CTL-I: Infrared Few-Shot Learning via Omnidirectional Compatible Class-Incremental

Biwen Yang[1], Ruiheng Zhang[1](✉), Yumeng Liu[2], Guanyu Liu[1], Zhe Cao[1], Zhidong Yang[1], Heng Yu[1], and Lixin Xu[1]

[1] Beijing Institute of Technology, No.5 Yard, Zhong Guan Cun South Street Haidian District, Beijing, China
ruiheng.zhang@bit.edu.cn

[2] Beijing Key Laboratory of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing, China
yumeng@iscas.ac.cn

Abstract. Accommodating infrared novel class in deep learning models without sacrificing prior knowledge of base class is a challenging task , especially when the available data for the novel class is limited. Existing infrared few-shot learning methods mainly focus on measuring similarity between novel and base embedding spaces or transferring novel class features to base class feature spaces. To address this issue, we propose Infrared (omnidirectional) Compatibility Training Learning (CTL-I). We suggest building a virtual infrared prototype in the basic model to preserve feature space for potential new classes in advance. We use a method of coupling virtual and real data to gradually update these virtual prototypes as predictions for potential new categories, resulting in a more powerful classifier that can effectively adapt to new categories while retaining knowledge about general infrared features learned from the base class. Our empirical results demonstrate that our approach outperforms existing few-shot incremental learning methods on various benchmark datasets, even with extremely limited instances per class. Our work offers a promising direction for addressing the challenges of few-shot incremental learning in infrared image.

Keywords: Infrared · Few-shot Learning · Class-incremental Learning

1 Introduction

In recent years, significant breakthroughs have been made using various neural network architectures [1]. These achievements hinge on extensive datasets with well-balanced class distribution. However, the challenge emerges when dealing with infrared image data, which often arrives as a continuous stream [2], introducing novel classes [3] in open-world scenarios, such as newly discovered frog species in rainforests. This necessitates flexible methods for incorporating novel

Z. Tan et al. (Eds.): BDTA 2023, LNICST 555, pp. 3–17, 2024.
https://doi.org/10.1007/978-3-031-52265-9_1

class knowledge, termed Class-Incremental Learning (CIL). Yet, training models with new class data triggers catastrophic forgetting [4], where accuracy on old classes sharply declines post-update. CIL mitigates this, but designing compatible CIL algorithms remains challenging due to limited access to high-quality infrared data [11,13].

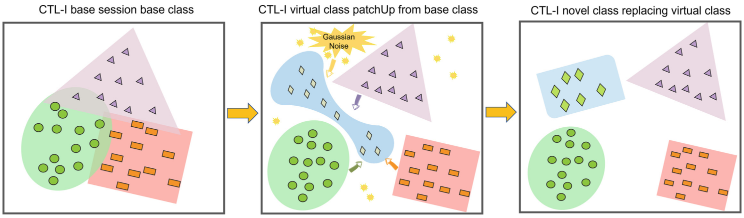

Fig. 1. The left figure represents the occupancy of base class data in the feature space. The middle represents the association between virtual class instances generated through collaboration between base class instances and random Gaussian noise, which exceeds the range of the base class due to the latter. The right shows that the new class data inherits a portion of the feature space of the virtual class and the resolution ability from the base class.

For the practical application of CIL, amassing sufficient new class instances is often unrealistic, particularly in scenarios like infrared scenes where factors like noise and ambient light limit data collection. This gives rise to incremental learning with few shots (FSCIL), where models learn new classes with limited instances, risking overfitting to new classes and forgetting old ones. To address overfitting, algorithms [5,14] have been proposed for less-shot learning.

In FSCIL, models learning new class data must retain distinctiveness for base classes, akin to retaining knowledge of solid geometry after learning calculus. This is termed backward compatibility [6]. The CIL method attempts to increase backward compatibility by reducing the distinguishability of old classes, while the FSCIL method achieves this by abandoning the update of old classifiers and adding novel class classifiers.

Currently, the main focus of research is on backward compatibility patterns, leaving the task of learning novel classes to subsequent models. Some work has shown that if the previous model is not well learned, the latter model will also degrade on this basis. This is indeed a potential issue to encounter, but it is not the focus of the current issue discussion. The fundamental problem with backward compatibility mode is that in the incremental stage, the data participating in learning is all novel classes, competing with the old class data, thus completely occupying the feature space of the old class data. In addition, there is a forward compatible method [15] that, although it solves the problem of accepting novel classes, lacks protection for the base class. In contrast, a better solution is to consider future additions and retention of growth space in the base class model. Therefore, another type of compatibility, omnidirectional compatibility, is more suitable for FSCIL, which is both backward and forward compatible.

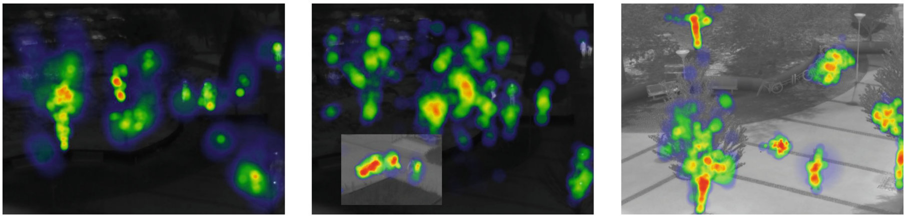

Fig. 2. This figure shows the visualization of the results, which are categorized into base class instances, virtual instances of base class patchUp and novel class instances. The classification ability on the base class instance can be transferred to the virtual instance with random Gaussian noise through the patchUp operation, and ultimately successfully transferred to the novel class instance.

Building upon Zhou's framework [15], we introduce Infrared (Omnidirectional) Compatibility Training Learning (CTL-I) for Infrared FSCIL. CTL-I establishes a model structure that ensures compatibility between new and existing classes. To facilitate seamless model updates, we pre-construct virtual prototypes, aligning with the count of newly introduced categories in the feature space. These virtual prototypes are derived from existing class prototypes. Through optimization, we not only condense the space occupied by same-class data, thus reserving more room for updated novel classes as depicted in Fig. 1, but also generate virtual instances corresponding to the virtual prototypes using the patchUp technique [7]. During the foundational training, the feature space of the virtual prototype is solely populated by virtual instances. Leveraging these instances, the model progressively learns classifiers effective for both new and base classes.

The main contributions of our CTL-I method can be concluded as follows:

1) We propose a new virtual prototype generation method that is compatible with base class and new class features in incremental learning models for few shot infrared classes, improving the classification accuracy of new classes and base classes.
2) We design a new Loss function, which can infer virtual prototype and improve the performance of incremental learning of infrared class with few lenses.
3) Our method has wide adaptability and excellent performance for various types of infrared images.

The rest of this article is organized as follows. The second section briefly introduces the few shot learning and class incremental learning in infrared images. In the third section, we provide a detailed description of the proposed CTL-I method. The fourth section introduces the comprehensive experiments on our infrared dataset. Finally, the fifth section provides the conclusion of this article.

2 Related Work

In this section, we first describe the specific methods of the few shot learning and class incremental learning in infrared images.

2.1 Few Shot Learning

Few-shot learning algorithms can be broadly categorized into three groups: data augmentation-based methods, fine-tuning-based methods, and meta-learning-based methods. Data augmentation-based approaches aim to generate additional data for novel classes, while fine-tuning-based methods focus on achieving rapid backward compatibility. On the other hand, meta-learning-based algorithms leverage extensive data to pre-train the backbone and employ suitable metrics to compute the distance between support and query sets [8]. These methods are collectively referred to as backward-compatible few-shot learning strategies.

Prototypical Network: The sample sizes of the base class and the novel class may differ, but they can be transformed into functional expressions, ensuring that each sample type has an equal number of specific metrics. ProtoNet [16] employs cross-entropy loss to train on base class data and subsequently adapts the same network for novel class data. This involves constructing embeddings using distinct depth feature sets $\phi(\cdot)$, resulting in the computation of average prototypes for each class. Notably, both novel and base classes adhere to the same $\phi(\cdot)$ parameters, thereby maintaining uniformity in parameter quantity and formulation.

$$p_i = \frac{1}{K} \sum_{j=1}^{|\mathcal{D}^b|} \mathbb{I}\left(y_j = i\right) \phi\left(\mathbf{x}_j\right) \tag{1}$$

$\mathbb{I}(\cdot)$ is the indicator function. The average embedding represents a classifier that stores the differences in the features of each class, i.e. $\mathcal{W}_i = \mathcal{P}_i$, which does not mean that the average embedding truly distinguishes between new and old classes, as well as different novel classes. The average embedding layer is nearly frozen throughout the entire novel class learning stage in order to maintain backward compatibility between new and old models. Therefore, it can be seen that the updated part is actually the backbone part, and the classifier is not significantly updated. This is a relatively conservative learning mode that prioritizes sacrificing the ability to learn novel classes. Knowledge Distillation and Prototypical Network approach backward compatibility from distinct angles. The Prototypical Network preserves the distinctiveness of old classes by aligning the new model with the established one, whereas Knowledge Distillation partially freezes the embedding layer to hinder excessive updates and replacements to the embedding layer associated with old classes. Both methodologies approach class incremental learning through the lens of backward compatibility. To elaborate further, their primary objective is to minimize alterations to the old class model while introducing new classes, ensuring stability and continuity throughout the update process.

However, these approaches did not account for the inherent trade-off between ensuring backward compatibility and preserving adaptability for new classes. This trade-off is evident when considering both base class accuracy and novel class accuracy.

Forward compatibility presents a fresh perspective to address this challenge. It involves preparing the model during the current training session to accommodate potential future updates.

2.2 Class Incremental Learning in Infrared Images

This novel perspective aims to address the challenge of learning from novel classes without forgetting old ones [9] within the context of infrared images. The currently available Class-Incremental Learning (CIL) algorithms can be broadly categorized into two groups. The first group focuses on assessing crucial parameters and safeguarding them against adverse updates [10,11]. The second group leverages techniques like knowledge distillation or meta-learning, blending novel and base class data to mitigate forgetting [12,13]. Notably, CEC [14] employs additional graph models to propagate contextual information among classifiers for adaptation. In the realm of CIL, compatibility learning was introduced [9] to enhance the model's backward compatibility, while FACT [15] represents the pioneering effort to address the forward compatibility challenge in the CIL model.

FACT: Inspired by the promise of forward compatibility, FACT endeavors to integrate forward compatibility techniques into the framework of Few-Shot Class-Incremental Learning (FSCIL) [15]. In essence, FACT adopts a dual-classifier approach involving two pre-constructed classifiers with identical structures but distinct update sequences. This configuration is reminiscent of the prototype method. During the foundational learning phase, the classifier's posterior probability is optimized to yield a bimodal distribution. Subsequently, a virtual prototype is formulated by amalgamating prototypes from the base class. The virtual prototype is then augmented by generating virtual data through instance mixing with base class data, facilitating virtual prototype updates. This augmented virtual prototype reserves a designated feature space to accommodate potential new classes.

$$
\begin{aligned}
\mathcal{L}_v(\mathbf{x}, y) &= \underbrace{\ell\left(f_v(\mathbf{x}), y\right)}_{\mathcal{L}_1} + \gamma \underbrace{\ell\left(\operatorname{Mask}\left(f_v(\mathbf{x}), y\right), \hat{y}\right)}_{\mathcal{L}_2} \\
\operatorname{Mask}\left(f_v(\mathbf{x}), y\right) &= f_v(\mathbf{x}) \otimes (\mathbf{1} - \operatorname{OneHot}(y)),
\end{aligned}
\tag{2}
$$

Forward compatibility methods operate under the premise that the base class shares a certain level of resemblance with the novel class, exhibiting either explicit or implicit similarities in their features. An analogy can be drawn from the similarity in deep features between kittens and puppies. When examined through the lens of feature space theory, the virtual prototype's feature representation for the base class is an approximation. Similarly, the features of the novel class must bear resemblance to the virtual prototype to facilitate a seamless transition from the old model trained on the base class to the updated new model enriched with novel class data. This ensures the retention of base class knowledge while facilitating the acquisition of novel class knowledge.

However, it's worth noting that forward compatibility has its limitations. If the disparity between the features of the novel class and the base class is substantial, it can result in a pronounced difference between the novel class and the virtual prototype. This disparity can subsequently constrain the learning potential of the novel class to some extent.

3 Proposed Method

3.1 Preliminary

Few shot incremental learning has two stages: base training session and incremental training session.

Base Training Session: In first stage, a model with random parameter initialization accepts a session 0 training set $D_{train}^{session=0} = \{(x_i, y_i)\}_{i=1}^{N_0, class_0}$ with sufficient base class instances firstly, and then evaluates it using the base class data testing set $D_{test}^{session=0} = \{(x_j, y_j)\}_{j=1}^{M_0, class_0}$. $D^{session=0}$ is called the base session. $x_i \in R^D$ is one of training data in class $y_i \in Y_0$. Y_0, which is the label set of session 0. Although x_i has a large amount of data, in order to maintain the compatibility of the model before and after the training session, it is necessary to maintain the same quantity of instances for each class. $x_j \in R^D$ is a testing instance of class $y_j \in Y_0$. Each type of test set and training set instance is divided by 7/3.The algorithm fits a model $f(x)$ to minimize the empirical risk over testing set.

$$\sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_{train}^{session=0}} \ell\left(f\left(\mathbf{x}_j\right), y_j\right) \tag{3}$$

where $\ell(\cdot, \cdot)$ represents the difference between the prediction and the ground-truth label, and the cross entropy loss function is used here. The output layer of the model is a Linear classifier and an embedding weight matrix: $f(x) = W^T\phi(x)$, where $\phi(\cdot) : R^D \longrightarrow R^d$. Both $\phi(x)$ and W^T are learnable parameters.

Incremental Session: In practical situations, novel classes usually surface incrementally with a limited number of instances. These instances are presented as a sequence of datasets $D^1, \cdots, D^B$, following a specific order. The practical approach of arranging the training categories' order is viable. We posit that access to a given dataset D^b is only possible during the training session corresponding to b. Alternatively, if the available number of new class categories falls short of the specified count, we can supplement the training data for that session with old class data. This mimics the chronological emergence of data and supports the intended order.

Each dataset contains a set of instances represented as $D^b = (x_i, y_i)_{i=1}^{NK}$, where $y_i \in Y_b$ belongs to the label space of the corresponding session b. The label spaces of different tasks are linearly independent, which means that if $b \neq b', Y_b \cap Y_{b'} = \emptyset$.

In order to handle limited instances, the dataset is structured in N-way K-shot format to avoid imbalanced sample sizes between new classes. Each dataset has N classes, and each class has K instances, which is also a commonly used small sample learning training setting.When using a new dataset, the model must effectively learn the new class while preserving its performance on the old class. The optimization goal of the model during the training process is to minimize the empirical risk of all training datasets.

$$\sum_{(\mathbf{x}_j, y_j) \in \mathcal{D}_t^0 \cup \cdots \mathcal{D}_t^b} \ell\left(f\left(\mathbf{x}_j\right), y_j\right) \tag{4}$$

3.2 Model of CTL-I

Through the lens of backward compatibility, the focus lies in safeguarding the integrity of base class information. However, the vantage point shifts when examining incremental learning from a forward compatibility standpoint-anticipating and accommodating future updates. Essentially, the learning process of base classes should remain 'unperturbed by what lies ahead.' By allocating embedding space for potential novel classes and projecting their potential patterns, the incurred costs of adaptation can be substantially minimized.

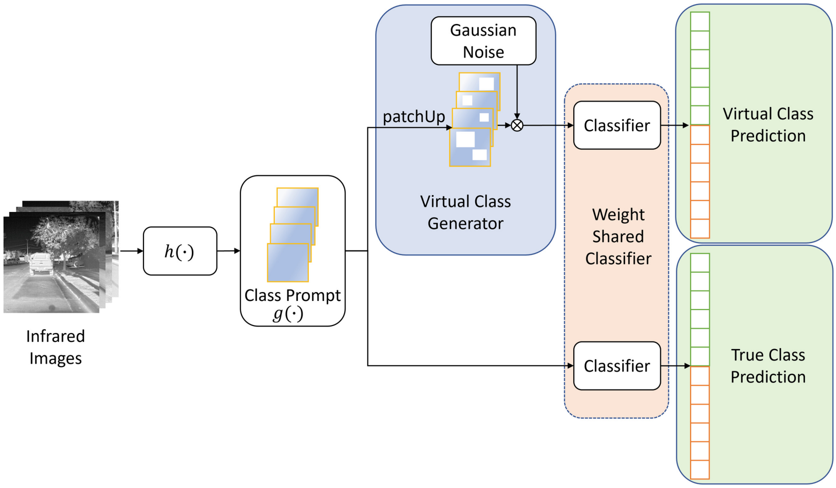

Fig. 3. CTL-I model Network architecture. The base class prediction is presented by green vector, base classifier. The virtual class prediction in Session 0 is presented by the orange vector, extending classifier. The novel prediction after Session 0 is presented by the orange vector. During the learning process, the base classifier and the extended classifier will be updated separately. (Color figure online)

Inspired by both two theories, our novel omnidirectional compatibility model, CTL-I, synergistically harnesses the benefits of both forward and backward compatibility. This model bifurcates the learning process for the base class and the novel class into two distinct and independent stages, as depicted in Fig. 3. This design ensures not only the preservation of the base class but also the effective acquisition of knowledge pertaining to the novel class. Consequently, a seamless transition through the update phase is facilitated, reducing the costs associated with adaptation and enhancing the processing capabilities for infrared images.

Initially, we introduce the approach of amalgamating virtual and real prototypes. Subsequently, we elucidate the process of utilizing these prototypes for logical inference.

3.3 Virtual Prototypes Assignment with Loss

In Session 0, we introduce the concept of metrics to gauge the resemblance between instance embedding and prototypes of the base class, denoted as $p(y \mid x) \propto sim\langle\omega_y, \phi(x)\rangle$. This similarity score reflects the likelihood that instance x belongs to class y, with a higher score indicating a stronger association. By employing prototypes, we can project K instances of novel classes into the deep feature space and aggregate them into a unified class representation. Similarly, base class instances are also aggregated, resulting in an equivalent number of clustering centers for novel and base classes. This approach effectively mitigates data imbalance concerns.

The process of calculating similarity involves convolving the prototype weights with instance depth features. In the context of Digital Signal Processing (DSP), this equates to filtering the instance's depth features using the prototype weight. A higher similarity value corresponds to more pronounced filtering outcomes.

To incorporate the concept of forward compatibility [15], we adopt distinct prototypes $|Y_0| \in D_0$ for virtual classes in the embedding space, augmented with Gaussian noise denoted as $noise_G$. These augmented prototypes, referred to as virtual prototypes P_v, correspond to virtual classes and are defined in $R^{d\times V}$, where V signifies the number of virtual classes. We represent the output of the current model as $f_v(x) = [W, P_v]^\top \phi(x)$, effectively establishing an embedded space for virtual classes. Consequently, the loss function adheres to Eq. 2, where $f_v(x) = [W, P_v]^\top \phi(x)$, and $\hat{y} = \text{argmax}\, v\boldsymbol{p}v^\top \phi(\mathbf{x}) + |Y_0|$ denotes the top logit for virtual class, which serves as the ground truth for the virtual prototype. Equation 2 encompasses both base class loss and ground truth. The second term involves a mask operation that minimizes the actual value and retains the remaining portion as a pseudo-label. By doing so, Eq. 2 reduces the impact of actual values on virtual classes while accentuating the influence of virtual labels.

Furthermore, to facilitate the update of the virtual prototype and enable its integration of novel knowledge, we introduce a fusion of base class instances into virtual instances, thereby enhancing the virtual prototype's capacity to assimilate new insights. Given that interpolation between base classes often lies far from the center of the two categories within the feature space, discerning the original classes can prove challenging. To address this, we employ the "patchUp" technique, segmenting the intermediate hidden layer output into blocks and interchanging them to meld the two instances.

The embedding layer $\phi(\cdot)$ is bifurcated into two components, $\phi(\cdot) = g\left(h\left(\cdot\right)\right)$. To bolster the virtual prototypes' adaptability to arbitrary novel classes, we introduce standard Gaussian noise, emulating the randomness inherent to novel classes:

$$\boldsymbol{z} = g\left[\lambda h\left(\mathbf{x}_i\right) + (1-\lambda)h\left(\mathbf{x}_j\right) + noise_G\right] \tag{5}$$

where λ denotes the hyperparameter governing the feature mixing ratio employed in patchUp. When combining n instance features to compose the features of a virtual prototype, the resulting representation of the virtual prototype can be expressed as follows:

$$\boldsymbol{z} = g\left[\left\|\sum_{i=1}^{n} \lambda_i h\left(\mathbf{x}_i\right)\right\|_2 + noise_G\right], \lambda = \{\lambda_1, \ldots, \lambda_n \mid \lambda_n \in (0,1)\} \tag{6}$$

where $\|\cdot\|_2$ signifies the utilization of 2-norm normalization for two-dimensional vectors. This normalization technique is employed to mitigate potential instability stemming from excessively large or minute random parameters. Its purpose is to strike a balance between the virtual prototype and the random noise, preventing one from overpowering the other. Given the low resolution of infrared images and the pronounced similarity across different categories, training models to emphasize the learning of more akin category features leads to improved testing outcomes.

We can build a loss function for virtual instance $\boldsymbol{z}$ to reserve embedded space:

$$\mathcal{L}_f(\mathbf{x}, y) = \underbrace{\ell\left(\mathrm{Mask}\left(f_v(\mathbf{z}), \hat{y}\right), \hat{\hat{y}}\right)}_{\mathcal{L}_3} \tag{7}$$

Equation 7 serves to strike a balance between mixed instances from known classes and the introduction of randomness through virtual classes. This equilibrium is crucial to avoid known classes from unduly constraining the adaptability of virtual classes to novel ones, effectively mitigating the risk of overfitting. The final loss formulation combines the contributions from Eq. 2 and Eq. 7, yielding $\mathcal{L} = \mathcal{L}_v + \mathcal{L}_f$. By introducing heightened randomness to virtual classes, the model gains a more efficient capability to simulate future instances, thereby enhancing the implementation of forward compatibility compared to the approach presented in [15].

To encompass the representation of both virtual and known classes within the model, we expand the classifier as follows: $C = [C_{base}; C_{novel}], novel = N * B]$. Here, C_{base} signifies the classifier dedicated to base classes, while C_{novel} pertains to the classifier tailored for novel classes. During Session 0, the update of C_{novel} is driven by virtual class data, while C_{base} is refined through real data from D^0.

3.4 Compatibility Update

We have elucidated the process of generating virtual prototypes and their impact on the model's grasp of base classes. Now, let's delve into the strategy for substituting acquired virtual data with data from new classes in subsequent updates.

During the acquisition of each batch of novel class data, we adhere to Eq. 1 for extracting class prototypes and populating the extended classifiers. To be specific, in Session 1, we utilize D^1 to update $C_{novel}(1, N)$, while concurrently maintaining the immutability of the C_{base} branch. Subsequent sessions follow a similar update procedure.

Freezing the base class classifier during the update with novel class data is a characteristic trait of backward compatibility methodologies. This approach serves to preserve the base class's discriminative capability. To harness the benefits of backward compatibility, we will introduce novel class classifiers alongside the ongoing training sessions. In a novel class session, both the base class classifier and embedding layer will coexist, effectively maintaining their frozen state. Leveraging the inclusion of virtual classes in the base class training phase allows us to attain a more comprehensive information distribution for $\phi(x)$, thus aiding the model in assimilating a greater volume of information during the incremental phase. This proposition is corroborated through testing.

Our approach capitalizes on the strengths of both backward compatibility and forward compatibility techniques, intelligently circumventing their individual limitations. In doing so, we strike a harmonious equilibrium between the precision of acquiring novel class knowledge and safeguarding the integrity of base classes.

4 Experiments

In this section, we present a comprehensive comparison of CTL-I with state-of-the-art methods on our infrared Few-Shot Class-Incremental Learning (FSCIL) dataset. Through extensive experimentation, we validate the efficacy of omnidirectional compatibility training and provide visual insights into the incremental process of CTL-I. Moreover, we introduce novel evaluation metrics for incremental learning, encompassing base accuracy and novel accuracy. These metrics offer a more nuanced perspective compared to the conventional full class accuracy, as they better capture the learning dynamics of both novel and base classes.

4.1 Implementation Details

Dataset: After benchmark setting [19], we evaluated the performance on our infrared street view dataset. Our infrared dataset contains 22000 images from 40 classes.

Dataset Split: For our infrared dataset, 40 classes are divided into 10 base classes and 30 novel classes. The base class is divided into 10-way 30-shot tasks, while the novel class is designated as four 5-way 5-shot incremental tasks. For fair comparison, we use the same training segmentation for each comparison method [19] (including basic and incremental sessions). The test set is sampled from the current training category, and the same session uses the same test set for overall evaluation.

Compared Methods: We first compared the classic CIL methods iCaRL [17]. In addition, we also compared the current SOTA FSCIL algorithms: SPPR [5], CEC [14], method[15]. The baseline method we reported is 'fine-tuning' by fine-tuning the model with a few instances of shots.

Training Details: All models are deployed using PyTorch [18]. We use the same backbone [19] for all comparison methods. For all datasets, we use ResNet18. The model is trained with a batch size of 256 for 600 epochs and optimized using SGD with driving capacity. The learning rate starts at 0.1 and decays with cosine annealing.

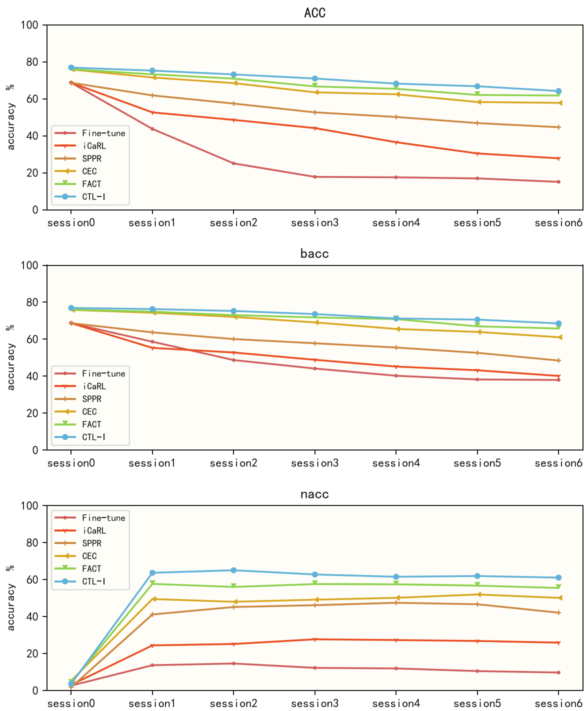

Fig. 4. Top-1 accuracy, bacc and nacc of each incremental session.We have shown the legend in (a) and annotated the performance gap between CTL-I and the runner up method after the last session at the end of each curve. The specific values are shown in Table 1 and the supplementary table.

Evaluation Protocol: After [19], we represent the Top-1 accuracy after the i-th session as A_i. We set the base class accuracy (bacc) and the novel class accuracy (nacc) to evaluate them separately. We also quantitatively measured the forgetting phenomenon using performance drop rate (bPD), which is $bPD = bA_0 - bA_B$, where bA_0 represents the accuracy after the basic session and bA_B represents the accuracy after the last session,visualized in Fig. 2. We calculated bPD for the base class.

4.2 Benchmark Comparison

We report the performance curves on the benchmark dataset [19] (i.e. our infrared dataset) in Fig. 4 and Table 1.

Table 1. The acc, bacc, and nacc of each incremental session on the our infrared dataset. The results of the comparison method are cited from [19] and [14].

Method	Acc	Accuracy in each session (%)							bPD ↓
		0	1	2	3	4	5	6	
Fine-tune	acc	68.69	43.71	25.08	17.80	17.56	16.96	15.10	30.71
	bacc	68.69	58.66	48.73	44.15	40.26	38.24	37.98	
	nacc	2.7	13.73	14.65	12.30	11.98	10.55	9.78	
iCaRL	acc	68.69	52.65	48.61	44.16	36.52	30.46	27.83	28.55
	bacc	68.69	55.32	52.82	48.87	45.22	43.21	40.14	
	nacc	3.1	24.45	25.21	27.74	27.34	26.84	25.94	
SPPR	acc	68.69	61.85	57.43	52.68	50.19	46.88	44.65	20.19
	bacc	68.69	63.71	60.09	57.82	55.51	52.64	48.50	
	nacc	1.8	41.17	45.20	46.18	47.50	46.69	42.14	
CEC	acc	75.86	71.54	68.48	63.52	62.45	58.29	57.80	14.76
	bacc	75.86	74.33	72.08	69.10	65.56	63.96	61.10	
	nacc	5.3	49.51	48.02	49.16	50.16	51.98	50.16	
FACT	acc	76.0	73.26	70.9	66.7	65.42	62.10	61.78	10.20
	bacc	76.0	74.89	73.02	71.80	70.86	66.97	65.80	
	nacc	4.28	57.71	56.05	57.60	57.46	56.76	55.50	
CTL-I(our)	acc	76.9	75.25	73.2	70.98	68.23	66.79	64.21	**8.34**
	bacc	76.9	**76.31**	**75.28**	**73.60**	**71.24**	**70.60**	**68.56**	
	nacc	3.65	**63.71**	**65.08**	**62.80**	**61.56**	**61.96**	**61.10**	

We can infer from Fig. 4 that CTL-I consistently outperforms the current SOTA method (i.e. FACT) by a margin of 2–4%, demonstrating its clear superiority. The subpar performance of the CIL method underscores its limitations in handling novel class compatibility. Moreover, CTL-I's superiority over the FSCIL approach reinforces the significance of considering forward compatibility for improved results.

Moreover, CTL-I surpasses the forward compatibility method, which leans towards compatibility with virtual novel classes. This underscores that omnidirectional compatibility is better aligned with FSCIL requirements. Furthermore, CTL-I outperforms FACT, which approaches forward compatibility from a different angle. Collectively, our training strategy is better suited for FSCIL. Detailed results for our infrared dataset are presented in Table. 1. Notably, CTL-I demonstrates minimal degradation in bPD metrics for both visible and infrared data. This highlights the effectiveness of omnidirectional compatibility training in combating base class forgetting during FSCIL, all the while mastering novel classes with high precision.

In summary, CTL-I always handles small-scale FSCIL tasks with SOTA performance.

4.3 Further Analysis

The performance metrics encompass precision evaluations spanning both old and novel classes. To delve into novel class learning and the resilience against base class forgetting, we present base class and novel class accuracies across sessions, along with their harmonic mean, for our infrared dataset. These measurements adhere to the identical benchmark testing conditions. It is evident that CTL-I exhibits superior performance in novel class learning, thereby validating the efficacy of omnidirectional compatibility training and its adeptness in effectively mastering infrared data.

5 Conclusion

FSCIL capability stands as the initial stride into open-world learning in Infrared images, demanding models to grasp new classes with limited data while retaining knowledge of old classes. This article emphasized the significance of developing an FSCIL omnidirectional compatible model. We devised a virtual prototype incorporating random noise for forthcoming new classes within the classifier, harmonizing the feature space between new and old classes from dual perspectives. This approach, which preserves the attributes of old classes, promotes a balanced growth between novel and old classes, consequently mitigating performance deterioration during updates. Surprisingly, virtual prototypes integrated into the embedding space unexpectedly enhance FSCIL's learning performance for base classes and effectively curtail overfitting concerns. CTL-I effectively integrates new knowledge into old models and achieves SOTA performance.

Acknowledgments. This work was funded by the STI 2030-Major Projects under grant 2022ZD0209600, the National Natural Science Foundation of China under grant 62201058, the Beijing Institute of Technology Research Fund Program for Young Scholars under grant 6120210047, and the China Postdoctoral Science Foundation under grant 2021M700399.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
2. Gomes, H.M., Barddal, J.P., Enembreck, F., Bifet, A.: A survey on ensemble learning for data stream classification. ACM Comput. Surv. **50**, 23:1–23:36 (2017). https://doi.org/10.1145/3054925

3. Chou, Y.Y., Lin, H.T., Liu, T.L.: Adaptive and generative zero-shot learning. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=ahAUv8TI2Mz
4. French, R.M.: Catastrophic forgetting in connectionist networks. Trends Cogn. Sci. **3**(4), 128–135 (1999). https://doi.org/10.1016/S1364-6613(99)01294-2, https://www.sciencedirect.com/science/article/pii/S1364661399012942
5. Zhu, K., Cao, Y., Zhai, W., Cheng, J., Zha, Z.J.: Self-promoted prototype refinement for few-shot class-incremental learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6797–6806 (2021). https://doi.org/10.1109/CVPR46437.2021.00673
6. Nagarakatte, S., Bb, A., Martin, M., Zdancewic, S.: SoftBound: highly compatible and complete spatial memory safety for c. ACM SIGPLAN Notices **44**, 245 (2009). https://doi.org/10.1145/1543135.1542504
7. Faramarzi, M., Amini, M., Badrinaaraayanan, A., Verma, V., Chandar, S.: PatchUp: a feature-space block-level regularization technique for convolutional neural networks. arXiv e-prints arXiv:2006.07794 (2020). https://doi.org/10.48550/arXiv.2006.07794
8. Liu, B., et al.: Negative margin matters: understanding margin in few-shot classification. arXiv e-prints arXiv:2003.12060 (2020). https://doi.org/10.48550/arXiv.2003.12060
9. Bansal, G., Nushi, B., Kamar, E., Weld, D.S., Lasecki, W.S., Horvitz, E.: Updates in human-AI teams: understanding and addressing the performance/compatibility tradeoff. In: Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 2429–2437 (2019). https://doi.org/10.1609/aaai.v33i01.33012429, https://ojs.aaai.org/index.php/AAAI/article/view/4087
10. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: learning what (not) to forget. arXiv e-prints arXiv:1711.09601 (2017). https://doi.org/10.48550/arXiv.1711.09601
11. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. CoRR abs/1612.00796 (2016). http://arxiv.org/abs/1612.00796
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv e-prints arXiv:1503.02531 (2015). https://doi.org/10.48550/arXiv.1503.02531
13. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Trans. Pattern Anal. Mach. Intell. **40**(12), 2935–2947 (2018). https://doi.org/10.1109/TPAMI.2017.2773081
14. Zhang, C., Song, N., Lin, G., Zheng, Y., Pan, P., Xu, Y.: Few-shot incremental learning with continually evolved classifiers. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12450–12459 (2021). https://doi.org/10.1109/CVPR46437.2021.01227
15. Zhou, D.W., Wang, F.Y., Ye, H.J., Ma, L., Pu, S., Zhan, D.C.: Forward compatible few-shot class-incremental learning. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9036–9046 (2022). https://doi.org/10.1109/CVPR52688.2022.00884
16. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. arXiv e-prints arXiv:1703.05175 (2017). https://doi.org/10.48550/arXiv.1703.05175
17. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: iCaRL: incremental classifier and representation learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5533–5542 (2017). https://doi.org/10.1109/CVPR.2017.587

18. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. arXiv e-prints arXiv:1912.01703 (2019). https://doi.org/10.48550/arXiv.1912.01703
19. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12180–12189 (2020). https://doi.org/10.1109/CVPR42600.2020.01220