Bayesian Learning-driven Prototypical Contrastive Loss for Class-Incremental Learning

Anonymous authors
Paper under double-blind review

Abstract

The primary objective of methods in continual learning is to learn tasks in a sequential manner over time (sometimes from a stream of data), while mitigating the detrimental phenomenon of catastrophic forgetting. This paper proposes a method to learn an effective representation between previous and newly encountered class prototypes. We propose a prototypical network with a Bayesian learning-driven contrastive loss (BLCL), tailored specifically for class-incremental learning scenarios. We introduce a contrastive loss that incorporates novel classes into the latent representation by reducing intra-class and increasing inter-class distance. Our approach dynamically adapts the balance between the cross-entropy and contrastive loss functions with a Bayesian learning technique. Experimental results conducted on both the CIFAR-10 and CIFAR-100 dataset for image classification and images of a GNSS-based dataset for interference classification validate the efficacy of our method, showcasing its superiority over existing state-of-the-art approaches.

1 Introduction

Numerous practical vision applications require the learning of new visual capabilities while maintaining high performance on existing ones. Examples include construction safety, employing reinforcement learning methodologies (Kirkpatrick et al., 2017), or adapting to novel types of interference (as image classification task) in global navigation satellite system (GNSS) operations (Ott et al., 2024; Brieger et al., 2022; Raichur et al., 2022; van der Merwe et al., 2023; Heublein et al., 2024; Gaikwad et al., 2024). *Continual Learning* describes the setting of sequentially learning novel tasks while avoiding *catastrophic forgetting*, i.e., forgetting how to perform well on previously seen tasks (Kirkpatrick et al., 2017), which commonly occurs when models are trained successively on multiple tasks without revisiting earlier tasks. Recent methodologies try to circumvent catastrophic forgetting, for instance by utilization of feature extraction and fine-tuning adaptation techniques (Li & Hoiem, 2018), by leveraging off-policy learning and behavioral cloning from replay to enhance stability (Rolnick et al., 2019), or by selectively decelerating learning on weights important for specific tasks (Kirkpatrick et al., 2017).

Previous work (Zhou et al., 2023b) investigated the effect of model size of the specialized components, i.e., shallow and deep layers, and concluded that deeper layers obtain diverse feature representations and is crucial for learning novel tasks. Specifically, the learned data representations between preceding and novel tasks hold considerable importance (Rebuffi et al., 2017). This serves as a motivation for our focus on refining the architecture of specialized blocks and enhancing dynamic representations for continual learning. Our proposed method, BLCL, enhances task adaptability by dynamically expanding specialized blocks. These blocks, optimized through a targeted hyperparameter search, focus primarily on task-specific learning.

A related yet distinct task is *contrastive learning*, which pairs positive and negative instances with corresponding anchors to enrich training (Chen et al., 2020; Ott et al., 2023). While continual learning with contrastive techniques is prevalent in semantic tasks (Ke et al., 2021), contrastive learning's application in continual learning for image classification tasks is limited (Cha et al., 2021). This limitation arises from the potential decrease in diversity of negative samples, affecting the task-specific loss, and the likelihood of the regularizer may hinder learning of new distinctive representations (Cha & Moon, 2023). Recognizing

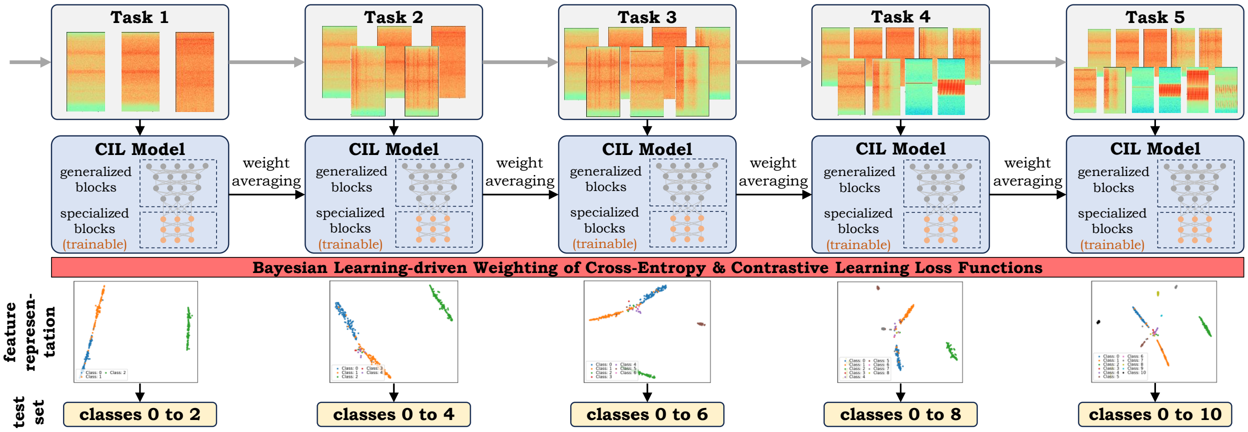


Figure 1: An overview of our continual learning approach is presented, which employs Bayesian weighting to balance cross-entropy and contrastive learning loss functions, facilitating efficient representation learning during the updating of specialized blocks.

the importance of weighting both the classification and contrastive loss functions to achieve a balance between previous and new tasks, our approach utilizes a Bayesian learning-driven strategy (Kendall & Gal, 2017; Kendall et al., 2018). A notable challenge arises from the inconsistency of negative samples between classification and representation learning tasks (Liu et al., 2024). A Bayesian framework facilitates the inference of probabilities for contrastive model samples, eliminating the necessity for computationally intensive hyperparameter tuning searches (Hasanzadeh et al., 2022). Figure 1 shows the main idea of our approach, exemplary for GNSS interference classification. For each task, two additional classes are trained and the model weights are averaged with the weights from the previous task to circumvent catastrophic forgetting by adapting only the specialized blocks. We combine the cross-entropy (CE) with a contrastive learning (CL) loss and dynamically weigh them with Bayesian learning. This leads to an optimal feature representation with improved clusters.

The main contribution of this work is to propose a continual learning approach capable of adapting to novel classes. Our contributions are as follows: (1) We introduce BLCL, a continual learning method similar to MEMO (Zhou et al., 2023b), which is founded on generalized and specialized blocks. In addition to this setting, we introduce dynamic specialized blocks by adding additional blocks as shown in Figure 2. (2) We combine the CE with a CL loss, incorporating class embedding distances in a prototypical manner (Snell et al., 2017). (3) We utilize a Bayesian learning-driven (Kendall et al., 2018) dynamic weighting mechanism for both loss functions, and (4) we conduct extensive evaluations on CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009), and a GNSS (Ott et al., 2024) dataset to demonstrate that BLCL yields an enhanced feature representation, resulting in improved classification accuracy for image-related tasks.

2 Related Work

2.1 Continual/Class-Incremental Learning

Class-incremental learning (CIL) aims to continuously develop a combined classifier for all encountered classes. The primary challenge in CIL lies in catastrophic forgetting, where direct optimization of the network with new classes leads to the erasure of knowledge pertaining to former tasks, thereby resulting in irreversible performance degradation. The objective is to effectively mitigate catastrophic forgetting (Zhou et al., 2023a). Conventional ML models address CIL through dual tasks by updating only the model with a single new stage (Lewandowsky & Li, 1995; Kuzborskij et al., 2013; Da et al., 2014). Feature extraction methods (Donahua et al., 2014) maintain the set of shared network parameters (Θ_s) and task-specific parameters for previously learned tasks (Θ_o) unchanged. Here, only the outputs of one or more layers serve as features for the new task, employing randomly initialized task-specific parameters (Θ_n). In finetune (Girshick et al., 2014) methods,

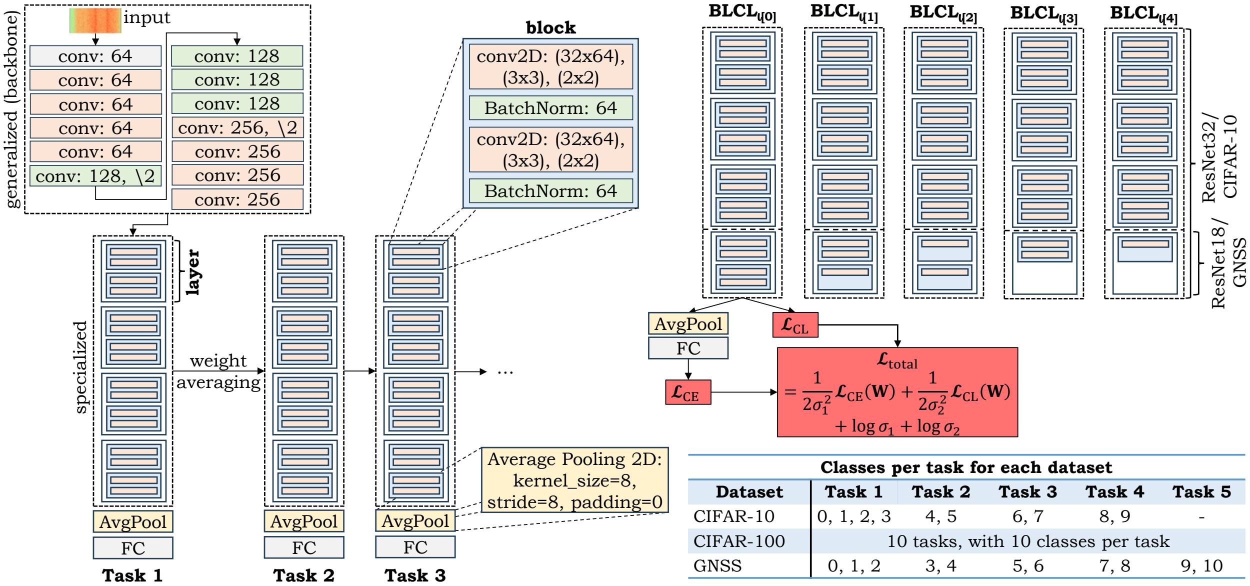
Θ_o remains fixed, while Θ_s and Θ_n are optimized for the new task. Kirkpatrick et al. (2017) proposed elastic weight consolidation (EWC), which selectively decelerates learning on weights crucial for previous tasks. The method learning without forgetting (LwF), introduced by Li & Hoiem (2018), employs solely new task data to train the network while retaining the original capabilities. Rolnick et al. (2019) proposed CLEAR (continual learning with experience and replay) for multi-task reinforcement learning, leveraging off-policy learning and behavioral cloning from replay to enhance stability, as well as on-policy learning to preserve plasticity. Rebuffi et al. (2017) introduced iCaRL, which learns such that only the training data for a small number of classes needs to be present concurrently, allowing for progressive addition of new classes. iCaRL combines a classification loss with a distillation loss to replicate the scores stored in the previous step. Replay, introduced by Ratcliff (1997), investigates manipulations of the network within a multilayer model and various variants thereof. Dynamically expandable representation (DER) (Yan et al., 2021) considers limited memory and aim to achieve better stability-plasticity trade-off utilizing a dynamically expandable representation. Inspired by gradient boosting, FOSTER (Wang et al., 2022) dynamically expands new modules to fit the residual between the target and output of the original model. Yu et al. (2020) compensated the semantic drift of features for embedding networks without exemplars. PASS (Zhu et al., 2021) and Zhu et al. (2022) mainly focus on previous class prototypes and updating the backbone model.

Conventional CIL methods typically prioritize representative exemplars from previous classes to mitigate forgetting (Luo et al., 2023). However, recent investigations indicate that preserving models from history can significantly enhance performance. Certain applications necessitate memory-efficient architectures. However, state-of-the-art CIL methods are not compared with regard to their memory budget. The memory-efficient expandable model (MEMO) (Zhou et al., 2023b) addresses this gap by considering accuracy and memory size, analyzing the impact of different network layers. Zhou et al. (2023b) discovered that shallow and deep layers exhibit distinct characteristics in the context of CIL. MEMO extends specialized layers based on shared generalized representations, efficiently extracting diverse representations at modest cost while maintaining representative exemplars. We adopt MEMO as our methodological baseline. However, we observe that results can be further enhanced by incorporating dynamic specialized blocks (the last layers in Figure 1 for which the weights are averaged with the previous task). In our experiments, we compare the performance of Finetune, EWC, LwF, iCarl, Replay, MEMO, DER, and FOSTER. For a comprehensive overview of methodologies, refer to Zhou et al. (2023a); Lange et al. (2022); Masana et al. (2023); Mai et al. (2022).

2.2 Bayesian Contrastive Learning

Contrastive learning establishes pairs of positive and negative instances along corresponding anchors to enhance training (Chen et al., 2020). This approach has found extensive application across various domains including multi-modal learning (Wei et al., 2016; Rasiwasia et al., 2010; Huang et al., 2020), domain adaptation (Thota & Leontidis, 2021), and semantic learning tasks (Ke et al., 2021; Sarafianos et al., 2019). The contrastive learning paradigm has been extended to triplet learning (Ott et al., 2023; Schroff et al., 2015), which utilizes a positive sample, and quadruplet learning (Ott et al., 2024; Chen et al., 2017), which employs a similar sample. The main focus within this field resolves around the optimal selection of pairs to augment training. However, the primary focus of our study resides in determining the optimal weighting between classification and contrastive loss functions.

Cha et al. (2021) demonstrated that representations learned contrastively exhibit greater robustness against catastrophic forgetting compared to those trained using the CE loss. Their approach integrates an asymmetric variant of supervised contrastive loss, mitigating model overfitting to a limited number of previous task samples by employing a dynamic architecture. However, contrastive learning requires a large number of diverse negative samples to be effective. On smaller datasets (as the CIFAR-10 and GNSS datasets), this lack of diversity can degrade performance (Cha et al., 2021). Similarly, Sy-CON (Cha & Moon, 2023) also integrates symmetric contrastive loss. Hasanzadeh et al. (2022) utilized distributional representations to provide uncertainty estimates in downstream graph analytic tasks. Lin et al. (2023) employ Monte Carlo dropout on skeleton data for action recognition, generating pairwise samples for model robustness. Liu et al. (2024) introduced a Bayesian contrastive loss to mitigate false negatives and mine hard negative samples, aligning the semantics of negative samples across tasks. Instead, our approach leverages Bayesian learning as proposed by Kendall et al. (2018) for weighting the CE and CL functions.


input
generalized (backbone)
conv: 64
conv: 64
conv: 64
conv: 64
conv: 64
conv: 128, \2
conv: 128
conv: 128
conv: 128
conv: 256, \2
conv: 256
conv: 256
conv: 256
block
conv2D: (32x64), (3x3), (2x2)
BatchNorm: 64
conv2D: (32x64), (3x3), (2x2)
BatchNorm: 64
layer
specialized
weight averaging
...
Average Pooling 2D: kernel_size=8, stride=8, padding=0
AvgPool
FC
Task 1
Task 2
Task 3
BLCL$_{l[0]}$
BLCL$_{l[1]}$
BLCL$_{l[2]}$
BLCL$_{l[3]}$
BLCL$_{l[4]}$
ResNet32/ CIFAR-10
ResNet18/ GNSS
$\mathcal{L}_{\text{CL}}$
$\mathcal{L}_{\text{CE}}$
$\mathcal{L}_{\text{total}} = \frac{1}{2\sigma_1^2}\mathcal{L}_{\text{CE}}(\mathbf{W}) + \frac{1}{2\sigma_2^2}\mathcal{L}_{\text{CL}}(\mathbf{W}) + \log\sigma_1 + \log\sigma_2$

Classes per task for each dataset

Dataset	Task 1	Task 2	Task 3	Task 4	Task 5
CIFAR-10	0, 1, 2, 3	4, 5	6, 7	8, 9	-
CIFAR-100	10 tasks, with 10 classes per task				
GNSS	0, 1, 2	3, 4	5, 6	7, 8	9, 10

Figure 2: Overview of our BLCL method. After the generalized component of ResNet18 or ResNet32 (He et al., 2016), follow up to eight blocks ($l = 8$) for the CIFAR (Krizhevsky & Hinton, 2009) datasets and two blocks ($l = 2$) for the GNSS (Ott et al., 2024) dataset in their respective specialized blocks. After each task the weights are averaged with the weights of the previous tasks. We propose a dynamic architecture with a different number of blocks and convolutional layers, indicated by BLCL$_{l[s]}$. After the FC layer, follows a CE loss in combination with a Bayesian learning-driven weighting of the contrastive loss (CL).

3 Methodology

This section provides a detailed description of our methodology. In Section 3.1, we mathematically formulate the problem of CIL. Subsequently, in Section 3.2, we provide an overview of our proposed method. We introduce a contrastive loss formulation and Bayesian learning-driven weighting strategy in Section 3.3. We describe data augmentation techniques in the Appendix A.1.

3.1 Problem Formulation

We consider T sequentially presented training tasks denoted as $\{\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^T\}$, where each $\mathcal{D}^t = \{(\mathbf{X}_i^t, y_i^t)\}_{i=1}^{n_t}$ represents the t-th incremental step containing n_t training instances of images. $\mathbf{X}_i^t$ signifies a sample belonging to class $y_i \in Y_t$, with Y_t representing the label space of task t. During the training of task t, only data from $\mathcal{D}^t$ is available. The objective is to continuously develop a classification model encompassing all classes encountered thus far. Subsequent to each task, the model is evaluated for all observed classes $\mathcal{Y}_t = Y_1 \cup \cdots \cup Y_t$ (Zhou et al., 2023a; Rebuffi et al., 2017). Typically, classes do not overlap between different tasks. However, in certain scenarios, previously encountered classes may reappear in subsequent tasks, constituting what is termed as blurry CIL (Xie et al., 2022; Koh et al., 2022; Bang et al., 2021; 2022). For the purpose of this paper, it is noted that the classes are overlapping as we have a fixed number of representative instances from the previous task classes, so-called *exemplars*.

3.2 Method Overview

Our methodology is outlined in Figure 2, designated as BLCL. It leverages the MEMO (Zhou et al., 2023b) architecture. The ResNet model, proposed by He et al. (2016), specifically ResNet18 or ResNet32, comprises both generalized and specialized components. The generalized component extracts features from the image dataset, featuring 13 convolutional layers, and is succeeded by one or more (up to four) additional layers. The specialized component consists of blocks whose weights are averaged with the weights from the previous task. These specialized layers contain different numbers of blocks l, adapted to the dataset at hand, are denoted as BLCL_l, where each block comprises of two convolutional layers and two batch normalizations. The output dimension of the convolutional layers is set to 512. We employ average pooling and a fully connected (FC) layer, and we train utilizing the CE loss $\mathcal{L}_{\text{CE}}$. The training process involves freezing the specialized component from previous tasks while updating the generalized and specialized component with classes pertinent to the specific tasks. Moreover, we average the weights of the specialized component from previous tasks. To maintain adaptability to varying complexities of new classes across tasks, we dynamically adjust each task's layer structure, denoted as $\text{BLCL}_{l[s]}$ (as depicted in the top right of Figure 2). This adaptation involves the removal of either one convolutional layer or an entire block from the specialized model's layer configuration, catering to the evolving requirements of each task. Furthermore, to facilitate a fair comparison with other state-of-the-art methods and to compensate for the additional blocks, we maintain a constant number of exemplars, fixed at 2,000 samples, throughout the training process for all datasets.

3.3 Loss Functions

The lower-level representation of the CIL model is continuously adapted with the addition of each new class, thereby potentially undergoing significant changes. The primary objective is to incrementally enhance class prototypes, thereby reducing intra-class distances while simultaneously increasing inter-class distances. To facilitate this objective, we introduce a contrastive loss term, denoted as $\mathcal{L}_{\text{CL}}$. Achieving an optimal balance between the CE and CL loss functions is crucial. Hence, we propose an automated weighting mechanism utilizing Bayesian learning techniques.

Contrastive Loss. Pairwise learning is characterized by the utilization of pairs featuring distinct labels, serving to enrich the training process by enforcing a margin between pairs of images belonging to different identities. Consequently, the feature embedding for a specific label lives on a manifold while ensuring sufficient discriminability, i.e., distance, from other identities (Ott et al., 2023; Do et al., 2019). CL, on the other hand, is a methodology aimed at training models to learn representations by minimizing the distance between positive pairs and maximizing the distance between negative pairs within a latent space. We define feature embeddings $f(\mathbf{X}) \in \mathbb{R}^{q\times h}$ to map the image input into a feature space $\mathbb{R}^{q\times h}$, where f represents the output of the last convolutional layer with a size of 512 from the specialized model, and $q \times h$ denotes the dimensionality of the layer output. Consequently, we define an *anchor* sample $\mathbf{X}_i^a$ corresponding to a specific label, a *positive* sample $\mathbf{X}_i^p$ from the same label, and a *negative* sample $\mathbf{X}_i^n$ drawn from a different label. For all training samples $\big(f(\mathbf{X}_i^a), f(\mathbf{X}_i^p), f(\mathbf{X}_i^n)\big) \in \Phi$, our objective is to fulfill the following inequality:

$$d\big(f(\mathbf{X}_i^a), f(\mathbf{X}_i^p)\big) + \alpha < d\big(f(\mathbf{X}_i^a), f(\mathbf{X}_i^n)\big), \tag{1}$$

where d is a distance function, specifically the cosine distance $\text{CD}(\mathbf{x}_1, \mathbf{x}_2) = 1 - \frac{\mathbf{x}_1\cdot\mathbf{x}_2}{||\mathbf{x}_1||\cdot||\mathbf{x}_2||}$ computed between two non-zero vectors $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ of size n, $\alpha = 1$ denotes a margin differentiating positive and negative pairs, and Φ is the set of all possible pairs within the training dataset, where N is the number of pairs. For each batch of size bs, we select all conceivable pairs and increase the batch size to $\frac{bs(bs+1)}{2}$.

Bayesian Learning. Typically, naïve methods employ a weighted combination of both loss functions to compute the total loss, denoted as $\mathcal{L}_{\text{total}} = w_1\mathcal{L}_{\text{CE}} + w_2\mathcal{L}_{\text{CL}}$. However, the model performance is extremely sensitive to the selection of weights w_i (Groenendijk et al., 2021). In our approach, we adopt a strategy wherein we concurrently optimize both objectives using homoscedastic task uncertainty, as defined by Kendall & Gal (2017); Kendall et al. (2018). This homoscedastic uncertainty pertains to the task-dependent aleatoric uncertainty, which remains invariant across input data but varies across distinct tasks (Klaß et al., 2022). Let $\text{f}^{\mathbf{W}}(\mathbf{X})$ denote the output of a neural network with weights $\mathbf{W}$ on input $\mathbf{X}$. In scenarios involving multiple model outputs, we factorize over the outputs

$$p\big(\mathbf{y}_1, \ldots, \mathbf{y}_K|\text{f}^{\mathbf{W}}(\mathbf{X})\big) = p\big(\mathbf{y}_1|\text{f}^{\mathbf{W}}(\mathbf{X})\big)\cdots p\big(\mathbf{y}_K|\text{f}^{\mathbf{W}}(\mathbf{X})\big), \tag{2}$$

with K model outputs $\mathbf{y}_1, \ldots, \mathbf{y}_K$ (Kendall et al., 2018). In a classification task, we sample from a probability vector from a scaled softmax function's output $p\big(\mathbf{y}|\text{f}^{\mathbf{W}}(\mathbf{X}), \sigma\big) = \text{softmax}\big(\frac{1}{\sigma^2}\text{f}^{\mathbf{W}}(\mathbf{X})\big)$. The log likelihood is then defined as

$$\log p\big(\mathbf{y} = c|\mathrm{f}^{\mathbf{W}}(\mathbf{X})\big) = \frac{1}{\sigma^2}\mathrm{f}_c^{\mathbf{W}}(\mathbf{X}) - \log \sum_{c'} \exp\big(\frac{1}{\sigma^2}\mathrm{f}_c^{\mathbf{W}}(\mathbf{X})\big), \tag{3}$$

where $\mathrm{f}_c^{\mathbf{W}}(\mathbf{X})$ is the c-th element of $\mathrm{f}^{\mathbf{W}}(\mathbf{X})$. In a regression task, we maximize the log likelihood of the model as

$$\log p\big(\mathbf{y}|\mathrm{f}^{\mathbf{W}}(\mathbf{X})\big) \propto -\frac{1}{2\sigma^2}||\mathbf{y} - \mathrm{f}^{\mathbf{W}}(\mathbf{X})||^2 - \log\sigma, \tag{4}$$

for a Gaussian likelihood $p\big(\mathbf{y}|\mathrm{f}^{\mathbf{W}}(\mathbf{X})\big) = \mathcal{N}\big(\mathrm{f}^{\mathbf{W}}(\mathbf{X}), \sigma^2\big)$, where σ is the model's observation noise parameter (Kendall et al., 2018). In our case, our model output is composed of two vectors, a discrete output $\mathbf{y}_1$ for the CE loss and continuous output $\mathbf{y}_2$ for the CL loss, which leads to the total minimization objective:

$$\begin{aligned}\mathcal{L}(\mathbf{W}, \mathbf{X}, \sigma_1, \sigma_2) &= -\log p\big(\mathbf{y}_1 = c, \mathbf{y}_2|\mathrm{f}^{\mathbf{W}}(\mathbf{X})\big) = -\mathrm{softmax}\big(\mathbf{y}_1 = c; \mathrm{f}^{\mathbf{W}}(\mathbf{X}), \sigma_1\big) \cdot \log\mathcal{N}\big(\mathbf{y}_2; \mathrm{f}^{\mathbf{W}}(\mathbf{X}), \sigma_2^2\big) \\ &= -\log p\big(\mathbf{y}_1 = c|\mathrm{f}^{\mathbf{W}}(\mathbf{X}), \sigma_1\big) + \frac{1}{2\sigma_2^2}||\mathbf{y}_2 - \mathrm{f}^{\mathbf{W}}(\mathbf{X})||^2 + \log\sigma_2 \\ &\approx \frac{1}{2\sigma_1^2}\mathcal{L}_{\mathrm{CE}}(\mathbf{W}, \mathbf{X}) + \frac{1}{2\sigma_2^2}\mathcal{L}_{\mathrm{CL}}(\mathbf{W}, \mathbf{X}) + \log\sigma_1 + \log\sigma_2.\end{aligned} \tag{5}$$

The final objective is to minimize Equation 5 with respect to σ_1 and σ_2, thereby learning the relative weights of the losses $\mathcal{L}_1(\mathbf{W})$ and $\mathcal{L}_2(\mathbf{W})$ in an adaptive manner (Kendall & Gal, 2017; Kendall et al., 2018). Figure 7 in the Appendix A.2 provides an illustrative demonstration of the weighting mechanism applied to both loss functions for the first task on the CIFAR-10 dataset. As σ_1, respectively σ_2, the noise parameters associated with variable $\mathbf{y}_1$, respectively $\mathbf{y}_2$, increase, the weight assigned to $\mathcal{L}_{\mathrm{CE}}$, respectively $\mathcal{L}_{\mathrm{CL}}$, diminishes. Such an adjustment serves as a regularizer for the noise term. This trend is observable at the early period of each task (see Figure 7 for the first task), as new classes introduce more noise, and hence, the weighting of both loss functions decreases. In general, both loss functions have converged after 200 epochs.

4 Experiments

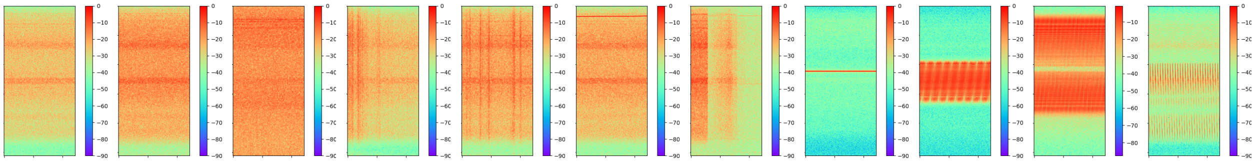
For fair comparability, we conduct experiments on two distinct image datasets: the CIFAR-10 and CIFAR-100 dataset and a GNSS-based dataset. Consequently, we encounter a diverse array of applications pertinent to continual learning, specifically within the realm of image classification tasks.

4.1 Image Classification: CIFAR-10 & CIFAR-100

The CIFAR-10 (Krizhevsky & Hinton, 2009) dataset consists of 60,000 colour images of size 32 × 32. The goal is to classify 10 distinct classes (i.e., airplane, automobile, bird, cat, deer, dog, etc.), with 6,000 images per class. The dataset is split into 50,000 training and 10,000 test images. We train four tasks: task 1 consists of the classes 0, 1, 2, and 3, task 2 consists of the classes 4 and 5, task 3 consists of the classes 6 and 7, and task 4 consists of the classes 8 and 9. The CIFAR-100 (Krizhevsky & Hinton, 2009) dataset has 100 classes containing 600 images each, and hence, we train 10 classes per task with 10 tasks in total.

4.2 Interference Classification: GNSS Snapshots

The GNSS-based (Ott et al., 2024) dataset contains short, wideband snapshots in both E1 and E6 GNSS bands. The dataset was captured at a bridge over a highway. The setup records 20 ms raw IQ (in-phase and quadrature-phase) snapshots triggered from the energy with a sample rate of 62.5 MHz, an analog bandwidth of 50 MHz and an 8 bit bit-width. Further technical details can be found in (Brieger et al., 2022). Figure 3 shows exemplary snapshots of the spectrogram. Manual labeling of these snapshots has resulted in 11 classes: Classes 0 to 2 represent samples with no interferences, distinguished by variations in background intensity, while classes 3 to 10 contain different interferences such as pulsed, noise, tone, and multitone. For instance, Figure 3k showcases a snapshot containing a potential chirp jammer type. The dataset's imbalance of 197,574 samples for non-interference classes and between 9 to 79 samples per interference class, emphasizing the under-representation of positive class labels. The goal is to adapt to new interference types



(a) 0. (b) 1. (c) 2. (d) 3. (e) 4. (f) 5. (g) 6. (h) 7. (i) 8. (j) 9. (k) 10.

Figure 3: Exemplary spectrogram samples without interference (classes 0 to 2) and with interference (classes 3 to 10) between intensity [0, -90] with logarithmic scale.

with continual learning. The challenge lies in adapting to positive class labels with only a limited number of samples available. We partition the dataset into a 64% training set, 16% validation set, and a 20% test set split (balanced over the classes). We train five tasks: task 1 consists of the classes 0, 1, and 2, task 2 consists of the classes 3 and 4, task 3 consists of the classes 5 and 6, task 4 consists of the classes 7 and 8, and task 5 consists of the classes 9 and 10.

5 Evaluation

Initially, we present the results obtained through our BLCL approach, in comparison with state-of-the-art methods, on both the CIFAR and the GNSS datasets. Subsequently, we conduct an analysis involving confusion matrices, model embeddings, cluster analysis, and parameter counts. Throughout, we provide the accuracy achieved after each task, the average accuracy over all tasks ($\overline{\text{Acc.}}$), as well as the F1-score and F2-score. For our hardware and training setup, see the Appendix A.3.

Evaluation Results. The evaluation results for the CIFAR datasets are summarized in Table 1 and Table 2, while those for the GNSS dataset are presented in Table 3. On the CIFAR-10 dataset, MEMO and DER achieve the highest classification results for each task, with the highest average accuracy of 84.78%, respectively 86.15%. iCarl also demonstrates strong performance with an average accuracy of 84.47%. However, MEMO with weight averaging slightly decreases results to 83.53%, leading us to train BLCL without weight averaging. Figure 4 shows results for different BLCL$_{l[s]}$ architectures. Increasing the number of layers in the specialized component from four (84.78%) to six (84.96%) and eight (85.02%) results in improved accuracy. Consequently, follow-up experiments are conducted with the BLCL$_{8[s]}$ model. Evaluation of dynamic specialized components reveals that smaller blocks are preferable for the first task, as BLCL$_{8[3,\ldots]}$ and BLCL$_{8[4,\ldots]}$ outperform BLCL$_{8[0,\ldots]}$. Larger layers for tasks 3 and 4 further improve results; for instance, BLCL$_{8[1,2,3,4]}$ achieves 85.11%. However, simply adding the contrastive loss with a naïve weighting of 0.9 decreases results, motivating the adoption of automatic Bayesian weighting. For all the architectures tested, including BLCL$_{8[1,1,1,2]}$, BLCL$_{8[1,1,2,2]}$, BLCL$_{8[0,0,3,3]}$, and BLCL$_{8[1,2,3,4]}$, our Bayesian loss outperforms the baseline model. Consequently, our method achieves 86.57% with high robustness across all tasks, significantly outperforming MEMO with 84.78% and DER with 86.15%. On the CIFAR-100 dataset, BLCL$_{8[s]}$ with $s = 0$ for all the 10 tasks, achieves 71.10% outperforming all state-of-the-art methods. On the GNSS dataset, Replay and DER achieve the highest accuracy of 95.45%, respectively 95.77%, although the F2-score of 0.661 is low due to dataset underrepresentation. We outperform MEMO (95.34% average accuracy) with weight averaging, achieving an accuracy of 95.67%. Combining this with the contrastive loss further increases accuracy to 96.16%. The F2-score can be improved to 0.678 by reducing the false negative rate.

Table 1: Evaluation results for CIFAR-100.

	Final				
Method	**$\overline{\text{Acc.}}$**	**F1**	**F2**	**DB**	**CH**
Finetune	26.41	0.01	0.03	4.34	68
EWC (Kirkpatrick et al., 2017)	35.81	0.20	0.20	3.11	77
LwF (Li & Hoiem, 2018)	43.62	0.15	0.17	2.99	72
iCarl (Rebuffi et al., 2017)	54.83	0.35	0.33	2.67	63
Replay (Ratcliff, 1997)	52.90	0.34	0.31	2.77	68
MEMO (Zhou et al., 2023b), w/o	68.86	0.57	0.57	3.55	55
DER (Yan et al., 2021)	67.17	0.58	**0.58**	3.37	61
FOSTER (Wang et al., 2022)	63.51	0.50	0.50	3.52	74
BLCL$_{8[s]}$, w/	**71.10**	**0.59**	**0.58**	**2.56**	**103**

Confusion Matrices. Figure 5 illustrates the confusion matrices subsequent to the final task to all test classes within the CIFAR-10 and GNSS datasets, employing MEMO (Zhou et al., 2023b) and our proposed

Table 2: Evaluation results for baseline methods and our Bayesian-driven (B) contrastive loss functions for the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset. We indicate the weighting with (w/) and without (w/o) averaging. Underlined is the best baseline method and **bold** is the best method in total. We present the final Davies-Bouldin (DB) (↓) and Caliński-Harbasz (CH) (↑) scores.

	Loss/	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Final**				
Method	**Weighting**	**Acc.**	**Acc.**	**Acc.**	**Acc.**	$\overline{\textbf{Acc.}}$	**F1**	**F2**	**DB**	**CH**
Finetune	CE	95.20	32.08	24.80	19.80	43.22	0.067	0.110	7.29	1,716
EWC (Kirkpatrick et al., 2017)		95.20	54.30	47.39	36.10	58.24	0.303	0.324	2.12	1,365
LwF (Li & Hoiem, 2018)		95.48	63.55	62.80	56.89	69.68	0.556	0.557	2.14	848
iCarl (Rebuffi et al., 2017)		95.35	81.85	80.12	79.57	84.47	0.792	0.791	1.28	1,907
Replay (Ratcliff, 1997)		95.35	78.95	76.69	76.72	81.93	0.766	0.764	1.18	2,327
MEMO (Zhou et al., 2023b), w/o	CE	**96.00**	80.68	80.79	81.68	84.78	0.810	0.812	1.78	2,327
DER (Yan et al., 2021)		95.32	84.23	**83.70**	83.48	86.15	0.833	**0.834**	1.38	1,822
FOSTER (Wang et al., 2022)		95.95	69.58	66.90	66.86	74.82	0.637	0.645	2.20	1,225
$\text{BLCL}_{8[1,1,1,2]}$, w/o	CE, B, CL	95.58	**84.50**	82.55	**83.62**	86.56	**0.834**	**0.834**	**0.79**	**5,085**
$\text{BLCL}_{8[1,1,2,2]}$, w/o	CE, B, CL	95.58	**84.50**	83.45	82.76	**86.57**	0.824	0.825	0.81	4,688

Table 3: Evaluation results for the GNSS (Ott et al., 2024) dataset.

	Loss/	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Final**				
Method	**Weighting**	**Acc.**	**Acc.**	**Acc.**	**Acc.**	**Acc.**	$\overline{\textbf{Acc.}}$	**F1**	**F2**	**DB**	**CH**
Finetune	CE	96.06	0.04	0.04	0.06	0.02	19.24	0.033	0.048	1.48	7,958
EWC (Kirkpatrick et al., 2017)		95.99	0.49	0.04	0.05	0.04	19.32	0.071	0.072	1.34	4,799
LwF (Li & Hoiem, 2018)		96.40	52.39	17.49	3.50	2.52	34.46	0.026	0.037	1.97	4,565
iCarl (Rebuffi et al., 2017)		94.75	95.43	95.23	95.14	95.14	95.14	0.601	0.651	0.81	29,076
Replay (Ratcliff, 1997)		94.55	95.97	95.49	95.50	95.75	95.45	0.594	0.661	0.96	27,622
MEMO (Zhou et al., 2023b), w/o	CE	95.95	95.41	94.99	95.14	95.19	95.34	0.579	0.651	1.15	14,066
DER (Yan et al., 2021)		96.08	95.73	95.61	95.67	95.78	95.77	0.562	0.642	1.15	8,828
FOSTER (Wang et al., 2022)		96.24	93.67	92.30	91.09	93.53	93.36	0.401	0.479	1.53	10,402
$\text{BLCL}_{2[1,1,1,1,1]}$, w/	CE, 0.9, CL	96.03	95.99	95.54	95.61	95.17	95.67	**0.619**	**0.678**	1.04	15,868
$\text{BLCL}_{2[1,1,1,1,1]}$, w/	CE, B, CL	95.12	95.35	94.58	94.39	93.74	94.63	0.547	0.596	0.79	98,877
$\text{BLCL}_{2[2,2,2,2,2]}$, w/	CE, 0.9, CL	96.21	95.37	95.13	95.44	**96.25**	95.68	0.612	0.648	0.94	28,598
$\text{BLCL}_{2[2,2,2,2,2]}$, w/	CE, B, CL	**96.74**	**96.16**	**95.75**	**95.96**	96.18	**96.16**	0.601	0.662	**0.78**	**163,420**

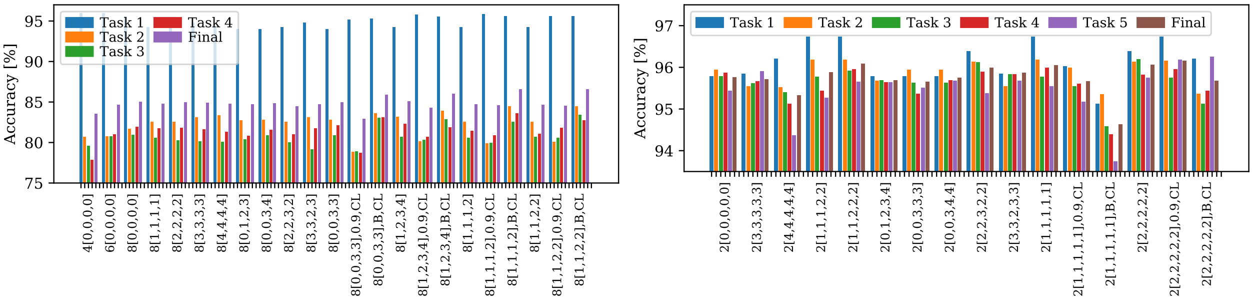
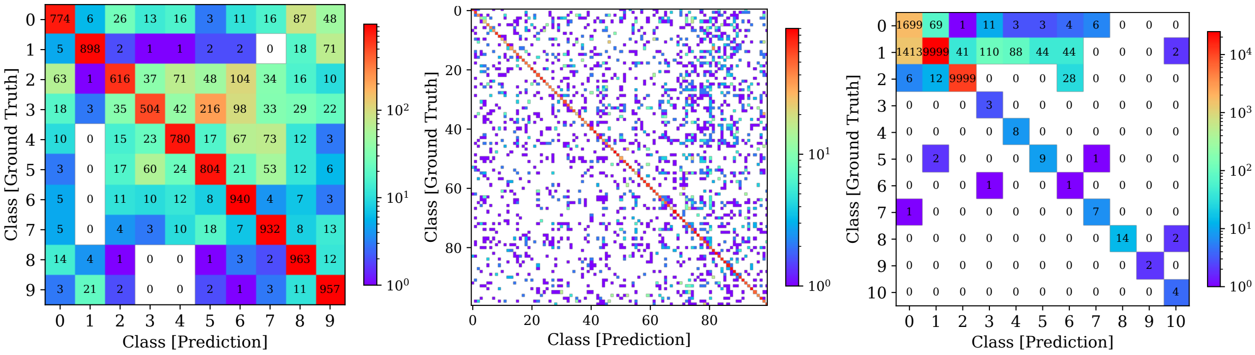
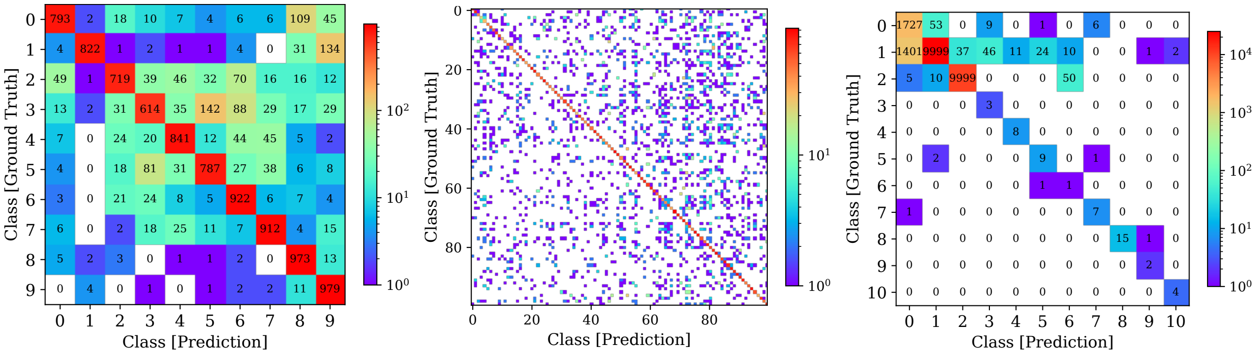


Figure 4: Evaluation of $\text{BLCL}_{l[s]}$ on the CIFAR-10 (a) and GNSS (b) datasets.

technique denoted as $\text{BLCL}_{8[1,1,2,2]}$ and $\text{BLCL}_{2[2,2,2,2,2]}$ with Bayesian weighting, respectively. See the appendix, for confusion matrices on all methods. On the CIFAR-10 dataset, MEMO demonstrates significant performance superiority over Replay for specific classes 0, 1, 4, 5, 6, and 7 (refer to Figure 9e compared to Figure 5a). Notably, Replay exhibits pronounced confusion between classes 8 and 9 as well as the remaining classes, a misclassification mitigated by MEMO. Additionally, BLCL achieves a higher true positive rate than MEMO for classes 0, 2, 3, 4, and 9, thereby attaining an average accuracy of 86.57%, surpassing MEMO's 84.78% (see Figure 5d). Our approach effectively mitigates misclassifications, particularly between classes 3 and 6. On the GNSS dataset, a distinct resemblance exists between classes 1 and other classes (refer to Figure 3), resulting in misclassifications. While MEMO exhibits a higher false positive rate compared to Replay, their false negative rates remain equal, leading to Replay's superior average accuracy of 95.45% versus MEMO's 95.34% (compare Figure 11e with Figure 5c). Leveraging enhanced representation learning



(a) MEMO (Zhou et al., 2023b). (b) MEMO (Zhou et al., 2023b). (c) MEMO (Zhou et al., 2023b).



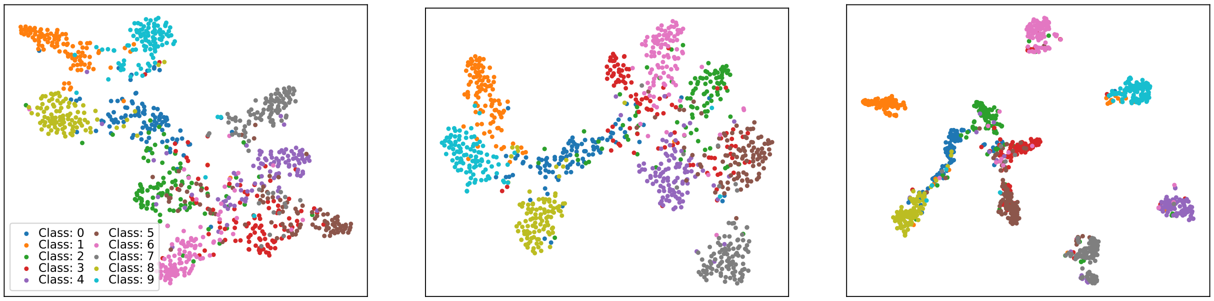
(d) $\text{BLCL}_{8[1,1,2,2]}$. (e) $\text{BLCL}_{8[s]}$. (f) $\text{BLCL}_{2[2,2,2,2,2]}$.

Figure 5: Confusion matrices for the CIFAR-10 (left), CIFAR-100 (middle), and GNSS (right) datasets. For confusion matrices of all methods, refer to the appendix.

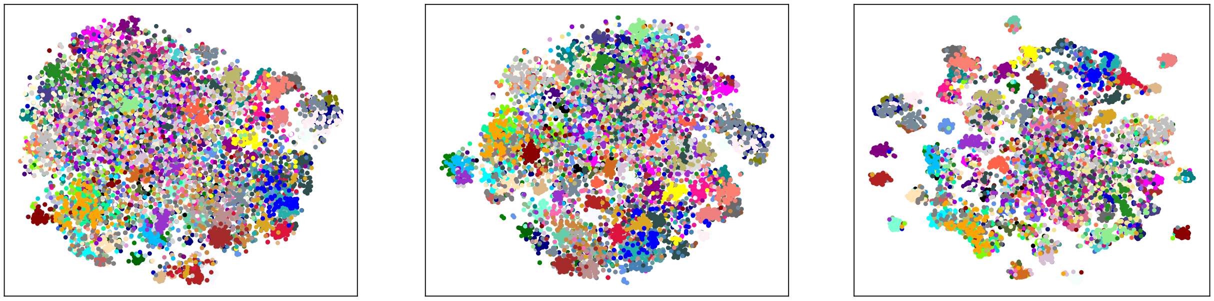
via contrastive loss, our BLCL method effectively separates positive classes (3 to 10) into distinct clusters, thus reducing the false positive rate and yielding a higher F2-score of 0.678 while simultaneously decreasing the false negative rate (refer to Figure 5f).

Cluster Analysis. Subsequently, we compare the Davies-Bouldin (DB) (Davies & Bouldin, 1979) and Caliński-Harbasz (CH) (Caliński & Harabasz, 1972) scores. DB is defined as the mean similarity measure between each cluster and its most similar cluster. This measure of similarity is the ratio of within-cluster distances to between-cluster distances. Consequently, clusters exhibiting greater separation and lesser dispersion will yield a superior score (Halkidi et al., 2001; Ros et al., 2023; Thrun, 2011). CH is a variance-ratio criterion, where a higher CH score relates to a model with better defined clusters, and is defined as the ratio of the sum of inter-clusters dispersion and of intra-cluster dispersion for all clusters. Table 1 to Table 3 summarizes the scores attributed to Finetune, EWC, LwF, iCarl, Replay, MEMO, DER, FOSTER, and BLCL. Across all three datasets, BLCL demonstrates a lower DB score and higher CH score in comparison to the baseline methods. This observation supports the assertion that BLCL results in a representation characterized by lower inter-class distances and enhanced clustering efficacy.

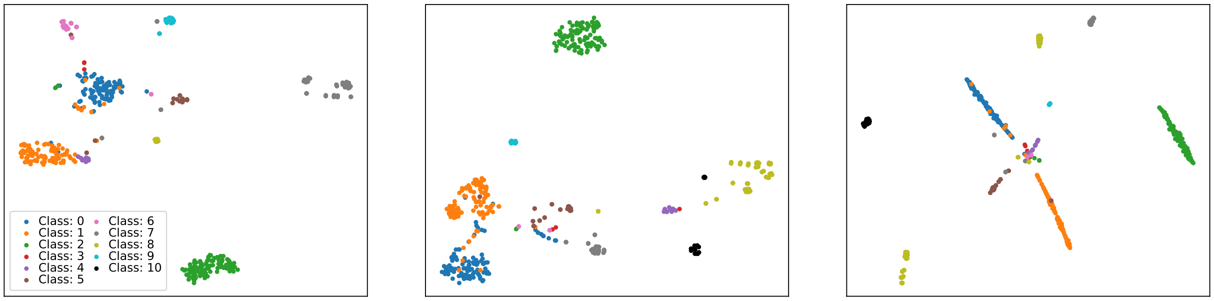
Embedding Analysis. The feature embedding of the last convolutional layer, with an output size of 512, is depicted in Figure 6. This embedding is generated using t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten & Hinton, 2008) with a perplexity of 30, an initial momentum of 0.5, and a final momentum of 0.8. On the CIFAR-10 dataset, the Replay method (Figure 6a) and MEMO method (Figure 6b) demonstrate poor separation of clusters, resulting in the overlap of classes 2, 3, and 4, thereby leading to misclassification. This observation aligns with the findings presented earlier using confusion matrices (refer to Figure 5a and 9e). Our proposed BLCL method (Figure 6c), which incorporates contrastive loss, significantly reduces the intra-class distance and increases inter-class distance. This is also evident for the CIFAR-100 dataset (refer to Figure 6d to 6f). However, some overlap persists, particularly for classes 0, 1, and 3,



(a) Replay (Ratcliff, 1997). (b) MEMO (Zhou et al., 2023b). (c) BLCL$_{8[1,1,2,2]}$.



(d) Replay (Ratcliff, 1997). (e) MEMO (Zhou et al., 2023b). (f) BLCL$_{8[s]}$.



(g) Replay (Ratcliff, 1997). (h) MEMO (Zhou et al., 2023b). (i) BLCL$_{2[2,2,2,2,2]}$.

Figure 6: t-SNE plots for the CIFAR-10 (a to c), CIFAR-100 (d to f), and GNSS (g to i) datasets.

resulting in a higher rate of misclassification. Similar trends are observed in the GNSS dataset, where both Replay and MEMO methods exhibit overlapping samples with class 0 (Figure 6g and 6h), while the BLCL method only forms a small cluster with specific samples in the center of the embedding (Figure 6i).

6 Conclusion

In the context of the continual learning task, we proposed a method characterized by a dynamic specialized component. This component combines the CE loss with a prototypical contrastive loss by utilizing Bayesian learning principles. By employing dynamic loss weighting, our method effectively achieves a feature representation with reduced intra-class disparity and higher inter-class distance. Evaluations conducted on the CIFAR-10, CIFAR-100, and GNSS datasets demonstrate that our proposed technique outperforms the state-of-the-art Replay, MEMO, and DER. Specifically, our method achieves accuracy rates of up to 86.57% on CIFAR-10, 71.10% on CIFAR-100, and 96.16% on the GNSS dataset with improved Davies-Bouldin and Caliński-Harabasz scores.

References

Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow Memory: Continual Learning with a Memory of Diverse Samples. In *CVPR*, Nashville, TN, June 2021. doi: 10.1109/CVPR46437.2021.00812.

Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. Online Continual Learning on a Contaminated Data Stream with Blurry Task Boundaries. In *CVPR*, New Orleans, LA, June 2022. doi: 10.1109/CVPR52688.2022.00906.

Tobias Brieger, Nisha L. Raichur, Dorsaf Jdidi, Felix Ott, Tobias Feigl, Johannes Rossouw van der Merwe, Alexander Rügamer, and Wolfgang Felber. Multimodal Learning for Reliable Interference Classification in GNSS Signals. In *ION GNSS+*, pp. 3210–3234, Denver, CO, September 2022. doi: 10.33012/2022.18586.

Tadeusz Caliński and J. Harabasz. A Dendrite Method for Cluster Analysis. In *Communications in Statistics*, volume 3(1), pp. 1–28, September 1972. doi: 10.1080/03610927408827101.

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co^{2}L: Contrastive Continual Learning. In *ICCV*, Montreal, QC, October 2021. doi: 10.1109/ICCV48922.2021.00938.

Sungmin Cha and Taesup Moon. Sy-CON: Symmetric Contrastive Loss for Continual Self-Supervised Representation Learning. In *arXiv:2306.05101*, June 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, volume 149, pp. 1597–1607, July 2020. doi: 10.5555/3524938.3525087.

Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-identification. In *CVPR*, Honolulu, HI, July 2017. doi: 10.1109/CVPR.2017.145.

Qing Da, Yang Yu, and Zhi-Hua Zhou. Learning with Augmented Class by Exploiting Unlabeled Data. In *AAAI*, volume 28, 2014.

David L. Davies and Donald W. Bouldin. A Cluster Separation Measure. In *TPAMI*, volume PAMI-1(2), pp. 224–227, April 1979. doi: 10.1109/TPAMI.1979.4766909.

Thanh-Toan Do, Toan Tran, Ian Reid, Vijay Kumar, Tuan Hoang, and Gustavo Carneiro. A Theoretically Sound Upper Bound on the Triplet Loss for Improving the Efficiency of Deep Distance Metric Learning. In *CVPR*, Long Beach, CA, June 2019. doi: 10.1109/CVPR.2019.01065.

Jeff Donahua, Yangqing Jia, Oriol Vinyals, Judy Hoffmann, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *JMLR*, volume 32(1), pp. 647–655, 2014.

Yaron Fairstein, oren Kalinsky, Zohar Karnin, Guy Kushilevitz, Alexander Libov, and Sofia Tolmach. Class Balancing for Efficient Active Learning in Imbalanced Datasets. In *Proc. of the Linguistic Annotation Workshop (LAW_XVIII)*, pp. 77–86, St. Julians, Malta, March 2024.

Nishant S. Gaikwad, Lucas Heublein, Nisha L. Raichur, Tobias Feigl, Christopher Mutschler, and Felix Ott. Federated Learning with MMD-based Early Stopping for Adaptive GNSS Interference Classification. In *arXiv preprint arXiv:2410.15681v2*, December 2024.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, pp. 580–587, 2014. doi: 10.1109/CVPR.2014.81.

Rick Groenendijk, Sezer Karaoglu, Theo Gevers, and Thomas Mensink. Multi-Loss Weighting with Coefficient of Variations. In *WACV*, pp. 1468–1477, 2021. doi: 0.1109/WACV48630.2021.00151.

Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On Clustering Validation Techniques. In *Journal of Intelligent Information Systems*, volume 17, pp. 107–145, December 2001. doi: 10.1023/A:1012801612483.

Arman Hasanzadeh, Mohammadreza Armandpour, Ehsan Hajiramezanali, Mingyuan Zhou, Nick Duffield, and Krishna Narayanan. Bayesian Graph Contrastive Learning. In *arXiv:2112.07823*, August 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, Las Vegas, NV, June 2016. doi: 10.1109/CVPR.2016.90.

Lucas Heublein, Nisha L. Raichur, Tobias Feigl, Tobias Brieger, Fin Heuer, Lennart Asbach, Alexander Rügamer, and Felix Ott. Evaluation of (Un-)Supervised Machine Learning Methods for GNSS Interference Classification with Real-World Data Discrepancies. In *ION GNSS+*, pp. 1260–1293, Baltimore, MD, September 2024. doi: 10.33012/2024.19887.

Xin Huang, Yuxin Peng, and Mingkuan Yuan. MHTN: Modal-Adversarial Hybrid Transfer Network for Cross-Modal Retrieval. In *Trans. on Cybernetics*, 2020. doi: 10.1109/TCYB.2018.2879846.

Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. CLASSIC: Continual and Contrastive Learning of Aspect Sentiment Classification Tasks. In *EMNLP*, November 2021. doi: 10.18653/v1/2021.emnlp-main.550.

Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *NIPS*, Long Beach, CA, 2017.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *CVPR*, 2018. doi: 0.1109/CVPR.2018.00781.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming Catastrophic Forgetting in Neural Networks. In *Applied Mathematics*, volume 114(13), pp. 3521–3526, March 2017. doi: 10.1073/pnas.1611835114.

Andreas Klaß, Sven M. Lorenz, Martin W. Lauer-Schmaltz, David Rügamer, Bernd Bischl, Christopher Mutschler, and Felix Ott. Uncertainty-aware Evaluation of Time-Series Classification for Online Handwriting Recognition with Domain Shift. In *IJCAI-ECAI STRL*, volume 3190, Vienna, Austria, July 2022.

Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online Continual Learning on Class Incremental Blurry Task Configuration with Anytime Inference. In *ICLR*, 2022.

Alex Krizhevsky and G. Hinton. Learning Multiple Layers of Features from Tiny Images. In *Technical Report, University of Toronto*, Toronto, Ontario, April 2009.

Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. From N to N+1: Multiclass Transfer Incremental Learning. In *CVPR*, Portland, OR, June 2013. doi: 10.1109/CVPR.2013.431.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A Continual Learning Survey: Defying Forgetting in Classification Tasks. In *TPAMI*, volume 44(7), July 2022. doi: 1109/TPAMI.2021.3057446.

S. Lewandowsky and S.-C. Li. Catastrophic Interference in Neural Networks: Causes, Solutions, and Data. In *F. N. Dempster & C. J. Brainerd (Eds.), Interference and Inhibition in Cognition*, pp. 329–361, 1995. doi: 10.1016/B978-012208930-5/50011-8.

Zhizhong Li and Derek Hoiem. Learning Without Forgetting. In *TPAMI*, volume 40(12), pp. 2935–2947, December 2018. doi: 10.1109/TPAMI.2017.2773081.

Lilang Lin, Jiahang Zhang, and Jiaying Liu. Bayesian Contrastive Learning with Manifold Regularization for Self-Supervised Skeleton Based Action Recognition. In *ISCAS*, Monterey, CA, May 2023. doi: 10.1109/ISCAS46773.2023.10181797.

Bin Liu, Bang Wang, and Tianrui Li. Bayesian Self-Supervised Contrastive Learning. In *arXiv:2301.11673*, January 2024.

Zilin Luo, Yaoyao Liu, Bernt Schiele, and Qianru Sun. Class-Incremental Exemplar Compression for Class-Incremental Learning. In *CVPR*, Vancouver, BC, June 2023. doi: 10.1109/CVPR52729.2023.01094.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online Continual Learning in Image Classification: An Empirical Survey. In *Neurocomputing*, volume 469, pp. 28–51, January 2022. doi: 10.1016/j.neucom.2021.10.021.

Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-Incremental Learning: Survey and Performance Evaluation on Image Classification. In *TPAMI*, volume 45(5), May 2023. doi: 1109/TPAMI.2022.3213473.

Felix Ott, David Rügamer, Lucas Heublein, Bernd Bischl, and Christopher Mutschler. Auxiliary Cross-Modal Representation Learning with Triplet Loss Functions for Online Handwriting Recognition. In *IEEE Access*, volume 11, pp. 94148–94172, August 2023. doi: 10.1109/ACCESS.2023.3310819.

Felix Ott, Lucas Heublein, Nisha Lakshmana Raichur, Tobias Feigl, Jonathan Hansen, Alexander Rügamer, and Christopher Mutschler. Few-Shot Learning with Uncertainty-based Quadruplet Selection for Interference Classification in GNSS Data. In *ICL-GNSS*, Antwerp, Belgium, June 2024.

Yiming Qin, Huangjie Zheng, Jiangchao Yao, Mingyuan Zhou, and Ya Zhang. Class-Balancing Diffusion Models. In *CVPR*, pp. 18434–18443, 2023.

Nisha L. Raichur, Tobias Brieger, Dorsaf Jdidi, Tobias Feigl, Johannes Rossouw van der Merwe, Birendra Ghimire, Felix Ott, Alexander Rügamer, and Wolfgang Felber. Machine Learning-assisted GNSS Interference Monitoring Through Crowdsourcing. In *ION GNSS+*, pp. 1151–1175, Denver, CO, September 2022.

Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R.G. Lanckriet, Roger Levy, and Nuno Vasconcelos. A New Approach to Cross-Modal Multimedia Retrieval. In *ACMMM*, pp. 251–260, October 2010. doi: 10.1145/1873951.1873987.

R. Ratcliff. Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. In *Psychological Review*, volume 97(2), April 1997. doi: 10.1037/0033-295x.97.2.285.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental Classifier and Representation Learning. In *CVPR*, pp. 2011–2010, July 2017. doi: 10.1109/CVPR.2017.587.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience Replay for Continual Learning. In *NIPS*, 2019.

Frédéric Ros, Rabia Riad, and Serge Guillaume. PDBI: A Partitioning Davies-Bouldin Index for Clustering Evaluation. In *Neurocomputing*, volume 528, pp. 178–199, April 2023. doi: 10.1016/j.neucom.2023.01.043.

Nikolaos Sarafianos, Xiang Xu, and Ioannis A. Kakadiaris. Adversarial Representation Learning for Text-to-Image Matching. In *ICCV*, pp. 5814–5824, Seoul, Korea, 2019. doi: 10.1109/ICCV.2019.00591.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *CVPR*, Boston, MA, June 2015. doi: 10.1109/CVPR.2015.7298682.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical Networks for Few-shot Learning. In *NIPS*, pp. 4080–4090, December 2017. doi: 10.5555/3294996.3295163.

Mamatha Thota and Georgios Leontidis. Contrastive Domain Adaptation. In *CVPRW*, Nashville, TN, June 2021. doi: 10.1109/CVPRW53098.2021.00250.

Michael C. Thrun. Distance-based Clustering Challenges for Unbiased Benchmarking Studies. In *Scientific Report*, volume 11(18988), 2011. doi: 10.1038/s41598-021-98126-1.

Dennis Treder-Tschechlov, Peter Reimann, Holger Schwarz, and Bernhard Mitschang. Approach to Synthetic Data Generation for Imbalanced Multi-class Problems with Heterogeneous Groups. In *Lecture Notes in Informatic (LNI), Gesellschaft für Informatik*, pp. 329–351, Dresden, Germany, 2023. doi: 10.18420/BTW2023-16.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data Using t-SNE. In *JMLR*, volume 9(86), pp. 2579–2605, November 2008.

Johannes Rossouw van der Merwe, David Contreras Franco, Jonathan Hansen, Tobias Brieger, Tobias Feigl, Felix Ott, Dorsaf Jdidi, Alexander Rügamer, and Wolfgang Felber. Low-Cost COTS GNSS Interference Monitoring, Detection, and Classification System. In *MDPI Sensors*, March 2023. doi: 10.3390/s23073452.

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. FOSTER: Feature Boosting and Compression for Class-Incremental Learning. In *ECCV*, October 2022. doi: 10.1007/978-3-031-19806-9_23.

Yunchao Wei, Yao Zhao, Canyi Lu, Shikui Wei, Luoqi Liu, Zhenfeng Zhu, and Shuicheng Yan. Cross-Modal Retrieval with CNN Visual Features: A New Baseline. In *Trans. on Cybernetics*, volume 47(2), pp. 449–460, March 2016. doi: 10.1109/TCYB.2016.2519449.

Jiangwei Xie, Shipeng Yan, and Xuming He. General Incremental Learning with Domain-aware Categorical Representations. In *CVPR*, pp. 14351–14360, 2022.

Shipeng Yan, Jiangwei Xie, and Xuming He. DER: Dynamically Expandable Representation for Class Incremental Learning. In *CVPR*, pp. 3014–3023, 2021.

Lu Yu, Bartłomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, and Yongmei Cheng. Semantic Drift Compensation for Class-Incremental Learning. In *CVPR*, Seattle, WA, June 2020. doi: 10.1109/CVPR42600.2020.00701.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep Class-Incremental Learning: A Survey. In *arXiv:2302.03648*, February 2023a.

Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning. In *ICLR*, 2023b.

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype Augmentation and Self-Supervision for Incremental Learning. In *CVPR*, Nashville, TN, June 2021. doi: 10.1109/CVPR46437.2021.00581.

Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-Sustaining Representation Expansion for Non-Exemplar Class-Incremental Learning. In *CVPR*, New Orleands, LA, June 2022. doi: 10.1109/CVPR52688.2022.00908.

A Appendix

A.1 Data Augmentation

Due to the potential for highly unbalanced training dataset, such as the interfered classes within the GNSS dataset, we address this issue by balancing each class through data augmentation techniques (Fairstein et al., 2024; Qin et al., 2023; Treder-Tschechlov et al., 2023). Specifically, we augment the data to ensure a uniform distribution of 1,000 samples per class. We employ techniques provided by *torchvision.transforms*, including color jittering with a brightness value of 0.5 and a hue value of 0.3, Gaussian blur with a kernel size of $(5, 9)$ and sigma values of $(0.1, 5.0)$, random adjustments of sharpness with a sharpness factor of 2, and random horizontal and vertical flipping with a probability of 0.4. Values are chosen after a hyperparameter search.

A.2 Bayesian Learning-driven Weighting

Figure 7 illustrates the dynamic weighting mechanism applied to the cross-entropy (CE) and contrastive learning (CL) loss functions during the training of the first task on the CIFAR-10 dataset. The CE loss quantifies performance on classification tasks, with a higher weight indicating that classification accuracy is either of greater importance or subject to higher uncertainty. Conversely, the CL loss facilitates the learning of meaningful feature representations by contrasting positive and negative pairs, where a higher weight signifies a focus on developing robust embeddings. The posterior distributions of the weights assigned to each loss function provide key insights. Broad posterior distributions indicate significant uncertainty regarding the relative importance of the losses. Furthermore, shifts in the weight distributions during training may reflect changing task priorities—for instance, emphasizing feature representation learning in the early stages and classification accuracy in later stages. When the CE loss diminishes and the model demonstrates confidence in classifications, the weighting of the CE loss may decrease, reallocating focus to the CL loss. Similarly, if the CL loss plateaus, its weighting may reduce to prioritize fine-tuning classification performance. As σ_1 and σ_2, the noise parameters associated with the variables $\mathbf{y}_1$ and $\mathbf{y}_2$, respectively (introduced in Section 3.3), increase, the weights assigned to $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{CL}}$ correspondingly decrease. This adjustment acts as a regularization mechanism for the noise terms. This phenomenon is particularly evident during the early stages of each task, where the introduction of new classes increases noise levels, leading to a reduction in the weighting of both loss functions. Early in training, the weight assigned to the CE loss is high, indicating a strong emphasis on optimizing classification accuracy when the model is still learning the basic structure of the task. The CL loss initially has a lower weight, suggesting that representation learning is of secondary importance during these early epochs. As training progresses, the weights for both loss functions gradually decrease, reflecting a reduction in the relative importance of their contributions as the model converges. The initial fluctuation in the weights, particularly for the CE loss, may signify the Bayesian framework's attempt to model and adapt to the high uncertainty in the early stages of training.

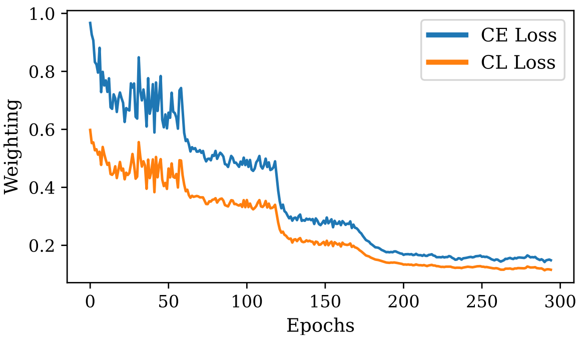


Figure 7: Weighting of the CE and CL loss functions.

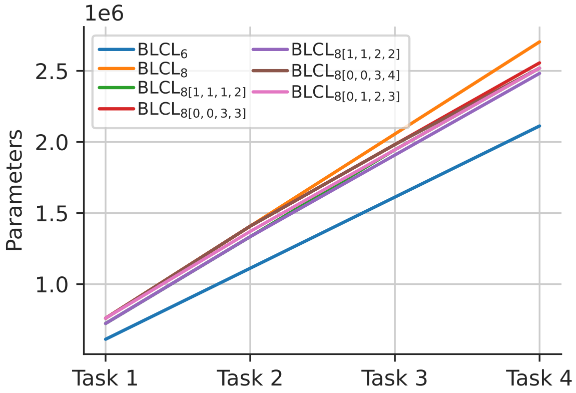


Figure 8: Number of model parameters for each task.

Table 4: Model properties.

Dataset	Method	Backbone	Model Size	Samples	Sample Size	Total
CIFAR-10 and CIFAR-100	Replay (Ratcliff, 1997)	ResNet32	1.75 MB	7,431	21.76 MB	23.51 MB
	iCarl (Rebuffi et al., 2017)	ResNet32	1.75 MB	7,431	21.76 MB	23.51 MB
	DER (Yan et al., 2021)	ResNet32	17.68 MB	2,000	5.85 MB	23.53 MB
	MEMO (Zhou et al., 2023b)	ResNet32	13.83 MB	3,312	9.70 MB	23.53 MB
	BLCL (ours)	ResNet50	15.85 MB	2,000	5.85 MB	**20.85** MB
GNSS	Replay (Ratcliff, 1997)	ResNet18	42.60 MB	4,970	713.50 MB	756.10 MB
	iCarl (Rebuffi et al., 2017)	ResNet18	42.60 MB	4,970	713.50 MB	756.10 MB
	DER (Yan et al., 2021)	ResNet18	468.00 MB	2,000	287.00 MB	755.00 MB
	MEMO (Zhou et al., 2023b)	ResNet18	362.80 MB	2,739	393.20 MB	755.00 MB
	BLCL (ours)	ResNet18	362.80 MB	2,000	287.00 MB	**649.80** MB

Over time, the smoother trends in the weights imply that the uncertainty has diminished as the model gains confidence in its predictions and representations. The decrease in weighting for both loss functions over epochs acts as an implicit regularization mechanism. This ensures that the model does not overfit to specific aspects of the loss functions and instead balances classification performance and representation learning. By around 200 epochs, both weights stabilize, indicating that the contributions of the CE and CL losses have reached an equilibrium. This suggests that the model has adequately learned both the task-specific and representation-related objectives.

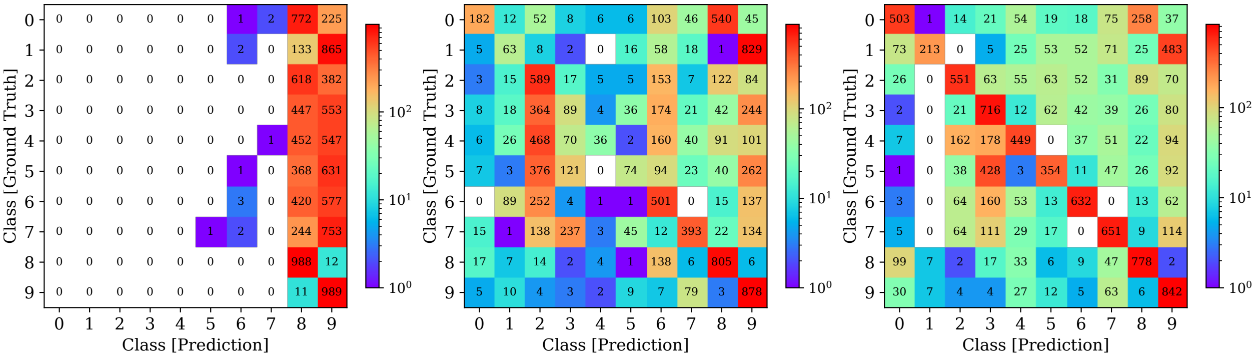
A.3 Hardware & Training Setup

All experiments are conducted utilizing Nvidia Tesla V100-SXM2 GPUs with 32 GB VRAM, equipped with Core Xeon CPUs and 192 GB RAM. We use the vanilla Adam optimizer with a learning rate set to 0.1, a decay rate of 0.1, a batch size of 128, and train each task for 300 epochs.

After each task we reduce the number of exemplars of previous classes. For example, the CIFAR-100 dataset has 10 tasks with each 10 classes. For the first task, the dataset contains 500 samples for each class for 10 classes. For the second task, the dataset contains 200 samples for each of the previous 10 classes and 500 samples for the new 10 classes. For the third task, the dataset contains 100 for each of the previous 20 classes and 500 samples for the new 10 classes. For the fourth task, the dataset contains 66 samples for each of the previous 30 classes and 500 samples for the new 10 classes. In this way, the number of samples of the previous classes decrease to accommodate samples of new classes in a fixed memory.

A.4 Parameter Counts

Figure 8 presents a detailed overview of the parameter counts for various configurations of the BLCLBLCL architecture, including BLCL_6, BLCL_8, and specialized configurations $\text{BLCL}_{8[s]}$, across all tasks within the CIFAR-10 dataset. The configurations differ based on the number and arrangement of dynamic specialized components. Transitioning from BLCL_6 to BLCL_8, which increases the number of blocks in the specialized component from 6 to 8, leads to a rise in parameter count from approximately $2.1 \cdot 10^6$ to $2.7 \cdot 10^6$. This increase in model complexity correlates with an improvement in accuracy (as reported in Table 2), highlighting the trade-off between model size and performance. The configurations BLCL_8 feature selective reductions in the number of specialized blocks for different tasks, as indicated in the legend. For example, $\text{BLCL}_8[0, 0, 0, 0]$ represents the smallest architecture, with minimal or no specialized components. Despite the reduced parameter count, these configurations maintain competitive accuracy, showcasing the efficiency of dynamic specialization in balancing model size and performance. As observed in Figure 8, the parameter count increases progressively with each additional task, reflecting the adaptive inclusion of specialized components. Configurations with more specialized blocks, such as $\text{BLCL}_8[1, 1, 1, 2]$ and $\text{BLCL}_8[1, 1, 2, 3]$, experience a steeper increase in parameter count compared to BLCL_6. This scaling behavior demonstrates the flexibility of the architecture to adapt to task complexity while controlling overall model growth. The comparison of $\text{BLCL}_8[1, 1, 2, 3]$ with simpler configurations like $\text{BLCL}_8[0, 0, 0, 0]$ or $\text{BLCL}_8[0, 0, 3, 3]$ illustrates how selective reduction of parameters can retain performance. This observation suggests that the dynamic allocation of specialized components effectively balances the need for task-specific learning and model gen-



	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	1	2	772	225
1	0	0	0	0	0	0	2	0	133	865
2	0	0	0	0	0	0	0	0	618	382
3	0	0	0	0	0	0	0	0	447	553
4	0	0	0	0	0	0	0	1	452	547
5	0	0	0	0	0	0	1	0	368	631
6	0	0	0	0	0	0	3	0	420	577
7	0	0	0	0	0	1	2	0	244	753
8	0	0	0	0	0	0	0	0	988	12
9	0	0	0	0	0	0	0	0	11	989

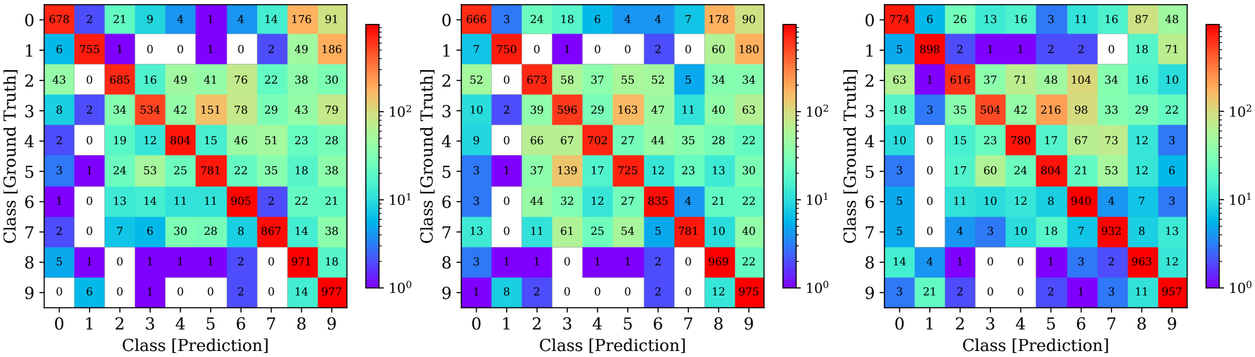
(a) Finetune.

	0	1	2	3	4	5	6	7	8	9
0	182	12	52	8	6	6	103	46	540	45
1	5	63	8	2	0	16	58	18	1	829
2	3	15	589	17	5	5	153	7	122	84
3	8	18	364	89	4	36	174	21	42	244
4	6	26	468	70	36	2	160	40	91	101
5	7	3	376	121	0	74	94	23	40	262
6	0	89	252	4	1	1	501	0	15	137
7	15	1	138	237	3	45	12	393	22	134
8	17	7	14	2	4	1	138	6	805	6
9	5	10	4	3	2	9	7	79	3	878

(b) EWC (Kirkpatrick et al., 2017).

	0	1	2	3	4	5	6	7	8	9
0	503	1	14	21	54	19	18	75	258	37
1	73	213	0	5	25	53	52	71	25	483
2	26	0	551	63	55	63	52	31	89	70
3	2	0	21	716	12	62	42	39	26	80
4	7	0	162	178	449	0	37	51	22	94
5	1	0	38	428	3	354	11	47	26	92
6	3	0	64	160	53	13	632	0	13	62
7	5	0	64	111	29	17	0	651	9	114
8	99	7	2	17	33	6	9	47	778	2
9	30	7	4	4	27	12	5	63	6	842

(c) LwF (Li & Hoiem, 2018).



	0	1	2	3	4	5	6	7	8	9
0	678	2	21	9	4	1	4	14	176	91
1	6	755	1	0	0	1	0	2	49	186
2	43	0	685	16	49	41	76	22	38	30
3	8	2	34	534	42	151	78	29	43	79
4	2	0	19	12	804	15	46	51	23	28
5	3	1	24	53	25	781	22	35	18	38
6	1	0	13	14	11	11	905	2	22	21
7	2	0	7	6	30	28	8	867	14	38
8	5	1	0	1	1	1	2	0	971	18
9	0	6	0	1	0	0	2	0	14	977

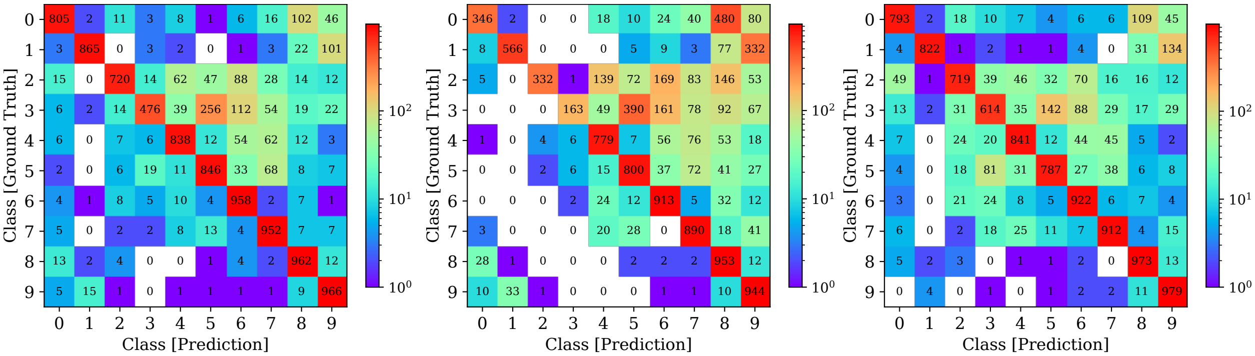
(d) iCarl (Rebuffi et al., 2017).

	0	1	2	3	4	5	6	7	8	9
0	666	3	24	18	6	4	4	7	178	90
1	7	750	0	1	0	0	2	0	60	180
2	52	0	673	58	37	55	52	5	34	34
3	10	2	39	596	29	163	47	11	40	63
4	9	0	66	67	702	27	44	35	28	22
5	3	1	37	139	17	725	12	23	13	30
6	3	0	44	32	12	27	835	4	21	22
7	13	0	11	61	25	54	5	781	10	40
8	3	1	1	0	1	1	2	0	969	22
9	1	8	2	0	0	0	2	0	12	975

(e) Replay (Ratcliff, 1997).

	0	1	2	3	4	5	6	7	8	9
0	774	6	26	13	16	3	11	16	87	48
1	5	898	2	1	1	2	2	0	18	71
2	63	1	616	37	71	48	104	34	16	10
3	18	3	35	504	42	216	98	33	29	22
4	10	0	15	23	780	17	67	73	12	3
5	3	0	17	60	24	804	21	53	12	6
6	5	0	11	10	12	8	940	4	7	3
7	5	0	4	3	10	18	7	932	8	13
8	14	4	1	0	0	1	3	2	963	12
9	3	21	2	0	0	2	1	3	11	957

(f) MEMO (Zhou et al., 2023b).



	0	1	2	3	4	5	6	7	8	9
0	805	2	11	3	8	1	6	16	102	46
1	3	865	0	3	2	0	1	3	22	101
2	15	0	720	14	62	47	88	28	14	12
3	6	2	14	476	39	256	112	54	19	22
4	6	0	7	6	838	12	54	62	12	3
5	2	0	6	19	11	846	33	68	8	7
6	4	1	8	5	10	4	958	2	7	1
7	5	0	2	2	8	13	4	952	7	7
8	13	2	4	0	0	1	4	2	962	12
9	5	15	1	0	1	1	1	1	9	966

(g) DER (Yan et al., 2021).

	0	1	2	3	4	5	6	7	8	9
0	346	2	0	0	18	10	24	40	480	80
1	8	566	0	0	0	5	9	3	77	332
2	5	0	332	1	139	72	169	83	146	53
3	0	0	0	163	49	390	161	78	92	67
4	1	0	4	6	779	7	56	76	53	18
5	0	0	2	6	15	800	37	72	41	27
6	0	0	0	2	24	12	913	5	32	12
7	3	0	0	0	20	28	0	890	18	41
8	28	1	0	0	0	2	2	2	953	12
9	10	33	1	0	0	0	1	1	10	944

(h) FOSTER (Wang et al., 2022).

	0	1	2	3	4	5	6	7	8	9
0	793	2	18	10	7	4	6	6	109	45
1	4	822	1	2	1	1	4	0	31	134
2	49	1	719	39	46	32	70	16	16	12
3	13	2	31	614	35	142	88	29	17	29
4	7	0	24	20	841	12	44	45	5	2
5	4	0	18	81	31	787	27	38	6	8
6	3	0	21	24	8	5	922	6	7	4
7	6	0	2	18	25	11	7	912	4	15
8	5	2	3	0	1	1	2	0	973	13
9	0	4	0	1	0	1	2	2	11	979

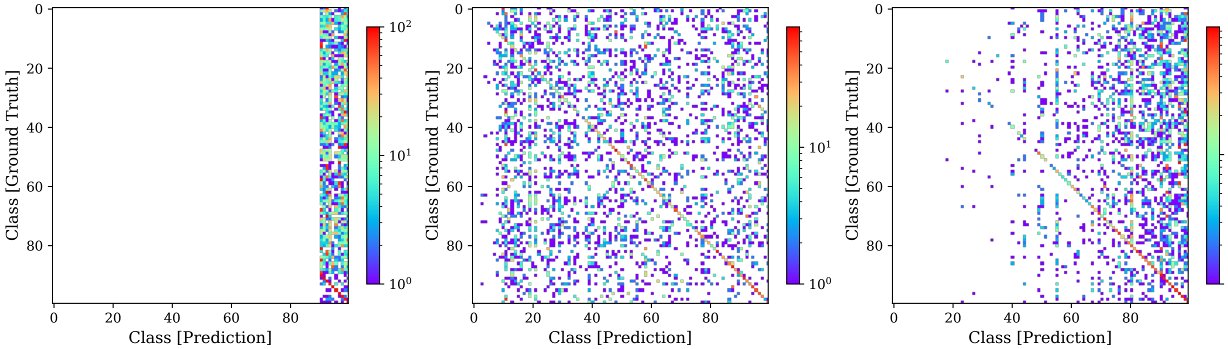
(i) $\text{BLCL}_{8[1,1,2,2]}$.

Figure 9: Confusion matrices of state-of-the-art methods and our Bayesian BLCL approach for the CIFAR-10 dataset.

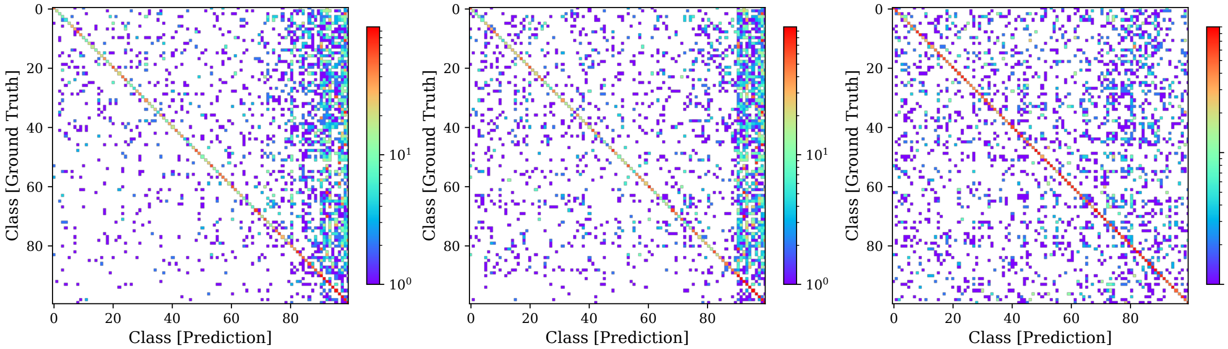
eralization. Table 4 contextualizes these configurations by comparing their parameter counts, model sizes, and memory footprints with baseline methods such as Replay, iCarl, and DER. The BLCL configurations achieve competitive or superior performance while maintaining manageable model sizes.

A.5 Embedding Analysis After Each Task

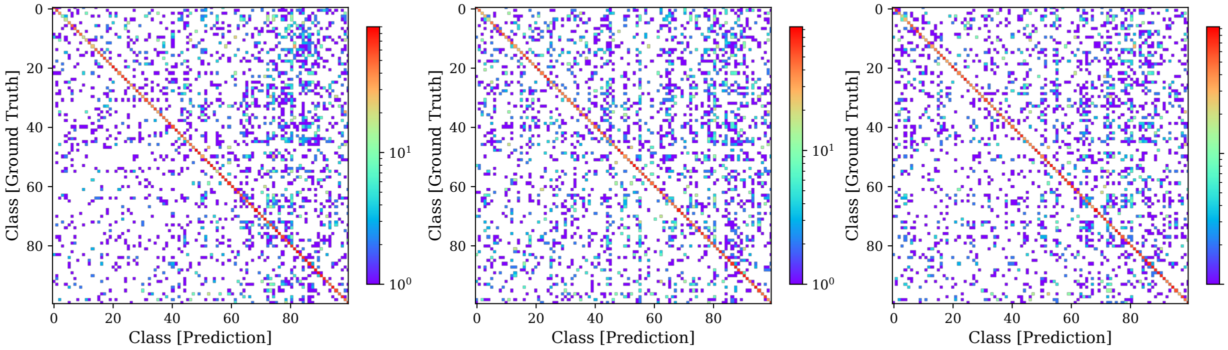
Figure 15 visualizes the feature embeddings after each of the tasks. Specifically for the CIFAR-10 dataset (a to d), when adding clusters from new tasks, a large inter-class distance between clusters from previous classes are maintained. However, for the GNSS dataset (e to i), the samples from classes 3 and 4 from task 2 are added in-between the classes 0 to 2 due to a high similarity between the classes.



(a) Finetune. (b) EWC (Kirkpatrick et al., 2017). (c) LwF (Li & Hoiem, 2018).

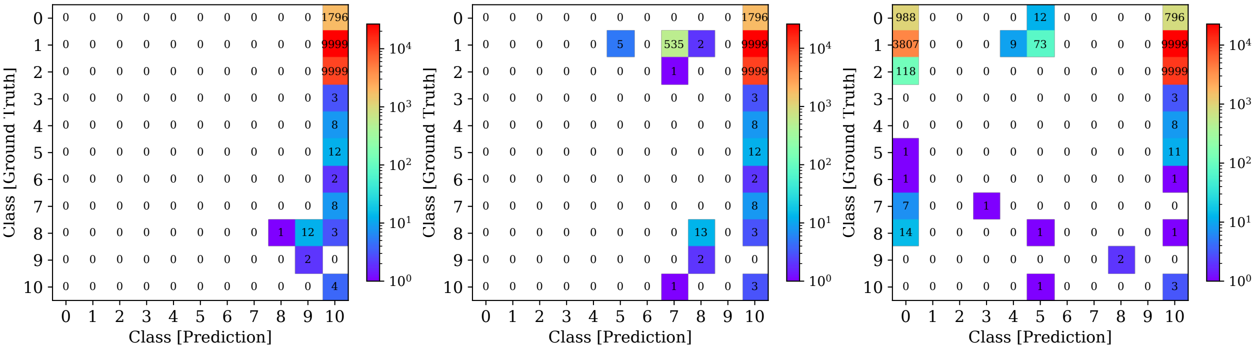


(d) iCarl (Rebuffi et al., 2017). (e) Replay (Ratcliff, 1997). (f) MEMO (Zhou et al., 2023b).



(g) DER (Yan et al., 2021). (h) FOSTER (Wang et al., 2022). (i) BLCL$_{8[*]}$.

Figure 10: Confusion matrices of state-of-the-art methods and our Bayesian BLCL approach for the CIFAR-100 dataset.



	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	1796
1	0	0	0	0	0	0	0	0	0	0	9999
2	0	0	0	0	0	0	0	0	0	0	9999
3	0	0	0	0	0	0	0	0	0	0	3
4	0	0	0	0	0	0	0	0	0	0	8
5	0	0	0	0	0	0	0	0	0	0	12
6	0	0	0	0	0	0	0	0	0	0	2
7	0	0	0	0	0	0	0	0	0	0	8
8	0	0	0	0	0	0	0	0	1	12	3
9	0	0	0	0	0	0	0	0	0	2	0
10	0	0	0	0	0	0	0	0	0	0	4

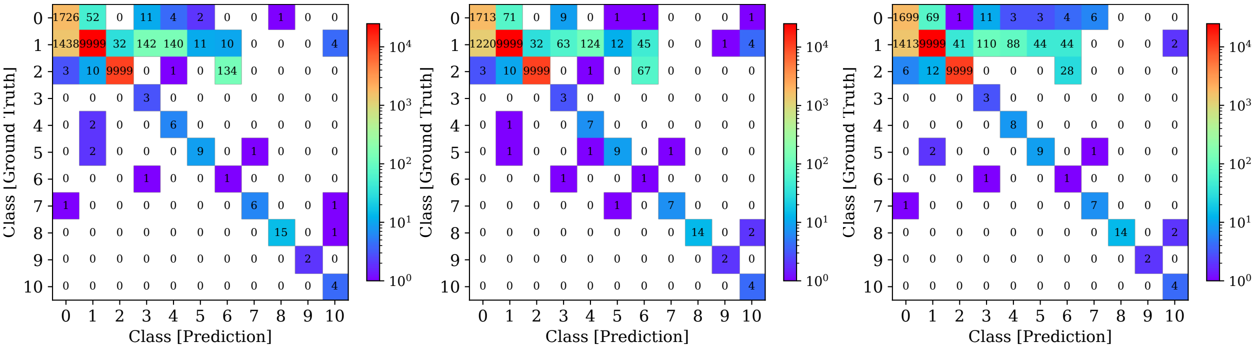
(a) Finetune.

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	1796
1	0	0	0	0	0	5	0	535	2	0	9999
2	0	0	0	0	0	0	0	1	0	0	9999
3	0	0	0	0	0	0	0	0	0	0	3
4	0	0	0	0	0	0	0	0	0	0	8
5	0	0	0	0	0	0	0	0	0	0	12
6	0	0	0	0	0	0	0	0	0	0	2
7	0	0	0	0	0	0	0	0	0	0	8
8	0	0	0	0	0	0	0	0	13	0	3
9	0	0	0	0	0	0	0	0	2	0	0
10	0	0	0	0	0	0	0	1	0	0	3

(b) EWC (Kirkpatrick et al., 2017).

	0	1	2	3	4	5	6	7	8	9	10
0	988	0	0	0	0	12	0	0	0	0	796
1	3807	0	0	0	9	73	0	0	0	0	9999
2	118	0	0	0	0	0	0	0	0	0	9999
3	0	0	0	0	0	0	0	0	0	0	3
4	0	0	0	0	0	0	0	0	0	0	8
5	1	0	0	0	0	0	0	0	0	0	11
6	1	0	0	0	0	0	0	0	0	0	1
7	7	0	0	1	0	0	0	0	0	0	0
8	14	0	0	0	0	1	0	0	0	0	1
9	0	0	0	0	0	0	0	0	2	0	0
10	0	0	0	0	0	1	0	0	0	0	3

(c) LwF (Li & Hoiem, 2018).



	0	1	2	3	4	5	6	7	8	9	10
0	1726	52	0	11	4	2	0	0	1	0	0
1	1438	9999	32	142	140	11	10	0	0	0	4
2	3	10	9999	0	1	0	134	0	0	0	0
3	0	0	0	3	0	0	0	0	0	0	0
4	0	2	0	0	6	0	0	0	0	0	0
5	0	2	0	0	0	9	0	1	0	0	0
6	0	0	0	1	0	0	1	0	0	0	0
7	1	0	0	0	0	0	0	6	0	0	1
8	0	0	0	0	0	0	0	0	15	0	1
9	0	0	0	0	0	0	0	0	0	2	0
10	0	0	0	0	0	0	0	0	0	0	4

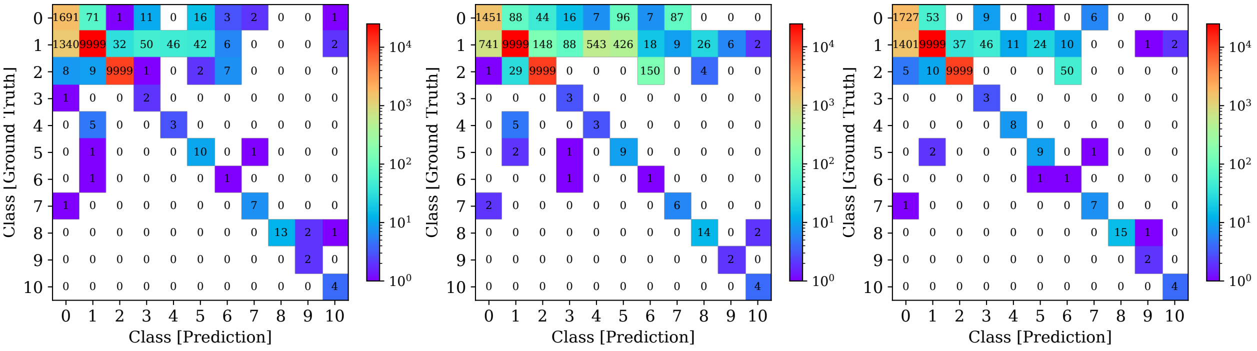
(d) iCarl (Rebuffi et al., 2017).

	0	1	2	3	4	5	6	7	8	9	10
0	1713	71	0	9	0	1	1	0	0	0	1
1	1220	9999	32	63	124	12	45	0	0	1	4
2	3	10	9999	0	1	0	67	0	0	0	0
3	0	0	0	3	0	0	0	0	0	0	0
4	0	1	0	0	7	0	0	0	0	0	0
5	0	1	0	0	1	9	0	1	0	0	0
6	0	0	0	1	0	0	1	0	0	0	0
7	0	0	0	0	0	1	0	7	0	0	0
8	0	0	0	0	0	0	0	0	14	0	2
9	0	0	0	0	0	0	0	0	0	2	0
10	0	0	0	0	0	0	0	0	0	0	4

(e) Replay (Ratcliff, 1997).

	0	1	2	3	4	5	6	7	8	9	10
0	1699	69	1	11	3	3	4	6	0	0	0
1	1413	9999	41	110	88	44	44	0	0	0	2
2	6	12	9999	0	0	0	28	0	0	0	0
3	0	0	0	3	0	0	0	0	0	0	0
4	0	0	0	0	8	0	0	0	0	0	0
5	0	2	0	0	0	9	0	1	0	0	0
6	0	0	0	1	0	0	1	0	0	0	0
7	1	0	0	0	0	0	0	7	0	0	0
8	0	0	0	0	0	0	0	0	14	0	2
9	0	0	0	0	0	0	0	0	0	2	0
10	0	0	0	0	0	0	0	0	0	0	4

(f) MEMO (Zhou et al., 2023b).



	0	1	2	3	4	5	6	7	8	9	10
0	1691	71	1	11	0	16	3	2	0	0	1
1	1340	9999	32	50	46	42	6	0	0	0	2
2	8	9	9999	1	0	2	7	0	0	0	0
3	1	0	0	2	0	0	0	0	0	0	0
4	0	5	0	0	3	0	0	0	0	0	0
5	0	1	0	0	0	10	0	1	0	0	0
6	0	1	0	0	0	0	1	0	0	0	0
7	1	0	0	0	0	0	0	7	0	0	0
8	0	0	0	0	0	0	0	0	13	2	1
9	0	0	0	0	0	0	0	0	0	2	0
10	0	0	0	0	0	0	0	0	0	0	4

(g) DER (Yan et al., 2021).

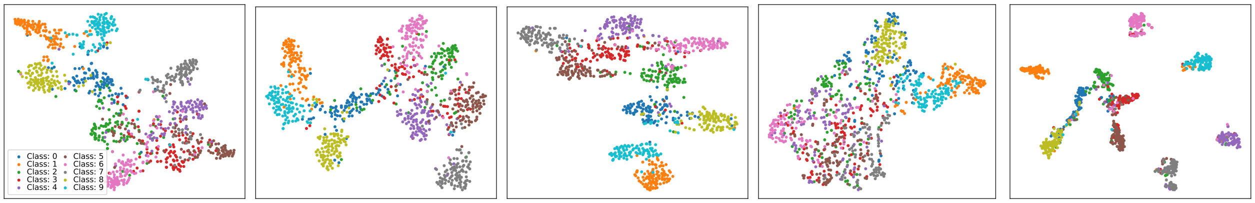
	0	1	2	3	4	5	6	7	8	9	10
0	1451	88	44	16	7	96	7	87	0	0	0
1	741	9999	148	88	543	426	18	9	26	6	2
2	1	29	9999	0	0	0	150	0	4	0	0
3	0	0	0	3	0	0	0	0	0	0	0
4	0	5	0	0	3	0	0	0	0	0	0
5	0	2	0	1	0	9	0	0	0	0	0
6	0	0	0	1	0	0	1	0	0	0	0
7	2	0	0	0	0	0	0	6	0	0	0
8	0	0	0	0	0	0	0	0	14	0	2
9	0	0	0	0	0	0	0	0	0	2	0
10	0	0	0	0	0	0	0	0	0	0	4

(h) FOSTER (Wang et al., 2022).

	0	1	2	3	4	5	6	7	8	9	10
0	1727	53	0	9	0	1	0	6	0	0	0
1	1401	9999	37	46	11	24	10	0	0	1	2
2	5	10	9999	0	0	0	50	0	0	0	0
3	0	0	0	3	0	0	0	0	0	0	0
4	0	0	0	0	8	0	0	0	0	0	0
5	0	2	0	0	0	9	0	1	0	0	0
6	0	0	0	0	0	1	1	0	0	0	0
7	1	0	0	0	0	0	0	7	0	0	0
8	0	0	0	0	0	0	0	0	15	1	0
9	0	0	0	0	0	0	0	0	0	2	0
10	0	0	0	0	0	0	0	0	0	0	4

(i) $\mathrm{BLCL}_{8[1,1,2,2]}$.

Figure 11: Confusion matrices of state-of-the-art methods and our Bayesian BLCL approach for the GNSS dataset.



(a) Replay (Ratcliff, 1997).

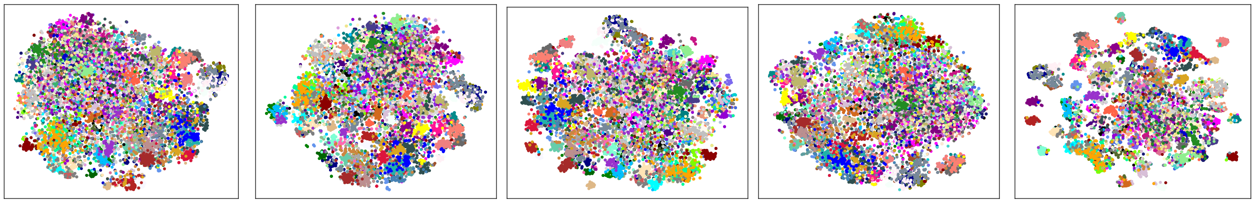
(b) MEMO (Zhou et al., 2023b).

(c) DER (Yan et al., 2021).

(d) FOSTER (Wang et al., 2022).

(e) BLCL$_{8[1,1,2,2]}$.

Figure 12: t-SNE (van der Maaten & Hinton, 2008) plots for the CIFAR-10 dataset.



(a) Replay (Ratcliff, 1997).

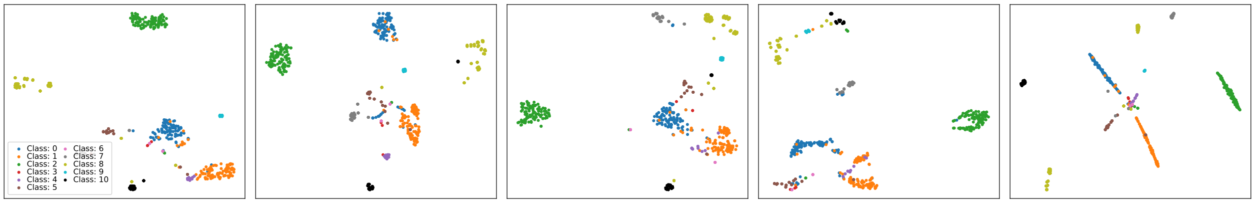
(b) MEMO (Zhou et al., 2023b).

(c) DER (Yan et al., 2021).

(d) FOSTER (Wang et al., 2022).

(e) BLCL$_{8[*]}$.

Figure 13: t-SNE (van der Maaten & Hinton, 2008) plots for the CIFAR-100 dataset.



(a) Replay (Ratcliff, 1997).

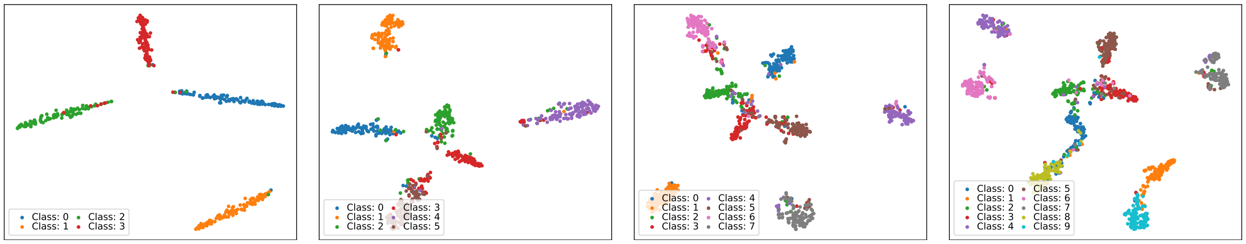
(b) MEMO (Zhou et al., 2023b).

(c) DER (Yan et al., 2021).

(d) FOSTER (Wang et al., 2022).

(e) BLCL$_{8[1,1,2,2]}$.

Figure 14: t-SNE (van der Maaten & Hinton, 2008) plots for the GNSS dataset.

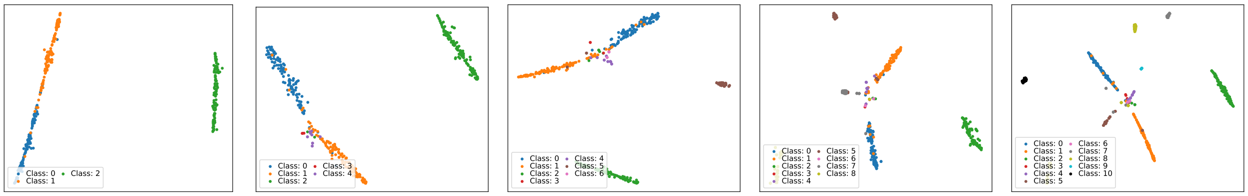


(a) Task 1.

(b) Task 2.

(c) Task 3.

(d) Task 4.



(e) Task 1.

(f) Task 2.

(g) Task 3.

(h) Task 4.

(i) Task 5.

Figure 15: t-SNE (van der Maaten & Hinton, 2008) plots for the CIFAR-10 (a to d) and GNSS (e to i) datasets after each task for our BLCL method.