Divide-and-Conquer for Enhancing Unlabeled Learning, Stability, and Plasticity in Semi-supervised Continual Learning

Yue Duan[1*] Taicai Chen[1*] Lei Qi[2] Yinghuan Shi[1†]
[1]Nanjing University [2]Southeast University
{yueduan@smail., taicaichen@smail., syh@}nju.edu.cn, qilei@seu.edu.cn

Abstract

Semi-supervised continual learning (SSCL) seeks to leverage both labeled and unlabeled data in a sequential learning setup, aiming to reduce annotation costs while managing continual data arrival. SSCL introduces complex challenges, including ensuring effective unlabeled learning (UL), while balancing memory stability (MS) and learning plasticity (LP). Previous SSCL efforts have typically focused on isolated aspects of the three, while this work presents USP, a divide-and-conquer framework designed to synergistically enhance these three aspects: (1) Feature Space Reservation (FSR) strategy for LP, which constructs reserved feature locations for future classes by shaping old classes into an equiangular tight frame; (2) Divide-and-Conquer Pseudo-labeling (DCP) approach for UL, which assigns reliable pseudo-labels across both high- and low-confidence unlabeled data; and (3) Class-mean-anchored Unlabeled Distillation (CUD) for MS, which reuses DCP's outputs to anchor unlabeled data to stable class means for distillation to prevent forgetting. Comprehensive evaluations show USP outperforms prior SSCL methods, with gains up to 5.94% in the last accuracy, validating its effectiveness. The code is available at https://github.com/NJUyued/USP4SSCL.

1. Introduction

Recently, *continual learning* (CL) has emerged as a promising approach for handling such sequential data arrival scenarios [47, 50]. Yet, most existing CL methods rely heavily on fully labeled data, which is often impractical in real-world applications due to high annotation costs, privacy concerns, and limitations in adapting to real-time online scenarios. To address these challenges, researchers have turned to *semi-supervised learning* (SSL) [34, 53] frameworks, where only a subset of samples requires labeling.

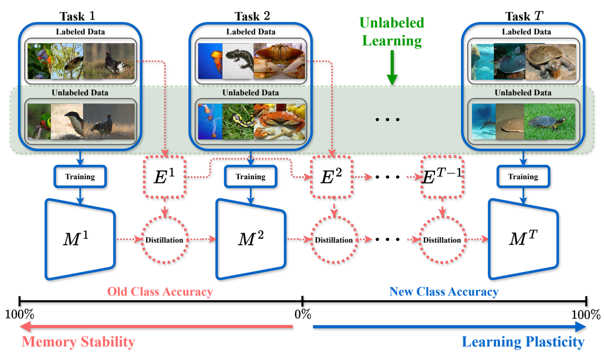


Figure 1. Illustration of the SSCL setting (E^t and M^t indicate the exemplar set and model in task t, respectively): in a dynamic data stream containing both labeled and unlabeled data, distinct tasks arrive sequentially with non-overlapping classes across tasks.

Inspired by this paradigm, a new direction called *semi-supervised continual learning* (SSCL) [2, 46] has emerged, aiming to leverage SSL setups across all tasks within the CL setting, which is illustrated in Fig. 1.

SSCL introduces distinct challenges due to the need to continuously learn from both labeled and unlabeled data. This setting demands careful management of the trade-off between *learning plasticity*—the model's ability to incorporate new knowledge, and *memory stability*—its capacity to retain past knowledge [24, 25, 61]. In SSCL, these dynamics become particularly complex as the model faces the risk of catastrophic forgetting of past tasks, while simultaneously being prone to overfitting on limited labeled samples [12, 17, 31]. Meanwhile, learning from a vast pool of unlabeled data is also challenging, as CL's anti-forgetting processes can disrupt learning under sparse supervision. Furthermore, standard CL techniques like experience replay face obstacles in SSCL: constrained replay buffers often prioritize labeled samples, limiting the use of valuable information from unlabeled data [5, 19, 39].

We argue that addressing SSCL effectively requires a holistic approach that does not overlook any of these

*Equal contribution.
†Corresponding author.

three aspects: **unlabeled learning (UL)**, **memory stability (MS)**, and **learning plasticity (LP)**. Although previous approaches have made strides in SSCL, most of them primarily focus on *just one or two of the three core challenges, often leaving the other aspects unaddressed*. For examples, for UL, [39] employs pseudo-labeling technique used in SSL to utilize unlabeled data for training and [13] apply consistency loss to boost the robustness of unlabeled learning; For MS, [17] proposes dynamic sub-graph distillation (DSGD) to leverages semantic and structural information from unlabeled data, helping the model remain robust to distribution shifts. Motivated by this, we propose a ***divide-and-conquer approach with three interlinked modules: USP**, each tailored to enhance one of these aspects while collectively improving the overall SSCL performance.*

The key to building a powerful SSCL lies in designing mechanisms that enable UL-, MS- and LP-components to complement and reinforce each other. With the established baseline SSCL learner, we first introduce a projection head to produce an additional feature branch, which serves as a unified feature output across all subsequent components aiming to strengthen their coupling for mutual enhancement. **(1) For LP**: We propose a simple yet effective Feature Space Reservation strategy (FSR). Leveraging the equiangular tight frame (ETF) for optimal feature geometry, we obtain anchor vectors that reserve space for future classes. A contrastive-like loss is proposed to align learned data features to these class-specific positions, laying a strong foundation for subsequent CL processes. **(2) For UL**: We propose a Divide-and-Conquer Pseudo-labeling strategy (DCP) to handle high- and low-confidence unlabeled data separately, leveraging two complementary pseudo-labeling techniques. This approach ensures effective utilization of all data while maintaining pseudo-label accuracy, ultimately delivering a robust UL process—*and even offering a "free lunch" benefit during testing phase*. **(3) For MS**: We repurpose intermediate calculations from DCP to introduce the Class-mean-anchored Unlabeled Distillation (CUD). CUD aggregates the latent relationship between labeled and unlabeled data, enhancing model's resistance to catastrophic forgetting on unlabeled data, thus supporting stable and reliable representation retention in SSCL.

Contribution. A divide-and-conquer framework, **USP**, is proposed that synergistically enhances unlabeled learning (**U**L), memory stability (M**S**), and learning plasticity (L**P**): (1) We introduce a novel pseudo-labeling scheme, which ensures high-quality pseudo-labels across confidence levels, fully utilizing all data to improve UL. (2) We propose a feature space reservation strategy and cross-labeled-unlabeled distillation to jointly enhance LP and MS, helping the model resist forgetting. (3) Extensive evaluations across diverse SSCL settings demonstrate the performance benefits of **USP**, with up to a 4.10% gain in average accuracy.

2. Related Works

2.1. Continual Learning

Continual learning (CL) addresses catastrophic forgetting during incremental learning. As summarized in [61], mainstream approaches fall into three categories: *replay*-based, *knowledge distillation*-based, and *dynamic network*-based methods. Replay-based methods retain/rehearse past data through stored exemplars [3, 13, 57] or synthetic generation [18, 21, 37]. Knowledge distillation-based methods transfer knowledge from old to new models through distillation of logits [38, 40, 56], features [22, 30, 35], or relations [1, 29]. Dynamic network-based methods expand architectures via neuron [28], backbone [45, 52], or prompt [41, 49] growth.

While CL methods are often categorized distinctly, their boundaries are fluid. Replay and distillation synergize as core anti-forgetting mechanisms across paradigms. Our method similarly harnesses their complementary strengths to combat catastrophic forgetting.

2.2. Semi-supervised Learning

Semi-supervised learning (SSL) aims to reduces the dependency of deep learning models on labeled data by leveraging abundant unlabeled data. Early SSL methods can be broadly categorized into *pseudo-labeling* and *consistency regularization* methods. Pseudo-labeling methods expand the training set by assigning predictions on unlabeled data as pseudo-labels [8, 16, 27, 51]. In contrast, consistency regularization methods enforce similar predictions across augmented versions of input samples, enhancing generalization boundaries through teacher-student interactions [43] or by applying diverse perturbations to inputs [15, 33, 42]. FixMatch [42] introduced a simple yet effective framework that integrates these two strategy: it applies weak and strong augmentations to unlabeled data and uses high-confidence predictions on weakly-augmented samples as pseudo-labels for the strongly-augmented counterparts, enforcing a strong-weak consistency regularization on unlabeled data. This approach demonstrated remarkable performance and has become a foundational benchmark in SSL, inspiring many subsequent methods that adjust pseudo-labeling strategies for unlabeled data [14, 58, 59] or modify confidence thresholding schemes [7, 9, 48].

While FixMatch and similar methods have achieved significant success in SSL, current SSL methods still fall short in addressing learning scenarios where data distribution or class compositions may shift over time.

2.3. Semi-supervised Continual Learning

Existing continual learning methods generally rely on a fully supervised setup, whereas semi-supervised continual learning (SSCL) more realistically assumes that only a limited number of samples are labeled at each task. The core challenge in SSCL is effectively utilizing unlabeled data to mitigate catastrophic forgetting. CNNL [4] fine-tunes its incremental learner by generating pseudo-labels for the unlabeled data, enabling self-training. DistillMatch [39] employs knowledge distillation with prediction consistency on unlabeled data. It also optimizes an out-of-distribution detector to identify task-specific representations. Pseudo-gradient learner [31] introduces a gradient predictor using labeled data to estimate gradients for unlabeled data, thereby avoiding the potential risks of pseudo-labeling. ORDisCo [46] learns continuously from partially labeled data using a classifier-equipped conditional GAN and performs online data replay. MCSSL [5] extends ORDisCo into a meta-learning framework. DSGD [17] introduces a dynamic subgraph distillation method that leverages semantic and structural information for more stable knowledge distillation on unlabeled data.

Unlike previous methods that focus on individual aspects of UL, MS, or LP, our proposed method integrates all three into a unified framework, aiming for a synergistic effect that amplifies their combined impact.

3. Methods

3.1. Baseline SSCL Learner

Denoting the input space as $\mathcal{X}$ and the label space as $\mathcal{Y} = \{1, ..., K\}$ over K classes, we formally define the problem of continual learning (CL) as follows: Given training data that arrives sequentially as a sequence of T tasks, each task t is associated with a dataset $D^t \subseteq \mathcal{X}^t \times \mathcal{Y}^t$, where $t \in \{1, \dots, T\}$ and $\mathcal{Y}^1 \cap \mathcal{Y}^2 \cap \dots \cap \mathcal{Y}^T = \emptyset$. Same below, the superscript of x^t is always used to indicate the variable x at different task t. The learning process is conducted task-by-task, and during the training of task t, only the current dataset D^t is accessible, while data from previous tasks is systematically discarded. Note that in actual continuous learning settings, a relatively small memory buffer is usually reserved to store past examples to help the model alleviate catastrophic forgetting, which is denoted as E^t. $E^t = E^{t,(1)} \cup \dots \cup E^{t,(K)}$ and $E^{t,(i)}$ is the exemplar set of old class i containing exemplars stored from the datasets in previous $t-1$ tasks.

In semi-supervised continual learning (SSCL), each dataset D^t is partially labeled and can be divided into two subsets: $D^t = D_l^t \cup D_u^t$, where $D_l^t \subseteq \mathcal{X}^t \times \mathcal{Y}^t$ denotes the labeled subset and $D_u^t \subseteq \mathcal{X}^t$ denotes the unlabeled subset, with $|D_l^t| \ll |D_u^t|$. In breif, we can review SSCL as a optimization task on the model parameterized by θ:

$$\min_{\theta} \sum_{t=1}^{T} \mathcal{L}_{\texttt{ssl}}(D^t) + \mathcal{L}_{\texttt{cl}}(E^t), \tag{1}$$

where $\mathcal{L}_{\texttt{ssl}}$ is the semi-supervised learning (SSL) loss and $\mathcal{L}_{\texttt{cl}}$ is the CL loss.

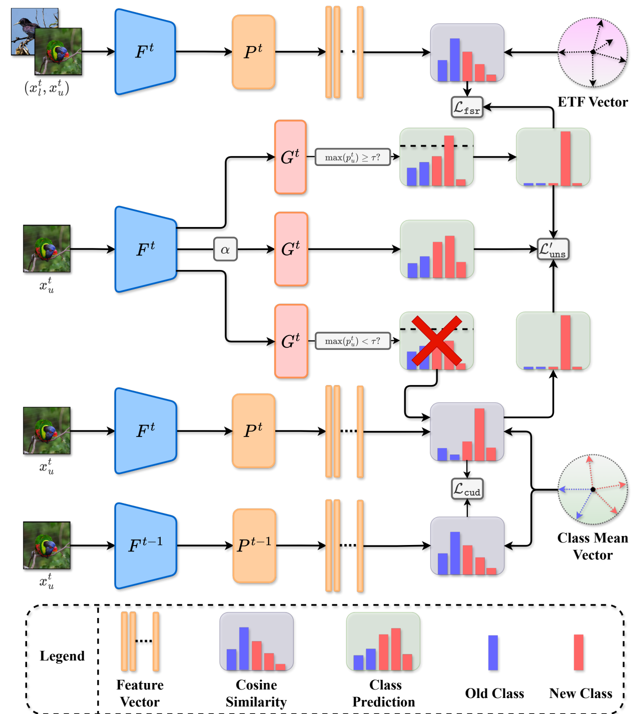


Figure 2. **Overview of USP.** All losses of USP uniformly utilize the features output by a projection head P^t to enhance synergy through coupling. (1) $\mathcal{L}_{\texttt{fsr}}$ for LP (Sec. 3.2): Given the labeled and unlabeled data (x_l^t, x_u^t), a contrastive loss aligns their features with pre-computed ETF vectors that represent the optimal geometric structure for classification, preventing feature space conflicts between new and old classes during learning; (2) $\mathcal{L}'_{\text{uns}}$ for UL (Sec. 3.3): For x_u^t, we compute its class prediction p_u^t and employ a confidence-based divide-and-conquer approach, which leverages complementary pseudo-labels based on classifier and NCM to provide a robust UL process; (3) $\mathcal{L}_{\text{cud}}$ for MS (Sec. 3.4): Reusing intermediate results from DCP, we anchor the features of unlabeled data to class mean vectors enriched with information of labeled data, mitigating catastrophic forgetting in SSCL.

We propose a divide and conquer approach to **U**nlabeled learning, **S**tability, and **P**lasticity (**USP**), which is shown in Fig. 2. Overall, we first set a feature extractor $F(\cdot)$, a classifier $G(\cdot)$ and a projection head $P(\cdot)$. Following [17], we build a basic SSCL learner using FixMatch [42] for $\mathcal{L}_{\texttt{ssl}}$, and adopt iCaRL [38] or DER [52] for $\mathcal{L}_{\texttt{cl}}$ (DER's details are deferred to Sec. A.3 of supplementary).

(1) $\mathcal{L}_{\texttt{ssl}}$. Given the labeled data $\{x_l^t, y_l^t\} \subset D_l^t$ and the unlabeled data $x_u^t \in D_u^t$, following the most popular pseudo-labeling based SSL method FixMatch, the terms $\mathcal{L}_{\texttt{ssl}}$ in Eq. (1) can be decomposed into two loss terms: $\mathcal{L}_{\texttt{ssl}}(D^t) = \mathcal{L}_{\text{sup}}(D_l^t) + \lambda_{\text{uns}} \mathcal{L}_{\text{uns}}(D_u^t)$, where λ_{uns} is the loss weight. Denoting the model prediction of x as $p_x^t = G^t(F^t(x))$, we define the supervised loss as

$$\mathcal{L}_{\text{sup}}(D_l^t) = \mathbb{E}_{(x_l^t, y_l^t) \sim D_l^t} \left[H \left(p_{x_l^t}^t, y_l^t \right) \right], \tag{2}$$

where $H(\cdot,\cdot)$ is the standard cross-entropy loss. Denoting $\tilde{x} = \max(x)$ and $\hat{x} = \arg\max(x)$, the unsupervised loss is defined as

$$\mathcal{L}_{\texttt{uns}}(D_u^t) = \mathbb{E}_{x_u^t \sim D_u^t}\left[\mathbb{1}\left(\tilde{p}_{x_u^t}^t \geq \tau\right) H\left(p_{\alpha(x_u^t)}^t, \hat{p}_{x_u^t}^t\right)\right], \tag{3}$$

where $\mathbb{1}(\cdot)$ is the indicator function, $\hat{p}_{x_u^t}^t$ is the hard pseudo-label of $p_{x_u^t}^t$ and $\alpha(\cdot)$ represents a strong data augmentation function[1] and τ is a confidence threshold for $\tilde{p}_{x_u^t}^t$ (*i.e.*, the maximum softmax probability) to select pseudo-labels that are more likely to be correct.

(2) $\mathcal{L}_{\texttt{cl}}$. With the exemplar management of iCaRL [38] (see Sec. A.1 of supplementary for details), given $x_e^t \in E^t$, we utilize the knowledge distillation loss for $\mathcal{L}_{\texttt{cl}}$ in Eq. (1) by encouraging the current model to output the same prediction $p_e^t = G^t(F^t(x_e^t))$ as that of the old model:

$$\mathcal{L}_{\texttt{cl}}(E^t) = \mathbb{E}_{x_e^t \sim E^t}\left[\texttt{KL}\left(\frac{p_e^t}{\beta} \parallel \frac{p_e^{t-1}}{\beta}\right)\right], \tag{4}$$

where $\texttt{KL}(\cdot \parallel \cdot)$ is the KL-divergence and β is the temperature parameter.

Note that from here on, all feature vectors extracted from an input x mentioned in the following text are denoted as f_x^t, which are output by the projection head $P(\cdot)$ and normalized using L_2-normalization, *i.e.*, $f_x^t = \frac{P^t(F^t(x))}{\|P^t(F^t(x))\|_2}$ (and after any operation on the features, they will be renormalized). All components of USP consistently leverage the output features from $P(\cdot)$, *aiming to strengthen the coupling of all components for mutual reinforcement.*

3.2. ETF-Based Feature Space Reservation

We begin by enhancing the plasticity of USP. Inspired by [36, 54, 60], we aim to design a **F**eature **S**pace **R**eservation (**FSR**) method for accommodating upcoming new classes without interfering with the feature patterns retained for previously learned classes.

We aim to utilize a simple yet effective contrastive learning loss to align sample features of each class with a set of predefined class prototype features derived from a simplex equiangular tight frame (ETF) for the entire label space. The ETF is inspired by neural collapse phenomenon (more details and the calculation of ETF can be found in Sec. A.4 of supplementary), which indicates that the final-layer features of samples within the same class collapse to a single vertex. And the vertices of all classes align with class prototypes that form an ETF, which refers to a matrix $\mathcal{E} \in \mathbb{R}^{d\times k}$ (d is a predefined parameter, k is the total number of classes). Crucially, the ETF structure pre-defines maximally separated prototype positions in the feature space, inherently reserving geometric capacity for future unseen classes while preserving semantic discrimination of learned ones. Each column vector $\mathcal{E}_{:,i} \in \mathbb{R}^d$ in ETF can be considered as the corresponding prototype for class i. As the ETF corresponds to the optimal geometric structure for classification, anchoring the continually arriving class features to the different ETF-prototype-vectors as learning targets fundamentally enhances the plasticity of SSCL models. With obtained $\mathcal{E}_{:,i}$, denoting the cosine similarity as $S(\cdot,\cdot)$, we define the following contrastive loss for feature alignment on both the labeled and unlabeled data:

$$\begin{aligned}\mathcal{L}_{\texttt{fsr}}(D^t) = \mathbb{E}_{(x_l^t, y_l^t)\sim D_l^t}&\left[-\log\frac{\exp\left(\frac{S(\mathcal{E}_{:,y_l^t}, f_{x_l^t}^t)}{\gamma}\right)}{\sum_{i=1}^k \exp\left(\frac{S(\mathcal{E}_{:,i}, f_{x_l^t}^t)}{\gamma}\right)}\right] \\ + \mathbb{E}_{x_u^t\sim D_u^t}&\left[\mathbb{1}\left(\tilde{p}_{x_u^t}^t \geq \tau\right) - \log\frac{\exp\left(\frac{S(\mathcal{E}_{:,\hat{p}_{x_u^t}^t}, f_{x_u^t}^t)}{\gamma}\right)}{\sum_{i=1}^k \exp\left(\frac{S(\mathcal{E}_{:,i}, f_{x_u^t}^t)}{\gamma}\right)}\right],\end{aligned} \tag{5}$$

where γ is the temperature parameter. We still use the confidence threshold to filter the pseudo-labels on unlabeled data to ensure the robustness of FSR as much as possible.

3.3. Divide and Conquer Pseudo-Labeling

For unlabeled learning, considering that we adopt a pseudo-labeling framework similar to FixMatch for $\mathcal{L}_{\texttt{uns}}$, ensuring the accuracy of pseudo-labels is paramount. Thus, we propose **D**ivide-and-**C**onquer **P**seudo-labeling (**DCP**) that combines the two complementary classification methods to leverage their respective strengths in SSCL for $\mathcal{L}_{\texttt{uns}}$.

3.3.1. Training Phase

In FixMatch and its adaptation as a basic learner in [17] (Eq. (3)), pseudo-labeling involves assigning hard labels by applying a threshold to the softmax outputs of classifier logits on unlabeled data. However, this will result in low-confidence samples not participating in training and resulting in a waste of data. This is a drawback of classifier-classification, because the pseudo-labels of low-confidence samples are likely to be wrong and harm training. Thus, we consider another nearest class mean (NCM) classification method in CL [32], which classifies samples by matching them to prototype vectors—incrementally computed as the average features of observed examples, for pseudo-labeling on samples with low confidence.

The work in [12] argues that NCM-like classification could maintain more stable performance in CL by measuring distances between test image features and class prototypes, which depend only on the parameters of the backbone model. In contrast, classifier-based predictions require feature input into an expanding fully connected layer, which is

[1] This implies that we employ *consistency regularization* to enhance the model's prediction stability for semantically similar images. More details of this technique can be found in Section 2.2.

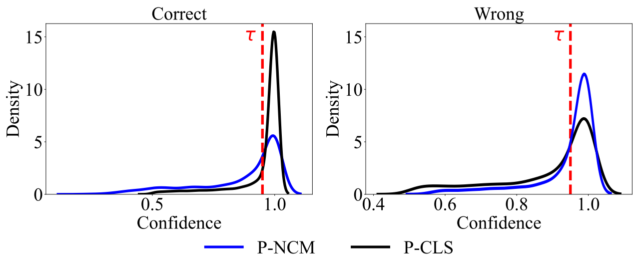


Figure 3. Kernel density estimation (KDE) of confidence distributions of pseudo-labels on 5-task CIFAR10-30 (see Sec. 4.1 for the experimental setting). We show the confidence of classifier-based pseudo-labels ("P-CLS"), divided into correct and incorrect parts, and the re-partitioning of these corresponding examples if the pseudo-labels are providing by NCM ("P-NCM"). The red dash lines indicate the confidence threshold τ in Eq. (3).

updated only for the current session, often leading to classification instability in incremental tasks. In this work, as illustrated in Fig. 3, we observe that classifier-based pseudo-labeling can yield highly reliable results at high-confidence levels but lack reliability for low-confidence predictions. NCM classification, on the other hand, maintains a relatively more robust accuracy across all confidence level of classifier predictions. Given discussed above, for high-confidence predictions, DCP apply hard pseudo-labels from the classifier outputs, while low-confidence predictions are assigned pseudo-labels using NCM classification.

In our NCM classification, considering that the ground-truth of unlabeled data is unknown, the prototype vector is computed for each observed class by averaging the feature vectors of all exemplars selected from labeled data in that class. First, we separate the labeled data by class, where $C^{t,(i)}$ represents the labeled set of class i. Then, we compute the feature mean of class i by $\mu_{C^{t,(i)}} = \frac{\sum_{c \in C^{t,(i)}} f_c^t}{|C^{t,(i)}|}$. Finally, the class label $q_{x_u^t}$ for an unlabeled sample x_u^t is assigned based on $\mu_{C^{t,(i)}}$ with the highest similarity to the input's features, *i.e.*,

$$q_{x_u^t} = \mathop{\arg\max}_{i=1,\cdots,|\mathcal{Y}^t|} S\Big(f_{x_u^t}^t, \mu_{C^{t,(i)}}\Big). \tag{6}$$

Hereafter, we can rewrite Eq. (3) as

$$\mathcal{L}_{\texttt{uns}}(D_u^t)' = \mathcal{L}_{\texttt{uns}}(D_u^t) + \mathbb{E}_{x_u^t \sim D_u^t}\left[\mathbb{1}\Big(\tilde{p}_{x_u^t}^t < \tau\Big) H\Big(p_{\alpha(x_u^t)}^t, q_{x_u^t}\Big)\right]. \tag{7}$$

3.3.2. Testing Phase

We can also treat the testing phase as a pseudo-labeling process, leveraging the benefits of DCP in this context. The key difference is that, for NCM classification of low-confidence samples, we rely on the full exemplar set E^t rather than the labeled dataset of a specific task. Given a test sample u^t, the predicted label q_{u^t} is assigned by

$$q_{u^t} = \begin{cases} \hat{p}_{u^t}^t, & \tilde{p}_{u^t}^t \geq \tau \\ \mathop{\arg\max}_{i=1,\cdots,k} S\Big(f_{u^t}^t, \mu_{E^{t,(i)}}\Big), & \tilde{p}_{u^t}^t < \tau \end{cases}, \tag{8}$$

where k is the number of class observed so far.

3.4. Class-Mean-Anchored Unlabeled Distillation

Finally, we enhance the stability of USP from a distillation perspective. Knowledge distillation (KD) is a common approach in CL that uses frozen models or stored features and probabilities from prior tasks as a "teacher" to guide the active "student" model on the new task. In traditional CL, KD typically focuses on independently distilling information from exemplars (*e.g.*, Eq. (11)), often focusing on labeled samples alone. However, this is insufficient in the SSCL setting, where the forgetting of unlabeled data contributes significantly to catastrophic forgetting. Thus, we introduce **C**lass-mean-anchored **U**nlabeled **D**istillation (**CUD**), which efficiently reuses the class mean features of labeled data computed in Sec. 3.3.1. Denoting the class mean feature matrix as $M = \begin{bmatrix} \mu_{C^{t,(1)}} \\ \vdots \\ \mu_{C^{t,(i)}} \end{bmatrix}$, we define CUD loss as

$$\mathcal{L}_{\texttt{cud}}(D_u^t) = \mathbb{E}_{x_u^t \sim D_u^t}\left[\texttt{KL}\left(\frac{S(f_{x_u^t}^t, M)}{\xi} \,\Big\|\, \frac{S(f_{x_u^t}^{t-1}, M)}{\xi}\right)\right], \tag{9}$$

where ξ is the temperature parameter. CUD distills the combined relationships between labeled and unlabeled data by anchoring unlabeled samples to the stable class mean features derived from labeled data. This encourages the model to develop more reliable and robust representations, effectively enhancing the stability of USP.

Finally, the total loss of USP can be presented as

$$\mathcal{L} = \mathcal{L}_{\texttt{sup}} + \lambda_{\texttt{uns}}\mathcal{L}_{\texttt{uns}}' + \lambda_{\texttt{cl}}\mathcal{L}_{\texttt{cl}} + \lambda_{\texttt{fsr}}\mathcal{L}_{\texttt{fsr}} + \lambda_{\texttt{cud}}\mathcal{L}_{\texttt{cud}}. \tag{10}$$

4. Experiments

4.1. Experimental setting

Dataset. We conduct comparative experiments on CIFAR-10, CIFAR-100, and ImageNet-100 to evaluate our method. CIFAR-10 [26] is a dataset with 10 classes, containing 50,000 training images and 10,000 test images, with each image sized 32 × 32. CIFAR-100 [26] is similar to CIFAR-10, but it contains 100 classes, with each class having 500 training images and 100 test images. ImageNet-100 [44] is a 100-class subset of the ImageNet-1k, with each class containing 1,300 training images and 500 test images. Additionally, we evaluate on a more challenging few-shot SSCL

Table 1. Average and last accuracy on 5-task CIFAR10-X and 10-task CIFAR100-X with X labels per class. We provide comparisons with multiple baseline methods reported in DSGD [17], which use the same baseline SSCL learner as ours. We mark out **the best result**.

Method	CIFAR10-30 Avg	CIFAR10-30 Last	CIFAR10-150 Avg	CIFAR10-150 Last	CIFAR100-20 Avg	CIFAR100-20 Last	CIFAR100-25 Avg	CIFAR100-25 Last	CIFAR100-80 Avg	CIFAR100-80 Last	CIFAR100-125 Avg	CIFAR100-125 Last
iCaRL [38]	34.16	21.84	60.86	53.65	26.43	13.92	28.14	15.29	36.32	19.10	44.14	30.73
DER [52]	40.41	31.48	64.77	61.06	31.01	23.53	32.82	26.53	53.32	41.55	57.21	48.86
CCIC [4]	-	55.20	-	74.30	-	-	-	29.50	-	-	-	44.30
ORDisCo [46]	-	-	74.77	65.91	-	-	-	-	-	-	-	-
NNCSL [23]	-	-	-	-	55.19	43.53	57.45	46.00	67.27	55.35	67.58	56.40
iCaRL&Fix [17]	45.98	30.71	78.36	69.08	45.75	23.40	49.83	31.25	53.46	32.21	56.87	41.38
+ DSGD [17]	77.33	**76.41**	84.14	**79.69**	52.80	35.47	53.42	35.95	57.92	37.81	58.08	43.14
+ USP (Ours)	79.66	70.43	**84.78**	78.21	53.20	41.30	54.36	38.25	58.59	44.20	59.96	43.80
DER&Fix [17]	66.71	61.41	81.10	77.00	51.76	40.86	52.03	44.47	64.03	50.25	66.69	53.57
+ DSGD [17]	75.04	72.59	83.08	79.39	55.63	44.63	57.94	46.68	65.48	55.40	69.14	58.50
+ USP (Ours)	**81.43**	73.65	84.43	77.74	**58.79**	**45.22**	**59.87**	**47.44**	**68.67**	**60.45**	**71.60**	**63.08**

task on the CUB [6] dataset, consisting of 200 bird species with 6,000 training images and 6,000 test images.

Task Settings. For CIFAR-10, CIFAR-100 and ImageNet-100 datasets, we follow DSGD [17], where we sequentially train all 10, 100 and 100 classes in increments of 2, 10 and 10 classes per task, respectively. For CIFAR-10, we use two levels of supervision, with 30 and 150 labeled images per class. For CIFAR-100, we use four levels of supervision, with 20, 25, 80 and 125 labeled images per class. For ImageNet-100, we mainly use two levels of supervision, with 13 and 100 labeled images per class. To simplify notation, we denote the benchmark as *Dataset-X (number of labeled samples per class)*. For example, CIFAR10-30 indicates CIFAR-10 with 30 labeled samples per class. For CUB, we follow UaD-CIE [12], a few-shot SSCL method, where the model trains on 100 classes under full supervision in the first task, and each subsequent task trains 10 classes with SSL settings, including 5 labeled images per class.

Metrics. We adopt the average incremental top-1 accuracy as our primary evaluation metric: $A_{Avg} = \frac{1}{T}\sum_{t=1}^{t} A_t$, where A_t is incremental accuracy for task t, defined as $A_t = \frac{1}{t}\sum_{t=1}^{t} a_{t,i}$ and $a_{t,i}$ is the accuracy on test set of i^{th} task after learning the t^{th} task. Additionally, we report the final model accuracy A_{Last} in the last task as a reference.

Implementation Details. We use ResNet-32 [20] as the feature extractor F for the CIFAR-10 and CIFAR-100 and ResNet-18 [20] for the ImageNet-100 and CUB. For unlabeled data, we apply the data augmentation approach from FixMatch [42]. The projection head P is a simple linear layer outputting 512-dimension features (*i.e.*, $d = 512$), which is consistent with the dimension of the generated ETF vectors across all datasets. All loss weights $\lambda_{\texttt{uns}}$, $\lambda_{\texttt{cl}}$, $\lambda_{\texttt{fsr}}$ and $\lambda_{\texttt{cud}}$ are set to 1, and all temperature parametes β, γ and ξ are set to 0.1. Following [42], the confidence threshold τ is set to 0.95. We utilize the SGD optimizer with a momentum of 0.9 and the weight decay is set to 10^{-5}. For SSCL tasks, we use a memory buffer of size 5120, set the batch size to 64, and use a learning rate of 0.03. We train for

Table 2. Comparisons on 10-task ImageNet-100. We re-run NNCSL under our setting for a direct comparison. See supplementary Sec. B.1.1 for results under the original NNCSL protocol.

Method	ImageNet100-13 Avg	ImageNet100-13 Last	ImageNet100-100 Avg	ImageNet100-100 Last
iCaRL [38]	19.89	12.88	30.78	16.68
NNCSL [23]	42.19	33.64	56.78	53.84
iCaRL&Fix [17]	26.37	15.58	37.49	21.02
+ DSGD [17]	28.35	19.14	50.53	32.10
+ USP (Ours)	43.91	35.40	56.84	50.36
DER&Fix [17]	35.40	29.22	61.96	52.91
+ DSGD [17]	35.73	31.53	62.27	52.82
+ USP (Ours)	**46.09**	**39.58**	**62.29**	**55.01**

200 epochs with a 10-epoch warm-up followed by a cosine scheduler to reduce the learning rate. For few-shot SSCL, we follow the settings of UaD-CIE [12] for the training parameters (see Sec. A.2 of supplementary for details).

Baselines. For SSCL tasks, we primarily consider published SSCL methods, including the previous SOTA methods: DSGD [17] and NNCSL [23], as well as classic methods like CCIC [4] and ORDisCo [46]. Our main competitors is DSGD [17]. For a fair comparison, we adopt the same base SSCL learners (*i.e.*, iCaRL&FixMatch and DER&FixMatch). Additionally, we consider converting traditional fully-supervised CL methods to SSCL setting for reference. Specifically, this adaptation involves using only labeled data during training and discarding all unlabeled data. We apply this approach to classic methods like iCaRL [38] and DER [52]. For few-shot SSCL task, we primarily compare against the previous SOTA method, UaD-CIE [12], and include several baselines used in its original paper.

4.2. Main Results

CIFAR-10 and CIFAR-100. We first report the results of different methods on CIFAR-10 and CIFAR-100 in Tab. 1. USP achieves the best performance across nearly all settings, showing even more pronounced advantages on the more challenging CIFAR-100. Both USP and DSGD are

Table 3. Performance comparisons on 11-task CUB. We provide the test accuracy on different tasks and average accuracy. We replace UaD-CIE's uncertainty-based distillation with CUD and incorporate FSR and CUP into its training pipeline. For fairness, we use the official code of UaD-CIE to build USP on top of it and report the results based on our re-run of UaD-CIE.

Method	Task ID											Avg
	1	2	3	4	5	6	7	8	9	10	11	
SS-iCaRL [10]	69.89	61.24	55.81	50.99	48.18	46.91	43.99	39.78	37.50	34.54	31.33	47.29
SS-NCM [10]	69.89	61.91	55.51	51.71	49.68	46.11	42.19	39.03	37.50	34.54	31.33	47.33
SS-NCM-CNN [10]	69.89	64.87	59.82	55.14	52.48	49.60	47.87	45.10	40.47	38.10	35.25	50.78
Semi-SPPR [62]	68.44	61.66	57.11	53.41	50.15	46.68	44.93	43.21	40.61	39.21	37.43	49.34
Semi-CEC [55]	75.82	71.91	68.52	63.53	62.45	58.27	57.62	55.81	54.85	53.52	52.26	61.32
Us-KD [11]	74.69	71.71	69.04	65.08	63.60	60.96	59.06	58.68	57.01	56.41	55.54	62.89
UaD-CIE [12]	75.87	73.05	69.50	65.61	64.37	61.84	61.49	58.93	56.95	56.21	55.66	64.33
+ USP (Ours)	**78.21**	**74.48**	**71.81**	**68.16**	**67.58**	**64.77**	**62.87**	**62.27**	**59.97**	**60.10**	**60.55**	**66.43**

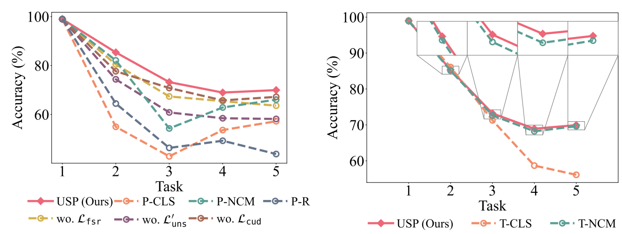
Table 4. Base and novel classes accuracy on 11-task CUB. Base classes refer to the classes used for fully supervised training on the first task, while novel classes refer to non-base classes trained on in subsequent tasks.

Method	Classes	Task ID											Avg
		1	2	3	4	5	6	7	8	9	10	11	
SS-iCaRL [10]	Base	69.89	62.32	60.62	58.99	58.59	57.77	59.88	56.21	54.46	50.54	46.11	57.76
	Novel	-	53.22	32.38	24.07	22.76	23.34	17.58	16.40	16.39	16.13	16.32	23.86
SS-NCM-CNN [10]	Base	69.89	65.80	64.97	63.79	63.81	61.08	65.24	63.73	58.77	55.74	51.88	62.24
	Novel	-	56.37	34.70	26.03	24.04	24.68	19.14	18.60	17.70	17.79	18.36	25.74
UaD-CIE [12]	Base	75.87	74.58	74.09	**73.46**	72.24	**71.68**	**71.33**	**70.50**	**70.15**	**69.27**	**69.13**	**72.03**
	Novel	-	57.35	46.29	39.58	45.02	42.54	45.37	42.75	40.82	41.98	42.49	44.42
+ USP (Ours)	Base	**78.21**	**75.00**	**74.13**	73.36	**72.42**	70.74	68.99	68.44	67.25	66.62	66.66	71.07
	Novel	-	**69.18**	**60.07**	**50.93**	**55.67**	**53.08**	**52.86**	**53.64**	**51.07**	**53.00**	**54.57**	**55.41**

based on distillation, but on CIFAR-10/-100, USP achieves an approximately 1-6% higher A_{Avg} and A_{Last} than DSGD using the same SSCL learner. These results indicate that USP is overall more robust than DSGD, specifically demonstrating a significant advantage in stability.

ImageNet-100. The results on ImageNet-100 are shown in Tab. 2, where it can be seen that USP significantly outperforms DSGD in both average accuracy and final task accuracy. As with the results on CIFAR, USP demonstrates a more pronounced advantage when the amount of labeled data is smaller, indicating higher efficiency in utilizing unlabeled data. Notably, compared to DSGD, USP shows a smaller difference between the average and the final accuracy under the same task settings, suggesting that USP achieves more stable performance across different training task and is better at mitigating forgetting than DSGD.

CUB. The results on CUB are shown in Tab. 3. We conduct experiments on CUB following few-shot SSCL setting, comparing USP with the sota method UaD-CIE. USP can effectively improve the performance of UaD-CIE. To further illustrate USP's effectiveness, we report base class and novel class accuracies across different training task, in Tab. 4. We observe that our method shows only a slight drop in base class accuracy compared to UaD-CIE (0.96%), yet achieves a substantial improvement in novel class accuracy (10.99%). UaD-CIE mitigates forgetting in base classes by applying uncertainty-based loss weighting to corresponding



(a) Training phase (b) Testing phase

Figure 4. Ablation studies on the main components of USP. The experiments are conducted on CIFAR-10 with 30 labels per class.

labeled samples. In contrast, USP leverages unlabeled samples and employs CUD distillation strategy to support old class learning. This effective distillation allows us to further enhance novel class learning through our DCP.

In addition to the standard settings, more realistic SSCL settings are provided in Sec. B.1 of supplementary.

4.3. Ablation Analysis and Discussions

We perform extensive ablation studies on the components and training strategies of USP. By default, our experiments are all based on iCaRL&Fix. More ablations on distillation methods (Sec. B.2.2), hyper-parameters (Sec. B.2.3), backbones/pre-training (Sec. B.2.4) and different memory buffer sizes (Sec. B.3) can be found in the supplementary.

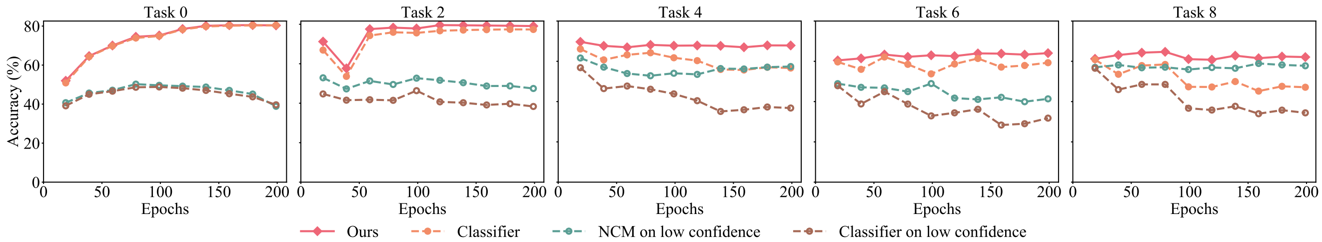


Figure 5. Pseudo-label accuracy of different pseudo-labeling strategies on CIFAR100-20. The y-axis represents the pseudo-label accuracy, and the x-axis represents the number of training epochs. We present the results for different training tasks.

Ablation on Method Components. We conduct ablation experiments on the main components of USP. With the basic SSCL learner training objective, USP incorporates the ETF-based feature space reservation (FSR) loss $\mathcal{L}_{\texttt{fsr}}$, the unsupervised training loss $\mathcal{L}'_{\texttt{uns}}$ with our divide and conquer pseudo-labeling (DCP) strategy, and the class-mean-anchored unlabeled distillation (CUD) loss $\mathcal{L}_{\texttt{cud}}$. To examine the effects of these modules, we ablate these losses, which is denoted as "wo. $\mathcal{L}_{\texttt{fsr}}$", "wo. $\mathcal{L}'_{\texttt{uns}}$" and "wo. $\mathcal{L}_{\texttt{cud}}$", respectively. As shown in Fig. 4a, each component of USP contributes to improved model performance, with the DCP strategy having the most substantial impact on USP performance. To further investigate this, we conduct additional experiments by replacing DCP with alternative approaches: a classifier-based pseudo-labeling strategy ("P-CLS"), an NCM-based pseudo-labeling strategy ("P-NCM"), and a hybrid strategy where high-confidence samples use NCM pseudo-labels while low-confidence samples use classifier pseudo-labels ("P-R"). All these replacements lead to varying degrees of performance degradation, with P-R showing the most substantial decline, further underscoring the rationale behind our DCP. In Fig. 5, we further experiment to illustrate the reasons for its effectiveness. Additionally, we perform an ablation study on DCP used in testing phase. Fig. 4b presents results for classifier-only inference ("T-CLS") and NCM-only inference ("T-NCM") in isolation. Our DCP again achieves the best performance. While the improvement over NCM alone is not substantial, our strategy is a "free lunch" benefit.

Effect of Pseudo-Labeling Strategy. To demonstrate the effectiveness of DCP, we present the pseudo-label accuracy across different tasks on CIFAR100-20 in Fig. 5, comparing DCP with the traditional classifier pseudo-labeling strategy. Considering DCP fully utilizes unlabeled data, for a direct comparison of pseudo-label quality, we do not apply a confidence threshold to the classifier pseudo-labeling approach (see Tab. 7 of supplementary for more comparison with thresholded pseudo-labeling). In the first task, our accuracy is comparable to the classifier's. However, as training progresses, DCP significantly outperforms the classifier, with an increasing advantage in later tasks.

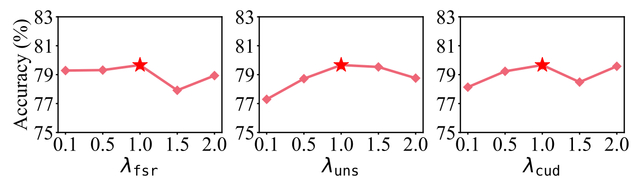


Figure 6. Average accuracy on CIFAR10-30 with different values of each loss weight (*i.e.*, $\lambda_{\texttt{fsr}}$, $\lambda_{\texttt{uns}}$ and $\lambda_{\texttt{cud}}$).

During incremental training, the classifier faces stability-plasticity trade-offs, resulting in performance degradation. In contrast, the NCM classifier, leveraging feature similarity, is less affected by incremental training and achieves higher accuracy on low-confidence samples. DCP effectively combines the strengths of both methods, enhancing overall pseudo-label accuracy. To illustrate this, we report the pseudo-label accuracy of both approaches on low-confidence samples (confidence $< \tau$), where the NCM's higher accuracy affirms the rationale behind DCP.

Loss Weights. We traverse the values of $\{0.1, 0.5, 1, 1.5, 2\}$ to separately set the each weights (*i.e.*, $\lambda_{\texttt{fsr}}$, $\lambda_{\texttt{uns}}$, and $\lambda_{\texttt{cud}}$) while keeping the other two weights fixed at 1.0. We report the average incremental accuracy of the model on CIFAR10-30 under different weight coefficients, as shown in Fig. 6. The model achieves the best performance when the weights are set to 1.0, and we observe that the model's final performance fluctuates within 2% across different weight coefficients, indicating that our method is not sensitive to the choice of weight coefficients.

5. Conclusion

We propose a divide-and-conquer SSCL framework called USP, comprising three main components: (1) ETF-based feature space reservation (FSR) strategy for learning plasticity; (2) divide-and-conquer pseudo-labeling (DCP) approach for unlabeled learning; and (3) class-mean-anchored unlabeled distillation (CUD) for memory stability, which are designed to synergistically enhance the SSCL model. In future work, we plan to adapt our USP into more SSCL paradigms to further contribute to the community.

Acknowledgments

Yue Duan, Taicai Chen and Yinghuan Shi are with the National Key Laboratory for Novel Software Technology and the National Institute of Healthcare Data Science, Nanjing University. Lei Qi is with the School of Computer Science and Engineering, Southeast University. This work is supported by NSFC Project (624B2063, 62222604, 62206052), China Postdoctoral Science Foundation (2024M750424), Fundamental Research Funds for the Central Universities (020214380120, 020214380128), State Key Laboratory Fund (ZZKT2024A14, ZZKT2025B05), Postdoctoral Fellowship Program of CPSF (GZC20240252), Jiangsu Funding Program for Excellent Postdoctoral Talent (2024ZB242) and Jiangsu Science and Technology Major Project (BG2024031).

References

[1] Nader Asadi, MohammadReza Davari, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Prototype-sample relation distillation: towards replay-free continual learning. In *International Conference on Machine Learning*, 2023. 2

[2] Benedikt Bagus, Alexander Gepperth, and Timothée Lesort. Beyond supervised continual learning: a review. *arXiv preprint arXiv:2208.14307*, 2022. 1

[3] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *IEEE/CVF Conference on Computer Cision and Pattern Recognition*, 2021. 2

[4] Matteo Boschini, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Continual semi-supervised learning through contrastive interpolation consistency. *Pattern Recognition Letters*, 162:9–14, 2022. 3, 6

[5] Dhanajit Brahma, Vinay Kumar Verma, and Piyush Rai. Hypernetworks for continual semi-supervised learning. *arXiv preprint arXiv:2110.01856*, 2021. 1, 3

[6] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2018. 6

[7] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In *International Conference on Learning Representations*, 2023. 2

[8] Taicai Chen, Yue Duan, Dong Li, Lei Qi, Yinghuan Shi, and Yang Gao. Pg-lbo: Enhancing high-dimensional bayesian optimization with pseudo-label and gaussian process guidance. In *AAAI Conference on Artificial Intelligence*, 2024. 2

[9] Yuhao Chen, Xin Tan, Borui Zhao, Zhaowei Chen, Renjie Song, Jiajun Liang, and Xuequan Lu. Boosting semi-supervised learning by exploiting all unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[10] Yawen Cui, Wuti Xiong, Mohammad Tavakolian, and Li Liu. Semi-supervised few-shot class-incremental learning. In *IEEE International Conference on Image Processing*, 2021. 7

[11] Yawen Cui, Wanxia Deng, Xin Xu, Zhen Liu, Zhong Liu, Matti Pietikäinen, and Li Liu. Uncertainty-guided semi-supervised few-shot class-incremental learning with knowledge distillation. *IEEE Transactions on Multimedia*, 25: 6422–6435, 2022. 7

[12] Yawen Cui, Wanxia Deng, Haoyu Chen, and Li Liu. Uncertainty-aware distillation for semi-supervised few-shot class-incremental learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 1, 4, 6, 7

[13] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *IEEE/CVF International Conference on Computer Vision*, 2021. 2

[14] Yue Duan, Zhen Zhao, Lei Qi, Luping Zhou, Lei Wang, and Yinghuan Shi. Towards semi-supervised learning with non-random missing labels. In *IEEE/CVF International Conference on Computer Vision*, 2023. 2

[15] Yue Duan, Zhen Zhao, Lei Qi, Lei Wang, Luping Zhou, Yinghuan Shi, and Yang Gao. Mutexmatch: Semi-supervised learning with mutex-based consistency regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):8441–8455, 2024. 2

[16] Yue Duan, Zhen Zhao, Lei Qi, Luping Zhou, Lei Wang, and Yinghuan Shi. Roll with the punches: Expansion and shrinkage of soft label selection for semi-supervised fine-grained learning. In *AAAI Conference on Artificial Intelligence*, 2024. 2

[17] Yan Fan, Yu Wang, Pengfei Zhu, and Qinghua Hu. Dynamic sub-graph distillation for robust semi-supervised continual learning. In *AAAI Conference on Artificial Intelligence*, 2024. 1, 2, 3, 4, 6

[18] Rui Gao and Weiwei Liu. Ddgr: Continual learning with deep diffusion-based generative replay. In *International Conference on Machine Learning*, 2023. 2

[19] Heitor Murilo Gomes, Maciej Grzenda, Rodrigo Mello, Jesse Read, Minh Huong Le Nguyen, and Albert Bifet. A survey on semi-supervised learning for delayed partially labelled data streams. *ACM Computing Surveys*, 55(4):1–42, 2022. 1

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6

[21] Quentin Jodelet, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. Class-incremental learning using diffusion model for distillation and replay. In *IEEE/CVF International Conference on Computer Vision*, 2023. 2

[22] Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[23] Zhiqi Kang, Enrico Fini, Moin Nabi, Elisa Ricci, and Karteek Alahari. A soft nearest-neighbor framework for con-

tinual semi-supervised learning. In *IEEE/CVF International Conference on Computer Vision*, 2023. 6

[24] Sanghwan Kim, Lorenzo Noci, Antonio Orvieto, and Thomas Hofmann. Achieving a better stability-plasticity trade-off via auxiliary networks in continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1

[25] Yajing Kong, Liu Liu, Zhen Wang, and Dacheng Tao. Balancing stability and plasticity through advanced null space in continual learning. In *European Conference on Computer Vision*, 2022. 1

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[27] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, International Conference on Machine Learning*, 2013. 2

[28] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, 2019. 2

[29] Yu Liu, Xiaopeng Hong, Xiaoyu Tao, Songlin Dong, Jingang Shi, and Yihong Gong. Model behavior preserving for class-incremental learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10):7529–7540, 2022. 2

[30] Yichen Lu, Mei Wang, and Weihong Deng. Augmented geometric distillation for data-free incremental person reid. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[31] Yan Luo, Yongkang Wong, Mohan Kankanhalli, and Qi Zhao. Learning to predict gradients for semi-supervised continual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 1, 3

[32] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *European Conference on Computer Vision*, 2012. 4

[33] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018. 2

[34] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020. 1

[35] Jaeyoo Park, Minsoo Kang, and Bohyung Han. Class-incremental learning for action recognition in videos. In *IEEE/CVF International Conference on Computer Vision*, 2021. 2

[36] Federico Pernici, Matteo Bruni, Claudio Baecchi, Francesco Turchini, and Alberto Del Bimbo. Class-incremental learning with pre-allocated fixed classifiers. In *International Conference on Pattern Recognition*, 2021. 4

[37] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023. 2

[38] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 4, 6

[39] James Smith, Jonathan Balloch, Yen-Chang Hsu, and Zsolt Kira. Memory-efficient semi-supervised continual learning: The world is its own replay buffer. In *International Joint Conference on Neural Networks*, 2021. 1, 2, 3

[40] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *IEEE/CVF International Conference on Computer Vision*, 2021. 2

[41] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[42] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 2020. 2, 3, 6

[43] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 2017. 2

[44] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, 2020. 5

[45] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *European Conference on Computer Vision*, 2022. 2

[46] Liyuan Wang, Kuo Yang, Chongxuan Li, Lanqing Hong, Zhenguo Li, and Jun Zhu. Ordisco: Effective and efficient usage of incremental unlabeled data for semi-supervised continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 3, 6

[47] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[48] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. In *International Conference on Learning Representations*, 2022. 2

[49] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[50] Buddhi Wickramasinghe, Gobinda Saha, and Kaushik Roy. Continual learning: A review of techniques, challenges and future directions. *IEEE Transactions on Artificial Intelligence*, 2023. 1

[51] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[52] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3, 6

[53] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2022. 1

[54] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *International Conference on Learning Representations*, 2022. 4

[55] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 7

[56] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020. 2

[57] Hanbin Zhao, Hui Wang, Yongjian Fu, Fei Wu, and Xi Li. Memory-efficient class-incremental learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5966–5977, 2021. 2

[58] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[59] Mingkai Zheng, Shan You, Lang Huang, Chen Luo, Fei Wang, Chen Qian, and Chang Xu. Simmatchv2: Semi-supervised learning with graph consistency. In *IEEE/CVF International Conference on Computer Vision*, 2023. 2

[60] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 4

[61] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 2

[62] Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-promoted prototype refinement for few-shot class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 7