(\mathcal{T}_r)$  and defined on  $\mathcal{D}^{m-train}$ . The generalization is then tested episode-wise on 10 test instances per class, on tasks  $\mathcal{T}$  sampled from  $p(\mathcal{T}_s)$  in  $\mathcal{D}^{m-test}$ , and  $p(\mathcal{T}_r)$ . All the experiments have been performed on 22,000 episodes. The built models have then been tested for 10,000 final tests on tasks sampled from  $p(\mathcal{T}_s)$  in  $\mathcal{D}^{m-test}$ .

## A.3 Results and state-of-the-art comparison omniglot

Also on Omniglot, to assess the generalization performance, box plots have been generated based on the average accuracy obtained for sets of 2,200 episodes. As an example, the trend obtained for the 5-shot, 5-way Weighting-Injection Net experiment is shown in Fig. 17. As the episodes progress, training on Omniglot sees a sharper increase in generalization in the early stages than radar-based people counting. This is most likely caused by the greater variety of classes among the handwritten characters, whose features take longer to be extracted by the relational network through the comparison mechanism.

The accuracy values achieved with Weighting-Injection Net and MAMW are listed for the various experiments in Table 13, in comparison with state-of-the-art methods. For the state-of-the-art algorithms, a TensorFlow<sup>TM</sup> implementation and the same testing pipeline as for the people counting

**Table 12** Network Layers Configuration - Omniglot

Module	Type	Filter Shape <sup>1</sup>	Output Shape
Injection	Conv2D	$2 \times 2 \times 1 \times 64$	$j \times 104 \times 104 \times 64$
	MaxPool	$2 \times 2$	$j \times 52 \times 52 \times 64$
	Conv2D	$3 \times 3 \times 64 \times 64$	$j \times 50 \times 50 \times 64$
	MaxPool	$2 \times 2$	$j \times 25 \times 25 \times 64$
	Conv2D	$2 \times 2 \times 64 \times 64$	$j \times 24 \times 24 \times 64$
	Conv2D	$3 \times 3 \times 64 \times 64$	$j \times 22 \times 22 \times g$
Comparison	Conv2D <sup>2</sup>	$3 \times 1 \times 2g \times g$	$jc \times 23 \times 24 \times g$
	MaxPool	$3 \times 3$	$jc \times 7 \times 8 \times g$
	Conv2D	$3 \times 3 \times g \times g$	$jc \times 5 \times 6 \times g$
	AvgPool	$1 \times 1$	$jc \times g$
Weighting	Dense	$ng \times 64$	$j \times 64$
	Dense	$1 \times n$	$j \times n$

The  $n$  and  $c$  represent the indices for class and shot number, respectively. The index of the  $j$ -th query shot is represented by  $j$ . The  $g$  represents the embedding size, which was set to 32 in the experiments

<sup>1</sup> For the Conv2D layers, the filter shape dimensions are, respectively, kernel height and width, and input and output channels

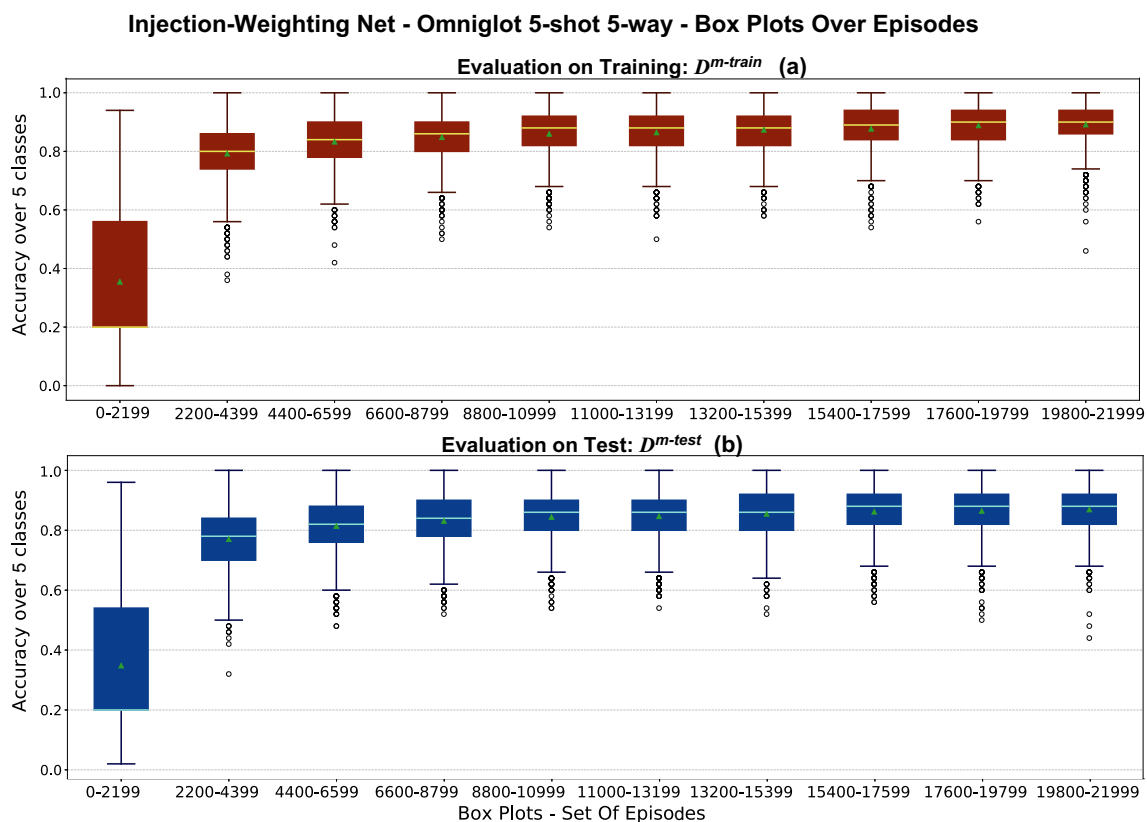
<sup>2</sup> In this layer, a symmetric zero-padding of 1 is applied to both the width and height of the samples

comparison have been adopted. The accuracy of the tasks is not calculated on a single query sample per class, as in Reptile [32], but on ten test instances per class in a step following the learning step. This allows a more fair comparison with relational algorithms, where the query example is not used in a step subsequent to the support ones. In addition, no data augmentation or scaling is performed on single inputs, in contrast to the MAML methods presented in [31, 52]. For the state-of-the-art methods, the same CNN topology and configurations presented in Section 5.1.1 for radar-based people counting have been used on Omniglot.

All experiments have been performed on an Nvidia<sup>®</sup> Tesla<sup>®</sup> P4 GPU and CUDA<sup>®</sup> Toolkit v11.1.0 for parallel computing.

Similarly to what has been observed in Section 5.1 for the radar-based people counting dataset, the MAMW seems to perform better than the Weighting-Injection Net in the 1-shot and 10-way scenarios (Table 13). For the 5-shot 5-way experiment, the two relation-based algorithms achieved similar accuracy, which is comparable to MAML 2<sup>nd</sup>. This may be inherent in the fact that for Omniglot, unlike radar data, there is no intrinsic background noise in the input instances. Consequently, the introduction of noise in the comparison between supports in MAMW does not promote generalization learning when many shots are fed to the network. Conversely, MAMW inner step may divert attention away from the learning goal of single tasks. Even for Omniglot,

<sup>2</sup> Available at <https://github.com/brendenlake/omniglot/>



**Fig. 17** Accuracy statistics box plots vs. episodes for a Weighting-Injection Net 5-shot 5-way experiment on Omniglot. The red box plots are constructed on validation tasks sampled from  $\mathcal{D}^{m-train}$  (a), whereas

the blue ones are constructed on test tasks sampled from  $\mathcal{D}^{m-test}$  (b). The median and mean values are represented by a horizontal line and a green triangle in each box plot

using the injection module seems to help generalization learning by making it easier to compare support features and queries in a higher dimension. Regardless of the number of ways and shots, the Weighting-Injection Net and MAMW outperform the other state-of-the-art in most of the Omniglot experiments. The presented methods, with about 30% fewer parameters, also perform significantly better in single-shot approaches than optimization-based methods. In the 1-shot 5-way experiment, MAMW leads to an average accuracy about 18% higher than MAML<sup>+</sup>.

## Appendix B: More people count details

This section analyzes a single episodic meta learning experiment for radar-based indoor people counting up to five people.

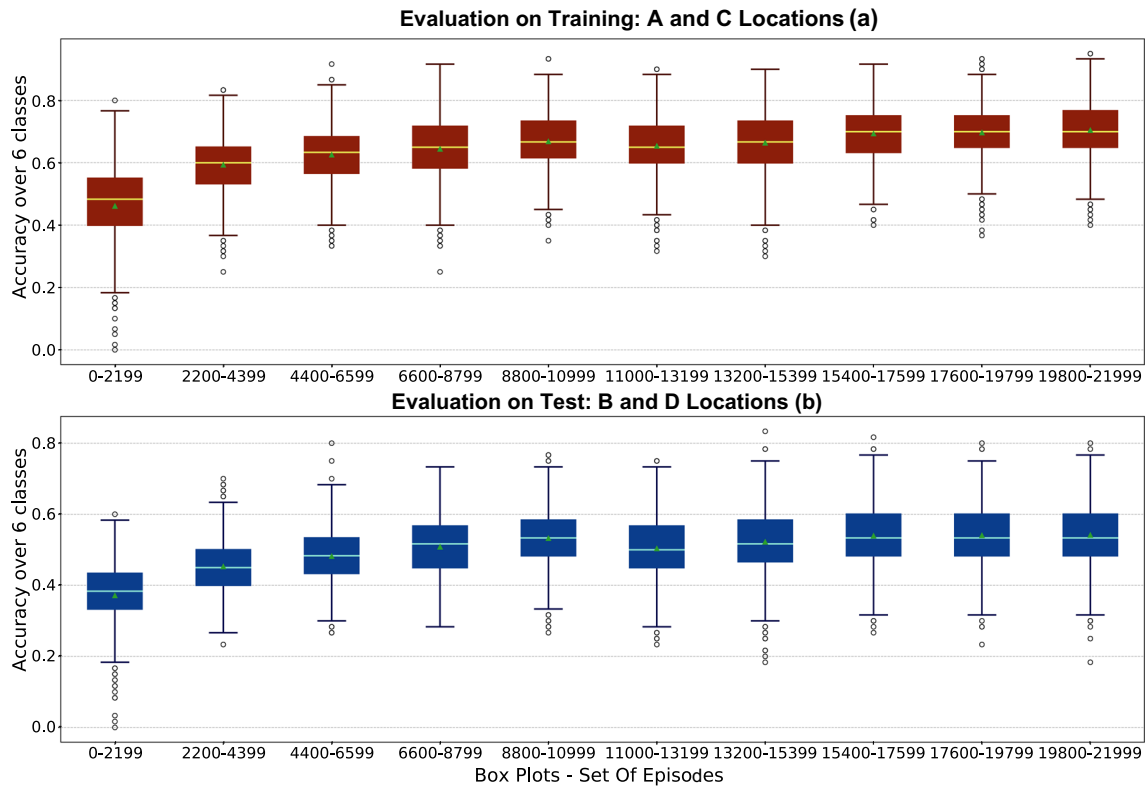
### B.1 Single experiment people counting analysis up to five individuals

The outcomes of the episodic adaptation on the five people meta-dataset of Section 5.1.1 can be analyzed at the level of the individual experiment. Every experiment consists of

**Table 13** The mean classification accuracy achieved by the various selected algorithms for experiments on Omniglot

Accuracy [%] on Omniglot Eval.	Reptile	MAML 2 <sup>nd</sup>	MAML <sup>+</sup>	Weighting-Injection Net	MAMW
1-shot 2-way	69.21 ± 0.30	74.18 ± 0.34	80.30 ± 0.32	76.65 ± 0.32	81.99 ± 0.31
1-shot 5-way	41.14 ± 0.10	55.76 ± 0.23	59.36 ± 0.23	71.46 ± 0.23	72.19 ± 0.23
5-shot 5-way	52.59 ± 0.20	84.99 ± 0.14	77.50 ± 0.18	85.76 ± 0.15	85.02 ± 0.16
5-shot 10-way	36.72 ± 0.15	78.60 ± 0.12	77.61 ± 0.13	79.11 ± 0.12	81.23 ± 0.12

### Weighting-Injection-Net - People Counting 10-shot - 5 People in B Room - Box Plots Over Episodes



**Fig. 18** Accuracy statistics box plots vs. episodes for a Weighting-Injection Net 10-shot 6-way experiment on radar-based people counting (*B* room). The red box plots are constructed on validation tasks (a),

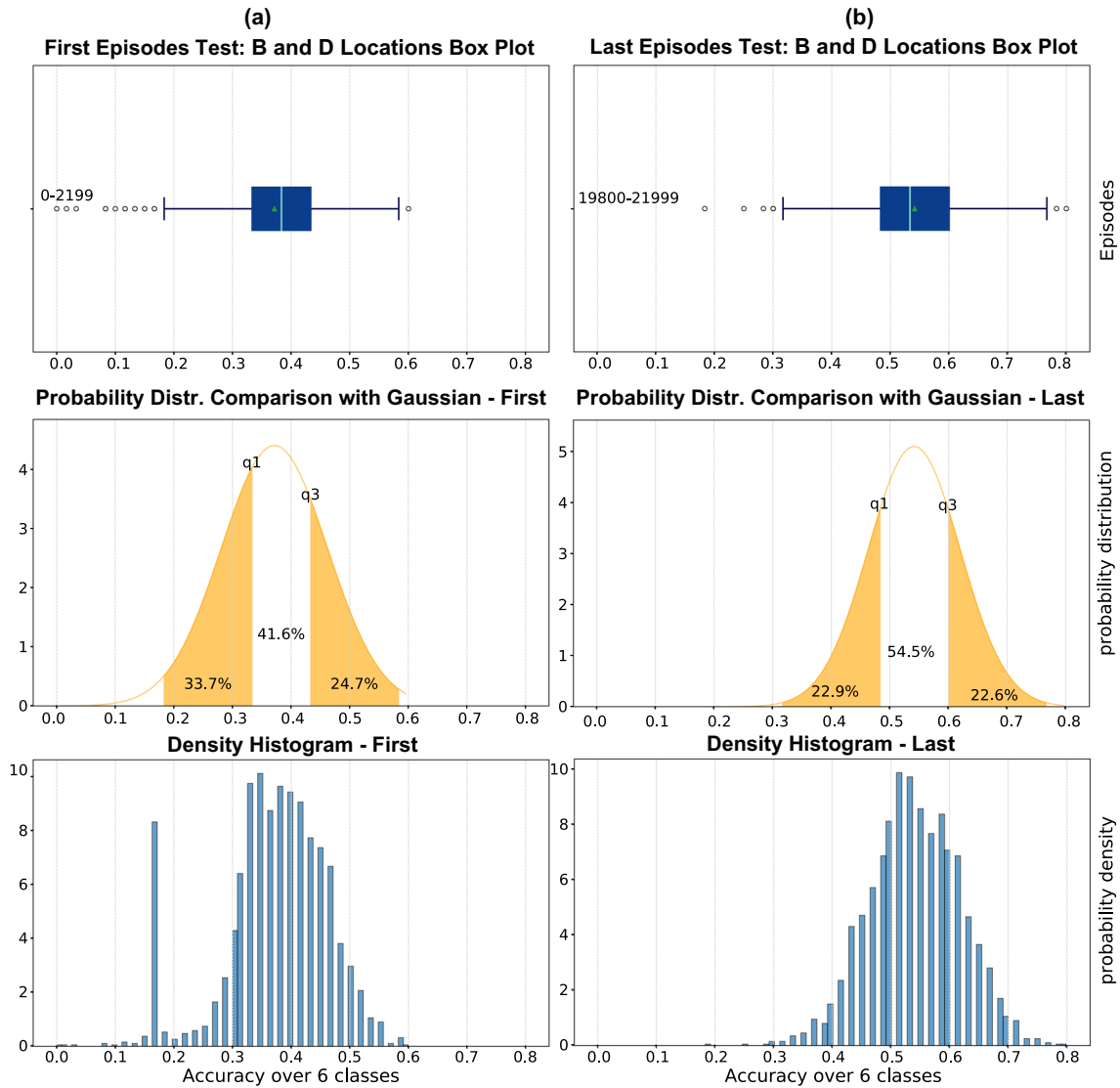
whereas the blue box plots are constructed on test tasks (b). The median and mean values are represented by a horizontal line and a green triangle in each box plot

22,000 episodes of meta-training in the room *B* (Fig. 5). Training and validation are performed on tasks sampled from locations *A* and *C* in the room, while testing is done on tasks sampled from locations *B* and *D*. The experiment is a 6-way, since zero individuals in the room is also considered a class. Figures 18, 19 and 20, show different statistical insights of a 10-shot Weighting-Injection Net experiment. Figure 18 displays the trend of box plots built on accuracy as episodes increase. Compared to the training up to three people (Fig. 12), the adaptation up to five people shows a less pronounced trend of improvement. In this case, the test fails to generalize better from 15,000 episodes onward, reaching a saturation of accuracy around 55%. Figure 19 reveals the density histograms underlying the first and last box plots constructed on the test in episodic learning. In comparison to the adaptation of up to three people Fig. 13, no marked reduction in whiskers or negative skew in the last histogram is

noticeable. Yet, there is an increase in average accuracy from 37% to 55% (18% improvement in generalization). A very interesting analysis can be done by analyzing the accuracy on individual classes, thus by generating the cumulative confusion matrices shown in Fig. 20. As in the confusion matrices generated for the 4-way approach (Figs. 14 and 15), the model easily succeeds in classifying the absence of people in the environment, reaching a solid 98% class accuracy in the test at the end of episodic learning. Further, as the episodes progress, the generalization approach yields higher accuracy in counting more than one person. Moreover, most of the miss-classifications lie around the main diagonal of the confusion matrix, which represents the  $\pm 1$  of accuracy. This means that most of the classification errors tend to under- or overestimate the number of people in the room by only one unit.

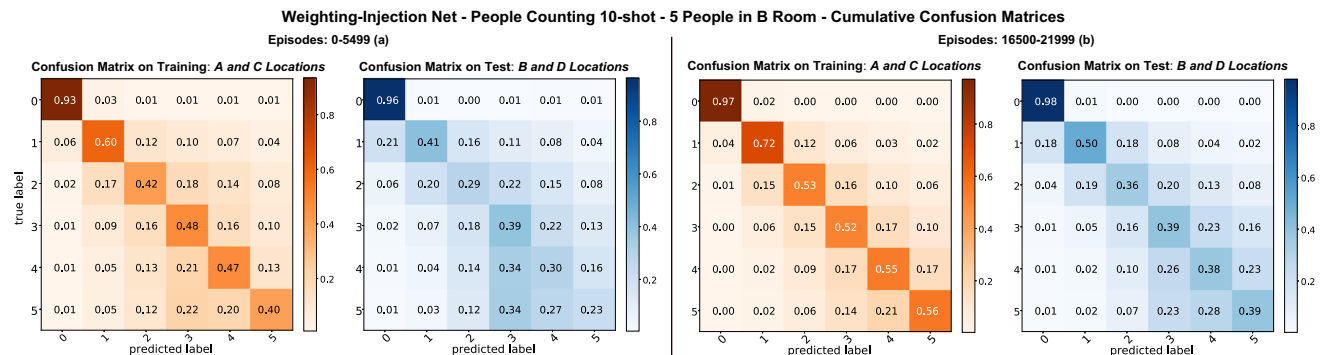


**Weighting-Injection-Net - People Counting 10-shot - 5 People in B Room - First Vs Last Test Box Plot: Accuracy Distribution**



**Fig. 19** Weighting-Injection Net 10-shot 6-way, first (a) and last (b) box plot underlying distributions, generated on people counting test tasks. The q1 and q2 values on the Gaussians indicate the first and third quartiles, respectively. The probability density histograms show

the actual non-Gaussian nature of the distribution. The accuracy probability density for the last box plot (b) has a mean value shifted towards higher accuracy as a result of the generalization learning



**Fig. 20** Cumulative confusion matrices for Weighting-Injection Net 10-shot 6-way people counting experiment. Confusion matrices are obtained on the first (a) and last (b) 5,550 meta-iterations in the validation phase for both training and test sampled tasks

**Acknowledgements** The authors would like to thank the reviewers for their time and efforts.

**Funding** This work has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No. 876925 (ANDANTE). The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Belgium, Germany, Netherlands, Portugal, Spain, Switzerland. Funding for open access publishing: Universidad de Granada/CBUA.

**Availability of Data and Materials** The data are not publicly available due to internal company board policy.

**Code Availability** The codes for the meta learning algorithms on Omniglot are available at: <https://github.com/GiancoMauro/TF-Meta-Learning>

## Declarations

**Ethics Approval** The work does not include any personal data relatable to identifiable living persons. The methods and results presented in this paper represent a general radar-based solution for nonuser-specific people counting. No personal information or photos have been obtained from participants.

**Consent to Participate** Informed consent was obtained from all subjects involved in the study.

**Conflicts of Interests** The authors declare no conflicts of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Moisello E, Malcovati P, Bonizzoni E (2021) Thermal sensors for contactless temperature measurements, occupancy detection, and automatic operation of appliances during the COVID-19 pandemic: A review. *Micromachines* 12(2):148
2. Rahman A, Yaakob S, Razak A, Ramlee R (2021) Post COVID-19 implementation of a bidirectional counter with reduced complexity for people counting application. In: *Journal of physics: conference series*, vol. 1878. IOP Publishing, pp 012040
3. Taha A, Krabicka J, Wu R, Kyberd P, Adams N (2019) Design of an occupancy monitoring unit: a thermal imaging based people counting solution for socio-technical energy saving systems in hospitals. In: *2019 11th Computer Science and Electronic Engineering (CEECE)*, IEEE pp 1–6
4. Hou YL, Pang GK (2010) People counting and human detection in a challenging situation. *IEEE Trans Syst Man Cybern-part Syst Hum* 41(1):24–33
5. Prathiba GT, Dhas Y (2013) Literature survey for people counting and human detection. *IOSR J Eng (IOSRJEN)* 3(1):05–10
6. Raghavachari C, Aparna V, Chithira S, Balasubramanian V (2015) A comparative study of vision based human detection techniques in people counting applications. *Procedia Comput Sci* 58:461–469
7. Stec M, Herrmann V, Stabernack B (2019) Using time-of-flight sensors for people counting applications. In: *2019 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, IEEE, pp 59–64
8. Brunetti A, Buongiorno D, Trotta GF, Bevilacqua V (2018) Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing* 300:17–33
9. Ilyas N, Shahzad A, Kim K (2019) Convolutional-neural network-based image crowd counting: review, categorization, analysis, and performance evaluation. *Sensors* 20(1):43
10. Abdullah F, Ghadi YY, Gochoo M, Jalal A, Kim K (2021) Multi-person tracking and crowd behavior detection via particles gradient motion descriptor and improved entropy classifier. *Entropy* 23(5):628
11. Basalamah S, Khan SD, Ullah H (2019) Scale driven convolutional neural network model for people counting and localization in crowd scenes. *IEEE Access* 7:71576–71584
12. Ivasic-Kos M, Kristo M, Pobar M (2019) Person Detection in thermal videos using YOLO. In: *Proceedings of SAI Intelligent Systems Conference*, Springer, pp 254–267
13. Kouyoumdjieva ST, Danielis P, Karlsson G (2019) Survey of non-image-based approaches for counting people. *IEEE Commun Surv Tutor* 22(2):1305–1336
14. Gupta A, Maurya S, Mehra N, Kapil D (2021) Covid-19: Employee fever detection with thermal camera integrated with attendance management system. In: *2021 11th International conference on cloud computing, data science & engineering (confluence)*, IEEE, pp 355–361
15. Lesani A, Nateghinia E, Miranda-Moreno LF (2020) Development and evaluation of a real-time pedestrian counting system for high-volume conditions based on 2D LiDAR. *Transp Res C Emerg Technol* 114:20–35
16. Günter A, Böker S, König M, Hoffmann M (2020) Privacy-preserving people detection enabled by solid state LiDAR. In: *2020 16th International conference on intelligent environments (IE)*, IEEE, pp 1–4
17. Zhao M, Li T, Abu Alsheikh M, Tian Y, Zhao H, Torralba A et al (2018) Through-wall human pose estimation using radio signals. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 7356–7365
18. Wang F, Zhou S, Panev S, Han J, Huang D (2019) Person-in-WiFi: Fine-grained person perception using WiFi. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 5452–5461
19. Rimmelspacher J, Ciocoveanu R, Steffan G, Bassi M, Issakov V (2020) Low power low phase noise 60 GHz multichannel transceiver in 28 nm CMOS for radar applications. In: *2020 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, IEEE, pp 19–22
20. Ciocoveanu R, Issakov V (2021) Low-Power 60GHz Receiver with an Integrated Analog Baseband for FMCW Radar Applications in 28nm CMOS Technology. In: *2021 IEEE 20th topical meeting on silicon monolithic integrated circuits in rf systems (SiRF)*, IEEE, pp 4–6
21. Trotta S, Weber D, Jungmaier RW, Baheti A, Lien J, Noppeney D et al (2021) Soli: a tiny device for a new human machine interface. In: *2021 IEEE International solid-state circuits conference (ISSCC)*, vol. 64, IEEE, pp 42–44
22. Santra A, Hazra S (2020) *Deep learning applications of short-range radars*. Artech House

23. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
24. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations (ICLR). San Diego, CA, USA, pp 1–14
25. Pan SJ, Yang Q (2009) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
26. Wang Y, Yao Q, Kwok JT, Ni LM (2020) Generalizing from a few examples: A survey on few-shot learning. *ACM Comput Surv (csur)* 53(3):1–34
27. Li X, Sun Z, Xue JH, Ma Z (2021) A concise review of recent few-shot meta-learning methods. *Neurocomputing* 456:463–468
28. Köksal A, Schick T, Schütze H (2022) MEAL: Stable and Active Learning for Few-Shot Prompting. *arXiv preprint arXiv:2211.08358*
29. Vanschoren J (2019) Meta-learning. *Automated machine learning: methods, systems, challenges*, pp 35–61
30. Hospedales T, Antoniou A, Micaelli P, Storkey A (2021) Meta-learning in neural networks: A survey. *IEEE Trans Pattern Anal Mach Intell* 44(9):5149–5169
31. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning, PMLR, pp 1126–1135
32. Nichol A, Achiam J, Schulman J (2018) On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*
33. Khadka R, Jha D, Hicks S, Thambawita V, Riegler MA, Ali S et al (2022) Meta-learning with implicit gradients in a few-shot setting for medical image segmentation. *Comput Biol Med* 105227
34. Mauro G, Chmurski M, Servadei L, Cuellar M, Morales-Santos DP (2022) Few-Shot User-definable Radar-based Hand Gesture Recognition at the Edge. *IEEE Access*
35. Sung F, Yang Y, Zhang L, Xiang T, Torr PH, Hospedales TM (2018) Learning to compare: Relation network for few-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1199–1208
36. Zeng X, Wu C, Ye WB (2021) User-Definable Dynamic Hand Gesture Recognition Based on Doppler Radar and Few-Shot Learning. *IEEE Sensors J* 21(20):23224–23233
37. Kumar P, Gupta A (2020) Active learning query strategies for classification, regression, and clustering: a survey. *J Comput Sci Technol* 35:913–945
38. Ren P, Xiao Y, Chang X, Huang PY, Li Z, Gupta BB et al (2021) A survey of deep active learning. *ACM Comput Surv (CSUR)* 54(9):1–40
39. Massa L, Barbosa A, Oliveira K, Vieira T (2021) LRCN-RetailNet: A recurrent neural network architecture for accurate people counting. *Multimedia Tools Appl* 80(4):5517–5537
40. Gomez A, Conti F, Benini L (2018) Thermal image-based CNN's for ultra-low power people recognition. In: Proceedings of the 15th ACM international conference on computing frontiers, pp 326–331
41. Kianoush S, Savazzi S, Rampa V, Nicoli M (2019) People counting by dense WiFi MIMO networks: Channel features and machine learning algorithms. *Sensors* 19(16):3450
42. Bao R, Yang Z (2021) CNN-based regional people counting algorithm exploiting multi-scale range-time maps with an IR-UWB radar. *IEEE Sensors J* 21(12):13704–13713
43. Stephan M, Hazra S, Santra A, Weigel R, Fischer G (2021) People counting solution using an FMCW radar with knowledge distillation from camera data. In: 2021 IEEE Sensors, IEEE, pp 1–4
44. Vandoni J, Aldea E, Le Hégarat-Masclé S (2017) Active learning for high-density crowd count regression. In: 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS), IEEE, pp 1–6
45. Zhao Z, Shi M, Zhao X, Li L (2020) Active crowd counting with limited supervision. In: European conference on computer vision, Springer, pp 565–581
46. Zhang Y, Zhou D, Chen S, Gao S, Ma Y (2016) Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 589–597
47. Reddy MKK, Hossain M, Rochan M, Wang Y (2020) Few-shot scene adaptive crowd counting using meta-learning. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 2814–2823
48. Zan C, Liu B, Guan W, Zhang K, Liu W (2021) Learn from object counting: crowd counting with meta-learning. *IET Image Process* 15(14):3543–3550
49. Hou X, Xu J, Wu J, Xu H (2021) Cross domain adaptation of crowd counting with model-agnostic meta-learning. *Appl Sci* 11(24):12037
50. Hou H, Bi S, Zheng L, Lin X, Wu Y, Quan Z (2022) DASECount: Domain-agnostic sample-efficient wireless indoor crowd counting via few-shot learning. *IEEE Internet Things J*
51. Zhang Y, Chen Y, Wang Y, Liu Q, Cheng A (2021) CSI-based human activity recognition with graph few-shot learning. *IEEE Internet Things J* 9(6):4139–4151
52. Antoniou A, Edwards H, Storkey A (2019) How to train your MAML. In: 7th International Conference on Learning Representations (ICLR). New Orleans, LA, USA, pp Poster
53. Lake BM, Salakhutdinov R, Tenenbaum JB (2015) Human-level concept learning through probabilistic program induction. *Science* 350(6266):1332–1338

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Gianfranco Mauro** received a B.Sc. degree in electronics engineering from Politecnico di Milano, Italy in 2017. Then, in 2020 he got the M.Sc. degree in Biomedical Engineering - Technologies for Electronics from Politecnico di Milano. He joined Infineon Technologies AG in August 2020 as a Ph.D. candidate in collaboration with the University of Granada, Spain. He is currently working on few-shot learning for radar-based applications, especially in the fields of activity

recognition and health monitoring. His main research interests involve few-shot learning, data feature extraction, and the innovative use of radar technologies in the everyday context.



**Ignacio Martinez-Rodriguez** received his B.Sc in Computer Science Engineering at the University of Granada in 2020. During his last year of Bachelor's studies, he traveled to Munich, Germany, where he completed an Erasmus program year at the Technical University of Munich. One year later, he received his Master's Degree in Data Science and Computer Engineering from the University of Granada. His main topics of interest involve

Data Science and Artificial Intelligence as well as other computer science topics like Blockchain technologies.



**Julius Ott** received his B.Sc. degree in electrical engineering from the Technical University of Munich, Germany in 2020. He joined Infineon Technologies in July 2021 as a student for his master's thesis. He is currently working on parameter optimization for radar-based tracking with Reinforcement Learning. In addition, his main research interests are representation learning and data sampling strategies.



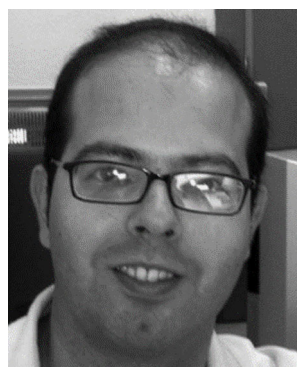
**Lorenzo Servadei** received his Ph.D. degree from the Johannes Kepler University Linz, Austria (Summa Cum Laude), in collaboration with Infineon Technologies AG. During his Ph.D. studies, his research focus has been Hardware Optimization with Machine Learning. He is currently working as a Senior Staff Machine Learning Engineer at Infineon Technologies AG and lecturing Machine Learning at the Technical University of Munich, Germany. He is IEEE as well as ACM

Member.



**Robert Wille** received the Diploma and Dr.-Ing. degrees in computer science from the University of Bremen, Germany, in 2006 and 2009, respectively. Since 2009, he has been with the University of Bremen, the German research center for artificial intelligence (DFKI), the University of Applied Science of Bremen, the University of Potsdam, and the Technical University Dresden. From 2015 to 2022, he was a Full Professor at Johannes Kepler University Linz, Austria. He is currently a Full and

Distinguished Professor at the Technical University of Munich, Germany, and the Chief Scientific Officer at the software competence center Hagenberg GmbH (SCCH), Austria. His research interests include design of circuits and systems for both conventional and emerging technologies and with more than 400 published papers. He was awarded an ERC Consolidator Grant, a Distinguished and a Lighthouse Professor Appointment, a Google Research Award, and more.



**Manuel P. Cuellar** graduated in computer engineering in 2003. He received his Ph.D. degree in 2006 with a focus on time series prediction, parameter identification, and neural networks. He is currently a full-time Teacher with the Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain. He has worked in multivariate image analysis and real-time control tasks. His research interests include neural and social


networks, evolutionary optimization, and fuzzy systems.



**Diego P. Morales-Santos** received the B.Sc., M.Eng., and Ph.D. degrees in electronics engineering from the University of Granada, Spain, in 2001 and 2011, respectively. Since 2001, he has been an Associate Professor with the Department of Computer Architecture and Electronics, University of Almeria, before joining the Department of Electronics and Computer Technology, University of Granada, in 2006, where he currently serves as a tenured Professor. He is the Co-Founder of

the Biochemistry and Electronics as Sensing Technologies (BEST) Research Group, at the University of Granada. He has co-authored more than 80 scientific contributions. His current research interests include low-power energy conversion, energy harvesting for wearable sensing systems, and new materials for electronics and sensors.

## Authors and Affiliations

Gianfranco Mauro<sup>1,2</sup>  · Ignacio Martinez-Rodriguez<sup>1</sup> · Julius Ott<sup>1,3</sup> · Lorenzo Servadei<sup>1,3</sup> · Robert Wille<sup>3</sup> · Manuel P. Cuellar<sup>4</sup> · Diego P. Morales-Santos<sup>2</sup>

Ignacio Martinez-Rodriguez  
ignacio.martinezrodriguez@infineon.com

Julius Ott  
julius.ott@infineon.com

Lorenzo Servadei  
lorenzo.servadei@infineon.com

Robert Wille  
robert.wille@tum.de

Diego P. Morales-Santos  
diegopm@ugr.es

<sup>1</sup> Infineon Technologies AG, Am Campeon 1-15, Neubiberg  
85579, Germany

<sup>2</sup> Department of Electronic and Computer Technology,  
University of Granada, Avenida de Fuente Nueva s/n, Granada  
18071, Spain

<sup>3</sup> Department of Electrical and Computer Engineering,  
Technical University of Munich, Arcisstrasse 21, Munich  
80333, Germany

<sup>4</sup> Department of Computer Science and Artificial Intelligence,  
University of Granada, C/. Pdta. Daniel Saucedo Aranda s/n,  
Granada 18015, Spain