Journal of Artificial Intelligence Research 82 (2025) 1807-1843 Submitted 09/2024; published 03/2025

Adaptive Few-Shot Class-Incremental Learning via Latent Variable Models

Tameem Adel TAMEEM.HESHAM@GMAIL.COM
National Physical Laboratory, Maxwell Centre, University of Cambridge
JJ Thomson Avenue, Cambridge, CB3 0HE, United Kingdom

Abstract

Approaches to class-incremental learning aim to successfully learn from continuously arriving classes. One added level of difficulty usually arises when the training data belonging to each class is scarce, which is the case in several open-world machine learning applications. In this paradigm, which is referred to as few-shot class-incremental learning, a typical learner needs to both be able to learn incrementally from the sequentially arriving classes, and preserve the knowledge which already exists about the old (i.e. already existing) classes. We propose a few-shot class-incremental learner which adapts the representations of the new few-shot classes as well as relevant previous knowledge based on a latent variable model. The proposed latent variable model is a form of a variational autoencoder that is designed to address the main challenges of the few-shot class-incremental learning paradigm, namely catastrophic forgetting and potential bias. During the few-shot learning of new classes, the amortization and high fidelity characteristics of the proposed model are leveraged to adapt not only the current class, but also the relevant previously encountered classes, in order to consistently mitigate the impact of catastrophic forgetting, bias and overfitting. We also derive a generalization upper bound on the error of an upcoming class. Experiments on several widely used few-shot class-incremental learning benchmarks, as well as a medical benchmark consisting of real-world medical images, demonstrate that the proposed model leads to improved performance, as measured by average overall and final classification accuracy, and in terms of alleviating catastrophic forgetting.

1. Introduction

Incremental learning, sometimes called continual learning or lifelong learning, refers to a machine learning paradigm where the knowledge obtained from previous tasks should be accumulated and potentially reused in the future (Ring, 1995; Srivastava et al., 2013; Schwarz et al., 2018; Hu et al., 2019; Adel et al., 2020; Wang et al., 2023a; Zhu et al., 2023a). This is particularly challenging for deep models whose ideal setting depends on large sizes of training data being available a priori, i.e. prior to the beginning of the training procedure. Incremental learning (IL) can be beneficial due to its time and resource (e.g. memory) management advantages over the alternative of having to retrain the model from scratch upon the arrival of new data. One of the standard assumptions in IL is that, both at the training time and the inference (test) time, the learner has access to the identity (ID) of the task to which each class belongs. In most real-world scenarios, this assumption can be impractical due to the likely non-availability of the task ID information at the inference time. To that end, the class-incremental learning (CIL, Belouadah & Popescu, 2019; Hou et al., 2019; Yu et al., 2020; Mai et al., 2021; Shim et al., 2021; van de Ven et al., 2021; Zhu et al., 2021; Masana et al., 2022; Liu et al., 2023; Rymarczyk et al., 2023; Wen et al., 2023; Zhou et al., 2023a) paradigm has arisen as a viable approach where the assumption that no task information is available at the inference time forces the learner to aim at distinguishing between all classes encountered so far, regardless of

their respective task identities. The CIL paradigm depicts a typical scenario which commonly takes place in open environments. Hence, addressing the challenges imposed by the CIL paradigm is of paramount significance in the quest to deploy reliable machine learning in open-world environments (Zhou, 2022).

The above problem is compounded when solely a scarce amount of data is available for every class. This setting is referred to in the literature as few-shot class-incremental learning (FSCIL, Rebuffi et al., 2017; Gidaris & Komodakis, 2018; Tao et al., 2020; Achituve et al., 2021; Ahmad et al., 2022; Peng et al., 2022; Song et al., 2023; Wang et al., 2023b; Zhou et al., 2023a; Zhao et al., 2024). For instance, think of a pedestrian attribute recognition model that is used for video surveillance where it is required to identify the characteristics of human appearance attributes such as age, gender and clothes (Xiang et al., 2019; Wang et al., 2023b). The class-incremental learner encounters the problem of classifying waistcoats (i.e. identifying pedestrians wearing waistcoats) vs. T-shirts as the first task. Given their rather different characteristics, it might not be so tricky to identify waistcoats during the first task. The class-incremental learner then encounters few new pedestrians. Each one of them wears either a jacket or a coat where the latter two garments constitute the two newly arriving classes. At this point, the learner's mission becomes more complicated since the new requirement is to address the challenge of differentiating between a waistcoat, a jacket and a coat during the inference time, without ever having the luxury to train on data belonging to all of such classes simultaneously.

Given the fact that it is typically prohibited for an FSCIL model to access data belonging to the previous classes, due to privacy and security constraints, a few-shot class-incremental learner should be capable of learning new classes without forgetting the previous (i.e. already learned) classes. A compromise must therefore be achieved between adapting to new classes and accomplishing a degree of stability that is sufficient to preserve the knowledge which has already been acquired about the previous classes. Unrestricted adaptation to the new classes can potentially result in the catastrophic forgetting of previous classes. In this context, catastrophic forgetting (McCloskey & Cohen, 1989; Ratcliff, 1990; Robins, 1993, 1995; French, 1999; Pape et al., 2011; Srivastava et al., 2013; Achille et al., 2018; Diaz-Rodriguez et al., 2018; Kemker et al., 2018; Zeno et al., 2018; Parisi et al., 2019; Pfulb & Gepperth, 2019; Ebrahimi et al., 2020; Gupta et al., 2020; Banayeeanzade et al., 2021; Ke et al., 2021; Ostapenko et al., 2021; Wang et al., 2021; Kalb & Beyerer, 2022; Kang et al., 2022a; Karakida & Akaho, 2022; Lin et al., 2022; Miao et al., 2022; Yasar & Iqbal, 2023) refers to the phenomenon when a learner already gains the necessary knowledge about (how to recognize) a certain class, then such knowledge eventually degrades upon encountering new classes. The aforementioned compromise is often referred to as the stability-plasticity dilemma; stability refers to the ability to maintain previous knowledge, whereas plasticity denotes the ability to adapt to new classes. Furthermore, FSCIL is also subject to additional challenges as a direct result of learning from few data samples per class, which are bias and overfitting.

We propose a few-shot class-incremental learner which aims at mitigating the above risks via establishing a latent variable model that is bespoke to fit the characteristics of the FSCIL paradigm. It has been demonstrated in previous works, e.g. (Wang et al., 2023b; Zhao et al., 2024), that adapting the representations of the classes which arrive during the few-shot sessions does not suffice to mitigate the risks of catastrophic forgetting and bias resulting from the scarcity of data during such sessions. As such, our proposed latent variable model is designed in a manner that permits the adaptation of not only the new few-shot classes, but also the previously encountered classes which are most likely to be impacted by the new classes. Hence, this strategy aims to establish a model which can

consistently balance the stability-plasticity tradeoff with the risks of potential bias and overfitting (which are fundamental in FSCIL), by controlling all the classes which are likely to be impacted upon the arrival of every new class.

The generalization upper bound that we derive on the error of an upcoming class supports our postulate regarding the added value of adapting the newly arriving few-shot classes along with relevant previous classes. In addition, we perform experiments on several widely used CIL benchmarks as well as the medical benchmark referred to as MedMNIST (Section 4). The performed experiments illustrate the capability of the proposed FSCIL model to achieve state-of-the-art results with respect to the overall classification performance, final classification performance and mitigating catastrophic forgetting. We also provide details of related work in Section 5.

Our main contributions can be summarized as follows:

1. An adaptive latent variable modeling-based FSCIL framework which does not only address the new classes encountered during the few-shot sessions, but also accordingly adjusts the knowledge previously acquired from the relevant previous classes (Section 2).

2. The proposed framework achieves a balanced treatment of the tradeoff between mitigating bias and catastrophic forgetting, and adapting to new classes.

3. A generalization upper bound on the error of an upcoming class (Section 3).

4. State-of-the-art results on the most widely used CIL benchmarks, as well as a real-world benchmark in the form of the medical dataset referred to as MedMNIST. These results are measured by metrics denoting classification accuracy (average overall, final, and after individual sessions) as well as metrics evaluating the degree of mitigating catastrophic forgetting (Section 4).

2. Our *CIAM* Approach

Given the several challenges a few-shot class-incremental learner should address, it is important for the proposed framework to first utilize the abundant data available during the base session to learn a representation of the base classes. Afterwards, it becomes crucial to adapt the learned representations to the few-shot classes while mitigating the potential risks of catastrophically forgetting the previously encountered classes, bias and overfitting. We propose a model, which we refer to as few-shot Class-Incremental learning Adaptation via latent variable Models (*CIAM*), to address such challenges of few-shot class-incremental learning.

2.1 Setup

We address a few-shot class-incremental learning (FSCIL) setup (Tao et al., 2020; Zhao et al., 2021; Yang et al., 2022; Wang et al., 2023b; Zhao et al., 2023; Zhou et al., 2023b) where the learner encounters a stream of m sequentially arriving sessions. Each session comprises a labelled training dataset $D_t = \{\boldsymbol{x}_t^n, \boldsymbol{y}_t^n\}_{n=1}^{N_t}$, where $t \in \{1, 2, \ldots, m\}$ is the session index and N_t is the size of the training dataset of session t. Data points are depicted by $\boldsymbol{x} \in \mathcal{X}$ and corresponding labels $\boldsymbol{y} \in \mathcal{L}_t$, where $\mathcal{X}$ is the input space. The set of classes (i.e. label space) of the t^{th} session is referred to as $\mathcal{L}_t$. Note that the sets of classes belonging to different sessions are disjoint, that is $\forall i, j \in [1,\ m]$ and $i \neq j$, $\mathcal{L}_i \cap \mathcal{L}_j = \varnothing$. During the training procedure for each session t, solely the corresponding training data D_t is available. Due to privacy constraints, data belonging to the previous sessions are

not available. Also note that D_1 is the large-scale training dataset consisting of many base classes and with relatively abundant data per class (compared to the subsequent sessions, $t > 1$), whereas $D_t,\ t > 1$, refers to few-shot training datasets with very scarce data per class. In other words, for sessions after the first session, $t > 1$: $N_1 >> N_t$. Furthermore, for such sessions $t > 1$, the amount of available data per class in the FSCIL setting is much more limited than the corresponding sessions in a vanilla CIL setting. For the incremental few-shot sessions, $D_t, t > 1$, the C-way K-shot FSCIL setting refers to the fact that the respective session comprises C classes and K training data points per class. This means that, for a few-shot session $t, t > 1$, the total size of the training dataset is $N_t = C \times K$. Refer to the test data and test labeling space belonging to a session t as D_t^{test} and $\mathcal{L}_t^{test}$, respectively. As is the case with all CIL frameworks, the model is tested on *all* the classes encountered by the learner so far during training (i.e. not only the classes of the current session). In other words, $\mathcal{L}_i^{test} = \bigcup_{j=1}^{i} \mathcal{L}_j$.

2.2 Base Session

Our *CIAM* framework consists of two models: a base network and a few-shot latent variable model. The base network learns representations of the base classes where it can relish the abundant data available during the base session. On the other hand, we develop a setting for both the base and few-shot sessions which enables our proposed latent variable model to adapt the base representations to the few-shot classes while mitigating the risks of catastrophic forgetting and potential bias. Let's focus here on the former model, i.e. the base network. At the bottom of the base network architecture, there is a feature extractor consisting of several nonlinear layers which learn a feature representation $f(\cdot; \boldsymbol{\omega})$ with a parameter set $\boldsymbol{\omega}$, defined on a feature space $\mathcal{F}$. The top part of the architecture of the base network comprises a label predictor in the form of a classification head with parameters $\boldsymbol{\nu}$ which produce output logits for every base class. Note that this predictor is solely used to train the base classes. In other words, this is not the classifier that is set to be used during the test phase of the proposed *CIAM* framework. For the base (initial) session with $t = 1$, we train the base network on the base training dataset D_1 with the cross-entropy loss which is denoted as follows:

$$J(\boldsymbol{x}_1^n, \boldsymbol{y}_1^n; \boldsymbol{\omega}, \boldsymbol{\nu}) = L_{CE}(\mathbf{O}(\boldsymbol{x}_1^n; \boldsymbol{\omega}, \boldsymbol{\nu}),\ \boldsymbol{y}_1^n), \tag{1}$$

where L_{CE} denotes the cross-entropy loss, $\mathbf{O}$ refers to the final output vector given input $\boldsymbol{x}_1^n$, $n = \{1, 2, \ldots, N_1\}$, and N_1 is the overall size of the training dataset in the base session.

For the incremental few-shot sessions $t > 1$, we no longer perform the aforementioned training procedure on the base network. Continuing to train the base network on the scarce data available in the few-shot sessions would lead to the detrimental impact of forgetting the base classes. This is the main reason why we stop training the network after the base session. Instead, during the few-shot sessions, we establish a robust Bayesian latent variable model to adapt the representations of the new and relevant previous classes such that catastrophic forgetting and potential bias can be mitigated.

2.3 Few-Shot Adaptation Based on Latent Variable Modeling

In this section, we describe the proposed few-shot adaptation procedure. As mentioned in Section 2.2, after the base session (where data was abundant), continuing to train the base network for the sake of learning the few-shot classes from their scarce data would be detrimental, and it can result in forgetting the already established knowledge about the base classes. On the other hand, not performing any learning at all from the few-shot data would signify a waste of potential knowledge

and a risk of underfitting. To that end, we propose to adapt the learned representations during the few-shot sessions via a latent variable model that can adapt both the few-shot classes and the relevant previous classes such that we can enforce a consistent update throughout (rather than solely updating the new few-shot classes). By adopting the proposed comprehensive adaptation strategy, we aim to capture all the potential sources of catastrophic forgetting and/or bias.

Recall that the network has learned a function representation $f(\cdot;\boldsymbol{\omega})$ which is a representation of a better predictive potential (i.e. it is optimized such that the respective classes can be accurately predicted). This representation has been exclusively learned on the base classes. There is a need to: i) adapt this representation to the few-shot classes, and to ii) ensure that this adaptation is aligned with the relevant classes, among the previously encountered classes. The gist here is to establish a latent variable model that can efficiently learn to adapt the representation of the current few-shot class as well as the relevant previous classes (whose data are no longer available), while avoiding the need to re-train the base network after the base session.

Several FSCIL algorithms completely freeze the representation learning and feature extraction after the base session (Zhang et al., 2021; Akyurek et al., 2022; Hersche et al., 2022; Wang et al., 2023b). Other previous works on FSCIL are based on forming prototypes of the few-shot classes (Mazumder et al., 2021; Shi et al., 2021; Zhou et al., 2022; Ji et al., 2023) or, at best, calibrating such prototypes (Zhu et al., 2021; Deng et al., 2022; Wang et al., 2023b; Zhang & Gu, 2023; Zhou et al., 2023b; Zhu et al., 2023b; Zhao et al., 2024). However, it can be challenging to directly obtain reliable prototypes in the few-shot sessions without applying any learning nor adaptation. We conjecture that even though this can help fitting the new few-shot classes, it can still lead to the adverse effect of having a negative impact on the previously learned classes. To that end, one further phase is needed between the feature representation learned during the base session and the ultimate few-shot classification (Zhao et al., 2024). We aim to utilize the few-shot data at our disposal to adapt *all* the potentially affected class representations.

For each base class, the feature representation function $f(\cdot;\boldsymbol{\omega})$ learned during the base session provides a mapping from the respective input $\boldsymbol{x}$ to $f(\boldsymbol{x};\boldsymbol{\omega})$. This representation has not been trained at all after the base session (for the reasons mentioned above). This is why directly expressing the few-shot classes in terms of this representation function (even in case of eventual calibration) is not fit for purpose. Instead, we aim to efficiently adapt the few-shot classes along with the previously encountered classes which are considered the most relevant to the current few-shot class. Refer to the number of classes of the base session as B. As mentioned before, the number of base classes B is much larger than the number of classes in any single few-shot session that follows. During the few-shot sessions, one must select the previous classes which are the most likely to be affected by the newly arriving classes since it can otherwise become prohibitively expensive to update the representations of every previous class when encountering a new few-shot class.

First, in order to enable the proposed latent variable model to understand the base classes (such that it can eventually adapt their representations), it learns a latent space of the base representations right after the end of the base session, and prior to encountering any few-shot session. After the base session, i.e. for $t > 1$, the first step in adapting a few-shot representation is to identify the most relevant classes, among the previously encountered classes, to the current few-shot class. This is important such that our updates can be focussed solely on the subset of classes which are the most similar to the current few-shot class, i.e. on the classes where such updates are deemed necessary. In addition, this is also beneficial to ensure a maximally efficient utilization of the latent variable model during the adaptation procedure. To select the most relevant previous classes, we use a Gaussian

kernel to assign weights to every previously encountered class based on their similarity to the current few-shot class. Corresponding to a base class with index $i, i \in \{1, 2, \ldots, B\}$, refer to the mean of the base network representations of the data points of this base class as $\boldsymbol{\mu}_{\mathbf{f_i}}$. Recall that we adopt a C-way K-shot few-shot setting, which means that the size of an arbitrary few-shot session is C classes. Assume that the universal indices for the few-shot classes i come in a consecutive fashion after the base classes, that is, the range $B < i <= B + C \times (m-1)$ spans all the universal indices for the few-shot classes (from session 2 up until session m). Our Gaussian kernel can therefore be described as follows:

$$k(\boldsymbol{\mu}_{\mathbf{f_i}}, \boldsymbol{\mu}_{\mathbf{f_j}}) = \exp\left(\frac{-\|\boldsymbol{\mu}_{\mathbf{f_i}} - \boldsymbol{\mu}_{\mathbf{f_j}}\|^2}{2\sigma^2}\right), \tag{2}$$

$1 <= j < B + C \times (m-1)$, $B < i <= B + C \times (m-1), i > j$, σ is the kernel width.

The main objective of the proposed FSCIL latent variable model is to adapt the learned representations of the current few-shot class as well as the most relevant previous classes, as a means to mitigate the potential risks of catastrophic forgetting and bias. The proposed FSCIL latent variable model consists of a two-branched variational autoencoder (VAE, Kingma & Welling, 2014; Kingma et al., 2014). Compared to a vanilla VAE which consists of solely one branch (connecting the data space and the latent space), the additional branch we introduce here is focussed on the predictive accuracy of the resulting representations where the latent space is connected to the class labels. As such, the simultaneous optimization of the two VAE branches can adapt all the relevant class representations while simultaneously optimizing for a high overall performance.

2.4 Inference on the Latent Variable Model

Normalizing flows (Tabak & Vanden-Eijnden, 2010; Tabak & Turner, 2013; Rezende & Mohamed, 2015; Kingma et al., 2016; Adel et al., 2018) are utilized to transform the base representation f into the latent representation $\mathbf{z}$. Normalizing flows are powerful transformations which are capable of establishing flexible posterior distributions through a chain of mappings. Each mapping is an invertible transformation. The resulting series of mappings, which is referred to here as $g_{\mathbf{s}},\ \mathbf{s} = \mathbf{1}, \ldots, \mathbf{S}$, is applied to the VAE inputs f and $\boldsymbol{y}$ such that the resulting latent representation $\mathbf{z}$ is adapted and better optimized for predictive accuracy. The index $\mathbf{s}$ refers to a transformation step, out of a total of $\mathbf{S}$ steps. Denote by $\mathbf{z_0}$ an initial random variable with density $\mathbf{q_0}(\mathbf{z_0})$ which, along with the VAE inputs, go through a series of successive transformations $g_{\mathbf{s}}$ that are expressed as follows:

$$\mathbf{z_s} = \mathbf{g_s}(\mathbf{z_{s-1}}, f, \boldsymbol{y}), \quad \mathbf{s} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{S}, \quad \mathbf{z} = \mathbf{z_S}. \tag{3}$$

Given that the determinant of the Jacobian of every transformation, $\mathbf{det}(\mathbf{g_s})$, can be computed, the probability density function $\mathbf{z} = \mathbf{z_S}$ can be computed straightforwardly (Rezende & Mohamed, 2015; Liao & He, 2021). Denote by $\mathbf{q}$ the normalizing flow-based approximation of the true posterior, the variational probability density distribution $\mathbf{q}(\mathbf{z}|f, \boldsymbol{y})$ can be described as follows:

$$\log \mathbf{q_S}(\mathbf{z_S}|f, \boldsymbol{y}) = \log \mathbf{q_0}(\mathbf{z_0}|f, \boldsymbol{y}) - \sum_{\mathbf{s=1}}^{\mathbf{S}} \log \left|\det \tfrac{d\mathbf{z_s}}{d\mathbf{z_{s-1}}}\right|, \quad \mathbf{s} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{S} \tag{4}$$

$$\mathbf{z} = \mathbf{z_S}. \tag{5}$$

Hence, the mapping from the VAE inputs, which comprise the base representation f and the labels $\boldsymbol{y}$, to the latent representation $\mathbf{z}$ can be expressed as follows:

$$\mathbf{z} = \mathbf{g_S} \circ \mathbf{g_{S-1}} \circ \ldots \circ \mathbf{g_1}(f, \boldsymbol{y}), \tag{6}$$

where each map $g_{\mathbf{s}},\ \mathbf{s} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{S}$, is a planar flow (Papamakarios et al., 2021):

$$\mathbf{g_s}(\mathbf{z_{s-1}}) = \mathbf{z_{s-1}} + \mathbf{uh}(\mathbf{w^T z_{s-1}} + \mathbf{b}), \tag{7}$$

where $\mathbf{w^T}$ is the transpose of $\mathbf{w}$, $\mathbf{b}$ is a scalar, $\mathbf{u}$ and $\mathbf{w}$ are vectors, and $\mathbf{h}$ is a nonlinearity. We opt for planar flows here since they are the least prone to overfitting, as per our FSCIL setting.

Our VAE is illustrated in Figure 1. To perform inference on our VAE, we define the generative model and the recognition model (Blei et al., 2017). The generative model involves the parameters $\boldsymbol{\theta}$ and ψ, where the latent space $\mathbf{z}$ is assumed to have generated both the base representation f and the labels $\boldsymbol{y}$. Regarding the recognition model, its variational parameters are referred to as $\boldsymbol{\phi}$. The recognition model aims to find a high-fidelity variational approximation $\mathbf{q}_{\boldsymbol{\phi}}(\mathbf{z}|f, \boldsymbol{y})$ of the true posterior. Based on the standard variational principle, we derive the evidence lower bound (ELBO) of our variational objective. This bound is responsible for ensuring that the corresponding optimization brings the approximate posterior as close as possible to the true posterior (which is unobserved).

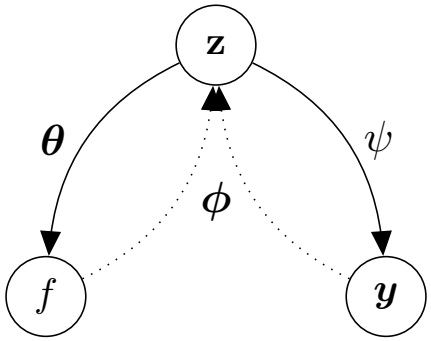

Figure 1: Description of the principal model of our proposed *CIAM* framework. This latent variable model is a two-branched variational autoencoder whose input consists of the representation f, which results from the base network that has been trained (solely once) prior to this model during the base session, as well as the labels $\boldsymbol{y}$. The goal of this model is to both learn the few-shot classes and adapt the most relevant previously encountered representations, while maximizing the overall predictive accuracy. The resulting latent representation is referred to as $\mathbf{z}$. The generative parameters are referred to as $\boldsymbol{\theta}$ and ψ, whereas the variational parameters are denoted as $\boldsymbol{\phi}$.

We describe the variational bound for our three modeling scenarios: first during the training phase of the base session, then during the training phase of the few-shot sessions, and finally during the inference (test) phase.

2.4.1 BASE SESSION

During the training procedure of the base session, the inputs to our VAE consist of the base representation f as well as the labels $\boldsymbol{y}$ of the training data points of all the base classes, which are both

observed at this point. According to the proposed model, the marginal likelihood for a single data point is expressed as follows:

$$\log\ \mathbf{p}_{\boldsymbol{\theta},\psi}(f,\boldsymbol{y}) = \log \int_{\mathbf{z}} \mathbf{p}_{\boldsymbol{\theta},\psi}(f,\boldsymbol{y},\mathbf{z})\, d\mathbf{z} = \log \int_{\mathbf{z}} \mathbf{p}(\mathbf{z})\mathbf{p}_{\boldsymbol{\theta}}(f|\mathbf{z})\mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z}) \frac{\mathbf{q}_{\boldsymbol{\phi}}(\mathbf{z}|f,\boldsymbol{y})}{\mathbf{q}_{\boldsymbol{\phi}}(\mathbf{z}|f,\boldsymbol{y})}\, d\mathbf{z} \tag{8}$$

$$= \log \mathbb{E}_{\mathbf{q}_{\boldsymbol{\phi}}(\mathbf{z}|f,\boldsymbol{y})}[\mathbf{p}(\mathbf{z})\mathbf{p}_{\boldsymbol{\theta}}(f|\mathbf{z})\mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z})/\mathbf{q}_{\boldsymbol{\phi}}(\mathbf{z}|f,\boldsymbol{y})] \tag{9}$$

$$\geq \mathbb{E}_{\mathbf{q}_{\boldsymbol{\phi}}(\mathbf{z}|f,\boldsymbol{y})}[\log \mathbf{p}(\mathbf{z}) + \log \mathbf{p}_{\boldsymbol{\theta}}(f|\mathbf{z}) + \log \mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z}) - \log \mathbf{q}_{\boldsymbol{\phi}}(\mathbf{z}|f,\boldsymbol{y})]. \tag{10}$$

The inequality in (10) is due to Jensen's inequality. Given the fact that $\mathbf{q}_{\boldsymbol{\phi}}(\mathbf{z}|f,\boldsymbol{y}) \equiv \mathbf{q}_{\mathbf{S}}(\mathbf{z}_{\mathbf{S}}|f,\boldsymbol{y})$, (4), and the basic characteristics of the normalizing flows, then the following can be obtained from (10):

$$\log\ \mathbf{p}_{\boldsymbol{\theta},\psi}(f,\boldsymbol{y}) \geq \mathbb{E}_{\mathbf{q}_{\mathbf{0}}(\mathbf{z}_{\mathbf{0}}|f,\boldsymbol{y})}[\log \mathbf{p}(\mathbf{z}_{\mathbf{S}}) + \log \mathbf{p}_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{z}_{\mathbf{S}}) + \log \mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z}_{\mathbf{S}}) - \mathbf{q}_{\mathbf{0}}(\mathbf{z}_{\mathbf{0}}|f,\boldsymbol{y}) + \sum_{\mathbf{s}=\mathbf{1}}^{\mathbf{S}} \log \left|\det \tfrac{d\mathbf{z}_{\mathbf{s}}}{d\mathbf{z}_{\mathbf{s}-\mathbf{1}}}\right|]. \tag{11}$$

A major computational advantage of the VAE is that, unlike the variational EM algorithm where the E-step is repeated for each data point, the cost of inference is amortized here via the the recognition network (Kingma & Welling, 2014; Rezende et al., 2014) whose parameters are referred to as $\boldsymbol{\phi}$. Based on the recognition network, a mapping is established from the VAE inputs, namely the base representation f and labels $\boldsymbol{y}$, first to the initial approximate density $\mathbf{q}_{\mathbf{0}}$, and then all the way through to $\mathbf{q}_{\boldsymbol{\phi}} = \mathbf{q}_{\mathbf{S}}$.

The probability distribution $\mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z})$ is depicted by a linear classifier. This is the ultimate few-shot[1] classifier based on which the FSCIL training and test procedures of our proposed *CIAM* framework are performed. A linear classifier for $\mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z})$ is preferred here for two reasons. First, it contributes towards making the proposed *CIAM* framework less prone to overfitting, which is one of the main challenges encountering any FSCIL framework. In addition, the nonlinear dependencies between all the elements of our framework have already been modeled while learning the representation $\mathbf{z}$. For an r-dimensional representation $\mathbf{z}$, the distribution $\mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z})$ is represented as $\sum_{\mathbf{j}=1}^{r} \psi_{\mathbf{j}}\mathbf{z}_{\mathbf{j}} + \psi_0$. Thus, the parameters ψ comprise $\psi_0, \psi_1, \ldots, \psi_r$.

2.4.2 FEW-SHOT SESSIONS

During the training procedure of the few-shot sessions, the introduced VAE aims to both learn the new few-shot classes and adapt the representations of the relevant previously encountered classes. The first step is to identify the relevant classes using the Gaussian kernel defined in (2). Note that solely one value per class is stored, which is the aforementioned mean of the base network representations of the training data points belonging to such a class, referred to as $\boldsymbol{\mu}_{\mathbf{f_i}}$ for class i. These are first used by the Gaussian kernel in (2) to identify the relevant classes. For the previously encountered classes with the largest values of k (where k is the outcome of the Gaussian kernel computations from (2)), which are then selected as the most relevant classes, the means $\boldsymbol{\mu}_{\mathbf{f_i}}$ are then used to adapt the VAE representations of their respective classes.

The corresponding base representations f of the few training data points available for the new few-shot class are obtained from the base network. At this stage, recall that the base network is fixed with no training whatsoever since the end of the base session. As such, it acts solely as a fixed

1. This is in contrast with the top classification layer of the base network which was solely used to learn the base representation f during the base session.

nonlinear function of the few-shot training data points during the training phase of the few-shot sessions. The (few) resulting base representation vectors f of the new few-shot class, along with the means of the base representations of the relevant previous classes (one point per relevant class) are used as points for the VAE such that a few optimization iterations can be performed so as to: i) learn the representation $\mathbf{z}$ of the new few-shot class that is optimized for predictive accuracy, and ii) correspondingly adapt the representations $\mathbf{z}$ of the relevant previous classes. The same VAE ELBO defined throughout (8) to (11) is also utilised during the training procedure of the few-shot sessions. Despite the fact that VAEs have already demonstrated an ability to learn in few-shot (Schonfeld et al., 2019), and even one-shot (Mocanu & Mocanu, 2018; Kim et al., 2019) or zero-shot (Schonfeld et al., 2019), settings , our arrangement here is further facilitated by the fact that the training previously applied to the already existing classes acts as a form of pre-training that enables the proposed VAE to efficiently adapt the (pre-trained) representations of the relevant previous classes upon the arrival of new few-shot classes.

2.4.3 TEST PHASE

During the test phase, the labels $\boldsymbol{y}$ are unknown. Thus, the ELBO defined in (8)-(11) does not apply during the test phase. We therefore develop the ELBO for the test phase here. The input $\boldsymbol{x}$ of every test data point is first fed as input to the base representation function (which is fixed at this stage) to obtain the corresponding base representation f. The latter is in turn fed as the only observed input to the VAE. The marginal likelihood for a single test data point can therefore be expressed as follows:

$$\log\, \mathbf{p}_{\boldsymbol{\theta},\psi}(f) = \log \int_{\boldsymbol{y},\mathbf{z}} \mathbf{p}_{\boldsymbol{\theta},\psi}(f,\boldsymbol{y},\mathbf{z})\, d\boldsymbol{y}\, d\mathbf{z} = \log \int_{\boldsymbol{y},\mathbf{z}} \mathbf{p}(\mathbf{z})\mathbf{p}_{\boldsymbol{\theta}}(f|\mathbf{z})\mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z}) \frac{\mathbf{q}_{\boldsymbol{\phi}}(\boldsymbol{y},\mathbf{z}|f)}{\mathbf{q}_{\boldsymbol{\phi}}(\boldsymbol{y},\mathbf{z}|f)}\, d\boldsymbol{y}\, d\mathbf{z} \tag{12}$$

$$= \log \mathbb{E}_{\mathbf{q}_{\boldsymbol{\phi}}(\boldsymbol{y},\mathbf{z}|f)}[\mathbf{p}(\mathbf{z})\mathbf{p}_{\boldsymbol{\theta}}(f|\mathbf{z})\mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z})/\mathbf{q}_{\boldsymbol{\phi}}(\boldsymbol{y},\mathbf{z}|f)] \tag{13}$$

$$\geq \mathbb{E}_{\mathbf{q}_{\boldsymbol{\phi}}(\boldsymbol{y},\mathbf{z}|f)}[\log \mathbf{p}(\mathbf{z}) + \log \mathbf{p}_{\boldsymbol{\theta}}(f|\mathbf{z}) + \log \mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z}) - \log \mathbf{q}_{\boldsymbol{\phi}}(\boldsymbol{y},\mathbf{z}|f)], \tag{14}$$

where Jensen's inequality is the reason for the inequality in (14). The classifier $\mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z})$ is then used to predict the classes (i.e. labels) given the optimized latent representation $\mathbf{z}$ of every test data point.

The key steps of the proposed *CIAM* algorithm are listed in Algorithm 1.

3. Theoretical Analysis

We shed light on the theoretical relationship between the current class in a class-incremental learning (CIL) setup, and the previously encountered classes. The overall objective is to provide a theoretical verification of the impact induced by the relevant (i.e. most similar) previously encountered classes on classifying the current class.

Given that our principal interest here is the characteristics of the relationship between the classes involved in a CIL setup, rather than the multi-class classification form, let's assume without loss of generality (WLOG) a one-vs-rest strategy where the multi-class classification problem is split into one binary classification problem per class. Corresponding to each class, consider a class task $\mathcal{T}$ which is a binary classification problem predicting whether or not a data point belongs to the respective class. Refer to the input data distribution of such a class as $\mathcal{D}_{\mathcal{T}}$ on $\mathcal{X}$, and to the true (i.e. ground truth) labeling function as l. The domain of the labeling function l is the input space $\mathcal{X}$, whereas its range is $[0, 1]$. The prediction function is referred to as h which is a learning hypothesis

Algorithm 1 The proposed *CIAM* algorithm for few-shot class-incremental learning adaptation via latent variable modeling

Input: m sequentially arriving datasets $D_t = \{\boldsymbol{x}_t^n, \boldsymbol{y}_t^n\}_{n=1}^{N_t}$, $t \in \{1, 2, \ldots, m\}$, and N_t is the size of D_t.
Base session $\leftarrow t = 1$.
$N_1 >> N_i, i \in \{2, \ldots, m\}$.
Parameters:
Generative parameters: $\boldsymbol{\theta}$ *and* ψ
Variational parameters: $\boldsymbol{\phi}$
$\log \mathbf{p}(f, \boldsymbol{y})$*: the training marginal likelihood.*
$\log \mathbf{p}(f)$*: the test marginal likelihood.*
for $t = 1$ ***do***
  // Base session Learning
  Observe $D_1 = \{\boldsymbol{x}_1^n, \boldsymbol{y}_1^n\}_{n=1}^{N_1}$
  Train the base network using (1)
  Compute $\boldsymbol{\mu}_{\mathbf{f_i}}$, $i \in \{1, 2, \ldots, N_1\}$
  Train the VAE on D_1 *to compute* $\log \mathbf{p}(f, \boldsymbol{y})$ *using (8)-(11):*
  repeat
    $\Delta\boldsymbol{\theta} \propto -\nabla_{\boldsymbol{\theta}} \log \mathbf{p}(f, \boldsymbol{y})$
    $\Delta\psi \propto -\nabla_{\psi} \log \mathbf{p}(f, \boldsymbol{y})$
    $\Delta\boldsymbol{\phi} \propto -\nabla_{\boldsymbol{\phi}} \log \mathbf{p}(f, \boldsymbol{y})$
  until $\boldsymbol{\theta}, \psi, \boldsymbol{\phi}$ *do not change*
end for
for $t = 2, \ldots, m$ ***do***
  // Few-shot session Learning
  Observe $D_t = \{\boldsymbol{x}_t^n, \boldsymbol{y}_t^n\}_{n=1}^{N_t}$
  Select the most similar previous classes using (2)
  Train the VAE on D_t *and* $\boldsymbol{\mu}_{\mathbf{f_j}}$*, where* j *denotes indices of the similar previous classes, to compute* $\log \mathbf{p}(f, \boldsymbol{y})$ *using (8)-(11)*
end for
// Test
Train the VAE on D^{test} *to compute* $\log \mathbf{p}(f)$ *using (12)-(14), and to predict the labels* $\boldsymbol{y}$*:*
repeat
  $\Delta\boldsymbol{\theta} \propto -\nabla_{\boldsymbol{\theta}} \log \mathbf{p}(f)$
  $\Delta\psi \propto -\nabla_{\psi} \log \mathbf{p}(f)$
  $\Delta\boldsymbol{\phi} \propto -\nabla_{\boldsymbol{\phi}} \log \mathbf{p}(f)$
until $\boldsymbol{\theta}, \psi, \boldsymbol{\phi}$ *do not change*
Predict the labels via $\mathbf{p}_{\psi}(\boldsymbol{y}|\mathbf{z})$

with the same domain and range as l. The hypothesis h belongs to a hypothesis space $\mathcal{H}$. We can then use the following to denote the error of the learning hypothesis h with respect to the true labeling function l under class task $\mathcal{T}$:

$$\varepsilon_{\mathcal{T}}(h, l) := \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{T}}}[|h(\mathbf{x}) - l(\mathbf{x})|] \tag{15}$$

We refer to the risk of a hypothesis h as the error $\varepsilon_{\mathcal{T}}(h)$ of h under class task $\mathcal{T}$ with respect to the true labeling function $f_{\mathcal{T}}$ of the same class task $\mathcal{T}$: $\varepsilon_{\mathcal{T}}(h) := \varepsilon_{\mathcal{T}}(h, f_{\mathcal{T}})$. In this analysis, we assume that the class-incremental learner has already encountered $\boldsymbol{r}$ classes so far. Refer to the collection of subsets of the input space $\mathcal{X}$ which represent the support of a learning hypothesis h, $h \in \mathcal{H}$, as $\mathcal{A}_{\mathcal{H}}$, where $\mathcal{A}_{\mathcal{H}} := \{h^{-1}(1) \mid h \in \mathcal{H}\}$. In similar paradigms such as domain adaptation and transfer learning, there is a widely used measure of distance between two distributions referred to as the $\mathcal{H}$-divergence (Ben-David et al., 2010; Sun et al., 2011; Ajakan et al., 2014; Zhao et al., 2019; Adel, 2024). We follow the definition of the $\mathcal{H}$-divergence in (Zhao et al., 2019; Adel, 2024) where the constant factor of 2 is excluded, since such a constant factor is not relevant for our setup either. The distance between two distributions $\mathcal{D}$ and $\mathcal{D}'$ can therefore be defined according to the $\mathcal{H}$-divergence as:

$$d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') := \sup_{A \in \mathcal{A}_{\mathcal{H}}} |\Pr_{\mathcal{D}}(A) - \Pr_{\mathcal{D}'}(A)| \tag{16}$$

3.1 Generalization Upper Bound

Define the difference between two class tasks as the summation of the distance between their data distributions $\mathcal{D}$ and the distance between their labeling functions l. We develop a theorem which states that the error of the current class (i.e. the class currently encountered by the class-incremental learner) is less than or equal to the summation of two terms. The first term does not involve any degrees of freedom related to the learner, whereas the second term depends on previous classes whose class tasks are as similar as possible to the current class with respect to the class task difference.

Theorem 1. *Let $\mathcal{T}_i$ refer to the current class task whose index is i, and $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{\boldsymbol{r}}\}$ refer to the previously encountered class tasks, that is $i = \boldsymbol{r} + 1$[2]. The following holds for any hypothesis $h \in \mathcal{H}$, where $\mathcal{H}$ is a hypothesis space:*

$$\varepsilon_{\mathcal{T}_i}(h) \leq$$

$$\sum_{j,k=1,k>j}^{\boldsymbol{r}} \Big(2min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k}), \varepsilon_{\mathcal{T}_k}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k})\} + 2d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k})\Big) + \tag{17}$$

$$\min_{j \in \{1,2,\ldots,\boldsymbol{r}\}} \Big\{\Big(\varepsilon_{\mathcal{T}_j}(h) + min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_j}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\} \tag{18}$$

Note that the first term of the bound in (17) completely depends on ground truth functions of the previously encountered classes, which are given as input to the learner. In other words, the first term in (17) does not depend at all on any learning hypothesis h, nor on any degrees of freedom that the class-incremental learner can control during its optimization procedure. In contrast, the second term of the bound, which is the minimum term in (18), consists of $\boldsymbol{r}$ clauses where each clause corresponds to one previous class. As derived in (27), each clause in (18) represents the upper bound on a class task had the corresponding class been the sole class previously encountered by the class-incremental learner. Consequently, given the fact that the class-incremental learner has previously encountered $\boldsymbol{r}$ classes: According to Theorem 1, identifying the most similar previous classes to the current class leads to minimizing (18), which in turn leads to tightening the bound on (i.e. minimizing) the error of the current class. Thus, Theorem 1 indicates a dependence between the error on the current class and the errors on the most similar (relevant) previous classes.

2. This is adopted herein to shorten the subscripts (with i rather than $\boldsymbol{r} + 1$).

3.2 Proof

Before delving into the proof of Theorem 1, we provide a few lemmata which will be needed throughout, along with their proofs.

Lemma 1. *For a hypothesis space $\mathcal{H}$ and distributions $\mathcal{D}$, $\mathcal{D}'$ and $\mathcal{D}''$ over $\mathcal{H}$, the $\mathcal{H}$-divergence distance measure satisfies the triangle inequality:*

$$d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') \leq d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}'') + d_{\mathcal{H}}(\mathcal{D}'', \mathcal{D}') \tag{19}$$

Proof.

$$\begin{aligned}
d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') &= \sup_{A \in \mathcal{A}_{\mathcal{H}}} |\Pr_{\mathcal{D}}(A) - \Pr_{\mathcal{D}'}(A)| \\
&= \sup_{A \in \mathcal{A}_{\mathcal{H}}} \Big(|\Pr_{\mathcal{D}}(A) - \Pr_{\mathcal{D}''}(A) + \Pr_{\mathcal{D}''}(A) - \Pr_{\mathcal{D}'}(A)| \Big) \\
&\leq \sup_{A \in \mathcal{A}_{\mathcal{H}}} \Big(|\Pr_{\mathcal{D}}(A) - \Pr_{\mathcal{D}''}(A)| + |\Pr_{\mathcal{D}''}(A) - \Pr_{\mathcal{D}'}(A)| \Big) \\
&= \sup_{A \in \mathcal{A}_{\mathcal{H}}} |\Pr_{\mathcal{D}}(A) - \Pr_{\mathcal{D}''}(A)| + \sup_{A \in \mathcal{A}_{\mathcal{H}}} |\Pr_{\mathcal{D}''}(A) - \Pr_{\mathcal{D}'}(A)| \\
&= d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}'') + d_{\mathcal{H}}(\mathcal{D}'', \mathcal{D}')
\end{aligned} \tag{20}$$

□

Lemma 2. *For a task $\mathcal{T}$ with a marginal (i.e. data) distribution $\mathcal{D}_{\mathcal{T}}$, and a hypothesis space $\mathcal{H}$, the learning hypothesis error satisfies the triangle inequality:*

$$\varepsilon_{\mathcal{T}}(h, h') \leq \varepsilon_{\mathcal{T}}(h, h'') + \varepsilon_{\mathcal{T}}(h'', h') \tag{21}$$

for any three hypotheses h, h', $h'' \in \mathcal{H}$.

Proof.

$$\begin{aligned}
\varepsilon_{\mathcal{T}}(h, h') &= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{T}}, h, h' \sim \mathcal{H}} \Big[|h(\mathbf{x}) - h'(\mathbf{x})| \Big] \\
&= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{T}}, h, h', h'' \sim \mathcal{H}} \Big[|h(\mathbf{x}) - h''(\mathbf{x}) + h''(\mathbf{x}) - h'(\mathbf{x})| \Big] \\
&\leq \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{T}}, h, h', h'' \sim \mathcal{H}} \Big[|h(\mathbf{x}) - h''(\mathbf{x})| + |h''(\mathbf{x}) - h'(\mathbf{x})| \Big] \\
&= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{T}}, h, h'' \sim \mathcal{H}} \Big[|h(\mathbf{x}) - h''(\mathbf{x})| \Big] + \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{T}}, h', h'' \sim \mathcal{H}} \Big[|h''(\mathbf{x}) - h'(\mathbf{x})| \Big] \\
&= \varepsilon_{\mathcal{T}}(h, h'') + \varepsilon_{\mathcal{T}}(h'', h')
\end{aligned} \tag{22}$$

□

Lemma 3. *For a hypothesis class $\mathcal{H}$, $\forall h, h' \in \mathcal{H}$, where h and h' are learning hypotheses: Let $\mathcal{D}_{\mathcal{T}_j}$ and $\mathcal{D}_{\mathcal{T}_k}$ be the marginal distributions for class tasks $\mathcal{T}_j$ and $\mathcal{T}_k$, respectively. Then the following holds:*

$$\varepsilon_{\mathcal{T}_j}(h, h') \leq \varepsilon_{\mathcal{T}_k}(h, h') + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k}) \tag{23}$$

Proof. For $h, h' \in \mathcal{H}$,

$$\begin{aligned}
\varepsilon_{\mathcal{T}_j}(h, h') &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{T}_j}, h, h'\sim\mathcal{H}}\Big[|h(\mathbf{x}) - h'(\mathbf{x})|\Big] \\
&= \mathbb{E}_{h,h'\sim\mathcal{H}} \int_{\mathbf{x}} \Pr_{\mathcal{D}_{\mathcal{T}_j}}(\mathbf{x})\Big[|h(\mathbf{x}) - h'(\mathbf{x})|\Big] \\
&= \mathbb{E}_{h,h'\sim\mathcal{H}} \int_{\mathbf{x}} \Big(\Pr_{\mathcal{D}_{\mathcal{T}_j}}(\mathbf{x}) - \Pr_{\mathcal{D}_{\mathcal{T}_k}}(\mathbf{x}) + \Pr_{\mathcal{D}_{\mathcal{T}_k}}(\mathbf{x})\Big)\Big[|h(\mathbf{x}) - h'(\mathbf{x})|\Big] \\
&= \varepsilon_{\mathcal{T}_k}(h, h') + \mathbb{E}_{h,h'\sim\mathcal{H}} \int_{\mathbf{x}} \Big(\Pr_{\mathcal{D}_{\mathcal{T}_j}}(\mathbf{x}) - \Pr_{\mathcal{D}_{\mathcal{T}_k}}(\mathbf{x})\Big)\Big[|h(\mathbf{x}) - h'(\mathbf{x})|\Big] \\
&\leq \varepsilon_{\mathcal{T}_k}(h, h') + \sup_{A\in\mathcal{A}_{\mathcal{H}}} |\Pr_{\mathcal{D}_{\mathcal{T}_j}}(A) - \Pr_{\mathcal{D}_{\mathcal{T}_k}}(A)| \\
&= \varepsilon_{\mathcal{T}_k}(h, h') + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k})
\end{aligned} \tag{24}$$

□

Proof of Theorem 1. We formalize the proof using mathematical induction. For the sake of further clarity, rather than beginning with solely one base case, we demonstrate that the form holds for two base cases, the first is when the class-incremental learner has previously encountered one class only, and the second case is after encountering two classes. Afterwards, we proceed with the induction step.

Let's begin with the case when the learner has previously encountered one class only. In this case, $\boldsymbol{r} = 1$. By applying Lemma 3 to $\varepsilon_{\mathcal{T}_i}(h)$, followed by Lemma 2, we obtain:

$$\varepsilon_{\mathcal{T}_i}(h) = \varepsilon_{\mathcal{T}_i}(h, f_{\mathcal{T}_i}) \leq \varepsilon_{\mathcal{T}_1}(h, f_{\mathcal{T}_i}) + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i}) \leq \varepsilon_{\mathcal{T}_1}(h) + \varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}) + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i}) \tag{25}$$

Recall that $\varepsilon_{\mathcal{T}_1}(h, f_{\mathcal{T}_1}) \equiv \varepsilon_{\mathcal{T}_1}(h)$, which was used in (25). Meanwhile, by reversing the order, i.e. by applying Lemma 2 first, followed by Lemma 3, the following is obtained:

$$\varepsilon_{\mathcal{T}_i}(h) = \varepsilon_{\mathcal{T}_i}(h, f_{\mathcal{T}_i}) \leq \varepsilon_{\mathcal{T}_i}(h, f_{\mathcal{T}_1}) + \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}) \leq \varepsilon_{\mathcal{T}_1}(h) + \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}) + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i}) \tag{26}$$

From (25) and (26), we can then obtain the following:

$$\varepsilon_{\mathcal{T}_i}(h) \leq \varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i}), \tag{27}$$

The bound in (27) represents an upper bound on the error of the current class task $\mathcal{T}_i$ given one previous task $\mathcal{T}_1$.

Let's now move on to the case when the class-incremental learner has encountered two previous tasks ($\boldsymbol{r} = 2$). By applying lemmata 2 and 3 in both orders, similar to the operations performed in (25), (26) and (27), $\varepsilon_{\mathcal{T}_1}(h)$ can then be expressed as follows:

$$\varepsilon_{\mathcal{T}_1}(h) \leq \varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1}), \varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_1}) \tag{28}$$

Using the expression in (28) for $\varepsilon_{\mathcal{T}_1}(h)$ back in (27), we get the following:

$$\begin{aligned}
\varepsilon_{\mathcal{T}_i}(h) \leq &\varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1}), \varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1})\} + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} \\
&+ d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_1}) + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i})
\end{aligned} \tag{29}$$

Recall that, by the time this analysis is performed, the class-incremental learner has previously encountered a set of $\boldsymbol{r}$ classes $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{\boldsymbol{r}}\}$. Our analysis is not dependent on the order of appearance of the previous classes to the class-incremental learner. Thus, any class task within this set can be arbitrarily chosen to begin our analysis with. As such, we can perform the above analysis with class task $\mathcal{T}_2$ as the starting point, rather than class task $\mathcal{T}_1$. We began above with $\mathcal{T}_1$ in (27), followed by $\mathcal{T}_2$ in (28). Let's now switch the order so that we begin with $\mathcal{T}_2$ instead, followed by $\mathcal{T}_1$. The outcome in such case is ultimately expressed as follows:

$$\begin{aligned}\varepsilon_{\mathcal{T}_i}(h) \leq & \varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1}), \varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1})\} + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i})\} \\ & + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_1}) + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_i}) \end{aligned} \tag{30}$$

From (29) and (30):

$$\begin{aligned}\varepsilon_{\mathcal{T}_i}(h) \leq & \, min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1}), \varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_1}) \\ & + min\Big\{\Big(\varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_i})\Big), \\ & \qquad \Big(\varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\} \end{aligned} \tag{31}$$

Lemmata 2 and 3 can be applied in both orders to obtain a bound for $\mathcal{T}_2$ by reversing the roles of $\mathcal{T}_1$ and $\mathcal{T}_2$, similar to what was performed for $\mathcal{T}_1$ to obtain (28). This operation leads to the following expression for $\mathcal{T}_2$:

$$\varepsilon_{\mathcal{T}_2}(h) \leq \varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_2}), \varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_1}, f_{\mathcal{T}_2})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_2}) \tag{32}$$

By using the inequalities in (28) and (32) as expressions for $\varepsilon_{\mathcal{T}_1}(h)$ and $\varepsilon_{\mathcal{T}_2}(h)$, respectively, in (31), we then obtain the following upper bound on the error $\varepsilon_{\mathcal{T}_i}(h)$ of the current class task $\mathcal{T}_i$ given two previous class tasks $\mathcal{T}_1$ and $\mathcal{T}_2$:

$$\begin{aligned}\varepsilon_{\mathcal{T}_i}(h) \leq & \; 2\, min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1}), \varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_2}, f_{\mathcal{T}_1})\} + 2\, d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_1}) \\ & + min\Big\{\Big(\varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_i})\Big), \\ & \qquad \Big(\varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\} \end{aligned} \tag{33}$$

By rearranging the clauses of the third (minimum) term on the R.H.S. of (33), we obtain the following:

$$\begin{aligned}\varepsilon_{\mathcal{T}_i}(h) \leq & \; 2\, min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_2}), \varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_1}, f_{\mathcal{T}_2})\} + 2\, d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_2}) \\ & + min\Big\{\Big(\varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i})\Big), \\ & \qquad \Big(\varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\} \end{aligned} \tag{34}$$

In (27), we have obtained an upper bound on the error $\varepsilon_{\mathcal{T}_i}(h)$ for the case when the class-incremental learner has solely encountered one previous class task. This one class task turned out to be $\mathcal{T}_1$ in (27), but had there been another class task involved rather than $\mathcal{T}_1$, as the sole class task, e.g. $\mathcal{T}_2$, the same bound in (27) could then be applied by replacing $\mathcal{T}_1$ with $\mathcal{T}_2$ in (27). Using this fact for both clauses

of the third (minimum) term on the R.H.S. of (34), the following upper bound on the error given two class tasks, $\mathcal{T}_1$ and $\mathcal{T}_2$, can then be expressed as follows:

$$\begin{aligned}\varepsilon_{\mathcal{T}_i}(h) \leq\ & 2\,min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_2}), \varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_1}, f_{\mathcal{T}_2})\} + 2\,d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_2}) \\ & + min\Big\{\text{bound with one class task } \mathcal{T}_1, \\ & \qquad \text{bound with one class task } \mathcal{T}_2\Big\} \end{aligned} \tag{35}$$

Finally, we proceed to the induction step. Assume that the form of the generalization bound on the error of the current class task $\mathcal{T}_i$, given two previous class tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ in (34), equivalently holds for $\boldsymbol{r}-1$ training tasks. In other words, assume that the following holds:

$$\begin{aligned}\varepsilon_{\mathcal{T}_i}(h) \leq & \sum_{j,k=1,k>j}^{\boldsymbol{r}-1} \Big(2\,min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k}), \varepsilon_{\mathcal{T}_k}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k})\} + 2\,d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k})\Big) \\ & + \min_{j\in\{1,2,\ldots,\boldsymbol{r}-1\}} \Big\{\Big(\varepsilon_{\mathcal{T}_j}(h) + min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_j}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\} \end{aligned} \tag{36}$$

Let's now involve $\mathcal{T}_{\boldsymbol{r}}$. Similar to the inequality form in (27), the following is an upper bound on the error of task $\mathcal{T}_{\boldsymbol{r}-1}$:

$$\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(h) \leq \varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}}), \varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-1}}, \mathcal{D}_{\mathcal{T}_{\boldsymbol{r}}}) \tag{37}$$

Using the expression in (37) for $\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(h)$ back into (36), we then obtain the following:

$$\begin{aligned}&\varepsilon_{\mathcal{T}_i}(h) \leq \sum_{j,k=1,k>j}^{\boldsymbol{r}-1} \Big(2\,min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k}), \varepsilon_{\mathcal{T}_k}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k})\} + 2\,d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k})\Big) \\ &+ \min\Big\{\Big(\varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i})\Big), \\ &\Big(\varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_i})\Big), \ldots, \\ &\Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-2}}, \mathcal{D}_{\mathcal{T}_i})\Big), \\ &\Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}}), \varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}})\} + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_i})\} \\ &+ d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-1}}, \mathcal{D}_{\mathcal{T}_{\boldsymbol{r}}}) + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-1}}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\} \end{aligned} \tag{38}$$

$$\begin{aligned}&\leq \sum_{j,k=1,k>j}^{\boldsymbol{r}-1} \Big(2\,min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k}), \varepsilon_{\mathcal{T}_k}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k})\} + 2\,d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k})\Big) \\ &+ \min\Big\{\Big(\varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i})\Big), \\ &\Big(\varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_i})\Big), \ldots, \\ &\Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-2}}, \mathcal{D}_{\mathcal{T}_i})\Big), \\ &\Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(h) + 2min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}}), \varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}})\} + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_i})\} \\ &+ 2d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-1}}, \mathcal{D}_{\mathcal{T}_{\boldsymbol{r}}}) + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-1}}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\}, \end{aligned} \tag{39}$$

where the switch from (38) to (39) is due to the inequality in (27) applied to $\varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(h)$. Recall again that the order by which the previous classes can be analyzed is arbitrary. Assuming that $\mathcal{T}_{\boldsymbol{r}}$ was involved in the analysis prior to the involvement of $\mathcal{T}_{\boldsymbol{r}-1}$, i.e. that the order of these two class tasks is switched, the outcome in such case would be as follows:

$$\begin{aligned}
\varepsilon_{\mathcal{T}_i}(h) \leq & \sum_{j,k=1,k>j}^{\boldsymbol{r}-1} \Big(2\, min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k}), \varepsilon_{\mathcal{T}_k}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k})\} + 2\, d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k})\Big) \\
& + \min\Big\{\Big(\varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i})\Big), \\
& \Big(\varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_i})\Big), \ldots, \\
& \Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-2}}, \mathcal{D}_{\mathcal{T}_i})\Big), \\
& \Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}}), \varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}})\} + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(f_{\mathcal{T}_{\boldsymbol{r}}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}}}, f_{\mathcal{T}_i})\} \\
& + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-1}}, \mathcal{D}_{\mathcal{T}_{\boldsymbol{r}}}) + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}}}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\} \qquad (40)
\end{aligned}$$

Again, using the inequality in (37) for the remaining values of $\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(h)$, then from (40), we obtain the following:

$$\begin{aligned}
\varepsilon_{\mathcal{T}_i}(h) \leq & \sum_{j,k=1,k>j}^{\boldsymbol{r}-1} \Big(2\, min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k}), \varepsilon_{\mathcal{T}_k}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k})\} + 2\, d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k})\Big) \\
& + \min\Big\{\Big(\varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i})\Big), \\
& \Big(\varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_i})\Big), \ldots, \\
& \Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-2}}, \mathcal{D}_{\mathcal{T}_i})\Big), \\
& \Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(h) + 2min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}}), \varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}})\} + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(f_{\mathcal{T}_{\boldsymbol{r}}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}}}, f_{\mathcal{T}_i})\} \\
& + 2d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-1}}, \mathcal{D}_{\mathcal{T}_{\boldsymbol{r}}}) + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}}}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\} \qquad (41)
\end{aligned}$$

Taking the common terms out of (39) and (41), the following is obtained:

$$\begin{aligned}
\varepsilon_{\mathcal{T}_i}(h) \leq & \sum_{j,k=1,k>j}^{\boldsymbol{r}-1} \Big(2\, min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k}), \varepsilon_{\mathcal{T}_k}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k})\} + 2\, d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k})\Big) + && (42) \\
& \Big(2\, min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}}), \varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_{\boldsymbol{r}}})\} + 2\, d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-1}}, \mathcal{D}_{\mathcal{T}_{\boldsymbol{r}}})\Big) + && (43) \\
& \min\Big\{\Big(\varepsilon_{\mathcal{T}_1}(h) + min\{\varepsilon_{\mathcal{T}_1}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_1}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_1}, \mathcal{D}_{\mathcal{T}_i})\Big), && (44) \\
& \Big(\varepsilon_{\mathcal{T}_2}(h) + min\{\varepsilon_{\mathcal{T}_2}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_2}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_2}, \mathcal{D}_{\mathcal{T}_i})\Big), \ldots, && (45) \\
& \Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-2}}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}-2}}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-2}}, \mathcal{D}_{\mathcal{T}_i})\Big), && (46) \\
& \Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}-1}}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}-1}}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}-1}}, \mathcal{D}_{\mathcal{T}_i})\Big), && (47) \\
& \Big(\varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(h) + min\{\varepsilon_{\mathcal{T}_{\boldsymbol{r}}}(f_{\mathcal{T}_{\boldsymbol{r}}}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_{\boldsymbol{r}}}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_{\boldsymbol{r}}}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\}, && (48)
\end{aligned}$$

which results in the following:

$$\varepsilon_{\mathcal{T}_i}(h) \leq \sum_{j,k=1,k>j}^{\boldsymbol{r}} \Big(2\, min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k}), \varepsilon_{\mathcal{T}_k}(f_{\mathcal{T}_j}, f_{\mathcal{T}_k})\} + 2\, d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_k})\Big) + \tag{49}$$

$$\min_{j\in\{1,2,\ldots,\boldsymbol{r}\}} \Big\{\Big(\varepsilon_{\mathcal{T}_j}(h) + min\{\varepsilon_{\mathcal{T}_j}(f_{\mathcal{T}_j}, f_{\mathcal{T}_i}), \varepsilon_{\mathcal{T}_i}(f_{\mathcal{T}_j}, f_{\mathcal{T}_i})\} + d_{\mathcal{H}}(\mathcal{D}_{\mathcal{T}_j}, \mathcal{D}_{\mathcal{T}_i})\Big)\Big\} \tag{50}$$

This concludes the induction step and therefore leads to the upper bound on the error of the current class (with index i) given $\boldsymbol{r}$ previous classes. □

4. Experiments

We empirically evaluate the performance of the proposed *CIAM* framework. We begin this section by explaining our experimental setup, prior to proceeding with the evaluation process which involves experiments on common FSCIL benchmarks, an ablative analysis, and an experiment on real-world medical data.

4.1 Experimental Setup

We mainly aim at evaluating the following aspects: i) the performance of *CIAM*, measured by both the final classification accuracy as well as the all-important average classification accuracy on three important benchmarks which represent the most commonly used benchmarks in FSCIL; ii) the capacity of the proposed *CIAM* to alleviate catastrophic forgetting, and the degree to which the latter can be reduced, measured by the commonly used performance dropping rate; iii) an ablation study gauging the impact of every component of *CIAM*; and iv) an application of the proposed *CIAM* framework to a real-world medical benchmark which is referred to as the MedMNIST dataset. State-of-the-art results obtained in terms of the average classification accuracy, final classification accuracy, and mitigating catastrophic forgetting demonstrate the efficacy of the proposed *CIAM* framework.

In all the experiments, we compare with the highest performing variation of every competing algorithm. Every reported result is an average of 10 runs. Statistical significance (highlighted in bold) is identified using a paired t-test with $p = 0.05$. As an optimizer, we use stochastic gradient descent (SGD) with momentum, with an initial learning rate of 0.01 for both miniImageNet and CIFAR100, and an initial learning rate of 0.001 for the CUB200 dataset. The kernel width parameter σ is tuned using cross-validation. The number of normalizing flow steps $\mathbf{s}$ is 2, which achieves a good balance between exploiting the high-fidelity prowess of normalizing flows and computational soundness. The computational environment consists of an NVIDIA A100 Tensor Core GPU and two AMD Rome CPUs based on the NVIDIA Mellanox ConnectX-6 interconnect technology.

4.2 Common Few-Shot Class-Incremental Learning Benchmarks

We evaluate *CIAM* here by performing experiments on the following FSCIL benchmarks: miniImageNet (Russakovsky et al., 2015), CIFAR100 (Krizhevsky & Hinton, 2009), and CUB200 (Wah et al., 2011). Here is a brief description of these 3 datasets:

miniImageNet It is 100-class subset of ImageNet (Russakovsky et al., 2015) which is commonly used in many few-shot learning settings (Vinyals et al., 2016; Finn et al., 2017), i.e. not solely FSCIL.

Each class consists of 500 training images and 100 test images. The image format is RGB with a size of 84×84. The base session, $t = 1$, comprises 60 classes. After the base session, there are 8 few-shot sessions, each comprising 5 classes. Following the FSCIL literature, e.g. (Tao et al., 2020; Yang et al., 2023; Zhao et al., 2024), each few-shot session, $2 \leq t \leq 9$, is a 5-way 5-shot session, which means that there are 5 training points available for each of the 5 classes.

CIFAR100 This is another dataset which is quite popular in FSCIL (Rebuffi et al., 2017; Castro et al., 2018). It contains a total of $60,000$ images divided into 100 classes. Each class comprises 500 training images and 100 test images. Each image is of size 32×32. Similar to miniImageNet, we also adopt the common setting of CIFAR100 in the FSCIL literature by beginning with a base session of 60 classes, followed by 8 5-way 5-shot sessions. Thus, there is a total of 9 CIFAR100 learning sessions (1 base session + 8 few-shot sessions).

CUB200 The original design of this dataset was made to address fine-grained image classification in an incremental learning setting (Chaudhry et al., 2019; Parisi et al., 2019; Tao et al., 2020). It consists of $6,000$ training images as well as $6,000$ test images. The size of each image is 224×224. It comprises 200 classes, each depicting a specific bird category; CUB refers to Caltech-UCSD Birds. We adopt the common FSCIL splitting format of CUB200 where the 200 classes are divided into 100 classes for the base session, followed by 10 incremental 10-way 5-shot sessions.

Following several previous works in FSCIL, e.g. (Tao et al., 2020; Zhao et al., 2024), we adopt the utilization of ResNet-18 (He et al., 2016) as the backbone (base) network. For CUB200, the ResNet-18 backbone is initialized by parameters which have been pre-trained on ImageNet (Deng et al., 2009). Also, following previous FSCIL works[3] (Yang et al., 2023), we train the model on all the experiments with a minibatch size of 512 during the base session, and a minibatch size of 64 during each incremental few-shot session. For the miniImageNet dataset, the number of epochs is 500 for the base session, and is 150 for each incremental few-shot session. For CIFAR100, we train the model for 200 epochs during the base session and for 100 epochs during each few-shot session. The number of epochs for the CUB200 dataset is 80 epochs for the base session and 60 epochs for each incremental few-shot session.

Evaluation Metrics The most influential evaluation metric for FSCIL is the average overall classification accuracy which depicts the average test accuracy over all the encountered classes (from the current as well as all the previous sessions) so far. The final classification accuracy, which refers to the test accuracy value after encountering all the classes of the final learning session, is another important metric for FSCIL which we also report below. We also estimate catastrophic forgetting based on its most widely used FSCIL metric: the performance dropping rate (PD, Zhang et al., 2021). The PD metric is defined as PD $=$ average overall accuracy after the base session $-$ average overall accuracy after the final few-shot session.

We evaluate how *CIAM* fares compared to multiple state-of-the-art FSCIL algorithms: CEC (Zhang et al., 2021), FACT (Zhou et al., 2022), C-FSCIL (Hersche et al., 2022), TEEN (Wang et al., 2023b), Bidist (Zhao et al., 2023), SAVC (Song et al., 2023), NC-FSCIL (Yang et al., 2023), TOPIC (Tao et al., 2020), FCIL (Gu et al., 2023), BM-FSCIL (Zhao et al., 2024), LIMIT (Zhou et al., 2023b), MetaFSCIL (Chi et al., 2022), iCaRL (Rebuffi et al., 2017), ALICE (Peng et al., 2022) and DF Replay (Liu et al., 2022).

3. As much as possible and feasible throughout the experiments, we aimed to follow the conventions adopted by previous works in FSCIL so that we can compare on common grounds.

Results of all the FSCIL metrics for the miniImageNet dataset are displayed in detail in Table 1. In addition, a summary including the bulk[4] of the highest performing algorithms is also portrayed in Figure 2. The proposed *CIAM* achieves significantly higher average overall classification accuracy as well as final classification accuracy (i.e. classification accuracy after the final session which is session 9 for miniImageNet). In addition, *CIAM* is capable of alleviating catastrophic forgetting more efficiently than all the previous state-of-the-art FSCIL algorithms. This is demonstrated by achieving significantly lower PD rates. Apparently, a higher PD rate signifies higher forgetting rates which is undesirable, unlike the accuracy-based metrics where a higher value is a testament for better performance. *CIAM* outperforms the previous state-of-the-art in terms of the average overall classification accuracy, final classification accuracy and PD rate by 3.77%, 5.38%, and 4.42%, respectively.

CIAM also significantly outperforms the previous state-of-the-art on CIFAR100 in terms of the (final and average overall) accuracy-based classification metrics as well as the PD rate. This is illustrated in Table 2 and in Figure 3. The improvement over the previous state-of-the-art is 5.02% in terms of the average overall classification accuracy, and 6.51% in terms of the final classification accuracy. In addition, *CIAM* achieves a PD rate which is 5.96% lower (denoting a lower forgetting rate) than the previous state-of-the-art. This is a considerable improvement in alleviating catastrophic forgetting, which demonstrates the ability of *CIAM* to address the stability-plasticity dilemma. The rich density estimators employed by the proposed *CIAM* latent variable model, and the way they are fused with probabilistic modelling, have contributed to an effective adaptation strategy, which is ultimately reflected in terms of the resulting predictive accuracy and performance retention.

The commonly used splitting of the CUB200 dataset in FSCIL comprises a total of 11 sessions, i.e. with two more few-shot sessions than the previous two benchmarks. Accordingly, the resulting improvement by *CIAM* in terms of the average overall classification accuracy is even more considerable on CUB200 than on the other two datasets, with a difference of 6.61% over the second highest FSCIL algorithm. The classification accuracy is also higher over all the learning sessions, with an improvement of 6.17% over the previous state-of-the-art in terms of the final classification accuracy (after session 11). The PD rate is also lower (better) by 4.41%. The results achieved on CUB200 are described in Table 3 and in Figure 4.

4.3 Ablation Study

We conduct an ablative analysis so as to analyze the role of every modeling component of the proposed *CIAM* framework in achieving the obtained levels of accuracy, performance retention, and in mitigating catastrophic forgetting. Results of the performed ablation study are displayed in Figures 5-7 for miniImageNet, CIFAR100 and CUB200, respectively. The classification performance of *CIAM* after learning from every session is compared with the following scenarios:

1. No base network, where the model does not have a proper opportunity to learn from the relatively large amounts of data samples and classes available during the base session.

2. No VAE, where the model hardly learns from the few data samples available during the few-shot sessions, and only utilizes the already learned base network (learned during the base session) to map the few-shot classes.

4. We opted for this for the sake of a better clarity of the figures.

Table 1: Classification accuracy after learning from every session (1 base session + 8 few-shot sessions) of the miniImageNet dataset. The classification accuracy value after session 9 is the final classification accuracy. We also list the values of the all-important average overall classification accuracy, which denotes the average accuracy over all the learned FSCIL sessions, and the values of the performance dropping rate (PD) as a catastrophic forgetting metric. Results are displayed for the proposed *CIAM* as well as 15 other seminal FSCIL algorithms. Bold entries indicate significance. For PD, the lower the value the better, whereas the higher the better for the other (accuracy-based) metrics. The proposed *CIAM* significantly outperforms the previous state-of-the-art in terms of the average overall classification accuracy, the forgetting rate measured by PD, as well the final classification accuracy (i.e. classification accuracy after session 9). The proposed *CIAM* algorithm achieves significantly higher classification accuracy values and higher performance retention levels than the other FSCIL algorithms when learning from all the few-shot sessions after the base session.

Method	Overall acc. after each session (%)									Avg. overall acc.	PD
	1	2	3	4	5	6	7	8	9		
CEC (Zhang et al., 2021)	72.0	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.75	24.37
FACT (Zhou et al., 2022)	72.56	69.63	66.38	62.77	60.60	57.33	54.34	52.16	50.49	60.69	22.07
C-FSCIL (Hersche et al., 2022)	76.40	71.14	66.46	63.29	60.42	57.46	54.78	53.11	51.41	61.61	24.99
TEEN (Wang et al., 2023b)	73.53	70.55	66.37	63.23	60.53	57.95	55.24	53.44	52.08	61.44	21.45
Bidist (Zhao et al., 2023)	74.65	70.43	66.29	62.77	60.75	57.24	54.79	54.79	54.79	61.42	22.43
SAVC (Song et al., 2023)	81.12	76.14	72.43	68.92	66.48	62.95	59.92	58.39	57.11	67.05	24.01
NC-FSCIL (Yang et al., 2023)	84.02	76.80	72.00	67.83	66.35	64.04	61.46	59.54	58.31	67.82	25.71
TOPIC (Tao et al., 2020)	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64	36.89
FCIL (Gu et al., 2023)	76.34	71.40	67.10	64.08	61.30	58.51	55.72	54.08	52.76	62.37	23.58
BM-FSCIL (Zhao et al., 2024)	86.22	77.38	73.90	70.13	67.85	65.11	62.84	61.61	60.47	69.50	25.75
LIMIT (Zhou et al., 2023b)	72.32	68.47	64.30	60.78	57.95	55.07	52.70	50.72	49.19	59.06	23.13
MetaFSCIL (Chi et al., 2022)	72.04	67.94	63.77	60.29	57.58	55.16	52.90	50.79	49.19	58.85	22.85
iCaRL (Rebuffi et al., 2017)	61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.21	33.29	44.10
ALICE (Peng et al., 2022)	80.60	70.60	67.40	64.50	62.50	60.00	57.80	56.80	55.70	63.99	24.90
DF Replay (Liu et al., 2022)	71.84	67.12	63.21	59.77	57.01	53.95	51.55	49.52	48.21	58.02	23.63
CIAM (Ours)	82.88	**80.74**	**76.55**	**75.63**	**72.82**	**70.01**	**68.28**	**66.68**	**65.85**	**73.27**	**17.03**

3. No adaptation of relevant classes, where the VAE is solely used to learn the representation of the current few-shot class, without leveraging one of the principal edges of our proposed model where the relevant previous classes are adapted accordingly.

4. Randomly selected relevant classes, where the VAE is used to adapt the current few-shot class as well as the relevant previous classes. However, the relevant classes are selected randomly, i.e. not via the Gaussian kernel-based similarity.

As shown in the results of the ablative analysis in Figures 5-7, significant differences in the performance levels (between *CIAM* and the other four scenarios) empirically demonstrate the significance of the proposed adaptation mechanism, along with the other modeling components, in achieving the performance levels obtained by *CIAM*.

4.4 Medical Benchmark

In addition to the commonly used FSCIL benchmarks, we apply the proposed *CIAM* framework to real-world medical images in the form of the MedMNIST dataset (Yang et al., 2021a, 2021b). MedMNIST is a standard medical imaging benchmark which consists of a collection of medical

Table 2: Classification accuracy after learning from the 9 sessions of the CIFAR100 dataset. The proposed *CIAM* achieves a significantly higher average overall classification accuracy and final (after session 9) classification accuracy. Furthermore, *CIAM* is also more efficient in terms of alleviating catastrophic forgetting, given its significantly lower PD rate. A summary of the results achieved by *CIAM*, along with most of the previous state-of-the-art, is further illustrated in Figure 3.

Method	Overall acc. after each session (%)									Avg. overall acc.	PD
	1	2	3	4	5	6	7	8	9		
CEC (Zhang et al., 2021)	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53	23.93
FACT (Zhou et al., 2022)	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	62.24	22.50
C-FSCIL (Hersche et al., 2022)	77.47	72.40	67.47	63.25	59.84	56.95	54.42	52.47	50.47	61.64	27.0
TEEN (Wang et al., 2023b)	78.92	72.32	68.16	64.43	61.19	58.48	56.11	54.03	51.87	62.83	27.05
Bidist (Zhao et al., 2023)	79.98	76.43	72.91	68.88	65.21	61.72	60.45	58.47	55.94	66.67	24.04
SAVC (Song et al., 2023)	78.47	72.31	67.49	62.41	59.10	55.95	53.81	51.54	49.16	61.14	29.31
NC-FSCIL (Yang et al., 2023)	82.52	76.82	73.34	69.68	66.19	62.85	60.96	59.02	56.11	67.50	26.41
TOPIC (Tao et al., 2020)	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62	34.73
FCIL (Gu et al., 2023)	78.31	73.07	69.84	66.29	62.93	58.99	56.18	53.70	51.91	63.47	26.40
BM-FSCIL (Zhao et al., 2024)	82.88	78.94	74.59	70.35	67.85	64.99	63.79	61.92	59.68	69.44	23.20
LIMIT (Zhou et al., 2023b)	73.81	72.09	67.87	63.89	60.70	57.77	55.67	53.52	51.23	61.84	22.58
MetaFSCIL (Chi et al., 2022)	74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97	60.79	24.53
iCaRL (Rebuffi et al., 2017)	64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	32.87	50.37
ALICE (Peng et al., 2022)	79.00	70.50	67.10	63.40	61.20	59.20	58.10	56.30	54.10	63.21	24.90
DF Replay (Liu et al., 2022)	74.40	70.20	66.54	62.51	59.71	56.58	54.52	52.39	50.14	60.78	24.26
CIAM (Ours)	82.73	**80.80**	**78.92**	**76.39**	**74.01**	**72.05**	**70.45**	**68.64**	**66.19**	**74.46**	**16.54**

Table 3: Classification accuracy after learning from the 11 sessions of the CUB200 dataset. The improvement achieved by the proposed *CIAM* is significant in terms of all the performance metrics (average overall classification accuracy, final classification accuracy and PD rate). Moreover, the larger number of incremental sessions (compared to the previous two datasets) has made the improvement in terms of the average overall classification accuracy even more considerable (6.61% improvement over the previous state-of-the-art). This is further clarified in Figure 4.

Method	Overall acc. after each session (%)											Avg. overall acc.	PD
	1	2	3	4	5	6	7	8	9	10	11		
CEC (Zhang et al., 2021)	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	23.57
FACT (Zhou et al., 2022)	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	64.42	18.96
C-FSCIL (Hersche et al., 2022)	75.88	75.51	72.42	70.55	66.09	64.58	62.11	59.71	57.98	56.87	56.42	65.28	19.46
TEEN (Wang et al., 2023b)	77.26	76.13	72.81	68.16	67.77	64.40	63.25	62.29	61.19	60.32	59.31	66.62	17.95
Bidist (Zhao et al., 2023)	79.98	75.24	73.80	69.44	67.85	65.09	64.14	63.72	62.42	62.17	61.51	67.76	18.47
SAVC (Song et al., 2023)	81.85	77.92	74.95	70.21	69.96	67.02	66.16	65.30	63.84	63.15	62.50	69.35	19.35
NC-FSCIL (Yang et al., 2023)	80.45	75.98	72.30	70.28	68.17	65.16	64.43	63.25	60.66	60.01	59.44	67.28	21.01
TOPIC (Tao et al., 2020)	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.26	43.92	42.42
FCIL (Gu et al., 2023)	79.86	74.37	71.88	66.49	67.57	64.28	62.91	60.01	59.16	58.93	58.18	65.79	21.68
BM-FSCIL (Zhao et al., 2024)	81.22	78.40	75.77	72.40	71.10	68.35	67.25	66.40	64.71	64.56	63.89	70.37	17.33
LIMIT (Zhou et al., 2023b)	76.32	74.18	72.68	69.19	68.79	65.64	63.57	62.69	61.47	60.44	58.45	66.67	17.87
MetaFSCIL (Chi et al., 2022)	75.90	72.41	68.78	64.78	62.96	59.99	58.30	56.85	54.78	53.82	52.64	61.93	23.26
iCaRL (Rebuffi et al., 2017)	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	36.67	47.52
ALICE (Peng et al., 2022)	77.40	72.70	70.60	67.20	65.90	63.40	62.90	61.90	60.50	60.60	60.10	65.75	17.30
DF Replay (Liu et al., 2022)	75.90	72.14	68.64	63.76	62.58	59.11	57.82	55.89	54.92	53.58	52.39	61.52	23.51
CIAM (Ours)	**82.95**	**81.70**	**81.14**	**80.02**	**78.95**	**77.67**	**75.83**	**74.84**	**72.01**	**71.56**	**70.06**	**76.98**	**12.89**

image datasets designed for the sake of benchmarking machine learning algorithms in the healthcare domain.

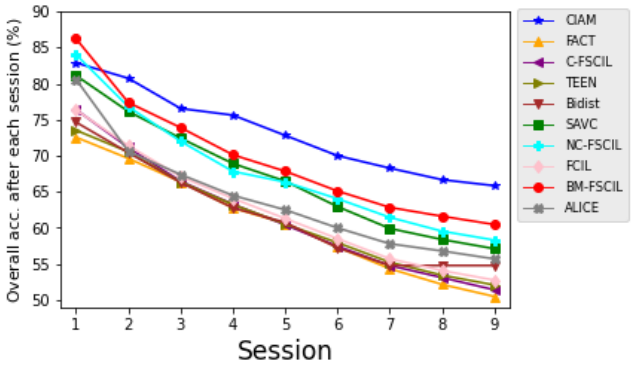

Figure 2: Comparison between state-of-the-art FSCIL algorithms on miniImageNet. The proposed *CIAM* significantly outperforms the previous state-of-the-art during all the incremental few-shot sessions. This is demonstrated by higher (average overall + final) classification accuracy values, as well as by achieving higher performance retention levels depicted by the significantly lower forgetting rate. Adapting the relevant previously encountered classes when learning new classes has contributed to achieving higher performance retention levels of the knowledge previously acquired on previous classes, ultimately leading to considerably lower forgetting rates. This will also be further clarified in the ablative analysis in Section 4.3. The difference in the performance becomes even more considerable after the first few incremental few-shot sessions, when it becomes more challenging to address the potential catastrophic forgetting of the already acquired knowledge.

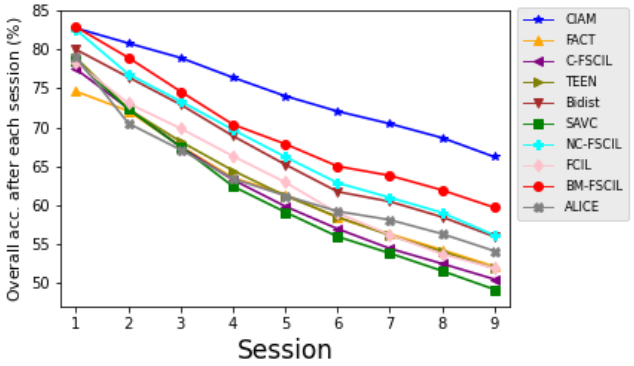

Figure 3: Performance of *CIAM* as well as other FSCIL algorithms on CIFAR100. *CIAM* achieves significantly higher classification accuracy results and better performance retention levels.

MedMNIST is a collection containing 12 2D medical datasets which have been pre-processed and standardized to perform classification tasks on 28×28 images (Yang et al., 2021a). The MedMNIST tasks cover numerous medical image modalities and diverse data scales. The corresponding labels are available and hence neither background knowledge nor manual tuning is required for users, which is a significant advantage, particularly when analyzing multiple datasets with different modalities.

Similar to what we adopted with the common benchmarks in Section 4.2, we also follow the experimental conventions pursued previously on the MedMNIST dataset within the same learning paradigm, which is FSCIL. The backbone (base) network we utilize here too is ResNet-18, similar

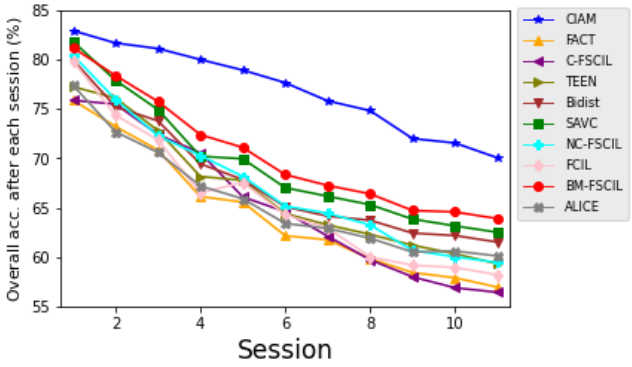

Figure 4: Comparison between *CIAM* and several other FSCIL algorithms on the CUB200 dataset. *CIAM* achieves significantly better performance in terms of classification accuracy throughout all the learning sessions (and consequently higher average overall classification accuracy), as well as better performance retention levels. A higher number of few-shot sessions (compared to the previous two datasets) has contributed to a more considerable improvement in terms of the average overall classification accuracy.

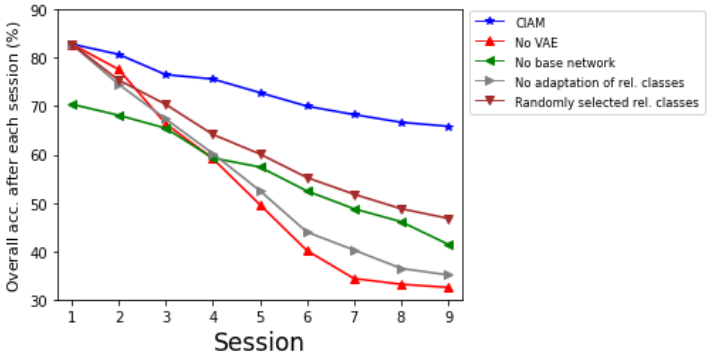

Figure 5: Ablations on miniImageNet. The significant differences in the performance levels, throughout all the learning sessions, between *CIAM* and the other four scenarios demonstrate the impactful role of its modeling components, and emphasize the value added by the proposed adaptation strategy.

to the work in (Yang et al., 2023) which, to the best of our knowledge, represents the sole few-shot class-incremental learning work with reported results on the MedMNIST dataset. The experiments performed on MedMNIST in (Derakhshani et al., 2022) follow a vanilla class-incremental learning setting with abundant data available during every incremental session. This is different from the adopted few-shot class-incremental learning setting with the immensely challenging aspect of having to learn from very few (one data point per class, as will be clarified shortly) data points during every incremental few-shot session. The minibatch size is 512 during the base session, and is 64 during each incremental few-shot session. The number of epochs is 150 epochs during the base session and 80 during every subsequent incremental few-shot session.

Following previous works on MedMNIST (Yang et al., 2023), we pursue an FSCIL setting consisting of 6 selected medical disease classification datasets. The classes involved in 3 (out of the 6) medical datasets are selected as the classes of the base session (with a total of 27 classes), whereas

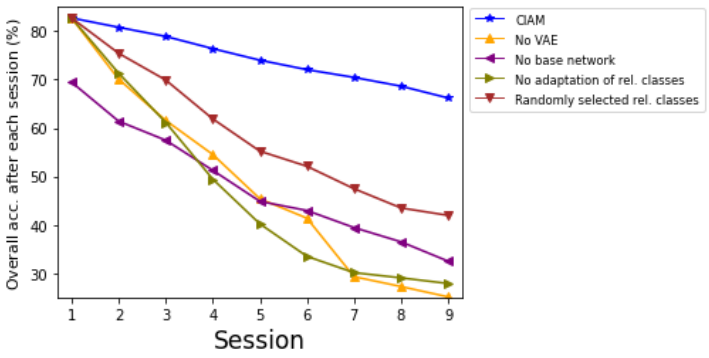

Figure 6: Ablations on CIFAR100. The results demonstrate that every modeling component of *CIAM* is influential. This is witnessed by the considerable differences with the other four scenarios.

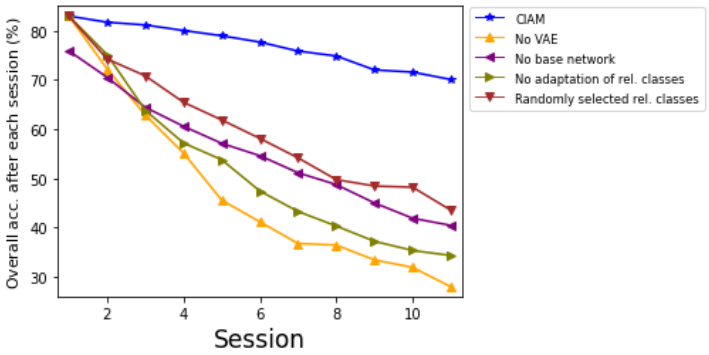

Figure 7: Ablations on CUB200. There are massive differences in performance between *CIAM* and the other four scenarios. This underlines the worth of every modeling component, including the proposed latent variable modeling-based adaptation strategy.

the classes of the other 3 datasets serve as the incremental few-shot classes (with a total of 15 classes). The base session datasets are: PathMNIST (Kather et al., 2019), DermaMNIST (Tschandl et al., 2018) and OrganAMNIST (Bilic et al., 2023). The few-shot datasets are: BloodMNIST (Acevedo et al., 2020), BreastMNIST (Al-Dhabyani et al., 2020) and RetinaMNIST (Dataset, 2020). After the base session which consists of 27 classes. We proceed with 15 1-way 1-shot incremental few-shot sessions each containing one out of the 15 few-shot classes.

PathMNIST (Kather et al., 2019) is a 9-class dataset which was collected for predicting survival from colorectal cancer histology slides. It consists of a total of $107,180$ images divided into $100,000$ training images and $7,180$ test images. DermaMNIST (Tschandl et al., 2018) is a 7-class dataset where the classes denote common pigmented skin lesions. It contains a total of $10,015$ images, $8,010$ for training and $2,005$ for the test procedure. OrganAMNIST is a 11-class dataset which is based on images from the Liver Tumor Segmentation benchmark (LiTS) (Bilic et al., 2023). The classes represent 11 body organs. The size of the dataset is $58,850$ images, divided into $41,072$ training and $17,778$ test images. The BloodMNIST (Acevedo et al., 2020) dataset contains 8 classes and is captured from individuals with either no infection, hematologic, or oncologic disease. It consists of a total of $17,092$ images divided into $13,671$ training and $3,421$ test images. BreastMNIST

(Al-Dhabyani et al., 2020) is a 2-class dataset of 780 breast ultrasound images. It was originally categorized into 3 classes: normal, benign, and malignant. The task was later simplified (Yang et al., 2021a) into binary classification by amalgamating the normal and benign classes together as one class, against the malignant class. It is divided into 624 training and 156 test images. RetinaMNIST (Dataset, 2020) is a 5-class dataset of retina fundus images whose original task was to predict a level out of a 5-level grading of diabetic retinopathy severity. It consists of $1,600$ images, divided into $1,200$ training and 400 test images.

The results of the experiments performed on the medical benchmark MedMNIST can be found in Table 4 and in Figure 8. *CIAM* achieves state-of-the-art average overall classification accuracy, final classification accuracy and PD rate. The final classification accuracy is 4.54% higher than the previous state-of-the-art. The forgetting rate, measured by PD (4.43% lower than the previous lowest PD), is particularly important with the MedMNIST medical benchmark for two reasons: i) this dataset of real-world images provides a proper simulation of a realistic scenario where an FSCIL framework must learn how to both adapt and be stable (i.e. not forget the previous knowledge), and ii) the degree of performance retention achieved by *CIAM* is accomplished in a rigorous few-shot setting with 15 1-way 1-shot sessions with one data point available per class. In addition, it can be noticed from Figure 8 that the last few sessions of the FSCIL procedure have witnessed a more significant superiority of *CIAM*, which demonstrates its ability to learn from a longer sequence of new classes consisting of real-world data, like MedMNIST. This is of paramount importance due to the widespread nature of real-world scenarios like the one studied here in the medical domain.

The average training run-time of *CIAM* on the miniImageNet dataset is 5.4 minutes on miniImageNet, 7.1 minutes on CIFAR100, 2.2 minutes on CUB200 and 14.8 minutes on MedMNIST. The run-time results, which are considered very compelling in the CIL paradigm, are mainly thanks to the proposed mechanism of dividing the training burden into a pre-trained representation during the base session phase, followed by VAE training phase which solely acts during the few-shot sessions and which benefits from the aforementioned pre-training phase.

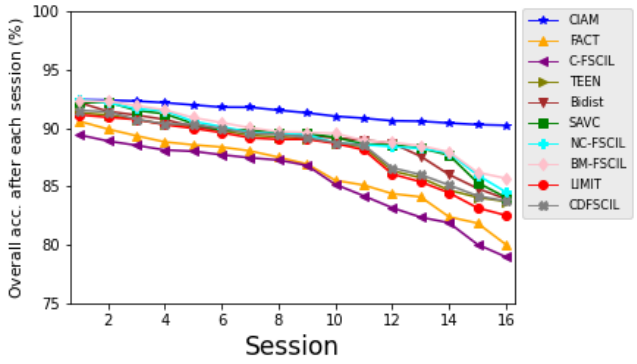

Figure 8: Performance of *CIAM* vs. 9 other FSCIL algorithms on the medical dataset MedMNIST. *CIAM* achieves higher classification accuracy results as well as significantly better performance retention levels in the rigorous few-shot setting including 15 1-way 1-shot sessions after the base session.

Table 4: Classification accuracy after learning from sessions (1 base session + 15 incremental few-shot sessions) of the MedMNIST dataset where each incremental few-shot session is 1-way 1-shot. A full illustration (i.e. with sessions $1, 2, \ldots 16$) is also provided in Figure 8. The classification accuracy after session 16 is the final classification accuracy. This is followed by the average overall classification accuracy and the performance dropping rate (PD) which is the catastrophic forgetting metric. Results are displayed for the proposed *CIAM* along with 9 other FSCIL algorithms. A bold entry indicates significance. *CIAM* achieves state-of-the-art results in terms of the aforementioned metrics. The forgetting rate, measured by PD, is of particular significance given the fact that this degree of performance retention (PD = 2.2) is achieved in a very rigorous few-shot setting where all the incremental few-shot sessions are 1-way 1-shot with solely one data point each.

Method	Overall acc. after each session (%)									Avg. overall acc.	PD
	1	3	5	7	9	11	13	15	16		
FACT	90.58	89.34	88.56	88.04	86.92	85.11	84.10	81.85	80.02	86.34	10.56
C-FSCIL	89.41	88.52	88.01	87.45	86.80	84.19	82.36	80.03	78.96	85.50	10.45
TEEN	91.30	90.74	90.16	89.38	89.11	88.38	85.74	84.06	83.71	88.30	7.59
Bidist	92.18	91.11	90.14	89.80	89.37	88.97	87.58	84.83	83.92	88.95	8.26
SAVC	92.15	91.51	90.38	89.81	89.53	88.63	88.28	85.33	84.05	89.25	8.10
NC-FSCIL	92.42	91.65	90.57	89.63	89.45	88.58	88.26	85.97	84.52	89.33	7.90
BM-FSCIL	92.32	91.92	90.89	90.02	89.62	88.93	88.49	86.20	85.69	89.65	6.63
LIMIT	91.13	90.73	89.97	89.16	89.04	88.13	85.39	83.16	82.52	88.02	8.61
CDFSCIL	91.51	90.75	90.33	89.55	89.18	88.54	86.01	84.19	83.75	88.46	7.76
CIAM (Ours)	92.43	**92.31**	**91.98**	**91.78**	**91.30**	**90.84**	**90.58**	**90.29**	**90.23**	**91.36**	**2.20**

5. Related Work

In incremental (continual) learning, training is divided into a set of subsequent tasks. During any training session, the learner can typically access the data of the current task only. Class-incremental learning (CIL) refers to the more challenging scenario of incremental learning where the learner has no access to the task ID at test (inference) time (Masana et al., 2022). Therefore, a class-incremental learner must learn how to distinguish between all the encountered classes from all tasks (Hou et al., 2019; Yu et al., 2020; Mai et al., 2021; Shim et al., 2021; Zhu et al., 2021; Liu et al., 2023; Rymarczyk et al., 2023; Wen et al., 2023; Zhou et al., 2023a). A class-incremental learner must also have the ability to continually learn new classes without forgetting the old (i.e. previously encountered) ones. Widely used approaches in CIL include replay strategies where several representative instances of the previously encountered classes are either stored or generated, and then replayed after encountering new classes so that the old knowledge can be maintained (Liu et al., 2020; Cha et al., 2021; Masana et al., 2022). Another approach, which is usually combined with replay strategies, is referred to as knowledge distillation. In knowledge distillation (Li & Hoiem, 2016; Rebuffi et al., 2017; Wu et al., 2019; Douillard et al., 2020; Cheraghian et al., 2021a; Kang et al., 2022b), the relationship between the representation changes along different tasks and the resulting losses is estimated. The objective is to enforce consistent loss changes such that the old representations do not change massively while adjusting to new tasks. A third CIL approach is based on expanding the model along with the arrival of new classes (Liu et al., 2021; Yan et al., 2021; Wang et al., 2023). Examples include the work in (Yan et al., 2021) where at each incremental step the previously learned representation is frozen and augmented with additional feature dimensions from a new feature extractor.

The challenging difficulties associated with CIL are further exacerbated when solely a scarce amount of data is available for each incremental class, which is the setting we adopt herein and is referred to as few-shot class-incremental learning (FSCIL). After training on a base session with sufficient sample sizes for every class, an FSCIL learner then encounters incremental few-shot sessions with limited sample sizes per class (Chen & Lee, 2020; Tao et al., 2020; Zhu et al., 2021; Liu et al., 2022; Tan et al., 2022; Zhu et al., 2022; Zou et al., 2022; Zhang et al., 2023; Zhuang et al., 2023; Ran et al., 2024; Tian et al., 2024).

Due to the scarcity of available data for all the incremental few-shot classes, the traditional CIL strategies cannot perform efficiently in FSCIL. Instead, after the base session, the bulk of FSCIL algorithms decouple the representation learning part from the classifiers. The feature extractor is often trained to learn representations during the base session, and is then frozen. The incremental few-shot sessions typically proceed with the sole task of optimizing the classifiers (Zhang et al., 2021; Akyurek et al., 2022). In addition to this decoupling which results in a pre-trained backbone, a non-parametric class mean classifier is deployed in (Zhang et al., 2021) to alleviate catastrophic forgetting. Pre-trained feature extractors are combined with subspace regularization schemes in (Akyurek et al., 2022) to encourage weight vectors of new classes to lie close to the space spanned by the weights of the previously encountered classes. The C-FSCIL algorithm in (Hersche et al., 2022) combines a frozen, meta-learned feature extractor with a rewritable and dynamically growing memory which stores vectors for all the encountered classes.

Other previous works on FSCIL are based on establishing representative prototypes of the few-shot classes. The algorithm presented in (Mazumder et al., 2021) aims to address catastrophic forgetting by minimizing the cosine similarity between the prototypes of the new classes and those of the old classes. It also adopts a selection strategy to choose a subset of the model parameters to train the new classes on, rather than training them on the whole model. Flat local minima of the objective function of the base session training is sought in (Shi et al., 2021). After the base session (i.e. during the incremental few-shot sessions), the prototypes are normalized and the model parameters are fine-tuned within the flat region. The FACT algorithm proposes a strategy for learning prospectively to prepare for future updates by assigning virtual prototypes and reserving embedding space for new classes (Zhou et al., 2022). A loss referred to as the Prototype Smoothing Hard-mining Triplet (PSHT) loss is developed in (Ji et al., 2023) to push the novel prototypes away from one another as well as from the prototypes of the old classes. The class prototypes are computed in (Cheraghian et al., 2021b) via a subspace computation strategy based on clustering in the (image) feature space to relate base and few-shot classes.

In spite of some success in alleviating catastrophic forgetting, the aforementioned approach of completely freezing the feature extractor and solely computing prototypes during the incremental few-shot sessions has been shown to be prone to potential bias basically due to the class imbalance between base and few-shot classes (Wang et al., 2023b; Zhao et al., 2024). A set of works have proposed approaches to mitigate this bias based on calibrating the class prototypes (Zhu et al., 2021; Deng et al., 2022; Wang et al., 2023b; Zhang & Gu, 2023; Zhou et al., 2023b; Zhu et al., 2023b). The incremental prototype learning scheme presented in (Zhu et al., 2021) consists of a random episode selection strategy that aligns the feature extractor with various generated incremental episodes, and a self-promoted prototype refinement mechanism to enhance the expressive ability of new classes. A prototype calibration strategy, which forms the main part of TEEN (Wang et al., 2023b), fuses the prototypes of the new classes with weighted base prototypes to enhance the discriminability of the new classes. However, they only update the prototype of the current class, leaving all the similar

previous classes unchanged, which can result in making the previous classes prone to potential bias. A technique to replay and calibrate prototypes, which is based on a series of augmentations by rotations and nonlinear transformations, is presented in (Zhang & Gu, 2023). New class prototypes and old class classifiers are calibrated into the same scale via a transformer-based calibration module in the LIMIT algorithm (Zhou et al., 2023b).

As argued in (Zhao et al., 2024), balanced and effective incremental classification is still tricky to achieve with freezing and/or prototype calibration. A balanced classification in this context refers to an FSCIL classifier which is not severely biased towards neither the base classes nor the incremental few-shot classes. It is also important to effectively utilise the few samples available for the incremental few-shot classes to express such classes with high fidelity. We conjecture that the stiff separation between the feature extractor and the classifier can play a role in enforcing this bias. Hence, our work can be viewed as a foundation of another representation, which is the latent representation $\mathbf{z}$, that acts as an intermediary between the feature extractor (i.e. the base representation) and the classifier. This latent representation is optimized in the proposed manner to adapt the representation of not only the current class but also the relevant previous classes, such that catastrophic forgetting as well as potential bias can be alleviated.

Other FSCIL algorithms include the work in (Tao et al., 2020) which uses a neural gas network to preserve the topology of the feature manifold of all the classes. The MetaFSCIL algorithm is based on meta-learning where a bi-level optimization procedure optimizes the network by sampling sequences of incremental tasks from the base classes and using such samples to simulate the evaluation protocol (Chi et al., 2022). The method in (Achituve et al., 2021) develops a growing tree-based hierarchical model in which each internal node fits a Gaussian process. A variational autoencoder was trained as part of a CIL model in (van de Ven et al., 2021). However, this was designed for the vanilla (not few-shot) CIL paradigm. More importantly, it encounters massive scalability issues since a completely new variational autoencoder is trained (and tested) for each and every new class. In addition, the cost of inference is further exacerbated by the need for $10,000$ importance samples for each likelihood estimation. The ALICE algorithm (Peng et al., 2022) is based on replacing the cross-entropy loss with an angular penalty loss to obtain more clustered features, adding a margin to improve discriminability, and combining this with augmentation mechanisms. The SAVC algorithm developed in (Song et al., 2023) is a virtual contrastive model which separates the base classes from the new few-shot classes by introducing virtual classes.

6. Conclusion

We introduced a few-shot class-incremental learning framework to address the main challenges of this paradigm, like catastrophic forgetting, potential bias and overfitting. The principal part of the proposed framework involves a latent variable model that we developed such that, upon the arrival of a new few-shot class, it enforces a consistent adaptation process covering not only the new class, but also the previous relevant classes, in light of the new knowledge that the class-incremental learner has just acquired. We also derived a generalization upper bound on the error of an upcoming class, providing theoretical evidence of the proposed strategy. Efficacy of the proposed adaptation strategy, measured by several performance metrics like the average overall classification accuracy, final classification accuracy, and the performance dropping rate to assess catastrophic forgetting, is demonstrated via powerful empirical performance over three widely used few-shot class-incremental learning benchmarks. In addition, the state-of-the-art accuracy and performance retention results

achieved on a real-world medical benchmark demonstrate the capability of the proposed adaptation strategy to learn in real-world scenarios.

References

Acevedo, A., Merino, A., Alferez, S., Molina, A., Boldu, L., & Rodellar, J. (2020). A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in brief*, *30*.

Achille, A., Eccles, T., Matthey, L., Burgess, C., Watters, N., Lerchner, A., & Higgins, I. (2018). Life-long disentangled representation learning with cross-domain latent homologies. *Advances in Neural Information Processing Systems (NIPS)*.

Achituve, I., Navon, A., Yemini, Y., Chechik, G., & Fetaya, E. (2021). GP-Tree: A Gaussian process classifier for few-shot incremental learning. *International Conference on Machine Learning (ICML)*.

Adel, T. (2024). Similarity-based adaptation for task-aware and task-free continual learning. *Journal of Artificial Intelligence Research (JAIR)*, *80*, 377–417.

Adel, T., Ghahramani, Z., & Weller, A. (2018). Discovering interpretable representations for both deep generative and discriminative models. *International Conference on Machine Learning (ICML)*.

Adel, T., Zhao, H., & Turner, R. (2020). Continual learning with adaptive weights (CLAW). *International Conference on Learning Representations (ICLR)*.

Ahmad, T., Dhamija, A., Jafarzadeh, M., Cruz, S., Rabinowitz, R., Li, C., & Boult, T. (2022). Variable few shot class incremental and open world learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 3688–3699.

Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., & Marchand, M. (2014). Domain adversarial neural networks. *CoRR, abs/1412.4446*.

Akyurek, A., Akyurek, E., Wijaya, D., & Andreas, J. (2022). Subspace regularizers for few-shot class incremental learning. *International Conference on Learning Representations (ICLR)*.

Al-Dhabyani, W., Gomaa, M., Khaled, H., & Fahmy, A. (2020). Dataset of breast ultrasound images. *Data in brief*, *28*.

Banayeeanzade, M., Mirzaiezadeh, R., Hasani, H., & Baghshah, M. (2021). Generative vs discriminative: Rethinking the meta-continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*.

Belouadah, E., & Popescu, A. (2019). IL2M: Class incremental learning with dual memory. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Ben-David, S., Blitzer, S., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. (2010). A theory of learning from different domains. *Machine learning*, *79*(2), 151–175.

Bilic, P., Christ, P., Li, H., Vorontsov, E., Ben-Cohen, A., Kaissis, G., Szeskin, A., Jacobs, C., Mamani, G., Chartrand, G., & et al. (2023). The liver tumor segmentation benchmark (LiTS). *Medical Image Analysis*, *84*.

Blei, D., Kucukelbir, A., & McAuliffe, J. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*, 859–877.

Castro, F., Marin-Jimenez, M., Guil, N., Schmid, C., & Alahari, K. (2018). End-to-end incremental learning. *European Conference on Computer Vision (ECCV)*.

Cha, S., Kim, B., Yoo, Y., & Moon, T. (2021). SSUL: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *Advances in Neural Information Processing Systems (NeurIPS)*.

Chaudhry, A., Ranzato, M., Rohrbach, M., & Elhoseiny, M. (2019). Efficient lifelong learning with A-GEM. *International Conference on Learning Representations (ICLR)*.

Chen, K., & Lee, C. (2020). Incremental few-shot learning via vector quantization in deep embedded space. *International Conference on Learning Representations (ICLR)*.

Cheraghian, A., Rahman, S., Fang, P., Roy, S., Petersson, L., & Harandi, M. (2021a). Semantic-aware knowledge distillation for few-shot class-incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2534–2543.

Cheraghian, A., Rahman, S., Ramasinghe, S., Fang, P., Simon, C., Petersson, L., & Harandi, M. (2021b). Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8661–8670.

Chi, Z., Gu, L., Liu, H., Wang, Y., Yu, Y., & Tang, J. (2022). MetaFSCIL: A meta-learning approach for few-shot class incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 14166–14175.

Dataset, D. R. I. (2020). The 2nd diabetic retinopathy–grading and image quality estimation challenge. *https://isbi.deepdr.org/data.html*.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, J., Zhang, H., Hu, J., & Wang, Y. (2022). Incremental prototype tuning for class incremental learning. *arXiv preprint arXiv:2204.03410*.

Derakhshani, M., Najdenkoska, I., van Sonsbeek, T., Zhen, X., Mahapatra, D., Worring, M., & Snoek, C. (2022). Lifelonger: A benchmark for continual disease classification. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 314–324.

Diaz-Rodriguez, N., Lomonaco, V., Filliat, D., & Maltoni, D. (2018). Don't forget, there is more than forgetting: new metrics for Continual Learning. *NIPS Continual Learning Workshop*.

Douillard, A., Cord, M., Ollion, C., Robert, T., & Valle, E. (2020). PODNet: Pooled outputs distillation for small-tasks incremental learning. *European Conference on Computer Vision (ECCV)*.

Ebrahimi, S., Elhoseiny, M., Darrell, T., & Rohrbach, M. (2020). Uncertainty-guided continual learning with Bayesian neural networks. *International Conference on Learning Representations (ICLR)*.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning (ICML)*, *34*.

French, R. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, *3*, 128–135.

Gidaris, S., & Komodakis, N. (2018). Dynamic few-shot visual learning without forgetting. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gu, Z., Xu, C., Yang, J., & Cui, Z. (2023). Few-shot continual infomax learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Gupta, P., Chaudhary, Y., Runkler, T., & Schutze, H. (2020). Neural topic modeling with continual lifelong learning. *International Conference on Machine Learning (ICML)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hersche, M., Karunaratne, G., Cherubini, G., Benini, L., Sebastian, A., & Rahimi, A. (2022). Constrained few-shot class-incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9047–9057.

Hou, S., Pan, X., Loy, C., Wang, Z., & Lin, D. (2019). Learning a unified classifier incrementally via rebalancing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, W., Lin, Z., Liu, B., Tao, C., Tao, Z., Ma, J., Zhao, D., & Yan, R. (2019). Overcoming catastrophic forgetting via model adaptation. *International Conference on Learning Representations (ICLR)*.

Ji, Z., Hou, Z., Liu, X., Pang, Y., & Li, X. (2023). Memorizing complementation network for few-shot class-incremental learning. *IEEE Transactions on Image Processing*, *32*, 937–948.

Kalb, T., & Beyerer, J. (2022). Causes of catastrophic forgetting in class-incremental semantic segmentation. *Proceedings of the Asian Conference on Computer Vision*, 56–73.

Kang, H., Mina, R., Madjid, S., Yoon, J., Hasegawa-Johnson, M., Hwang, S., & Yoo, C. (2022a). Forget-free continual learning with winning subnetworks. *International Conference on Machine Learning (ICML)*.

Kang, M., Park, J., & Han, B. (2022b). Class-incremental learning by knowledge distillation with adaptive feature consolidation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 16071–16080.

Karakida, R., & Akaho, S. (2022). Learning curves for continual learning in neural networks: Self-knowledge transfer and forgetting. *International Conference on Learning Representations (ICLR)*.

Kather, J., Krisam, J., Charoentong, P., Luedde, T., Herpel, E., Weis, C., Gaiser, T., Marx, A., Valous, N., Ferber, D., & et al. (2019). Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, *16*.

Ke, Z., Liu, B., Ma, N., Xu, H., & Shu, L. (2021). Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*.

Kemker, R., McClure, M., Abitino, A., Hayes, T., & Kanan, C. (2018). Measuring catastrophic forgetting in neural networks. *AAAI Conference on Artificial Intelligence*, *32*.

Kim, J., Oh, T., Lee, S., Pan, F., & Kweon, I. (2019). Variational prototyping-encoder: One-shot learning with prototypical images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kingma, D., Rezende, D., Mohamed, S., & Welling, M. (2014). Semi-supervised learning with deep generative models. *Advances in Neural Information Processing Systems (NIPS)*, *28*, 3581–3589.

Kingma, D., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems (NIPS)*, *30*.

Kingma, D., & Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*.

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*.

Li, Z., & Hoiem, D. (2016). Learning without forgetting. *European Conference on Computer Vision (ECCV)*.

Liao, H., & He, J. (2021). Jacobian determinant of normalizing flows. *arXiv preprint arXiv:2102.06539*.

Lin, S., Yang, L., Fan, D., & Zhang, J. (2022). TRGP: Trust region gradient projection for continual learning. *International Conference on Learning Representations (ICLR)*.

Liu, H., Gu, L., Chi, Z., Wang, Y., Yu, Y., Chen, J., & Tang, J. (2022). Few-shot class-incremental learning via entropy-regularized data-free replay. *European Conference on Computer Vision (ECCV)*.

Liu, X., Wu, C., Menta, M., Herranz, L., Raducanu, B., Bogdanov, A., Jui, S., & van de Weijer, J. (2020). Generative feature replay for class-incremental learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Liu, Y., Li, Y., Schiele, B., & Sun, Q. (2023). Online hyperparameter optimization for class-incremental learning. *arXiv preprint arXiv:2301.05032*.

Liu, Y., Schiele, B., & Sun, Q. (2021). Adaptive aggregation networks for class-incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2544–2553.

Mai, Z., Li, R., Kim, H., & Sanner, S. (2021). Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 3589–3599.

Masana, M., Liu, X., Twardowski, B., Menta, M., Bogdanov, A., & van de Weijer, J. (2022). Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI)*, *45*(5), 5513–5533.

Mazumder, P., Singh, P., & Rai, P. (2021). Few-shot lifelong learning. *AAAI Conference on Artificial Intelligence*.

McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*.

Miao, Z., Wang, Z., Chen, W., & Qiu, Q. (2022). Continual learning with filter atom swapping. *International Conference on Learning Representations (ICLR)*.

Mocanu, D., & Mocanu, E. (2018). One-shot learning using mixture of variational autoencoders: A generalization learning approach. *arXiv preprint arXiv:1804.07645*.

Ostapenko, O., Rodriguez, P., Caccia, M., & Charlin, L. (2021). Continual learning via local module composition. *Advances in Neural Information Processing Systems (NeurIPS)*.

Papamakarios, G., Nalisnick, E., Rezende, D., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research (JMLR)*, *22*, 2617–2680.

Pape, L., Gomez, F., Ring, M., & Schmidhuber, J. (2011). Modular deep belief networks that do not forget. *IEEE International Joint Conference on Neural Networks (IJCNN)*.

Parisi, G., Kemker, R., Part, J., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*.

Peng, C., Zhao, K., Wang, T., Li, M., & Lovell, B. (2022). Few-shot class-incremental learning from an open-set perspective. *European Conference on Computer Vision (ECCV)*.

Pfulb, B., & Gepperth, A. (2019). A comprehensive, application-oriented study of catastrophic forgetting in DNNs. *International Conference on Learning Representations (ICLR)*.

Ran, H., Li, W., Li, L., Tian, S., Ning, X., & Tiwari, P. (2024). Learning optimal inter-class margin adaptively for few-shot class-incremental learning via neural collapse-based meta-learning. *Information Processing and Management*, *61*.

Ratcliff, R. (1990). Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review*.

Rebuffi, S., Kolesnikov, A., Sperl, G., & Lampert, C. (2017). iCaRL: Incremental classifier and representation learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rezende, D., & Mohamed, S. (2015). Variational inference with normalizing flows. *International Conference on Machine Learning (ICML)*, *32*, 1530–1538.

Rezende, D., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning (ICML)*, *31*.

Ring, M. (1995). *Continual learning in reinforcement environments*. Ph.D. thesis, University of Texas, Austin.

Robins, A. (1993). Catastrophic forgetting in neural networks: The role of rehearsal mechanisms. *IEEE Artificial Neural Networks and Expert Systems*, 65–68.

Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, *7*, 123–146.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*.

Rymarczyk, D., van de Weijer, J., Zielinski, B., & Twardowski, B. (2023). ICICLE: Interpretable Class Incremental Continual Learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., & Akata, Z. (2019). Generalized zero-and few-shot learning via aligned variational autoencoders. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8247–8255.

Schwarz, J., Luketina, J., Czarnecki, W., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., & Hadsell, R. (2018). Progress & compress: A scalable framework for continual learning. *International Conference on Machine Learning (ICML)*.

Shi, G., Chen, J., Zhang, W., Zhan, L., & Wu, X. (2021). Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. *Advances in Neural Information Processing Systems (NeurIPS)*.

Shim, D., Mai, Z., Jeong, J., Sanner, S., Kim, H., & Jang, J. (2021). Online class-incremental continual learning with adversarial Shapley value. *AAAI Conference on Artificial Intelligence*.

Song, Z., Zhao, Y., Shi, Y., Peng, P., Yuan, L., & Tian, Y. (2023). Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 24183–24192.

Srivastava, R., Masci, J., Kazerounian, S., Gomez, F., & Schmidhuber, J. (2013). Compete to compute. *Advances in Neural Information Processing Systems (NIPS)*.

Sun, Q., Chattopadhyay, R., Panchanathan, S., & Ye, J. (2011). A two-stage weighting framework for multi-source domain adaptation. *Advances in Neural Information Processing Systems (NIPS)*.

Tabak, E., & Turner, C. (2013). A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, *66*(2), 145–164.

Tabak, E., & Vanden-Eijnden, E. (2010). Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, *8*(1), 217–233.

Tan, Z., Ding, K., Guo, R., & Liu, H. (2022). Graph few-shot class-incremental learning. *ACM International Conference on Web Search and Data Mining (WSDM)*.

Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., & Gong, Y. (2020). Few-shot class-incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tian, S., Li, L., Li, W., Ran, H., Ning, X., & Tiwari, P. (2024). A survey on few-shot class-incremental learning. *Neural Networks*.

Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, *5*, 1–9.

van de Ven, G., Li, Z., & Tolias, A. (2021). Class-incremental learning with generative classifiers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 3611–3620.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. *Advances in Neural Information Processing Systems (NIPS)*.

Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). The Caltech-UCSD Birds-200-2011 dataset. .

Wang, F., Zhou, D., Liu, L., Ye, H., Bian, Y., Zhan, D., & Zhao, P. (2023). Beef: Bi-compatible class-incremental learning via energy-based expansion and fusion. *International Conference on Learning Representations (ICLR)*.

Wang, L., Zhang, M., Jia, Z., Li, Q., Ma, K., Bao, C., Zhu, J., & Zhong, Y. (2021). AFEC: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*.

Wang, L., Zhang, X., Su, H., & Zhu, J. (2023a). A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*.

Wang, Q., Zhou, D., Zhang, Y., Zhan, D., & Ye, H. (2023b). Few-shot class-incremental learning via training-free prototype calibration. *Advances in Neural Information Processing Systems (NeurIPS)*.

Wen, H., Cheng, H., Qiu, H., Wang, L., Pan, L., & Li, H. (2023). Optimizing mode connectivity for class incremental learning. *International Conference on Machine Learning (ICML)*.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., & Fu, Y. (2019). Large scale incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xiang, L., Jin, X., Ding, G., Han, J., & Li, L. (2019). Incremental few-shot learning for pedestrian attribute recognition. *International Joint Conference on Artificial Intelligence (IJCAI)*.

Yan, S., Xie, J., & He, X. (2021). DER: Dynamically expandable representation for class incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3014–3023.

Yang, B., Lin, M., Zhang, Y., Liu, B., Liang, X., Ji, R., & Ye, Q. (2022). Dynamic support network for few-shot class incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI)*, *45*(3), 2945–2951.

Yang, H., Huang, W., Liu, J., Li, C., & Wang, S. (2023). Few-shot class-incremental learning for cross-domain disease classification. *arXiv preprint arXiv:2304.05734*.

Yang, J., Shi, R., & Ni, B. (2021a). MedMNIST classification decathlon: A lightweight AutoML benchmark for medical image analysis. *IEEE International Symposium on Biomedical Imaging (ISBI)*, *18*.

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., & Ni, B. (2021b). MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, *10*.

Yang, Y., Yuan, H., Li, X., Lin, Z., Torr, P., & Tao, D. (2023). Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. *International Conference on Learning Representations (ICLR)*.

Yasar, M., & Iqbal, T. (2023). CoRaL: Continual representation learning for overcoming catastrophic forgetting. *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., & van de Weijer, J. (2020). Semantic drift compensation for class-incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zeno, C., Golan, I., Hoffer, E., & Soudry, D. (2018). Task agnostic continual learning using online variational Bayes. *NIPS Bayesian Deep Learning Workshop*.

Zhang, C., Song, N., Lin, G., Zheng, Y., Pan, P., & Xu, Y. (2021). Few-shot incremental learning with continually evolved classifiers. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 12455–12464.

Zhang, J., Liu, L., Silven, O., Pietikainen, M., & Hu, D. (2023). Few-shot class-incremental learning: A survey. *arXiv preprint arXiv:2308.06764*.

Zhang, W., & Gu, X. (2023). Few shot class incremental learning via efficient prototype replay and calibration. *Entropy*, *25*.

Zhao, H., des Combes, R. T., Zhang, K., & Gordon, G. (2019). On learning invariant representations for domain adaptation. *International Conference on Machine Learning (ICML)*.

Zhao, H., Fu, Y., Kang, M., Tian, Q., Wu, F., & Li, X. (2021). MgSvF: Multi-grained slow vs. fast framework for few-shot class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI)*.

Zhao, L., Chen, Z., Zhang, Z., Luo, X., & Xu, X. (2024). Bias mitigating few-shot class-incremental learning. *arXiv preprint arXiv:2402.00481*.

Zhao, L., Lu, J., Xu, Y., Cheng, Z., Guo, D., Niu, Y., & Fang, X. (2023). Few-shot class-incremental learning via class-aware bilateral distillation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 11838–11847.

Zhou, D., Wang, F., Ye, H., Ma, L., Pu, S., & Zhan, D. (2022). Forward compatible few-shot class-incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9036–9046.

Zhou, D., Wang, Q., Qi, Z., Ye, H., Zhan, D., & Liu, Z. (2023a). Deep Class-Incremental Learning: A Survey. *arXiv preprint arXiv:2302.03648*.

Zhou, D., Ye, H., Ma, L., Xie, D., Pu, S., & Zhan, D. (2023b). Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI)*, *45*(11).

Zhou, Z. (2022). Open-environment machine learning. *National Science Review*, *9*(8).

Zhu, F., Cheng, Z., Zhang, X., & Liu, C. (2021). Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhu, J., Yao, G., Zhou, W., Zhang, G., Ping, W., & Zhang, W. (2022). Feature distribution distillation-based few shot class incremental learning. *International Conference on Pattern Recognition and Artificial Intelligence*.

Zhu, K., Cao, Y., Zhai, W., Cheng, J., & Zha, Z. (2021). Self-promoted prototype refinement for few-shot class-incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6801–6810.

Zhu, X., Hao, J., Guo, Y., & Liu, M. (2023a). AUC maximization in imbalanced lifelong learning. *Uncertainty in Aritifical Intelligence (UAI)*.

Zhu, Z., Wang, P., Diao, W., Yang, J., & Wang, H. (2023b). Few-shot incremental learning with continual prototype calibration for remote sensing image fine-grained classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 180–191.

Zhuang, H., Weng, Z., He, R., Lin, Z., & Zeng, Z. (2023). GKEAL: Gaussian kernel embedded analytic learning for few-shot class incremental task. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zou, Y., Zhang, S., Li, Y., & Li, R. (2022). Margin-based few-shot class-incremental learning with class-level overfitting mitigation. *Advances in Neural Information Processing Systems (NeurIPS)*.