CoRaL: Continual Representation Learning for Overcoming Catastrophic Forgetting

Mohammad Samin Yasar
University of Virginia
Charlottesville, Virginia, USA
msy9an@virginia.edu

Tariq Iqbal
University of Virginia
Charlottesville, Virginia, USA
tiqbal@virginia.edu

ABSTRACT

Humans have the ability to acquire, retain and transfer knowledge over their lifespan. For intelligent agents to achieve fluent longitudinal interaction, they need to continually retain, refine and acquire new knowledge. However, current learning approaches, in particular Deep Neural Networks, are prone to catastrophic forgetting, a phenomenon where the network forgets its past representation as the data distribution changes. To address this challenge, in this work, we propose CoRaL, a novel continual learning framework that considers the past response of the network when learning a new task. CoRaL comprises a Representation Learning module that learns representations that are robust to distribution shifts and a Knowledge Distillation module that encourages the network to retain past knowledge. The Representation Learning module is a Siamese Network setup that maximizes the similarity between two augmented versions of the input. The Knowledge Distillation module buffers past inputs and penalizes divergence between past and current network output. We evaluated CoRaL on three challenging Continual Learning scenarios across four datasets. The results suggest that CoRaL outperformed all evaluated state-of-the-art methods, achieving the highest accuracy and lowest forgetting. Finally, we conducted extensive ablation studies to highlight the importance of the proposed modules in addressing catastrophic forgetting.

KEYWORDS

Continual Learning, Catastrophic Forgetting, Representation Learning

ACM Reference Format:
Mohammad Samin Yasar and Tariq Iqbal. 2023. CoRaL: Continual Representation Learning for Overcoming Catastrophic Forgetting. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 10 pages.

1 INTRODUCTION

Intelligent agents have to adapt and interact with their environment using a continuous stream of observations, which requires that representations be learned in a continual manner [43]. However, continual learning does not suit current learning paradigms, which involve training Deep Neural Networks with the assumption that the training distribution is stationary and that the data samples are independent and identically distributed (i.i.d.) [41]. As such, current optimization strategies for training these networks focus on learning a representation from the existing data only and do not account explicitly for past observed data [23, 40]. As such, when these networks are tasked to learn from a sequential non-stationary data stream, they suffer from *catastrophic forgetting*, when the network forgets representation salient to the past task/data distribution [46].

Continual or Lifelong Learning approaches try to address the problem of catastrophic forgetting by acquiring new knowledge and refining existing representations from continuous non-stationary data such that the past knowledge is not completely overwritten [19, 26, 54, 57, 58]. Prior methods for addressing this problem can be grouped into three categories: *regularization-based*, *network expansion-based*, and *rehearsal-based* approaches. *Regularization-based* approaches induce a stability-plasticity trade-off in the network by penalizing the updates of specific parameters that are deemed important for past tasks [37, 55, 64]. *Network expansion-based* approaches instantiates new networks or modules for each new task [42, 51]. Lastly, *rehearsal-based* approaches mitigate forgetting by using a memory buffer of past data samples. These data are then replayed along with the samples of the current task to build optimization constraints during backpropagation [6, 8, 10, 44, 49, 50].

Although the previous works have all contributed to reducing catastrophic forgetting, their performances have yet to match offline learning. These methods predominantly focus on reducing the negative backward transfer of past tasks without explicitly improving the Representation Learning of the network, which is crucial for intelligent agents to generalize in incremental settings. However, we posit that the key to reducing backward transfer is to learn rich representations that can be shared among all tasks. Recently, work on self-supervised learning has shown promising results in learning robust representations using a pretext task [11, 13, 16, 27, 36]. While these methods can generate robust representations, they are prone to catastrophic forgetting when a new task is introduced.

To reduce catastrophic forgetting while maintaining performance, we propose CoRaL, a Continual Representation Learning approach for Overcoming Catastrophic Forgetting, that unifies Representation Learning with Continual Learning (CL). Our approach tackles the problems of CL from two different aspects: learning effective representations that can be retained, refined, and transferred in incremental settings; and encouraging the model to retain its past responses. CoRaL introduces Representation Learning for non-stationary distributions to learn a robust representation. The Representation Learning module is a Siamese network setup [7, 11] comprising an encoder, projection, and predictor network. This is trained using the Cosine Similarity loss, which is used to minimize the distance between representations of the same class.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

While learning transferable features can mitigate catastrophic forgetting, it may not explicitly direct the network to retain its past response to old training samples. To address this issue, we introduce a knowledge-distillation loss that compares the network's current output to its past output and penalizes divergence. The distillation loss imposes constraints on the parameter update, which prevents the network from forgetting the weights on the past samples. Thus, our overall framework unifies Representation Learning with Knowledge Distillation and is trained end-to-end with a novel objective function. The proposed objective function balances stability using the distillation loss and plasticity via the Representation Learning loss, which is now added to the existing Cross-Entropy loss. CoRaL is the first approach to efficiently combine Supervised Learning, Representation Learning, and Knowledge Distillation in an end-to-end manner through a novel objective function.

We performed extensive experiments to evaluate the efficacy of CoRaL, across three CL scenarios: incremental task, incremental class and incremental domain, on four widely used datasets in Continual Learning: permuted-MNIST [64], rotated-MNIST [44], Split-TinyImageNet [14] and split-CIFAR10 [64]. The results underline CoRaL's effectiveness in addressing catastrophic forgetting, as it outperformed all evaluated CL algorithms across all benchmarks attaining the highest accuracy with low standard deviation and the lowest forgetting. Furthermore, through extensive experiments, we demonstrated that these three objectives could be combined in a complementary manner for Continual Learning (CL). Finally, we conducted extensive ablation and stability-plasticity analyses to assess the efficacy of each of our modules across different scenarios and datasets. The ablation studies underline the importance of CoRaL's learning modules and provide empirical support for the objective function for Representation Learning. Our results provide promising direction for intelligent agents to learn continually.

2 RELATED WORK

Continual Learning Strategies: Prior works in CL have commonly been evaluated in three scenarios: *incremental task*, *incremental class* and *incremental domain*. In *incremental task*, the output spaces (and task-learning layers) are disjoint, and task boundaries are explicitly stated [29]. Dissolving the class-boundaries leads to *incremental class*, where the model needs to infer both classes (new and old) and the shift in task. Finally, in *incremental domain*, the classes remain the same, but the inputs undergo a distribution shift.

Towards addressing the challenges brought about by these scenarios, recent work in CL can be grouped into *regularization-based*, *network expansion-based*, and *rehearsal-based* methods. *Regularization-based* approaches aim to address catastrophic forgetting by imposing constraints on the update of specific model parameters via additional regularization terms [2, 21, 37, 52, 55, 64]. For example, Elastic Weight Consolidation (EWC) identifies important parameters using the diagonal values of the Fisher information matrix, which are then regularized when learning on new tasks [37, 55]. Synaptic Intelligence takes a different approach to identify important parameters for each task, relying on loss sensitivity with respect to the particular parameters [64]. While regularization-based approaches have shown promising results, they are known to perform poorly when the number of tasks is high or in incremental-class settings.

While regularization approaches focus on constraining the updates of a fixed-capacity network, *network expansion-based* techniques add to the existing architecture every time there is a change in task [42, 51, 53, 62, 63]. For example, Progressive Neural Networks expand the architecture by allocating new sub-networks with fixed capacity for each new task while freezing previously trained networks [51]. Li et al. [42] proposed a learn to grow framework that employs a neural architecture search to find the optimal architecture for each sequential task. The key limitation with network expansion approaches is the increase in computational overhead, and the added complexity of performing a hyper-parameter search for each new task.

Lastly, *rehearsal-based* methods use a memory buffer of past data which is replayed when learning new tasks [6, 8, 10, 44, 49, 50]. Buzzega et al. [8] proposed Dark Experience Replay, which added distillation loss and cross-entropy on previous task samples to reduce forgetting. Lopez-Paz et al. [44] proposed Gradient Episodic Memory, which uses past data to recall gradient directions and then project new gradients in a region that ensures that past representation is not over-written. These past data can also be used to add an additional objective term that can limit the forgetting on pivotal learned data points, as proposed in Hindsight Anchor Learning [9]. Sokar et al. [56] proposed a self-attention meta-learner, which incorporates an attention mechanism that learns to select particular representation for each task. Cha et al. [20] proposed Co^{2}L, that combines knowledge distillation with representation learning using the supervised Contrastive Learning objective [36]. The authors used a two-phase approach to train their framework, first for learning the representation and second for training the classifier.

Continual Learning in Intelligent Agents: For intelligent agents to become fully autonomous, they need to perceive and adapt to the changes in environmental dynamics [18, 24, 30, 60, 61]. Along this line, progress has been made in detecting changes and generalizing to new environments [1, 31–35, 45]. Recently, CL techniques have been introduced to applications ranging from object detection [4, 5, 15] to knowledge embeddings [17] to motion prediction [38, 59]. To mitigate catastrophic forgetting in object classification, Ayub et al. [4] proposed a centroid-based concept learning approach (CBCL), which uses a pre-trained feature extractor to obtain features for every input, on which an AggVar clustering algorithm is applied to generate centroids. Knoedler et al. [38] proposed a self-supervised approach to predicting pedestrian trajectories that uses online streams of data of pedestrian trajectories to continuously refine the model's prediction. Pellegrini et al. [47] proposed the use of latent replay, which combines with naive rehearsal, to classify objects on video benchmarks.

Although the aforementioned works have shown promising results for CL, learning effective representations that can be retained, refined, and transferred incrementally remains a long-standing challenge. Furthermore, certain approaches are only effective in specific scenarios, such as regularization-based approaches perform best on incremental-task and fail to achieve competitive results in other settings. Although recent works on improving representation learning have shown promising results [20, 22, 48], they require several changes which in turn relaxes the Continual Learning assumption, such as a two-phase training scheme for CL, class-balancing [20, 22]. To address these shortcomings, we propose a novel framework that

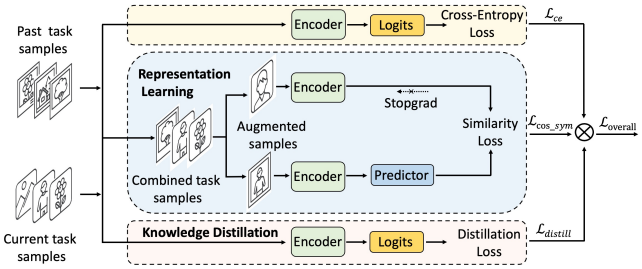

Figure 1: CoRaL: Continual Representation Learning for Overcoming Catastrophic Forgetting. The Representation Learning module maximizes the similarity between two augmented views of the input, which leads to more robust features. The Knowledge Distillation module distills the knowledge of previous tasks by buffering past input-output tuples of the network from memory. The two modules combine to reduce catastrophic forgetting.

unifies the Representation Learning for learning robust representation with a memory buffer that allows replaying of past samples and enables the network to optimize over a small set of past data.

3 PROBLEM FORMULATION

Formally defined, a Continual Learning problem comprises a sequence of T distinct tasks containing non-overlapping input-output pairs. The overall goal for the agent is to accurately predict new classes as they appear without forgetting the discriminative ability of past classes. We use superscripts to represent the task and subscripts to represent the index in all our formulations.

Let us denote inputs as X and labels as Y. As such, (x_i^t, y_i^t) represent an input-label tuple for a given task t. For *incremental task* (IL-Task) scenarios, the output (or label) space is disjoint, i.e., $Y_i^n \neq Y_i^m$ for two different tasks m and n. The same also applies for *incremental class* (IL-Class) scenarios. In *incremental domain* (IL-Domain) settings, the output space remains the same but the input space changes with each domain, i.e., $Y_i^n = Y_i^m$ and $X_i^n \neq X_i^m$.

For each task, input-label $(x_i^t, y_i^t) \sim D^t$ pairs are independently drawn from some task-specific distribution D^t. The learner is tasked to learn a non-linear mapping function using an encoder f_θ and a g_θ, which would correctly predict the output label for the input. Here, θ represents the parameters of the non-linear functions. For IL-Task settings, the learner is trained using the following objective:

$$\mathcal{L}(\theta) := \sum_{t=1}^{T} \mathbf{E}_{D^t}\left[l(y^t, g_\theta(f_\theta(x^t, t)))\right] \tag{1}$$

Here, the learner has access to the task label and will have different task-learning layers per task. $l(\cdot,\cdot)$ represents the loss function that needs to be minimized. For IL-Class and IL-Domain, the learner has one task-specific layer and is trained as follows:

$$\mathcal{L}(\theta) := \sum_{t=1}^{T} \mathbf{E}_{D^t}\left[l(y^t, g_\theta(f_\theta(x^t)))\right] \tag{2}$$

4 CORAL: CONTINUAL REPRESENTATION LEARNING

We now introduce our proposed framework, CoRaL: Continual Representation Learning, an end-to-end representation learning framework to tackle catastrophic forgetting in CL. The overall algorithm for our framework is provided in Algo. 1 and illustrated in Fig. 1. There are two primary components of CoRaL, which work in tandem with the supervised learning objective : i) Representation Learning (Algo. 1, Lines 4-10), and, ii) Knowledge Distillation (Algo. 1, Lines 11-13).

Algorithm 1: CoRaL : Continual Representation Learning for Overcoming Catastrophic Forgetting

Input: Dataset D, Networks: Encoder f_θ, Predictor h_θ, Task Layer g_θ, Memory Buffer M, Scalars: α, β, Learning rate γ

1 **for** x^t, y^t, t *in* D^t **do**
2 $\quad x_1^t, x_2^t \leftarrow aug(x^t)$
3 $\quad \hat{y}^t \leftarrow g_\theta(f_\theta(x_1^t))$
4 $\quad$ **# Representation Learning:**
5 $\quad\quad (x^\tau, \hat{y}^\tau) \leftarrow sample(M)$
6 $\quad\quad x_1^\tau, x_2^\tau \leftarrow aug(x^\tau)$
7 $\quad\quad x^{t,\tau} \leftarrow cat([x_1^t, x_2^t], [x_1^\tau, x_2^\tau])$
8 $\quad\quad z_1, z_2 \leftarrow f_\theta(x^{t,\tau})$
9 $\quad\quad p_1, p_2 \leftarrow h_\theta(z_1), h_\theta(z_2)$
10 $\quad\quad \mathcal{L}_{cos\_sym} \leftarrow \frac{1}{2}\mathcal{L}_{cos}(p_1, sg(z_2)) + \frac{1}{2}\mathcal{L}_{cos}(p_2, sg(z_1))$
11 $\quad$ **# Knowledge Distillation:**
12 $\quad\quad y^\tau \leftarrow g_\theta(f_\theta(x_1^\tau))$
13 $\quad\quad \mathcal{L}_{distill} \leftarrow ||y^\tau - \hat{y}^\tau||_2^2$
14 $\quad \mathcal{L}_{overall} \leftarrow \mathcal{L}_{ce}(\hat{y}^t, y^t) + \alpha \cdot \mathcal{L}_{cos\_sym} + \beta \cdot \mathcal{L}_{distill}$
15 $\quad \theta \leftarrow \theta + \gamma \nabla_\theta \mathcal{L}$
16 $\quad M \leftarrow reservoir(M, (x^t, \hat{y}^t))$
17 **end**

The Representation Learning module is a Siamese network setup comprising an encoder, a projection, and a prediction network. The projection and prediction networks are MLPs, while the encoder consists of a backbone (e.g., ResNet). To aid the Representation Learning module, we propose a memory buffer that replays past input samples using reservoir sampling [8]. The input samples from the buffer undergo augmentations before being fed to the Representation Learning module, which is trained using the Cosine Similarity loss to encourage the encoder to minimize the embedding distance between similar inputs under changing distributions.

The memory buffer also stores the model's past output logits, which is used in the second part of the framework: the Knowledge Distillation module. The storing of past outputs ensures that even when the encoder learns robust representations, the task learning layer can map it to the correct class. In this module, past input samples are fed to the overall network (encoder + task-learning layer), and the output is compared to the past output from the buffer, with the objective being to penalize divergence between the two values. We will first describe the Representation Learning module of our framework, followed by the Knowledge Distillation module, and finally, the modified objective function.

4.1 Representation Learning

4.1.1 Objective Function. Contrastive learning [11, 27] have proven to be an effective technique for learning instance discrimination without labels. The core idea behind these works is the following:

for every input, minimize the distance between the positive sample pairs and maximize the distance between the negative sample pairs. The positive sample pairs are the embeddings of the two augmented versions of the input, while all other embeddings are considered negative. Let (i, j) be the positive pairs. The contrastive learning objective can be defined using the InfoNCE loss:

$$\mathcal{L}_{i,j} = -\log \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(sim(z_i, z_k)/\tau)} \tag{3}$$

Here, z_i, z_j are the embeddings of the positive sample pairs, $1_{k \neq i}$ is an indicator function which is 1 for the $2N - 1$ negative sample pairs, i.e., when $k \neq i$. τ represents the temperature parameter used to scale the gradient. *sim* represents the dot product between the l_2 normalized embeddings.

While this formulation has proven effective in learning instance discrimination in the absence of labels, methods based on this contrastive formulation are sensitive to the choice of data augmentations [25]. This motivates the need to develop techniques that are robust to data augmentations and distribution shifts and is a key component for Continual Representation Learning. Here, due to the non-stationary data distribution, the encoder output for the positive samples is continuously changing, *along with the negative samples*, making the objective function in Eq. 3 challenging to optimize. Furthermore, such contrastive learning methods rely on a large number of negative samples, which require a large batch size, making their adoption intractable for a CL setup.

As our primary objective is to learn effective and robust representations under non-stationary settings, we introduce the modified Cosine Similarity loss [13] for CL, which only relies on the positive samples. The task then reduces to maximizing the similarity between two augmented versions of the same input, say z_1, z_2. As such, our objective function is:

$$\mathcal{L}_{cos}(p_1, z_2) = -\frac{p_1}{||p_1||_2} \cdot \frac{z_2}{||z_2||_2} \tag{4}$$

Here, p_1 is a non-linear transformation of z_1, which is processed through a predictor network.

4.1.2 Siamese Network Setup. We now describe our proposed representation learning module, which is trained using the modified Cosine Similarity loss (Eq. 4.) The input to the module is the augmented image samples. We first apply data augmentation on each input sample, in line with prior works [11, 12, 25, 27]. This augmentation effectively doubles the input sample size. The augmented samples are then passed to the Representation Learning module.

The Representation Learning module is a Siamese Network setup, comprised of one encoder f_θ, one projection network d_θ, and a prediction network h_θ. Inspired from BYOL [25] and SimSiam [13], our Siamese Network setup has one online network and one target network. The target network (f_θ & d_θ) provides the regression targets to train the online network, which is then used to update the gradients of the online network (f_θ, d_θ & h_θ). This is then followed by swapping the roles, i.e., the previously online network is now the target network and has to provide the regression targets, leading to two passes of optimization.

We use the same encoder and projection network for both online and target networks. For a given task t, our architecture takes as input two randomly augmented views from an image x_1^t, x_2^t, which is processed by the encoder network f_θ, followed by the projection network d_θ to get two embeddings z_1, z_2. As CoRaL is a rehearsal-based approach, the image can come from the stream of the current task t or from the input buffer τ. For a given pass, the prediction network h_θ processes one of the embeddings, for example, z_1, to output p_1 and matches it to the other embedding, z_2.

The outputs, p_1 and z_2, are normalized before calculating the Cosine Similarity loss (Eq. 4). Our framework uses symmetric loss, whereby we update the gradient of one network in one pass and then update the gradient of the other network in the next pass. In either pass, only the online network (one with the predictor) is updated ***end-to-end*** using backpropagation. The target network is not updated and is tasked to provide regression targets. The symmetric loss is an extension of Eq. 4 and is formulated below:

$$\mathcal{L}_{cos\_sym} = \frac{1}{2}\mathcal{L}_{cos}(p_1, sg(z_2)) + \frac{1}{2}\mathcal{L}_{cos}(p_2, sg(z_1)) \tag{5}$$

This is performed for all input samples, and the total loss is averaged. Here, *sg* refers to the stopgrad function and is used to prevent the target network from getting updated [13], meaning that the encoder on x_2 receives no gradients from z_2 and the encoder on x_1 receives no gradients from z_1.

Prior works on representation learning have leveraged self- supervised learning, with the frameworks trained in two phases: representation learning and task learning [11, 25, 27]. In the representation learning phase, the network has no access to labels and relies on a *pretext* task to distinguish between the unlabelled classes. In the task learning phase, the network has access to a small number of labels. Recent works on CL have also used a similar approach by using a two-phase training scheme and relaxing the non-i.i.d. assumption by introducing class-balancing strategies when training the classifier. A key difference between our work and prior works [20, 22] is that we do not decouple the representation learning from the task learning and in fact, unify the two objective functions, i.e., the Cross-Entropy loss and the Cosine Similarity loss. This approach has the benefit of learning robust features while also mapping the representations to their respective labels.

4.2 Knowledge Distillation

The Representation Learning module for CoRaL has the explicit objective of improving the robustness of the learned features at the encoder using the Cosine Similarity loss. However, standard backpropagation with the dual objective of Cross-Entropy and Cosine Similarity may not prevent catastrophic forgetting. As such, even if the representations are robust to changing distributions, the weights and output of the task-learning network may be prone to changes. To address this challenge, we introduce a Knowledge Distillation module to our framework.

Although knowledge distillation [28] has been mostly deployed in a teacher-student setting, where the teacher network distills its knowledge to a student network, in this work, we rely on self-distillation in CL settings [43, 49]. To perform self-distillation, we store the final network response along with the corresponding inputs in the memory buffer using reservoir sampling. This retention of past input and network response allows the network to have similar outputs, even if there is a shift in representation.

Table 1: Performance comparison (averaged across 10 runs) of various CL methods on different scenarios (Accuracy in %)

Approach	Method	IL-Task		IL-Class		IL-Domain	
		S-CIFAR10	S-Tiny-ImageNet	S-CIFAR10	S-Tiny-ImageNet	P-MNIST	R-MNIST
Non-CL	JOINT	98.31 ± 0.12	82.04 ± 0.10	92.20 ± 0.15	59.99 ± 0.19	94.33 ± 0.17	95.76 ± 0.04
	SGD	61.02 ± 3.33	18.31 ± 0.68	19.62 ± 0.05	7.92 ± 0.26	40.70 ± 2.33	67.66 ± 8.53
Architectural	PNN [51]	95.13 ± 0.72	67.84 ± 0.29	-	-	-	-
Regularization	oEWC [55]	68.29 ± 3.92	19.20 ± 0.31	19.49 ± 0.12	7.58 ± 0.10	75.79 ± 2.25	77.35 ± 5.77
	SI [64]	68.05 ± 5.91	36.32 ± 0.13	19.48 ± 0.17	6.58 ± 0.31	65.86 ± 1.57	71.91 ± 5.83
	LwF [43]	63.29 ± 2.35	15.85 ± 0.58	19.62 ± 0.05	8.46 ± 0.22	-	-
Rehearsal	ER [50]	91.19 ± 0.94	38.17 ± 2.00	44.79 ± 1.86	8.49 ± 0.6	72.37 ± 0.87	85.01 ± 1.90
	GEM [44]	90.44 ± 0.94	-	25.54 ± 0.76	-	66.93 ± 1.25	80.80 ± 1.15
	A-GEM [10]	83.88 ± 1.49	22.77 ± 0.03	20.04 ± 0.34	8.07 ± 0.08	66.42 ± 4.00	81.91 ± 0.76
	iCARL [49]	88.99 ± 2.13	28.19 ± 1.47	49.02 ± 3.20	7.53 ± 0.79	-	-
	FDR [6]	91.01 ± 0.68	40.36 ± 0.68	30.91 ± 2.74	8.70 ± 0.19	74.77 ± 0.83	85.22 ± 3.35
	GSS [3]	88.80 ± 2.89	-	39.07 ± 5.59	-	63.72 ± 0.70	79.50 ± 0.41
	HAL [9]	82.51 ± 3.20	-	32.36 ± 2.70	-	74.15 ± 1.65	84.02 ± 0.98
	DER [8]	91.40 ± 0.92	40.22 ± 0.67	61.93 ± 1.79	11.87 ± 0.78	81.74 ± 1.07	90.04 ± 2.61
	DER++ [8]	91.92 ± 0.60	40.87 ± 1.16	64.88 ± 1.17	10.96 ± 1.17	83.58 ± 0.59	90.43 ± 1.87
	CoRaL (Ours)	**92.01 ± 0.32**	**41.37 ± 0.91**	**65.24 ± 1.09**	**14.06 ± 0.57**	**84.60 ± 0.48**	**91.79 ± 0.92**

For every data x^τ sampled from the memory buffer, we forward propagate the sample through the current network to obtain the final output before computing the softmax probability, y^τ. This output is then compared with the network's past response, $\hat{y}^\tau$, which is obtained from the memory buffer. Unlike prior distillation-based approaches [8] which have shown to benefit from storing both the network's output logits and the class label, we only store the network's output logits, thus simplifying the objective function. As we are computing the loss on pre-softmax outputs, we use the Mean Square Error between the logits of the current model and the past model. The overall operations in this module can be formulated as:

$$\mathcal{L}_{distill} = ||y^\tau - \hat{y}^\tau||_2^2 \tag{6}$$

4.3 Overall Objective for End-to-End Learning

In CoRaL, we introduce a new approach to combine two different modules for end-to-end training. Moreover, these modules are used to overcome catastrophic forgetting on past task samples. Learning a mapping between the inputs and the labels for the current task is done using the *Cross-Entropy* loss function for each mini-batch that is sampled from the current distribution.

Overall, CoRaL is comprised of three different loss functions that is trained end-to-end: the standard Cross-Entropy loss for supervised learning, the modified Cosine Similarity loss from Eq. 5, and the Distillation loss from Eq. 6 for the distillation learning. For the initial task, the framework uses only the Cross-Entropy loss. For every incremental task/class/domain that follows, the model is trained using the following objective function:

$$\mathcal{L}_{overall} = L_{ce} + \alpha \cdot \mathcal{L}_{cos\_sym} + \beta \cdot \mathcal{L}_{distill} \tag{7}$$

Here, α, β are hyper-parameters for the different losses.

5 EXPERIMENTAL SETUP

5.1 Datasets

We evaluated our approach by comparing its performance to several state-of-the-art CL methods on four widely benchmarked datasets: Rotated MNIST (R-MNIST) [44], Permuted MNIST (P-MNIST) [37] which are variants of the MNIST dataset, Split CIFAR-10 (S-CIFAR-10) [64] which is a variant of the CIFAR10 [39] and Split TinyImageNet (S-Tiny-ImageNet) [14]. Please check the supplementary materials for more details on the datasets.

5.2 Continual Learning Scenarios

We consider three challenging CL scenarios for conducting evaluation inline with prior works [29, 64]. For all scenarios, the original dataset is split into separate tasks. For S-CIFAR-10, the original dataset is split into five 2-way classification tasks, whereas for S-Tiny-ImageNet, the original dataset is split into ten 20-category classification tasks. For P-MNIST and R-MNIST, the image pixels in the original dataset are permuted or rotated for 20 rounds, resulting in a shift in input while the classes remain unchanged.

For *incremental task* (IL-Task), models have access to the task label, and as a result, they are trained with task-specific components. For *incremental class* (IL-Class), models need to perform both classification of new samples as they arrive and infer the change in task. Lastly, for *incremental domain* (IL-Domain), models do not have access to task labels and need to only perform the classification of the input images, which may undergo perturbations.

5.3 Architectures

We use different encoders depending on the complexity of the dataset. On R-MNIST and P-MNIST, we use a fully connected neural network with two hidden layers of 100 ReLU units, following prior works [44]. On the CIFAR-10 and TinyImageNet, we use ResNet18. For implementation details, please look at the supplementary.

5.4 Evaluation Protocol

To ensure fair evaluation, we used a similar learning schedule for all evaluated methods and conducted a hyper-parameter search to ensure the best average accuracy. We compared CoRaL with state-of-the-art approaches that use a similar end-to-end training scheme. As such, we did not evaluate against techniques that require two-phases of training or relax the non-i.i.d assumption of CL by using

Table 2: Backward Transfer (BWT) comparison (averaged across 10 runs) on Incremental Domain (in %).

Approach	Method	P-MNIST	R-MNIST
Non-CL	SGD	-57.65 ± 4.32	-20.34 ± 2.50
Architectural	PNN [51]	-	-
Regularization	oEWC [55]	-36.69 ± 2.34	-24.59 ± 5.37
	SI [64]	-27.91 ± 0.31	-22.91 ± 0.26
	LwF [43]	-	-
Rehearsal	ER [50]	-22.54 ± 0.95	-8.24 ± 1.56
	GEM [44]	-29.38 ± 2.56	-11.51 ± 4.75
	A-GEM [10]	-31.69 ± 3.92	-19.32 ± 1.17
	FDR [6]	-20.62 ± 0.65	-13.31 ± 2.60
	GSS [3]	-47.85 ± 1.82	-20.19 ± 6.45
	HAL [9]	-15.24 ± 1.33	-11.71 ± 0.26
	DER [8]	-13.79 ± 0.80	-5.99 ± 0.46
	DER++ [8]	-11.47 ± 0.33	-5.27 ± 0.26
	CoRaL (Ours)	**-9.92 ± 0.51**	**-4.65 ± 0.86**

class-balancing strategies[20, 48] . For the S-MNIST and R-MNIST datasets, we trained all methods for one epoch per task, using a mini-batch size of 128 following prior work [8]. For the S-CIFAR-10 and S-Tiny-ImageNet datasets, we used a mini-batch size of 32 and trained for 50 epochs per task, following prior works [8, 64]. We used a memory buffer of 200 samples using reservoir sampling. For all scenarios, the evaluation metric is the test-set accuracy after being trained on all the tasks (Acc.), averaged over ten independent runs.

6 RESULTS AND DISCUSSION

6.1 Incremental Task

Results: We present the average accuracy over ten independent runs of all frameworks on IL-Task settings for the Split-CIFAR-10 (S-CIFAR-10) and Split-TinyImageNet (S-Tiny-ImageNet) datasets in Tab. 1. The results suggest that CoRaL outperformed all other methods on both the evaluated datasets. CoRaL achieved the highest average accuracy of 92.01% and 41.37% on S-CIFAR-10 and S-Tiny-ImageNet, respectively, while having low standard deviation.

Discussion: The results in Tab. 1 suggest the efficacy of the Representation Learning module in learning robust representation. The Representation Learning module increases the similarity between the positive samples, whereas the Knowledge Distillation module replays past samples and allows the network to optimize over them simultaneously. CoRaL achieved a performance improvement of 0.09% and 0.50% on S-CIFAR-10 and S-Tiny-ImageNet, while having a relatively low standard deviation, suggesting consistency in the results and the stability of the objective function in Eq. 7. Although we report methods that have an architectural expansion, such as PNN [51], it is not a fair comparison as PNN progressively adds a new learning network for each task, incurring significant memory overhead. In contrast, our work does not require a new network for each task and maintains the same buffer size as other approaches.

We also observed in Tab. 1 the effectiveness of rehearsal-based approaches (FDR,ER, DER++, CoRaL) compared to regularization-based approaches (oEWC, SI, LwF) over both the datasets. This is due to the network having access to past data samples and optimizing over them as well as the current data samples, providing a more effective way of recalling past representations. Moreover, regularization-based approaches add a penalty to parameter updates, which constrains the network from learning new tasks. As a result, the number of unregularized parameters decreases with each task, which leads to relatively low average accuracy.

6.2 Incremental Class

Results: We present the average accuracy over ten runs of all methods on IL-Class settings for the S-CIFAR10 and S-Tiny-ImageNet datasets in Tab. 1. The results suggest that CoRaL outperformed all other methods, further highlighting CoRaL's ability to mitigate catastrophic forgetting.

Discussion: The results underline the generalizability of CoRaL to different scenarios and posit a strong case for Representation Learning frameworks in CL. CoRaL's improved representation learning allows the encoder to learn a more robust representation, which along with the distillation loss, allows it to attain the best performance. As observed in Tab. 1, CoRaL significantly outperformed all other approaches by 2.19% in terms of average accuracy on the S-Tiny-ImageNet dataset. This provides empirical evidence of the benefit of the Siamese Network setup, which leads to a more robust Representation Learning under non-stationary distributions. Furthermore, the low standard deviation leads to more consistent results.

We observed that IL-Class presents a more significant challenge for all CL frameworks with a performance drop compared to IL-Task. This is because there are no task boundaries for IL-Class, leading to only one task-learning layer. This means that models need to *infer* the current task in addition to classifying the inputs, making it more challenging to recall knowledge over past *inferred* tasks.

6.3 Incremental Domain

Results: We present the average accuracy over ten runs of all models on IL-Domain settings for the P-MNIST and R-MNIST datasets in Tab. 1. Consistent with previous CL-scenarios, the results suggest that CoRaL achieved the highest average accuracy on both the datasets, further highlighting its superiority for addressing catastrophic forgetting in IL-Domain settings. On average, CoRaL outperformed all other approaches by 1.02% on the P-MNIST dataset and 1.36% on the R-MNIST dataset.

Discussion: The results reinforce the benefits of the Representation Learning module in CoRaL. As IL-Domain introduces input perturbation, the addition of Representation Learning is particularly effective as it explicitly directs the model to reduce the distance between samples that have undergone different perturbations but belong to the same class. This is not available in other evaluated approaches, which try to distinguish between these perturbations using the Cross-Entropy or other distillation losses, which are not explicitly targeted toward learning robust representations.

6.4 Backward Transfer

Results: Lastly, we present the average Backward Transfer (BWT) of all models, which is calculated in line with prior works [8, 44]. BWT is expected to increase with new tasks as the network no longer has access to all the data samples of the past tasks and is a good estimator for *catastrophic forgetting*. As such, we chose the CL scenario with the highest number of tasks: IL-Domain. We present the results for all the methods in Tab. 2 for the P-MNIST and R-MNIST datasets. For BWT, a negative value indicates forgetting, and as such, a lower negative value is desirable. As can be observed, CoRaL attained the lowest BWT, with an average BWT of -9.92% and -4.65% on P-MNIST and R-MNIST, respectively.

Table 3: Impact of Representation Learning techniques.

Approach	IL-Task		IL-Class	
	S-CIFAR-10	S-Tiny-ImageNet	S-CIFAR-10	S-Tiny-ImageNet
CoRaL with CrL (MoCo)	90.52 ± 0.51	35.88 ± 1.61	61.20 ± 1.02	11.16 ± 1.07
CoRaL with CSL (SimSiam)	**92.01 ± 0.32**	**41.37 ± 0.91**	**65.24 ± 1.09**	**14.06 ± 0.57**

Discussion: The results in Tab. 2 highlight that CoRaL attained the lowest BWT over all approaches on both datasets. For the P-MNIST dataset, CoRaL outperformed all baselines by 1.55%, and for R-MNIST, CoRaL attained the lowest BWT, outperforming all approaches by 0.62% on average. This suggests that the augmentations introduced for the Representation Learning module allow CoRaL to learn more robust features resulting in less forgetting.

7 ABLATION STUDY

7.1 Analyzing Different Representation Learning Approaches

Continual Learning requires frameworks to strike the right blend of stability and plasticity when learning on continuous data streams. Such frameworks strive to be *stable* to changing data distributions, retaining information on past tasks while exuding the requisite plasticity to learn new tasks efficiently. In this work, we presented a general framework for investigating the effectiveness of Representation Learning frameworks under non-stationary distributions. To assess the applicability of current Representation Learning frameworks, we compare two popular approaches: MoCo [27] and SimSiam[13], while fixing the parameters of the Knowledge Distillation module.
Results: Tab. 3 reports the average accuracy after ten runs on the S-CIFAR-10 and S-Tiny-ImageNet datasets. We conducted our analysis for both IL-Task and IL-Class scenarios for extensive evaluation. We compared two conceptually different approaches, *SimSiam* [13] and *MoCo* [27]. SimSiam is a Siamese Network setup trained using the negative symmetric cosine similarity. On the other hand, MoCo is also a Siamese setup, where one of the encoders is a momentum encoder, and the other is a standard encoder. MoCo is trained using the Contrastive loss.
Discussion: The results in Tab. 3 suggest that SimSiam, which is the approach used in this paper, outperformed MoCo for all the datasets and scenarios. The improvement is especially significant for IL-Class, with 4.04% and 2.90% gain for S-CIFAR-10 and S-Tiny-ImageNet, respectively. We posit that this is due to the Cosine Similarity objective, which does not rely on negative samples but tries to maximize the similarity between two augmentations. On the other hand, MoCo uses a momentum encoder along with a feature queue to maintain a consistent queue of negative samples, which is optimized using the InfoNCE loss (Eq. 3). As is the case with Continual Learning, the distribution for the negative samples keeps changing, making it challenging for the network to learn stable representations. The results justify the use of the Cosine Similarity loss for Representation Learning in CL, which provides CoRaL with the ideal blend of stability and plasticity when learning new tasks.

7.2 Impact of CoRaL's Learning Modules

Results: We extensively experimented across different scenarios and datasets to assess the importance of the two primary learning modules of CoRaL. Tab. 4 presents the accuracy while ablating a specific module, given the same encoder network and learning protocols.

Table 4: Ablation results (top-1) over different learning modules.

Method	IL-Domain		IL-Task	IL-Class
	P-MNIST	R-MNIST	S-CIFAR-10	S-CIFAR-10
CoRaLw/o R.L.	83.45	90.19	91.23	59.43
CoRaLw/o K.D.	41.99	69.45	72.92	19.67
CoRaL	**85.60**	**92.89**	**92.49**	**66.97**

Discussion: First, we ablate the Representation Learning module, i.e., we no longer have a Siamese Network setup. The Cosine Similarity loss is removed from the objective function, which is now comprised of only the Cross-Entropy and the distillation learning loss. The results in Tab. 4 suggest that the absence of the Representation Learning module results in a drop in accuracy across all scenarios and datasets. The drop is most significant in IL-Class scenarios (7.54% for S-CIFAR-10), which is the most challenging of all CL scenarios, asserting the importance of learning robust representations that are transferable.

We next ablate the Knowledge Distillation module, removing the distillation loss from the objective function. The results suggest that removing the Knowledge Distillation module has a significant impact on overall performance. There is a significant drop in accuracy for all scenarios and datasets, with the framework suffering most in IL-Class. This reinforces the importance of using a memory buffer which allows the network to retain its knowledge over past tasks. Our results also highlight that combining the Representation Learning and Knowledge Distillation modules provide the right balance between stability and plasticity, with the combination attaining the best performance.

8 ANALYSIS OF THE STABILITY-PLASTICITY

We conducted extensive experiments to evaluate the effect of stability and plasticity on the average accuracy. We achieved this by varying the values of α and β of the objective function in Eq. 7.

8.1 Effect of Varying the Plasticity on the Accuracy

Results: We varied the weight of the Cosine Similarity Loss, α, keeping the weight for the Knowledge Distillation loss, β fixed at 0.1. Varying the α provides us the flexibility of increasing or decreasing the plasticity of CoRaL and allows us to assess the subsequent impact on the forgetting. Tab. 5 presents the average accuracy after all five tasks on the S-CIFAR-10 for the IL-Task and IL-Class scenarios. In addition, we also tracked the average accuracy after learning each new task, as shown in Fig. 2 for different values of α.
Discussion: The results in Tab. 5 suggest that increasing the plasticity of the framework initially allows CoRaL to learn robust representations with the highest accuracy for IL-Task at $\alpha = 0.4$. However, with a further increase in the plasticity, the accuracy drops, suggesting the need to constrain the plasticity of the framework in order to improve the stability of the learned representations. The trend is also similar for IL-Class, with the accuracy increasing initially, with the best value at $\alpha = 0.4$.

When we track the average accuracy after each task in Fig. 2, we see that increasing the plasticity results in the network attaining

Table 5: Effect of varying the plasticity parameter (α) on the average accuracy (after 5 independent runs) for S-CIFAR-10.

α	β	IL-Task	IL-Class
0.1	0.1	90.33 ± 1.22	63.80 ± 1.20
0.2	0.1	90.69 ± 0.91	64.19 ± 0.14
0.3	0.1	91.72 ± 0.29	65.31 ± 1.24
0.4	0.1	**92.10 ± 0.12**	**66.05 ± 0.71**
0.5	0.1	91.25 ± 0.79	63.01 ± 1.10
1.0	0.1	90.63 ± 0.89	62.69 ± 0.77

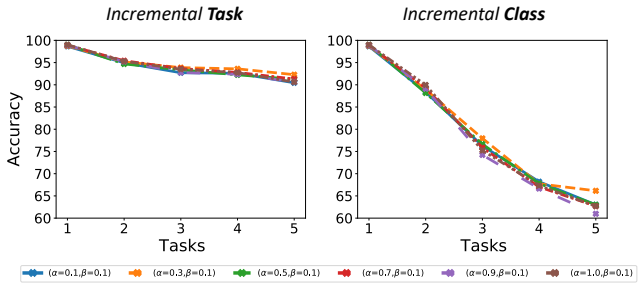

Figure 2: Plasticity Analysis on the S-CIFAR-10 dataset. We varied the weight (α) for the Representation Learning module

higher average accuracy for the initial tasks, especially for Task 1 and 2, where higher values of ($\alpha = 0.7 \sim 1.0$) led to higher accuracy for both IL-Task and IL-Class. However, as Continual Learning requires the network to retain past knowledge, a high plasticity coefficient may result in higher forgetting. As seen in Fig. 2, the average accuracy for later tasks decreases at a faster rate, for high values of α. As such, there needs to be a trade-off between high plasticity, which may provide better accuracy initially, and moderate plasticity, which may provide better accuracy at later stages.

8.2 Effect of Varying the Stability on the Accuracy

Results: We varied the weight of the Knowledge Distillation Loss, β, keeping the weight for the Representation Learning loss, α fixed at 0.1. This provided us with a mechanism to tune the stability of our framework, with a higher value of β resulting in stronger optimization constraints when updating the network parameters. Tab. 6 presents the final accuracy after all five tasks on the S-CIFAR-10 for the IL-Task and IL-Class scenarios. In addition, we also tracked the average accuracy after learning each new task, as depicted in Fig. 3 for different values of β.

Discussion: The results in Tab. 6 suggest that increasing the stability parameter β initially allows CoRaL to put optimization constraints when learning new tasks and results in improved knowledge retention over past tasks. The Distillation Loss $\mathcal{L}_{distill}$ acts as a regularizer during the parameter update while replaying the past samples also relaxes the non-i.i.d assumption. The highest accuracy for IL-Task was at $\beta = 0.3$, whereas the highest accuracy for IL-Class was at $\beta = 0.2$. However, with a further increase in the stability ($\beta > 0.3$), the accuracy drops for both IL-Task and IL-Class, suggesting over-regularization.

When we track the average accuracy after each task in Fig. 3, we see that increasing the stability ($\beta = 0.1 \sim 0.3$) leads to the network attaining higher average accuracy. However, a further increase sin β results in over-constraining the network and leads to lower accuracy for all the tasks, as observed for $\beta > 0.3$. The lowest average

Table 6: Effect of varying the stability parameter (β) on the average accuracy (after 5 independent runs) for S-CIFAR-10.

α	β	IL-Task	IL-Class
0.1	0.1	90.33 ± 1.22	63.80 ± 1.20
0.1	0.2	90.72 ± 0.72	**65.18** ± 1.09
0.1	0.3	**90.94** ± 0.60	64.77 ± 0.72
0.1	0.4	90.74 ± 1.18	62.11 ± 2.22
0.1	0.5	89.23 ± 1.44	61.88 ± 0.39
0.1	1.0	86.77 ± 0.82	51.18 ± 5.14

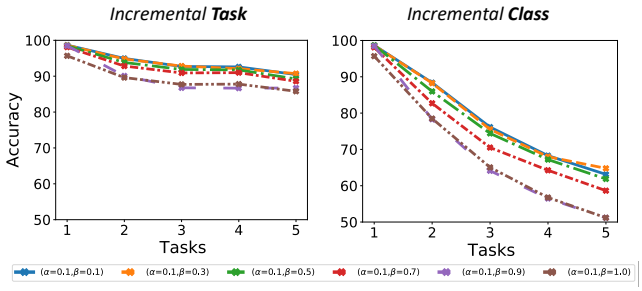

Figure 3: Stability Analysis on the S-CIFAR-10 dataset. We varied the weight (β) for the Knowledge Distillation module

accuracy after each task and after all five tasks was for $\beta = 1.0$. Interestingly, we also observed that the drop in accuracy after each new task is also lowest for $\beta = 1.0$. The combination of the network attaining low accuracy and low forgetting implies over-regularization, whereby the network is too stable to learn efficiently.

8.3 Discussion on the Stability-Plasticity Trade-off

Our experimental results in (Tabs. 5, 6 and Figs. 2, 3) underline the challenges of finding the right blend of stability and plasticity to mitigate catastrophic forgetting. An increase in plasticity, by increasing the weight of the Representation Learning loss, $\mathcal{L}_{cos\_sym}$ leads to the network learning transferable features, which results in higher average accuracy over the next task. However, a further increase in plasticity may lead to drops in accuracy. Similarly, an increase in the stability, by increasing the weight of the Knowledge Distillation loss, $\mathcal{L}_{distill}$, can improve the knowledge retention of the network by acting as a regularizer, up to a certain value. Further increase in the stability parameter might constrain the network from updating its weights, resulting in lower average accuracy.

9 CONCLUSION

In this work, we introduced CoRaL, a novel Continual Learning framework for addressing catastrophic forgetting. Our framework provides the right blend of stability in CL scenarios through the Knowledge Distillation module and plasticity via the Representation Learning module, thus providing a promising approach for intelligent agents to learn continually. CoRaL is trained end-to-end with a novel objective function that comprises the modified Cosine Similarity loss and the Distillation loss on top of the Cross-Entropy loss. Our results across three scenarios and four datasets suggest the efficacy of CoRaL, with our proposed approach outperforming all other techniques on all evaluated benchmarks. The ablation studies further validates the relevance of the Cosine Similarity loss for Continual Representation Learning and CoRaL's two proposed modules.

REFERENCES

[1] Elahe Aghapour and Nora Ayanian. 2021. Double meta-learning for data efficient policy optimization in non-stationary environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 9935–9942.

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 139–154.

[3] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. 2019. Gradient based sample selection for online continual learning. *NeurIPS* (2019), 11816–11825.

[4] Ali Ayub and Alan R Wagner. 2020. Tell me what this is: few-shot incremental object learning by a robot. In *IEEE/RSJ IROS*.

[5] Ali Ayub and Alan R Wagner. 2021. Continual learning of visual concepts for robots through limited supervision. In *Companion of the 2021 ACM/IEEE HRI*.

[6] Ari Benjamin, David Rolnick, and Konrad Kording. 2018. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*.

[7] Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence* 7, 04 (1993), 669–688.

[8] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *NeurIPS*.

[9] Arslan Chaudhry, Albert Gordo, Puneet Dokania, Philip Torr, and David Lopez-Paz. 2021. Using Hindsight to Anchor Past Knowledge in Continual Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6993–7001.

[10] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2018. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations*.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).

[13] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15750–15758.

[14] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. 2017. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819* (2017).

[15] Nikhil Churamani, Sinan Kalkan, and Hatice Gunes. 2020. Continual learning for affective robotics: Why, what and how?. In *29th IEEE RO-MAN*.

[16] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. 2021. UniMoCo: Unsupervised, Semi-Supervised and Full-Supervised Visual Representation Learning. *arXiv preprint arXiv:2103.10773* (2021).

[17] Angel Daruna, Mehul Gupta, Mohan Sridharan, and Sonia Chernova. 2021. Continual learning of knowledge graph embeddings. *IEEE RA-L* (2021).

[18] Celso M de Melo, Stacy Marsella, and Jonathan Gratch. 2017. Increasing Fairness by Delegating Decisions to Autonomous Agents.. In *AAMAS*. 419–425.

[19] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[20] Cha et al. 2021. Co^{2}L: Contrastive Continual Learning. In *ICCV*.

[21] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. 2020. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3762–3773.

[22] Jhair Gallardo, Tyler L Hayes, Christopher Kanan, and Cornell Tech. 2021. Self-Supervised Training Enhances Online Continual Learning. (2021).

[23] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[24] Haley N Green, Md Mofijul Islam, Shahira Ali, and Tariq Iqbal. 2022. Who's laughing nao? examining perceptions of failure in a humorous robot partner. In *ACM/IEEE HRI*.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 21271–21284. https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf

[26] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. 2017. Neuroscience-inspired artificial intelligence. *Neuron* 95, 2 (2017), 245–258.

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[28] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*. http://arxiv.org/abs/1503.02531

[29] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. 2018. Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines. In *NeurIPS Continual learning Workshop*. https://arxiv.org/abs/1810.12488

[30] T. Iqbal, S. Li, C. Fourie, B. Hayes, and J. A. Shah. 2019. Fast Online Segmentation of Activities from Partial Trajectories. In *ICRA*.

[31] T. Iqbal, S. Rack, and L. D. Riek. 2016. Movement Coordination in Human-Robot Teams: A Dynamical Systems Approach. *IEEE T-RO* (2016).

[32] Tariq Iqbal and Laurel D. Riek. 2017. Coordination Dynamics in Multi-human Multi-robot Teams. *IEEE RA-L* (2017).

[33] Md Mofijul Islam and Tariq Iqbal. 2020. HAMLET: A Hierarchical Multimodal Attention-based Human Activity Recognition Algorithm. In *IROS*.

[34] Md Mofijul Islam and Tariq Iqbal. 2021. Multi-GAT: A Graphical Attention-based Hierarchical Multimodal Representation Learning Approach for Human Activity Recognition. In *IEEE RA-L*.

[35] Md Mofijul Islam and Tariq Iqbal. 2022. MuMu: Cooperative Multitask Learning-based Guided Multimodal Fusion. In *AAAI*.

[36] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. *Advances in Neural Information Processing Systems* 33 (2020).

[37] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[38] Luzia Knoedler, Chadi Salmi, Hai Zhu, Bruno Brito, and Javier Alonso-Mora. 2022. Improving Pedestrian Prediction Models with Self-Supervised Continual Learning. *IEEE RA-L* (2022).

[39] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[40] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[41] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. 2020. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion* 58 (2020), 52–68.

[42] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. 2019. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*. PMLR, 3925–3934.

[43] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.

[44] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30 (2017), 6467–6476.

[45] Hang Ma, Wolfgang Hönig, TK Satish Kumar, Nora Ayanian, and Sven Koenig. 2019. Lifelong path planning with kinematic constraints for multi-agent pickup and delivery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7651–7658.

[46] Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Vol. 24. Elsevier, 109–165.

[47] Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. 2020. Latent replay for real-time continual learning. In *IEEE/RSJ IROS*.

[48] Quang Pham, Chenghao Liu, and Steven Hoi. 2021. DualNet: Continual Learning, Fast and Slow. *NeurIPS* (2021).

[49] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2001–2010.

[50] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2019. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *In International Conference on Learning Representations (ICLR)*.

[51] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671* (2016).

[52] Gobinda Saha, Isha Garg, Aayush Ankit, and Kaushik Roy. 2021. SPACE: Structured Compression and Sharing of Representational Space for Continual Learning. *IEEE Access* 9 (2021), 150480–150494. https://doi.org/10.1109/ACCESS.2021.3126027

[53] Syed Shakib Sarwar, Aayush Ankit, and Kaushik Roy. 2019. Incremental learning in deep convolutional neural networks using partial network sharing. *IEEE Access* 8 (2019), 4615–4628.

[54] Jeffrey C Schlimmer and Douglas Fisher. 1986. A case study of incremental concept induction. In *AAAI*, Vol. 86. 496–501.
[55] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*. PMLR, 4528–4537.
[56] Ghada Sokar, Decebal Constantin Mocanu, and Mykola Pechenizkiy. 2021. Self-Attention Meta-Learner for Continual Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 1658–1660.
[57] Richard S Sutton, Steven D Whitehead, et al. 2014. Online learning with random representations. In *Proceedings of the Tenth International Conference on Machine Learning*. 314–321.
[58] Sebastian Thrun. 1998. Lifelong learning algorithms. In *Learning to learn*. Springer, 181–209.
[59] Mohammad Samin Yasar and Tariq Iqbal. 2021. Improving human motion prediction through continual learning. *arXiv preprint arXiv:2107.00544* (2021).
[60] Mohammad Samin Yasar and Tariq Iqbal. 2021. A Scalable Approach to Predict Multi-Agent Motion for Human-Robot Collaboration. In *IEEE RA-L*.
[61] Mohammad Samin Yasar and Tariq Iqbal. 2022. Robots That Can Anticipate and Learn in Human-Robot Teams. In *ACM/IEEE HRI*.
[62] Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. 2019. Scalable and Order-robust Continual Learning with Additive Parameter Decomposition. In *International Conference on Learning Representations*.
[63] Jaehong Yoon, Eunho Yang, Jungtae Lee, and Sung Ju Hwang. 2018. Lifelong Learning with Dynamically Expandable Networks. In *Sixth International Conference on Learning Representations*. ICLR.
[64] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*. PMLR, 3987–3995.