Pattern Recognition Letters
journal homepage: www.elsevier.com

Continual Semi-Supervised Learning through Contrastive Interpolation Consistency

Matteo Boschini[a], Pietro Buzzega[a], Lorenzo Bonicelli[a,**], Angelo Porrello[a], Simone Calderara[a]

[a] *University of Modena and Reggio Emilia, Via Vivarelli 10, Modena, Italy*

Article history:

Continual learning, deep learning, semi-supervised learning, weak supervision, catastrophic forgetting

ABSTRACT

Continual Learning (CL) investigates how to train Deep Networks on a stream of tasks without incurring *forgetting*. CL settings proposed in literature assume that every incoming example is paired with ground-truth annotations. However, this clashes with many real-world applications: gathering labeled data, which is in itself tedious and expensive, becomes infeasible when data flow as a stream. This work explores *Continual Semi-Supervised Learning* (CSSL): here, only a small fraction of labeled input examples are shown to the learner. We assess how current CL methods (*e.g.*: EWC, LwF, iCaRL, ER, GDumb, DER) perform in this novel and challenging scenario, where overfitting entangles forgetting. Subsequently, we design a novel CSSL method that exploits *metric learning* and *consistency regularization* to leverage unlabeled examples while learning. We show that our proposal exhibits higher resilience to diminishing supervision and, even more surprisingly, relying only on 25% supervision suffices to outperform SOTA methods trained under full supervision.

1. Introduction

Perceptual information flows as a continuous stream, in which a certain data distribution may occur once and not recur for a long time. Unfortunately, this violates the i.i.d. assumption at the foundation of most Deep Learning algorithms and leads to the catastrophic forgetting [1] problem, where the acquired knowledge is rapidly overwritten by the new one. In practical scenarios, we would prefer a system that learns incrementally from the raw and non-i.i.d. stream of data, possibly ready to provide answers at any moment. The design of such lifelong-learning algorithms is the aim of Continual Learning (CL) [2].

Works in this field typically test the proposed methods on a series of image-classification tasks presented sequentially. The latter are built on top of image classification datasets (*e.g.*: MNIST, CIFAR, etc.) by allowing the learner to see just a subset of classes at once. While these experimental protocols validly highlight the effects of forgetting, they assume that all incoming data are labeled.

In some scenarios, this condition does not represent an issue and can be easily met. This may be the case when ground-truth annotations can be directly and automatically collected (*e.g.*: a robot that explores the environment and learns to avoid collisions by receiving direct feedback from it [3]). However, when the labeling stage involves human intervention (as holds in a number of computer vision tasks such as classification, object detection [4], etc.), relying only on full supervision clashes with the pursuit of lifelong learning. Indeed, the adaptability of the learner to incoming tasks would be bottlenecked by the speed of the human annotator: updating the model continually would lose its appeal w.r.t. the trivial solution of re-training from scratch. Therefore, we advocate taking into account the rate at which annotations are available to the learner.

To address this point, the adjustment of the prediction model can be simply limited to the fraction of examples that can be labeled in real-time. Our experiments show

**Corresponding author
e-mail: matteo.boschini@unimore.it (Matteo Boschini), pietro.buzzega@unimore.it (Pietro Buzzega), lorenzo.bonicelli@unimore.it (Lorenzo Bonicelli), angelo.porrello@unimore.it (Angelo Porrello), simone.calderara@unimore.it (Simone Calderara)

that this results in an expected degradation in terms of performance. Fortunately, the efforts recently made in *semi-supervised learning* [5, 6] come to the rescue: by revising these techniques to an incremental scenario, we can still benefit from the remaining part of the data represented by unlabeled observations. We argue that this is true to the lifelong nature of the application and also allows for exploiting the abundant source of information given by unlabeled data. To sum up, our work incorporates the features described above in a new setting called **Continual Semi-Supervised Learning (CSSL)**: a scenario where just **one out of k** examples is presented with its ground-truth label. At training time, this corresponds to providing a ground-truth label for any given example with uniform probability $1/k$ (as shown in Fig. 1 for $k = 2$).

Taking one more step, our proposal aims at filling the gap induced by partial annotations: **Contrastive Continual Interpolation Consistency** (CCIC), which imposes consistency among augmented and interpolated examples while exploiting secondhand information peculiar to the Class-Incremental setting. Doing so, we grant performance that matches and even surpasses that of the fully-supervised setting. We finally summarize our contributions:

- We propose CSSL: a scenario in which the learner must learn continually by exploiting both supervised and unsupervised data at the same time;
- We empirically review the performance of SOTA CL models at varying label-per-example rates, highlighting the subtle differences between CL and CSSL;
- Exploiting semi-supervised techniques, we introduce a novel CSSL method that successfully addresses the new setting and learns with limited labels;
- Surprisingly, our evaluations show that full supervision does not necessarily upper-bound partial supervision in CL: 25% labels can be enough to outperform SOTA methods using all ground truth.

2. Related Work

2.1. Continual Learning Protocols

Continual Learning is an umbrella term encompassing several slightly yet meaningfully different experimental settings [7, 8]. *Van de Ven et al.* produced a taxonomy [8] describing the following three well-known scenarios. **Task-Incremental Learning (Task-IL)** organizes the dataset in tasks comprising of disjoint sets of classes. The model must only learn (and remember) how to correctly classify examples within their original tasks. **Domain-Incremental Learning (Domain-IL)** presents all classes since the first task: distinct tasks are obtained by processing the examples with distinct transformations (*e.g.*: pixel permutations or image rotations) which change the input distribution. **Class-Incremental Learning (Class-IL)** operates on the same assumptions as Task-IL, but requires the learner to classify an example from any of the previously seen classes with no hints about its original task. Unlike Task-IL, this means that the model must learn the joint distribution from partial observations, making this the hardest scenario [8]. For such a reason, we focus on limited labels within the Class-IL formulation.

Towards realistic setups. Several recent works point out that these classic settings lack realism [9] and consequently define new scenarios by imposing restrictions on what models are allowed to do while learning. **Online Continual Learning** forbids multiple epochs on the training data on the grounds that real-world CL systems would never see the same input twice [10, 11, 12]. **Task-Free Learning** does not provide task identities either at inference or at training time [9]. This is in contrast with the classic settings that signal task boundaries to the learner while training, thus allowing it to prepare for the beginning of a new task.

This work also aims at providing a more realistic setup: instead of focusing on model limitations, we acknowledge that requiring fully labeled data can hinder the extension of CL algorithms to real-time and in-the-wild scenarios.

Continual Learning with Unsupervised Data. Some attempts have been recently made at improving CL methods by exploiting unlabeled data. *Zhang et al.* proposed the **Deep Model Consolidation** framework [13]; in it, a new model is first specialized on each new encountered task, then a unified learner is produced by distilling knowledge from both the new specialist and the previous incremental model. Alternatively, *Lechat et al.* introduced **Semi-Supervised Incremental Learning** [14], which alternates unsupervised feature learning on both input and auxiliary data with supervised classification.

We remark that both these settings are significantly different from our proposed **CSSL** as we do not separate the supervised and unsupervised training phases. On the contrary, we intertwine both kinds of data in all drawn batches in varying proportions and require that the model learns from both at the same time. Additionally, we do not exploit auxiliary unsupervised external data to supplement the training set; instead, we reduce the original supervised data to a fraction, thus modeling supervision becoming available on the input stream at a much slower rate.

2.2. Continual Learning Methods

Continual Learning methods have been chiefly categorized in three families [7, 2].

Architectural methods employ tailored architectures in which the number of parameters dynamically increases [15, 16] or a part of them is devoted to a distinct task [17]. While being usually very effective, they depend on the availability of task labels at prediction time to prepare the model for inference, which limits them to Task-IL.

Regularization methods condition the evolution of the model to prevent it from forgetting previous tasks. This is attained either by identifying important weights for each task and preventing them from changing in later ones [18, 19] or by distilling the knowledge from previous model snapshots to preserve the past responses [20, 21].

Rehearsal methods maintain a fixed-size working memory of previously encountered exemplars and recall them to prevent forgetting [22]. This simple solution has been

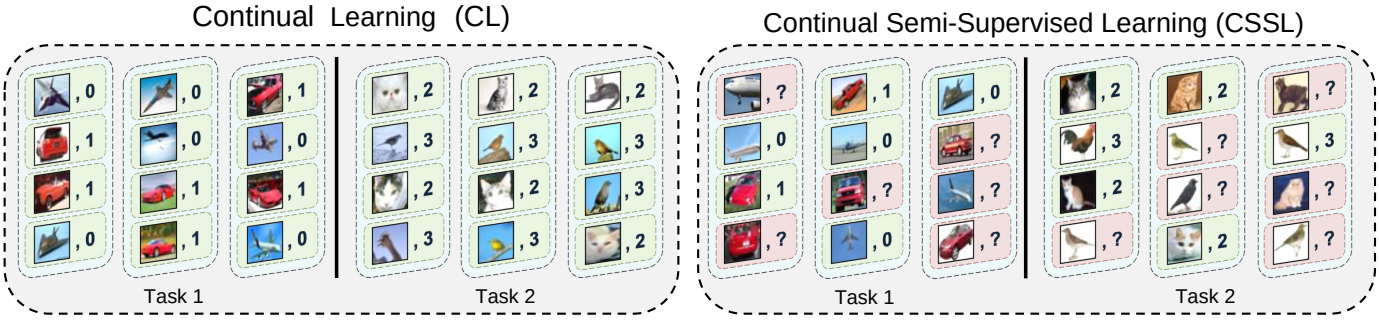

Fig. 1. Overview of the Continual Semi-Supervised Learning (CSSL) setting. Input batches include both labeled (green) and unlabeled (red) examples.

expanded upon in many ways, *e.g.* by adopting advanced memory management policies [9, 23], exploiting meta-learning algorithms [11], combining replay with knowledge distillation [24, 25], or using the memory to train the model in an offline fashion [26].

2.3. Semi-Supervised Learning

Semi-Supervised Learning studies how to improve supervised learning methods by leveraging additional unlabeled data. We exploit the latter in light of specific assumptions on how input and labels interact [5]. By assuming that close input data-points should correspond to similar outputs, **consistency regularization** encourages the model to produce consistent predictions for the same datapoint. This principle can be applied either by comparing the predictions on the same exemplar by different learners [27, 6] or the predictions on different augmentations of the same data-point by the same learner [28].

Recently, several works investigated the refinement of such regularization through **adversarial training**, producing either more challenging perturbations [29] or additional unsupervised samples for regularization purposes [30].

Our proposal, which we introduce in Sec. 4.2, combines within-task consistency regularization with the dual strategy of maximizing cross-task feature dissimilarity. The latter reinforces deep representation learning according to the high-level structure of the target problem – specifically, cross-task class disjunction. This can be seen as a form of Multi-Knowledge Representation [31] through the application of descriptive knowledge; on the other hand, our proposal remains open for further enrichment if additional knowledge on the target task were available [32].

3. Continual Semi-Supervised Learning

A supervised **Continual Learning** classification problem can be defined as a sequence S composed of T tasks. During each of the latter ($S = t \in \{1, ..., T\}$), input samples x and their corresponding ground truth labels y are drawn from an i.i.d. distribution D_t. Considering a function f with parameters θ, we indicate its responses (logits) with $h_\theta(x)$ and the corresponding probability distribution over the classes with $f_\theta(x) \triangleq \text{softmax}(h_\theta(x))$. The goal is to find the optimal value for the parameters θ such that f performs best on average on all tasks without incurring catastrophic forgetting; formally, we need to minimize the empirical risk over all tasks:

$$\underset{\theta}{\text{argmin}} \sum_{t=1}^{t_c} \mathcal{L}_t, \text{where } \mathcal{L}_t \triangleq \mathbb{E}_{(x,y)\sim D_t}\big[\ell(y, f_\theta(x))\big]. \quad (1)$$

In **Continual Semi-Supervised Learning**, we propose to distribute the samples coming from $\mathcal{D}_t$ into two sets: $\mathcal{D}_t^s$, which contains a limited amount of pairs of labeled samples and their ground-truth labels (x_s, y_s) and $\mathcal{D}_t^u$, containing the rest of the unsupervised samples. We define this split according to a given proportion $p_s = {|\mathcal{D}_t^s|}/{(|\mathcal{D}_t^s|+|\mathcal{D}_t^u|)}$ that remains fixed across all tasks. The objective of CSSL is optimizing Eq. 1 without having access to the ground-truth supervision signal for $\mathcal{D}_t^u$. Data from the stream consists of labeled pairs $\mathcal{S} \subset \mathcal{D}_t^s$ and unlabeled items $\mathcal{U} \subset \mathcal{D}_t^u$.

We are interested in shedding further light on CL models by understanding *i)* how they perform under partial lack of supervision and *ii)* how Semi-Supervised Learning approaches can be combined with them to exploit unsupervised data. Question *i)* is investigated experimentally in Sec. 5.1 and 5.2 by evaluating methods that simply *drop* unlabeled examples x_u. Differently, question *ii)* opens up many possible solutions that we address by proposing **Contrastive Continual Interpolation Consistency** (CCIC).

4. Method

We build our proposal upon two state-of-the-art approaches: on the one hand, we take advantage of **Experience Replay** (ER) [22, 11] to mitigate catastrophic forgetting; on the other, we exploit **MixMatch** [28] to learn useful representations also from unlabeled examples. In the following: *i)* to help the reader, we briefly recap the main traits of these algorithms (and let the original papers provide a deeper comprehension); *ii)* we discuss how these two former approaches can be favorably complemented.

4.1. Technical background

As a first step, we equip the learner with a small memory buffer $\mathcal{M}$ (based on *reservoir* sampling) and interleave a batch of examples drawn from it with each batch of the current task. Among all possible approaches, we opt for ER due to its lightweight design and effectiveness [11, 23].

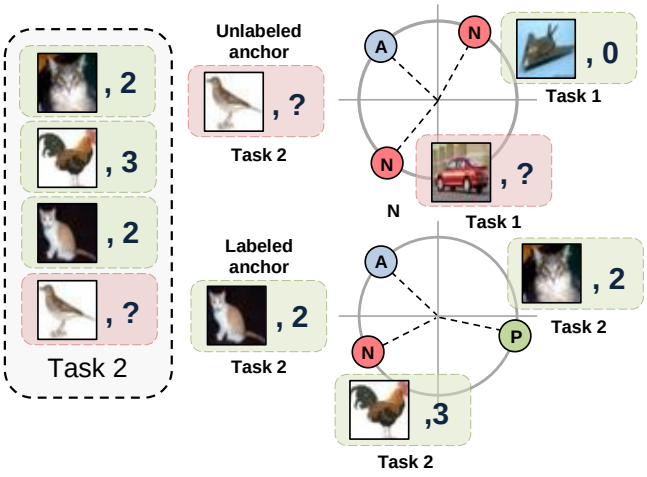

Fig. 2. CCIC exploits task identifiers to enforce semantic constraints: for each anchor (A), it asks the network to push away representations of different (N) tasks and move closer representations of the same one (P).

When dealing with lack of supervision, *self-training* represents a trivial strategy: here, the model itself produces the targets (*pseudo-labels*) for unlabeled examples [33, 34]. Unfortunately, this tends to become unstable with only a few annotations at disposal: as shown in our experiments, this encourages the model to overfit the limited supervised data available [35].

Such an issue raises the need for a different objective, the latter being independent from the accuracy of the model on unlabeled examples. Consequently, we supplement our proposal with **MixMatch** [28]: the predictions of the network are not meant as training targets, but rather as means for applying *consistency regularization* [6, 29]. Briefly, a soft-label is assigned to each unsupervised element by averaging and then sharpening the pre-softmax predictions of several different augmentations.

To promote consistent responses to considerable variations of the data-points, labeled and unlabeled samples are combined through the mixUp procedure [36][1]. Starting from the original sets $\mathcal{S}$ and $\mathcal{U}$ (respectively, labeled and unlabeled examples from the current batch), we thus obtain two final augmented and mixed sets of examples $\mathcal{S}^*$ and $\mathcal{U}^*$: in order to compute the loss terms $\mathcal{L}_{\mathtt{S}^*}$ and $\mathcal{L}_{\mathtt{U}^*}$, we use the ground truth labels for the examples of the former set and the soft-labels generated through response-averaging for the ones of the latter.

4.2. Contrastive Continual Interpolation Consistency

Supposing that boundaries between tasks are provided, we can associate in-memory exemplars with the task they come from. In the following, we discuss how this allows an additional weak form of supervision for unsupervised examples even if we do not know their classes exactly.

[1]Differently from MixMatch, we apply mixUp only on the input images (and not to the corresponding labels).

Unsupervised mining. As tasks are disjoint, examples from different tasks necessarily belong to different classes: we account for that by adding a contrastive loss term, which pushes their responses away from each other (Fig. 2). In details, we wish to maximize the Euclidean distance $D_\theta(x, x') \triangleq \|h_\theta(x) - h_\theta(x')\|_2^2$ between embeddings of examples of different tasks. Hence, we minimize:

$$\mathcal{L}_{\mathtt{UM}} = \mathbb{E}_{\substack{x\sim\mathcal{D}^u_{t_c} \\ x_N\sim\mathcal{M}_{t<t_c}}} \big[\max(\alpha - D_\theta(x, x_N), 0)\big], \qquad (2)$$

where t_c is the index of the current task $\mathcal{D}_{t_c}$, $\mathcal{M}_{t<t_c}$ indicates past examples from the memory buffer, and α is a constant margin beyond which no more efforts should be put into enlarging the distance between negative pairs.

Supervised mining. For each incoming labeled example, we also encourage the network to move its representation close to those belonging to the same class. We look for positive candidates x_P within both the current batch and the memory buffer. In formal terms:

$$\mathcal{L}_{\mathtt{SM}} = \mathbb{E}_{x\sim\mathcal{D}^s_{t_c}\cup\mathcal{M}} \big[\,\mathrm{relu}(\beta - D_\theta(x, x_N) + D_\theta(x, x_P))\big]. \quad (3)$$

Overall objective. To sum up, the objective of CCIC combines the consistency regularization term delivered by MixMatch with the two additional ones (Eq. 2 and Eq. 3) applied in feature space; the overall optimization problem can be formalized as follows:

$$\mathrm{argmin}_\theta \quad \mathcal{L} = \mathcal{L}_{\mathtt{S}} + \lambda\mathcal{L}_{\mathtt{U}} + \mathcal{L}_{\mathtt{SM}} + \mu\mathcal{L}_{\mathtt{UM}}, \qquad (4)$$

where λ and μ are hyperparameters setting the importance of the unsupervised examples.

Exploiting distance metric learning during inference. Once we have introduced constraints in feature space (Eq. 2, 3), we can also exploit them by devising a different inference schema, which further contributes to relieve forgetting. Similarly to [24], we employ the k-Nearest Neighbors algorithm as final classifier, thus decoupling classification from feature extraction. This has been shown beneficial in Continual Learning, as it saves the final fully-connected layer from continuously keeping up with the changing features (and *vice versa*). As kNN is non-parametric and builds upon the feature space solely, it fits in harmony with the rest of the model, controlling the damage caused by catastrophic forgetting. We fit the kNN classifier using the examples of memory buffer as training set.

5. Experiments

We conduct our experiments on three standard datasets[2]. **Split SVHN**: five subsequent binary tasks built on top of the Street View House Numbers (SVHN) dataset [37]; **Split CIFAR-10**: equivalent to the previous one, but using the CIFAR-10 dataset [38]. **Split CIFAR-100**: a longer

[2]Code available at `https://github.com/loribonna/CSSL`.

Table 1. Average Accuracy of CL Methods and of Our Proposals on CSSL Benchmarks.

Class-IL	**SVHN** (UB: 86.2±1.8)				**CIFAR-10** (UB: 92.1±0.1)				**CIFAR-100** (UB: 67.7±0.9)			
Labels %	*0.8%*	*5%*	*25%*	*100%*	*0.8%*	*5%*	*25%*	*100%*	*0.8%*	*5%*	*25%*	*100%*
Fine Tuning	9.9±1.7	9.9±8.4	17.5±9.4	17.8±1.2	13.6±2.9	18.2±0.4	19.2±2.2	19.6±8.4	1.8±0.2	5.0±0.3	7.8±0.1	8.6±0.4
LwF	9.9±0.3	9.9±1.9	14.8±3.6	16.9±0.1	13.1±2.2	17.7±3.2	19.4±1.7	19.6±10.3	1.6±0.1	4.5±0.1	8.0±0.1	8.4±0.5
oEWC	9.9±0.2	9.9±0.7	14.7±0.5	17.9±0.2	13.7±1.2	17.6±1.2	19.1±0.8	19.6±7.5	1.4±0.1	4.7±0.1	7.8±0.4	7.8±0.1
SI	9.9±1.2	10.2±5.9	17.1±7.7	18.2±0.2	12.4±0.4	15.9±1.0	19.2±1.3	19.5±3.3	1.3±0.2	3.4±0.2	7.5±0.5	8.1±1.2
ER_{500}	32.5±7.1	56.0±2.0	59.7±1.8	66.5±2.8	36.3±1.1	51.9±4.5	60.9±5.7	62.2±2.6	8.2±0.1	13.7±0.6	17.1±0.7	21.3±0.2
$iCaRL_{500}$	8.9±0.4	10.0±1.5	19.9±1.2	23.1±2.4	24.7±2.3	35.8±3.2	51.4±8.4	61.0±0.4	3.6±0.1	11.3±0.3	**27.6±0.4**	**37.8±0.3**
DER_{500}	11.9±1.7	54.6±2.6	56.9±5.8	**70.8±3.7**	29.1±0.4	35.3±8.3	50.0±2.3	**67.1±1.6**	1.7±0.1	5.1±0.9	13.0±5.3	28.8±7.2
$GDumb_{500}$	34.6±5.1	41.8±8.3	59.2±8.5	59.9±9.7	39.6±9.6	40.9±11.8	44.8±5.4	47.9±1.6	8.6±0.1	9.9±0.4	10.1±0.4	11.0±1.8
$PseudoER_{500}$	23.2±0.7	48.9±1.2	63.6±2.7	-	37.8±1.6	44.9±2.3	56.3±1.6	-	5.1±0.6	14.3±0.1	18.5±0.5	-
$CCIC_{500}$	**55.3±3.2**	**70.1±3.9**	**75.9±1.5**	-	**54.0±0.2**	**63.3±1.9**	**63.9±2.6**	-	**11.5±0.7**	**19.5±0.2**	20.3±0.3	-
ER_{5120}	44.4±1.4	69.9±3.6	77.6±8.7	**80.5±3.2**	37.4±2.3	64.1±5.3	79.7±1.2	83.3±2.8	9.6±0.6	22.8±0.3	37.9±0.6	**49.0±0.2**
$iCaRL_{5120}$	9.3±0.2	11.5±0.5	19.5±3.7	23.9±4.5	20.7±3.3	35.5±5.6	56.3±2.2	61.9±1.5	4.3±0.1	12.2±0.3	30.9±1.0	41.2±0.4
DER_{5120}	23.1±1.0	67.8±5.2	74.7±2.4	75.3±7.6	32.9±0.9	47.6±2.2	73.9±4.5	**84.5±2.1**	1.6±0.1	4.7±0.6	11.9±3.4	38.6±3.6
$GDumb_{5120}$	46.5±8.0	74.4±2.3	74.6±3.8	78.3±2.3	40.8±12.7	71.2±2.6	81.4±0.8	82.5±0.5	9.6±1.1	23.3±0.1	33.2±2.2	42.9±1.7
$PseudoER_{5120}$	45.8±2.8	74.6±2.4	77.9±0.8	-	**62.2±2.1**	72.9±2.0	80.4±0.1	-	8.2±1.4	25.1±1.6	40.0±0.4	-
$CCIC_{5120}$	**59.3±5.3**	**81.0±2.3**	**83.9±0.2**	-	55.2±1.4	**74.3±1.7**	**84.7±0.9**	-	**12.0±0.3**	**29.5±0.4**	**44.3±0.1**	-

and more challenging evaluation in which the model is presented ten subsequent tasks, each comprising of 10 classes from the CIFAR-100 dataset [38].

We vary the fraction of labeled data shown to the model (p_s) to encompass different degrees of supervision (0.8%, 5%, 25%, and 100%, *i.e.*, 400, 2500, 25000, and 50000 samples for CIFAR-10/100). For fairness, we keep the original balancing between classes in both train and test sets; in presence of low rates, we make sure that each class is represented by a proportional amount of labels.

Architectures. As in [39], experiments on Split SVHN are conducted on a small CNN, comprising of three ReLU layers interleaved by max-pooling. Instead, we rely on ResNet18 for CIFAR-10 and CIFAR-100, as done in [25].

Metrics. We report the performance in terms of average final accuracy, as done in [10, 9]. Accuracies are averaged across 5 runs (we also report standard deviations).

Implementation details. As discussed in Sec. 4, our proposals rely on data augmentation to promote consistency regularization. We apply random cropping and horizontal flipping (except for Split SVHN); the same choice is applied to competitors to ensure fairness. To perform hyperparameters selection (learning rate, batch size, optimization algorithm, and regularization coefficients), we perform a grid search on top of a validation set (corresponding to 10% of the training set), as done in [11, 25, 24]. For CCIC, we keep the number of augmentations fixed to 3 and report chosen values for λ and μ in Tab. 2. To guarantee fairness, we fix the batch size and memory minibatch size to 32 for all models. We train on each task for 10 epochs on SVHN, for 50 on CIFAR-10, and 30 on CIFAR-100. All methods use SGD as an optimizer with the only exception of CCIC, which employs Adam.

5.1. Baselines

Lower/Upper bounds. We bound the performance for our experiments by including two reference measures. As a lower bound, we evaluate the performance of a model trained by *Fine Tuning* exclusively on the set of supervised examples, without any countermeasure to catastrophic forgetting. We also provide an upper-bound (UB) given by a model trained jointly, *i.e.*, without dividing the dataset into tasks or discarding any ground-truth annotation.

Table 2. Values of (λ, μ) for CCIC chosen after the grid-search.

Lab. %	$\lvert\mathcal{M}\rvert$	**SVHN**	**CIFAR-10**	**CIFAR-100**
0.8%	500	$(0.5, 0.5)$	$(0.5, 0.5)$	$(0.3, 1.0)$
	5120	$(0.1, 0.5)$	$(0.5, 0.5)$	$(0.3, 0.3)$
5%	500	$(0.1, 0.5)$	$(0.3, 0.5)$	$(0.5, 0.5)$
	5120	$(0.1, 0.5)$	$(0.5, 0.5)$	$(0.5, 0.5)$
25%	500	$(0.5, 0.5)$	$(0.1, 0.5)$	$(0.5, 0.7)$
	5120	$(0.5, 0.5)$	$(0.1, 1.0)$	$(0.5, 0.5)$

Drop-the-unlabeled. The most straightforward approach to adapt existing methods to our setting consists in simply discarding unlabeled examples from the current batch. In this regard, we compare our proposal with Learning Without Forgetting (LwF) [20], online Elastic Weight Consolidation (oEWC) [21], Synaptic Intelligence (SI) [19], Experience Replay (ER) [11], iCaRL [24], Dark Experience Replay (DER) [25] and GDumb [26]. By so doing, we can verify whether our proposal is able to better sustain a training regime with reduced supervision.

Pseudo-Labeling. Inspired by the line of works relying on self-labeling [33, 34], we here introduce a simple CSSL baseline that allows ER to profit from the unlabeled examples: given an unlabeled example x_u, it pins as a *pseudo-label* $\tilde{y}_u$ [34] the prediction of the model itself. Formally,

$$\tilde{y}_u = \text{argmax}_{c \in C_t} \; h^c_\theta(x_u), \tag{5}$$

where C_t is the set of classes of the current task. As discussed in Sec. 4.1, self training is likely to cause model instability (especially at task boundaries, when the model starts to experience new data): we mitigate this by applying a threshold η to discard low-confidence outputs and their relative x_u. Specifically, we estimate the confidence

as the difference between the two highest values of $h^c_\theta(x_u)$. After this step, a pair $(x_u, \tilde{y}_u)$ is considered on a par with any supervised pair (x_s, y_s), and is therefore inserted into the memory buffer. We refer to this baseline as **PseudoER**.

5.2. Experimental Results

As revealed by the results in Tab. 1, CSSL proves to be a challenging scenario. Unsurprisingly, its difficulty increases when fewer labels are provided to the learner.

Regularization methods are generally regarded as weak in the Class-IL scenario [7, 9]. This conforms with our empirical observations, as LwF, oEWC and SI underperform across all datasets. Indeed, these methods rarely outperform our lower bound (Fine Tuning), indicating that they are not effective outside of Task-IL and Domain-IL. This becomes especially evident in the low-label regime.

Rehearsal methods overall show an expected decrease in performance as supervision diminishes. This is especially severe for DER and iCaRL, as their accuracy drops on average by more than 70% between 100% and 0.8% labels. As the model underfits the task when less supervision is provided, it produces less reliable targets that cannot be successfully used for replay by these methods. In contrast, ER is able to replay information successfully as it exploits hard targets; thus, it learns effectively even after initially underfitting the task. Indeed, its accuracy with 5% labels and buffer 5120 is always higher than its fully-supervised accuracy with a smaller buffer. While ER is able to overcome the lack of labels when paired with an appropriate buffer, knowledge-distillation based approaches remarkably encounter a major hindrance in this setting.

We attribute the failure of iCaRL on SVHN to the low complexity of the backbone network. Indeed, a shallow backbone provides for a latent space that is less suitable for its nearest-mean-of-exemplars classifier. Conversely, this method proves quite effective even with a reduced memory buffer on CIFAR-100. In this benchmark, the *herding sampling* of iCaRL ensures that all classes are fairly represented even in a small memory buffer.

Finally, GDumb does not suffer from lower supervision as long as its buffer can be filled completely: its operation is not disrupted by unlabeled examples on the stream, as it ignores the latter entirely. While it outperforms other methods when few labels are available, CCIC surpasses it consistently. This suggests that the stream offers potential for further learning and should not be dismissed.

CSSL Methods. Our PseudoER baseline performs notably well on CIFAR-10, maintaining high accuracy as the amount of supervision decreases. However, while CIFAR-10 is a complex benchmark, it only features two classes for each task, which makes it easy for pseudo-labeling to produce reasonable responses (it is noted that a random guess would result in 50% accuracy). Conversely, PseudoER struggles to produce valid targets and exhibits a swift performance drop on CIFAR-100 as the availability of labeled data decreases. Similarly, we find the application of pseudo-labeling beneficial for SVHN only as the space reserved for the buffer increases, demonstrating the pitfalls of this approach in the online setting.

On the contrary, the compelling performance of CCIC indicates successful blending of supervised information and semi-supervised regularization. While ER encounters an average performance drop of 47%, going from 25% to 0.8% labels on CIFAR-10, CCIC only loses 26% on average. Surprisingly, we observe that – for the majority of evaluated benchmarks – 25% supervision is enough to approach the results of fully-supervised methods, even outperforming the state-of-the-art in some circumstances (CIFAR-10 with buffer size 5120, SVHN with buffer size 500 and 5120).

This hints that, when learning from a stream of data, striving to provide full supervision is not as essential as it might be expected: differently from the offline scenario, a greater amount of labels might not produce a proportionate profit due to *catastrophic forgetting*. In this respect, our experiments suggest that pairing few labeled examples with semi-supervised techniques represents a more efficient paradigm to achieve satisfying performance.

Table 3. Unsupervised Mining Techniques for CCIC on CIFAR-100.

Labels % ($\lvert\mathcal{M}\rvert = 5120$)	*5%*	*25%*
Across-Task Mining (Eq. 2)	29.5±0.4	44.3±0.1
Within-Task Mining	29.3±0.2	44.0±0.2
Task-Agnostic Mining	29.1±0.7	43.9±0.8

Table 4. Average Accuracy of alternative CSSL proposals on CIFAR-10

Labels %	*0.8%*	*5%*	*25%*
ER+EMA$_{500}$	21.4±0.5	26.3±1.0	43.3±1.2
CCIC$_{500}$	54.0±0.2	63.3±2.1	63.9±2.6
ER+EMA$_{5120}$	25.9±0.8	40.8±2.1	64.8±0.4
CCIC$_{5120}$	55.2±1.2	74.3±1.7	84.7±0.9

Unsupervised Mining in CCIC. In its unsupervised mining loss term $\mathcal{L}_{\texttt{UM}}$, CCIC takes examples of previous tasks in the memory buffer as negatives (*Across-Task Mining*) and requires their representations to be pushed away from current data. In Tab. 3, we compare this design choice with two alternative strategies: *i) Within-Task Mining*, where we let the model choose the negatives from the current task only; and *ii) Task-Agnostic Mining*, where the model can freely pick a negative example from either the memory or the current batch without any task-specific prior. As can be observed, Task-Agnostic Mining and Within-Task Mining lead to a small but consistent decrease in performance, while $\mathcal{L}_{\texttt{UM}}$ proves to be the most rewarding strategy.

Model-driven Consistency. In addition with combining a contrastive form of consistency regularization with ER, we propose an additional *temporal consistency* baseline which requires the activations of the model to match a slower moving-average checkpoint. Results in Tab. 4 show, however, that such approach under-performs consistently, not even reaching the performance of ER. This suggests that, differently from fully-supervised scenarios [6], exponential moving average approaches do not necessarily scale to CL.

6. Conclusion

Catastrophic forgetting prevents most current state-of-the-art models from sequentially learning multiple tasks, forcing practitioners to heavy resource-demanding training processes. Moreover, many of the applications that might benefit from CL algorithms are often characterized by label scarcity. For this reason, we investigate the possibility of leveraging unlabeled data-points to enhance the performance of Continual Learning models, a scenario that we name **Continual Semi-Supervised Learning (CSSL)**.

We further propose **Constrastive Continual Interpolation Consistency (CCIC)**, an incremental approach that combines the benefits of rehearsal with consistency regularization and distance-based constraints. Remarkably, our experiments suggest that well-designed methods can effectively exploit the unlabeled examples to prevent forgetting. This indicates that the effort of annotating all data may be unnecessary in a continual scenario.

Acknowledgement

This work was supported by the FF4EuroHPC: HPC Innovation for European SMEs, Project Call 1. Project FF4EuroHPC has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 951745.

References

[1] M. McCloskey, N. J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, Psychol Learn Motiv doi:10.1016/S0079-7421(08)60536-8 (1989).

[2] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, IEEE TPAMI doi:10.1109/TPAMI.2021.3057446 (2021).

[3] R. Aljundi, K. Kelchtermans, T. Tuytelaars, Task-free continual learning, in: CVPR, 2019.

[4] W. Zhou, S. Chang, N. Sosa, H. Hamann, D. Cox, Lifelong object detection, arXiv:2009.01129 (2020).

[5] C. Olivier, S. Bernhard, Z. Alexander, Semi-supervised learning, 2006, doi:10.7551/mitpress/9780262033589.001.0001.

[6] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: ANIPS, 2017.

[7] S. Farquhar, Y. Gal, Towards robust evaluations of continual learning, in: ICML Workshop, 2018.

[8] G. M. van de Ven, A. S. Tolias, Three continual learning scenarios, in: ANIPS Workshop, 2018.

[9] R. Aljundi, M. Lin, B. Goujaud, Y. Bengio, Gradient based sample selection for online continual learning, in: ANIPS, 2019.

[10] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, in: ANIPS, 2017.

[11] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, G. Tesauro, Learning to learn without forgetting by maximizing transfer and minimizing interference, in: ICLR, 2019.

[12] A. Chaudhry, A. Gordo, P. K. Dokania, P. Torr, D. Lopez-Paz, Using hindsight to anchor past knowledge in continual learning, in: AAAI Conf. Artif. Intell., 2021.

[13] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, C.-C. J. Kuo, Class-incremental learning via deep model consolidation, in: WACV, 2020.

[14] A. Lechat, S. Herbin, F. Jurie, Semi-supervised class incremental learning, in: ICPR, 2021.

[15] J. Serra, D. Suris, M. Miron, A. Karatzoglou, Overcoming catastrophic forgetting with hard attention to the task, in: ICML, 2018.

[16] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, D. Wierstra, Pathnet: Evolution channels gradient descent in super neural networks, arXiv:1701.08734 (2017).

[17] A. Mallya, S. Lazebnik, Packnet: Adding multiple tasks to a single network by iterative pruning, in: CVPR, 2018.

[18] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, PNAS doi:10.1073/pnas.1611835114 (2017).

[19] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: ICML, 2017.

[20] Z. Li, D. Hoiem, Learning without forgetting, IEEE TPAMI doi:10.1109/TPAMI.2017.2773081 (2017).

[21] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, R. Hadsell, Progress & compress: A scalable framework for continual learning, in: ICML, 2018.

[22] R. Ratcliff, Connectionist models of recognition memory: constraints imposed by learning and forgetting functions., Psychol. Rev. doi:10.1037/0033-295x.97.2.285 (1990).

[23] P. Buzzega, M. Boschini, A. Porrello, S. Calderara, Rethinking experience replay: a bag of tricks for continual learning, in: ICPR, 2020.

[24] S. Rebuffi, A. Kolesnikov, G. Sperl, C. Lampert, icarl: Incremental classifier and representation learning, in: CVPR, 2017.

[25] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, in: ANIPS, 2020.

[26] A. Prabhu, P. H. Torr, P. K. Dokania, Gdumb: A simple approach that questions our progress in continual learning, in: ECCV, 2020.

[27] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, in: ICLR, 2017.

[28] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C. A. Raffel, Mixmatch: A holistic approach to semi-supervised learning, in: ANIPS, 2019.

[29] T. Miyato, S.-i. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, IEEE TPAMI doi:10.1109/TPAMI.2018.2858821 (2018).

[30] Z. Zheng, L. Zheng, Y. Yang, Unlabeled samples generated by gan improve the person re-identification baseline in vitro, in: ICCV, 2017.

[31] Y. Yang, Y. Zhuang, Y. Pan, Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies, Front. Inf. Technol. Electron. Eng. doi:10.1631/FITEE.2100463 (2021).

[32] Y. Pan, Multiple knowledge representation of artificial intelligence, Engineering doi:10.1016/j.eng.2019.12.011 (2020).

[33] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: ACL, 1995, doi:10.3115/981658.981684.

[34] D.-H. Lee, Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: ICML Workshop, 2013.

[35] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, I. Goodfellow, Realistic evaluation of deep semi-supervised learning algorithms, in: ANIPS, 2018.

[36] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: ICLR, 2018.

[37] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: ANIPS, 2011.

[38] A. Krizhevsky, et al., Learning multiple layers of features from tiny images, Tech. rep. (2009).

[39] D. Abati, J. Tomczak, T. Blankevoort, S. Calderara, R. Cucchiara, B. E. Bejnordi, Conditional channel gated networks for task-aware continual learning, in: CVPR, 2020.