On the Geometry of Forgetting in Continual Self-Supervised Learning

Arjan Chakravarthy ARJAN_CHAKRAVARTHY@BROWN.EDU
Department of Computer Science
Brown University
Providence, RI 02912, USA

Abstract

Catastrophic forgetting in continual learning has been extensively studied in supervised settings, yet its formulation in self-supervised learning (SSL) remains poorly understood. In this work, we investigate catastrophic forgetting in continual SSL from an explicitly geometric perspective, moving beyond downstream linear probe accuracy to characterize the evolution of the representation space directly. We evaluate three representative SSL methods under a Class-Incremental Learning protocol on CIFAR-100, and introduce a suite of geometric diagnostic tools, covariance decomposition, Grassmannian subspace analysis, and gradient alignment analysis, all of which provide insights into specific changes in the model's embedding space. Our results reveal that forgetting in SSL is not a uniform representational collapse but a geometrically specific phenomenon. Inter-class discriminative structure degrades faster than within-sample invariance across all methods, SSL gradient updates at task boundaries are structurally biased toward overwriting precisely this inter-class geometry, and task-specific subspaces drift substantially from their initial configurations with no mechanism to recover them. We further show that experience replay improves downstream accuracy, but very slightly addresses the important geometric objectives necessary to mitigate forgetting. These findings inform the development of novel methods that specifically preserve the embedding space geometry.

1 Introduction

Over the last few years, self-supervised Learning (SSL) has emerged as a new field, aimed at producing robust representations of data. Unlike traditional supervised methods, SSL models train without access to a datapoint's label. Instead, the training paradigm involves tasks where the supervision is derived from the structure of the data itself. There exist various approaches to construct strong representations, often rivaling accuracy on supervised models [30, 9, 4, 7, 3, 16, 10, 12], while also remaining more generalizable. Ultimately, these models learn intrinsic geometric properties of the underlying embedding space manifold, yielding features that generalize better when transferred to downstream tasks.

Separately, Continual Learning (CL) is a problem where the primary objective is to enable models to learn from a non-stationary stream of data. Unlike standard machine learning, which assumes that data is independent and identically distributed (i.i.d) and all data is presented at once, CL systems must be robust when data may not be i.i.d and are presented sequentially. This adaptation poses challenges with the risk of catastrophic forgetting, where the optimization of the model's parameters for a current task T_n significantly degrades the performance on a previous task T_{n-1}.

Evaluating models in the CL setup typically can be categorized into three distinct scenarios based on the availability of task-specific information at inference time: Task-Incremental, Domain-Incremental, and Class-Incremental Learning. In Task-Incremental Learning (Task-IL), the model is provided with a "task-id" at both training and testing, allowing it to utilize task-specific output heads or isolated parameter sets. Specifically, during evaluation, the model only has to distinguish classes within a single task. Domain-Incremental Learning (Domain-IL) occurs when the high-level task remains constant, but the input distribution undergoes a shift. Finally, Class-Incremental Learning (Class-IL) is a setting where the model must distinguish between all classes encountered across the entire task history without any task-related information at test time. This setting is particularly challenging for SSL, as the model must integrate new categories into a unified embedding space while preventing the features of previous classes from collapsing or being overwritten. While extensive research has addressed catastrophic forgetting in supervised learning, the dynamics of forgetting in Self-Supervised Learning remain less understood. In supervised settings, forgetting is often attributed to the final classification layers [20]. However, in SSL, the absence of explicit labels means that the entire geometric structure of the embedding space is subject to drift or collapse. As the model optimizes for data presented in the current task, latent features essential for characterizing previous data distributions may drift from their initial positions or be overwritten entirely, leading to a loss of representational stability.

This work seeks to approach SSL in the CL setup from a geometric perspective, investigating the different phenomena that could be responsible for catastrophic forgetting in SSL models. By evaluating different models in a Class-Incremental setup, we investigate how different collapse-prevention mechanisms influence the model's susceptibility to catastrophic forgetting. Beyond standard accuracy metrics, we explore different diagnostic tools such as the covariance decomposition, Grassmannian distances, and gradient projection norms to characterize the geometric evolution of the embedding space. Our findings aim to identify which mathematical constraints best preserve a robust and separable manifold throughout this non-stationary training regime.

2 Related Work

2.1 Self-Supervised Learning Models

Recent SSL methods have demonstrated strong representational performance, often approaching or matching supervised models [9, 30, 4, 16, 6, 7, 3]. The dominant paradigm across these methods involves generating two augmented views of an input image and training a network to produce consistent representations across those views. This encourages models to generate latent representations that are similar for augmented samples of the same image. To prevent representational collapse, different methods have adopted fundamentally distinct strategies. Contrastive methods such as SimCLR [9] and MoCo [17] rely on sample discrimination, explicitly pushing apart representations of different images (negatives) while pulling together those of augmented views of the same image. While effective, these approaches typically require large batch sizes to maintain a sufficient pool of negatives. Negative-free methods have since addressed this limitation. BYOL [16] and SimSiam [10] adapt the model architecture with stop-gradient operations, while Barlow Twins [30]

and VICReg [4] impose information-theoretic constraints directly on the feature dimensions, decorrelating or explicitly regularizing the variance and covariance of the embedding space. Clustering-based methods such as SwAV [6], DeepCluster [5], and DINO [7] take a different approach, comparing views through a cross-entropy loss using cluster prototypes or self-distillation targets as a proxy. Despite this large corpus of work dedicated to creating these robust latent representations, understanding how these models act in the CL setting, is still largely underexplored.

2.2 Continual Learning in Supervised Settings

A substantial body of work has developed methods to counteract catastrophic forgetting in supervised settings [18, 31, 19, 21, 8, 26, 23, 27]. These methods can be broadly organized into two categories. Regularization-based methods penalize changes to parameter updates deemed important for previous tasks. Weight-regularization approaches such as EWC [18] and SI [31] operate by estimating parameter importance and constraining updates accordingly. However, these methods can be brittle for deep networks where the mapping from weights to function is highly nonlinear. Functional regularization approaches [24, 29] address this more directly by constraining the network's outputs at specific input points. In particular, these methods try to ensure that the network's outputs do not significantly change as the model trains on new tasks. Replay-based methods [26, 21, 8] either store a subset of past data in an episodic memory buffer and interleave it with current training, or train generative models to create "examples" from previous tasks. While these methods have demonstrated success in supervised settings, the majority of these rely on labels to identify what should be preserved, limiting their direct applicability to SSL.

2.3 Continual Self-Supervised Learning

The intersection of continual learning and SSL has only recently begun to receive dedicated attention. Early work on unsupervised continual learning [25, 1] established foundational ideas but was largely limited to simple datasets such as MNIST, and the proposed methods do not scale to more complex visual benchmarks. More recent work has explored SSL pretraining as an initialization strategy for supervised continual learning [11], leveraging the transferability of self-supervised representations rather than addressing forgetting within the SSL training process itself. Two concurrent works most directly relevant to the CSSL problem are Projected Functional Regularization [15] and Lifelong Unsupervised Mixup [22]. The former extends supervised contrastive continual learning to the unsupervised setting, but is specifically tailored to contrastive SSL methods and does not generalize readily to non-contrastive paradigms. The latter is similarly constrained, demonstrating results only at small scale and within the class-incremental setting with a limited selection of SSL methods. Recent methods such as CaSSLe [13] and Kaizen [28] proposing distillation-based approaches that operate across multiple SSL paradigms, using feature extractors to prevent forgetting. However, these works primarily assess forgetting through downstream linear probe accuracy, and do not investigate the geometric evolution of the embedding space itself. This geometric perspective, and its relationship to the collapse-prevention mechanisms intrinsic to different SSL methods, is the focus of this work.

3 Preliminaries

In this section, we formalize the Continual Learning (CL) problem setting and outline the mathematical foundations of the three Self-Supervised Learning (SSL) paradigms evaluated in this work, namely SimCLR, Barlow Twins, and VICReg.

3.1 Continual Learning Formulation

We consider a task-incremental data stream defined as a sequence of T tasks, $\mathcal{S} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T\}$. Each task $t \in \{1, \ldots, T\}$ consists of a dataset $\mathcal{D}_t = \{x_i^{(t)}\}_{i=1}^{N_t}$ drawn from a task-specific underlying distribution. While the underlying data may belong to a specific set of classes Y_t, in the unsupervised setting, the model does not have access to these labels during training. In the Class-Incremental Learning (Class-IL) setting, the label spaces of different tasks are disjoint such that $Y_i \cap Y_j = \emptyset$ for $i \neq j$ [2]. During training on task t, the model only has access to data from $\mathcal{D}_t$, and historical data $\mathcal{D}_{1:t-1}$ is strictly unavailable. Furthermore, at inference time, the model is not provided with a task identifier, preventing information leakage. The objective is to continually update the parameters θ of a feature encoder $f_\theta : \mathcal{X} \to \mathbb{R}^d$ such that it maintains a robust and discriminative representation space capable of differentiating all seen distributions without suffering from catastrophic forgetting.

3.2 Self-Supervised Learning Frameworks

Modern joint-embedding architectures for SSL follow a common pre-training pipeline. Given an input image $x \in \mathcal{D}_t$, an augmentation pipeline is applied to generate two correlated views, x^A and x^B. These views are processed by an encoder f_θ (typically a ResNet or ViT) to produce feature representations $h^A = f_\theta(x^A)$ and $h^B = f_\theta(x^B)$. A projection head g_ϕ subsequently maps these features into a lower-dimensional latent space to yield embeddings $z^A = g_\phi(h^A)$ and $z^B = g_\phi(h^B)$.The network is optimized by minimizing a specific self-supervised objective dependent on these embeddings $\mathcal{L}_{SSL}(z^A, z^B)$. The primary distinction between SSL methodologies lies in how this objective prevents representational collapse.

3.2.1 Barlow Twins

Barlow Twins [30] operates directly on the empirical cross-correlation matrix $\mathcal{C}$ computed between the batch embeddings Z^A and Z^B along the feature dimension. The matrix elements are defined as:

$$\mathcal{C}_{ij} = \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b (z^A_{b,i})^2} \sqrt{\sum_b (z^B_{b,j})^2}}$$

where b indexes batch samples, and i, j index the vector dimensions. The Barlow Twins loss function forces this cross-correlation matrix to approximate the identity matrix:

$$\mathcal{L}_{BT} \triangleq \sum_i (1 - \mathcal{C}_{ii})^2 + \lambda \sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2$$

The first term (invariance) aligns the diagonal elements to 1, ensuring representations are invariant to augmentations. The second term (redundancy reduction) pushes off-diagonal

elements to 0, decorrelating the feature dimensions to maximize the information content of the embedding space.

3.2.2 SimCLR

Unlike Barlow Twins, SimCLR [9] prevents collapse through explicit negative sampling. It optimizes an InfoNCE loss, which maximizes the similarity between the positive pair (z^A, z^B) while minimizing the similarity between z^A and all other "negative" samples in the current mini-batch. The objective is defined as:

$$\mathcal{L}_{SimCLR} = -\frac{1}{2N}\sum_{i=1}^{2N}\log\frac{\exp(\text{sim}(z_i^A, z_i^B)/\tau)}{\sum_{j=1}^{2N}\mathbb{1}_{[j\neq i]}\exp(\text{sim}(z_i, z_j)/\tau)}$$

where N is the batch size, $\text{sim}(\cdot,\cdot)$ denotes cosine similarity, τ is a temperature scaling parameter, and the denominator sums over all $2N-1$ negative pairs in the augmented batch.

3.2.3 VICReg

VICReg's objective [4] explicitly focuses on decomposing the collapse-prevention objective into three distinct regularization terms applied to the embeddings $Z = [Z^A, Z^B]$.The Invariance term is the mean-squared error between positive pairs:

$$s(Z^A, Z^B) = \frac{1}{N}\sum_i ||z_i^A - z_i^B||_2^2$$

The Variance term prevents dimension collapse by using a hinge loss to maintain the standard deviation of each feature dimension above a threshold γ:

$$v(Z) = \frac{1}{d}\sum_{j=1}^{d}\max(0, \gamma - \text{S}(z^j, \epsilon))$$

where S is the standard deviation.The Covariance term prevents informational redundancy by penalizing the off-diagonal coefficients of the covariance matrix of Z, similar to Barlow Twins:

$$c(Z) = \frac{1}{d}\sum_{i\neq j}(\text{Cov}(Z)_{i,j})^2$$

The overall VICReg objective is a weighted sum of these three components:

$$\mathcal{L}_{VICReg} = \lambda s(Z^A, Z^B) + \mu\left[v(Z^A) + v(Z^B)\right] + \nu\left[c(Z^A) + c(Z^B)\right]$$

where λ, μ, and ν are hyperparameters balancing the objective.

4 Methodology

4.1 Models and Training

We evaluate three SSL frameworks that represent the dominant paradigms for collapse prevention: Barlow Twins [30], SimCLR [9], and VICReg [4]. These methods are chosen deliberately to span the landscape of collapse-prevention strategies. SimCLR operates

via explicit contrastive negative sampling, while Barlow Twins and VICReg are negative-free methods that instead impose information-theoretic constraints directly on the feature dimensions. Within the negative-free family, Barlow Twins acts on the cross-correlation matrix of paired embeddings, whereas VICReg decomposes the objective into three explicit regularization terms targeting variance, invariance, and covariance. This diversity of mechanisms allows us to assess whether susceptibility to catastrophic forgetting is a general property of SSL under non-stationary data, or whether it is mediated by the specific form of the collapse-prevention constraint.

For the purpose of our exploration, all models share a common backbone (ResNet-18 encoder) f_θ modified for low-resolution inputs by replacing the initial 7×7 convolution with a 3×3 kernel and removing the first max-pooling layer, following standard practice for CIFAR-scale images. The backbone produces 512-dimensional feature representations. These are projected by a non-linear multi-layer perceptron (MLP) head g_ϕ mapping to a 2048-dimensional embedding space, in which all SSL loss functions are computed. Following standard SSL pre-training protocol, the projection head is discarded after training and downstream evaluations are performed on the 512-dimensional backbone representations.

Each model is trained sequentially across tasks using the Adam optimizer with a cosine annealing learning rate schedule. Training hyperparameters for each method follow the settings from their respective original papers, scaled appropriately for CIFAR-resolution inputs. We also evaluate each model with a random replay buffer and analyze the corresponding geometric landscape of the model's embedding space on that setting. Model weights are saved after completing each task, enabling retrospective evaluation of the encoder's representation quality at each stage of the training sequence.

4.2 Datasets and Task Construction

To simulate a continual learning environment, we use the CIFAR-100 benchmark, partitioned into sequences of disjoint tasks consistent with the Class-Incremental Learning (Class-IL) protocol described in Section 3.1. The hundred classes are partitioned into five tasks of twenty classes each.

At each task boundary, the model loses access to all data from previous tasks and is trained exclusively on the current task's data. No task identifier is provided at inference time. Within each task, images are not presented with their class labels and the only supervision signal available to the model is derived from the structure of the SSL objective itself.

For data augmentation, we follow the standard SSL protocol used in the original implementations of each method. Each input image $x \in \mathcal{D}_t$ is processed by an augmentation pipeline to generate two correlated views x^A and x^B, consisting of random cropping with resizing, horizontal flipping, color jitter (adjusting brightness, contrast, saturation, and hue), and random grayscale conversion. This augmentation policy is held constant across all methods to ensure comparability.

4.3 Evaluation Protocol

We adopt an evaluation protocol tailored to the Class-Incremental Learning (Class-IL) setting, where the model must maintain a unified representation space across all previously encountered classes without access to task identity at inference time.

Following each task $t \in 1, \ldots, T$, the model θ_t is frozen and evaluated using a linear probing protocol. A single linear classifier is trained on top of the learned representations using labeled data from all classes observed up to task t, i.e., $\bigcup_{\tau=1}^{t} Y_\tau$. The resulting accuracy measures the extent to which the representation space remains globally discriminative as new classes are introduced.

To assess the impact of continual updates on previously learned knowledge, we additionally report accuracy restricted to subsets of data corresponding to earlier tasks. Concretely, after training the classifier at step t, we evaluate its performance on samples belonging to each previously observed task $\tau < t$. Importantly, this evaluation relies on a single shared classifier over the cumulative label space and does not assume access to task identity at test time.

Beyond linear probe accuracy, our primary evaluation tools are the geometric diagnostic measures described in Section 5, which directly characterize the structural evolution of the embedding space rather than relying solely on downstream label-based proxies.

5 Geometric Diagnostic Framework

A central thesis of this work is that the standard evaluation of SSL models in the continual learning setting (which measures downstream linear probe accuracy) provides an incomplete picture of what is happening geometrically to the learned representation space as tasks accumulate. A model may retain reasonable classification accuracy while its underlying feature geometry has degraded significantly, or conversely, may exhibit accuracy degradation that does not fully reflect the extent of structural collapse. To address this, we develop a suite of geometric diagnostic tools that operate directly on the encoder's representation space, without reference to labels. The three complementary analyses described below each illuminate a distinct aspect of the embedding geometry and its evolution under task shifts.

5.1 Covariance Decomposition

The covariance structure of the encoder's representation space encodes fundamental information about how the model distributes features across the data manifold. Following the framework of Gamba et al. [14], we decompose the full representation covariance Σ into two geometrically interpretable components that separately measure within-sample and between-sample variation.

Let $f_\theta : \mathcal{X} \to \mathbb{R}^d$ denote the encoder and, for a dataset $\mathcal{D} = \{x_i\}_{i=1}^n$, let $\{x_i^1, \ldots, x_i^m\}$ denote the m augmented views generated from base sample x_i. The full representation covariance is

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \left(f_\theta(x_i^j) - \bar{f}_\theta \right) \left(f_\theta(x_i^j) - \bar{f}_\theta \right)^T \tag{1}$$

where $\bar{f}_\theta = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m} f_\theta(x_i^j)$ is the global mean representation. Applying the law of total covariance and a local first-order linearization of f_θ around each sample centroid, this quantity admits the decomposition

$$\Sigma = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \nabla_x f_\theta(\bar{x}_i)\, \Sigma^{(i)}\, \nabla_x f_\theta(\bar{x}_i)^T}_{\Sigma_{\text{intra}}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n} \left(f_\theta(\bar{x}_i) - \bar{f}_\theta\right)\left(f_\theta(\bar{x}_i) - \bar{f}_\theta\right)^T}_{\Sigma_{\text{inter}}} = \Sigma_{\text{intra}} + \Sigma_{\text{inter}} \tag{2}$$

where $\bar{x}_i = \frac{1}{m}\sum_{j=1}^{m} x_i^j$ is the centroid of the augmentation views of sample x_i, and $\Sigma^{(i)} = \frac{1}{m}\sum_{j=1}^{m}(x_i^j - \bar{x}_i)(x_i^j - \bar{x}_i)^T$ is the input-space covariance of the positive views for sample i.

The two terms carry distinct geometric meanings. The intra-class term Σ_{intra} measures the expected covariance of representations within the set of augmented views of each sample, propagated through the Jacobian of the encoder. It therefore captures the degree to which the encoder spreads representations of augmentations of the same image, reflecting local sensitivity of the feature map to input perturbations and encoding the invariance attained by the model. The inter-class term Σ_{inter}, by contrast, measures the covariance of the per-sample centroid representations around the global mean, capturing the extent to which the model separates different inputs in the embedding space. In the ideal setting, we expect Σ_{intra} to be small and Σ_{inter} to be large.

In practice, computing the Jacobian $\nabla_x f_\theta(\bar{x}_i)$ for each sample is prohibitively expensive for large datasets. We therefore approximate both terms using their trace. We compute $\text{tr}(\Sigma_{\text{intra}})$ as the mean squared pairwise distance between augmented views of the same sample in feature space, and $\text{tr}(\Sigma_{\text{inter}})$ as the variance of the per-sample mean representations around the global mean. These trace approximations are computed over held-out evaluation batches after each task checkpoint, yielding a pair of scalar diagnostics that we track over the course of continual training.

The trace ratio $\text{tr}(\Sigma_{\text{inter}})/\text{tr}(\Sigma)$ serves as a summary statistic for the geometric quality of the representation. Specifically, a high ratio indicates that most of the total variance in the embedding space is attributable to between-sample differences rather than within-sample augmentation noise, which is precisely the condition a discriminative representation should satisfy. A declining ratio over the course of continual training would indicate that the model is progressively losing its ability to discriminate between distinct inputs.

5.2 Grassmannian Subspace Analysis

While the covariance decomposition provides a scalar summary of representational quality at each checkpoint, it does not directly characterize how the geometric structure of the embedding space changes across tasks. To address this, we employ Grassmannian subspace analysis, which provides a principled distance metric between the subspaces spanned by the representations of different tasks.

The Grassmannian $G(n,k)$ is the manifold of all k-dimensional linear subspaces of $\mathbb{R}^n$. Given two subspaces $\mathcal{A}, \mathcal{B} \in G(n,k)$, represented by orthonormal bases $A, B \in \mathbb{R}^{n\times k}$, the principal angles $\theta_1 \le \theta_2 \le \cdots \le \theta_k \in [0, \pi/2]$ between them are defined recursively as

$$\cos(\theta_i) = \max_{\substack{a_i \in \mathcal{A} \\ b_i \in \mathcal{B}}} a_i^T b_i \quad \text{s.t.} \quad \|a_i\| = \|b_i\| = 1, \quad a_i^T a_j = 0, \quad b_i^T b_j = 0 \quad \forall j < i \tag{3}$$

In practice, the principal angles are computed via the singular value decomposition of $A^T B$, whose singular values are exactly $\{\cos(\theta_i)\}_{i=1}^{k}$. The geodesic distance on the Grassmannian is then

$$d_G(\mathcal{A}, \mathcal{B}) = \sqrt{\left(\sum_{i=1}^{k} \theta_i^2\right)} \tag{4}$$

A distance of zero indicates that the two subspaces are identical, while the maximum distance $d_G = \pi\sqrt{k}/2$ occurs when the subspaces are mutually orthogonal.

We apply this framework to measure subspace drift across tasks as follows. For each model's weights θ_t, we compute the k-dimensional principal subspace of the encoder's representation covariance on each task's held-out data, obtained as the top-k eigenvectors of Σ evaluated at that checkpoint. In practice, we use a value k that captures 95% of the embedding space variance. The pairwise Grassmannian distances between these task subspaces across all pairs of checkpoints are then assembled into a distance matrix, which reveals the degree to which the representational geometry of different tasks is preserved or overwritten as training proceeds. Large off-diagonal entries in this matrix indicate that representations learned for task τ occupy a subspace that is nearly orthogonal to the subspace of the checkpoint produced after task $t > \tau$.

We track two complementary versions of this analysis. In the first, subspaces are computed from each task's data separately and evaluated at each checkpoint (subspace drift), providing a more fine-grained view of how task-specific representational subspaces drift as new tasks are learned. In the second, subspaces are computed from all data seen up to the final checkpoint, which denotes the final encoder (subspace separability), measuring how distinct task subspaces are in the final embedding space. Together, these two views allow us to separate global subspace rotation from targeted erasure of task-specific subspace directions.

5.3 Gradient Alignment Analysis

The covariance decomposition and Grassmannian analysis characterize the static geometry of the representation space at each checkpoint. Our third diagnostic complements these by examining the optimization dynamics at task boundaries. Specifically, it analyzes when the model transitions from task t to task $t+1$, whether the parameter gradients induced by the new task tend to preserve or destroy the representational geometry that was useful for prior tasks.

We formalize this through a gradient alignment analysis, in which we measure the alignment between the gradients of the current task's SSL loss and the principal directions of the prior task's covariance structure. Let $g = \nabla_\theta \mathcal{L}_t(\theta)$ denote the gradient of the current task's SSL loss with respect to the encoder parameters θ, evaluated at the beginning of task $t+1$. We compute the alignment of this gradient with the inter-class covariance Σ_{inter} and intra-class covariance Σ_{intra} from the prior checkpoint via the normalized quadratic forms

$$\alpha_{\text{inter}} = \frac{g^T \hat{\Sigma}_{\text{inter}}\, g}{g^T g} \qquad \text{and} \qquad \alpha_{\text{intra}} = \frac{g^T \hat{\Sigma}_{\text{intra}}\, g}{g^T g} \tag{5}$$

where $\hat{\Sigma}$ denotes the projection of the covariance matrix onto the parameter space of the encoder through its Jacobian, approximated empirically using the saved encoder represen-

tations. A high value of α_{inter} indicates that the new task's gradient has strong components in the directions that previously encoded between-class discrimination. This would mean that the gradient is likely to overwrite the structure most responsible for discriminability of prior tasks. A high ratio $\alpha_{\text{inter}}/\alpha_{\text{intra}} > 1$ thus signals a particularly destructive update, one that disproportionately perturbs the embedding dimensions that carry between-class information while leaving within-sample invariance relatively unaffected.

All gradient quantities are computed on evaluation batches from the portions of each task's data at the beginning of each new task, before any parameter updates have been applied, to ensure that the measurements reflect the representational geometry established by the prior task rather than the partially updated geometry of the new one.

6 Results

We report results on CIFAR-100 across all three SSL frameworks under two training regimes: a fine-tuning baseline (FT) in which the model is trained sequentially on each task with no access to prior data, and an experience replay baseline (ER) in which a random replay buffer provides a small buffer of data from previous tasks at training. We evaluate our metrics defined above (namely the covariance decomposition, Grassmannian subspace analysis, and gradient alignment ratios) across these two methods. All results use a ResNet-18 backbone producing 512-dimensional representations across five tasks, each comprising 20 disjoint CIFAR-100 classes.

6.1 Linear Probe Accuracy

Figure 1 summarizes the final class-incremental linear probe accuracy for each method under FT and ER conditions. Under FT, all three methods achieve competitive accuracies in the Continual Learning setup. Adding a random replay buffer (ER) yields modest but consistent improvements across all methods. Class-incremental accuracy increases by roughly 2–3% across methods.

6.2 Covariance Decomposition

Figure 2 reports the ratio of the inter-class and intra-class trace values, $\frac{\text{tr}(\Sigma_{\text{inter}})}{\text{tr}(\Sigma_{\text{intra}})}$, computed after each task. Under FT, all three methods show a consistent decline in the ratio as training progresses across tasks, indicating that the encoder progressively loses the ability to separate different inputs in embedding space. Under FT, Barlow Twins exhibits a declining ratio across tasks (from approximately 0.24 at Task 0 to 0.16 at Task 4), indicating that the inter-class covariance degrades faster than the intra-class variance throughout training. VICReg displays a similar trajectory but with a steeper initial decline and partial recovery at later tasks. SimCLR, by contrast, shows an interestingly different trend. Rather than declining, its trace ratio increases steadily across tasks from approximately 0.36 to 0.45. This reflects the distinct geometry induced by contrastive negative sampling, which explicitly maintains inter-sample distinctions even as the model learns new tasks. This also perhaps justifies the higher linear probing accuracy on SimCLR compared to the other SSL methods studied here.

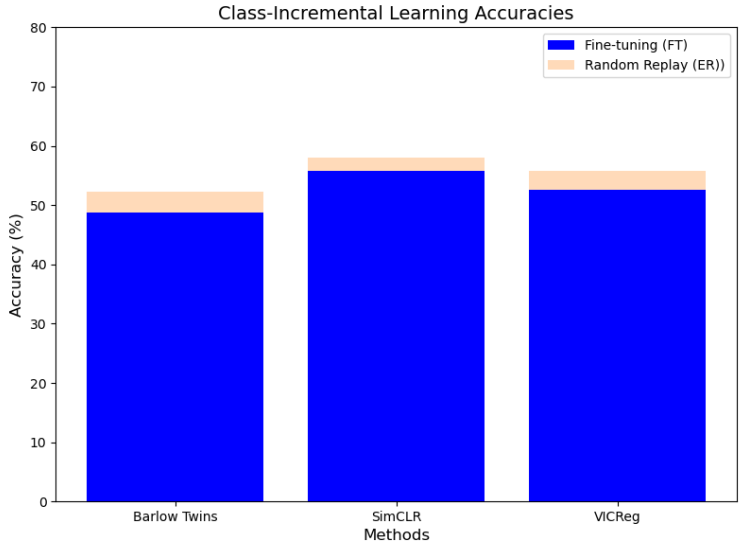


Figure 1: Linear Probing accuracy on Barlow Twins, SimCLR, and VICReg with finetuning and random replay on CIFAR-100. The setup has 5 different tasks and 20 classes per task.

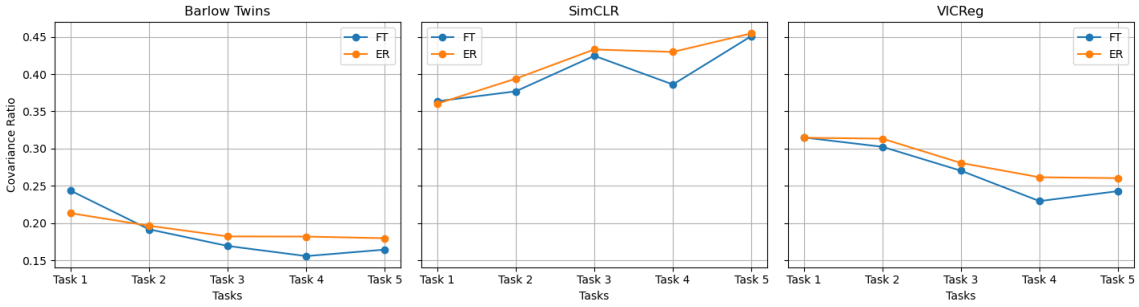
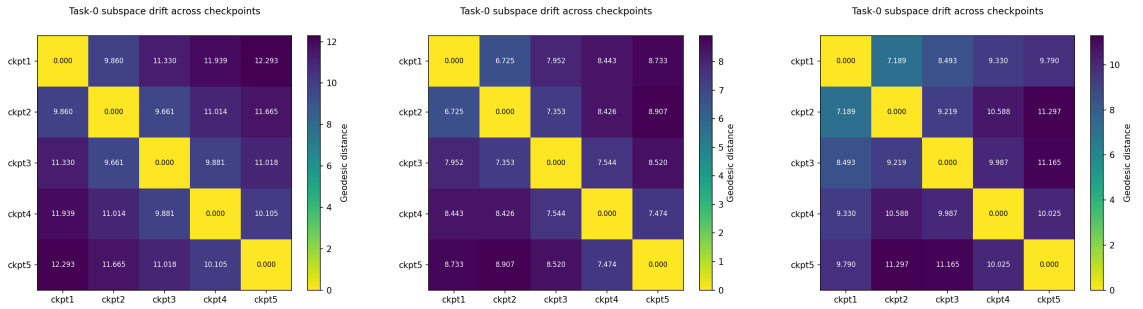


Figure 2: Ratio of $\text{tr}(\Sigma_{inter})/\text{tr}(\Sigma_{intra})$ values for each method, both with and without a replay buffer. Higher values generally indicate healthier subspaces.

The ER condition partially mitigates the trace ratio decay observed in Barlow Twins and VICReg. The random buffer substantially preserves both inter- and intra-class variance relative to the FT baseline, with the trace ratio for Barlow Twins under ER remaining more stable across the task sequence. With SimCLR, the ER ratio maintains consistently higher than standard fine-tuning, demonstrating some greater preservation of variance in embedding space. Nonetheless, all methods suggest that the replay buffer provides only partial recovery of the geometry embedding space in the continual learning setup.

6.3 Grassmannian Subspace Analysis

Figures 3 and 4 present two complementary Grassmannian distance analyses. Both of these figures provide insights into how the geometry of the embedding space drifts over time. The first measures subspace drift over time as new data is learned, and the second



(a) Barlow Twins (FT) (b) SimCLR (FT) (c) VICReg (FT)

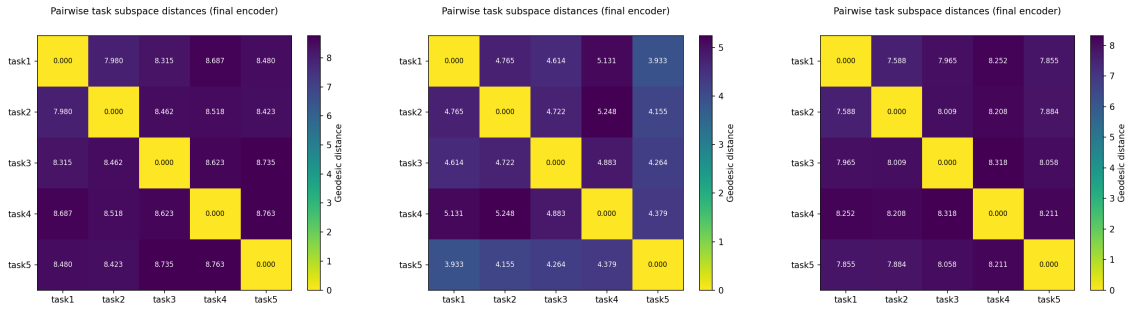
(d) Barlow Twins (ER) (e) SimCLR (ER) (f) VICReg (ER)

Figure 3: Task 0 subspace drift across checkpoints under fine-tuning (FT, top row) and experience replay (ER, bottom row) for all three SSL methods. Each cell reports the Grassmannian distance between the Task 0 subspace at the initial checkpoint and the subspace at each subsequent checkpoint.

measures the separability of task subspaces after training on all tasks. Each of these matrices are symmetric (since the Grassmannian distance is a symmetric metric). The main diagonal values are 0 throughout since the distance between any subspace to itself is 0, as defined by the metric.

Subspace Drift In Figure 3, we plot the drift of the first task's subspace across checkpoints as the model trains on more tasks. The subspace drift analysis reveals that task-specific subspaces drift substantially from their original configurations as new tasks are trained. As expected, as each model (regardless of whether optimized with replay) encounters more tasks, the drift of embedding space created by the first task increases, since the model has little incentive to preserve old representations. However, between each method's FT Grassmannian matrix and the corresponding matrix for the method with ER, we notice that the drift in the first task's subspace is less, indicating that the injected variance from replaying past data prevents the model from drifting significantly from the best representations it has of the data. This pattern can be observed for all pairwise Grassmannian distances across all three methods.

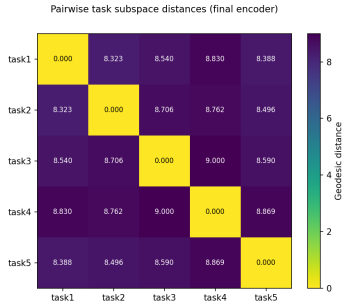
Subspace Separability In Figure 4, we compare the distances between the subspaces created by each task's data on the final encoder. Under FT, all three methods produce small and similar subspace Grassmannian distances. In this evaluation, small pairwise



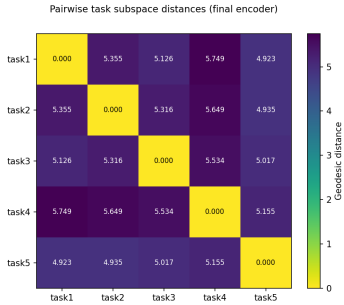
(a) Barlow Twins (FT)

(b) SimCLR (FT)

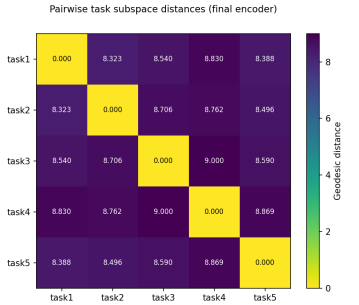
(c) VICReg (FT)



(d) Barlow Twins (ER)



(e) SimCLR (ER)



(f) VICReg (ER)

Figure 4: Pairwise Grassmannian distances between all task subspaces evaluated at the final checkpoint, under fine-tuning (FT, top row) and experience replay (ER, bottom row). Higher pairwise distances under ER indicate that task-specific subspaces remain more geometrically separated in the final embedding space, suggesting that replay better preserves task-discriminative representational structure across all three SSL paradigms.

Grassmannian distances imply similar task subspaces in the final encoder. Even though the magnitude of the values vary between the different methods (Barlow Twins values vary between 7.98-8.763, SimCLR values vary between 3.933-5.248, and VICReg values vary between 7.588-8.211), in all cases, when a replay buffer is used, the subspaces are farther apart in embedding space. In some cases (particularly in SimCLR), the difference in magnitudes between Grassmannian distances of embeddings are significantly higher when using a replay buffer. This would suggest that the replay buffer effectively makes task subspaces more separable. Greater subspace separability implies that the encoder retains more task-discriminative structure under ER, which aligns with the accuracy improvements observed and suggests that replay acts not only as a data augmentation but as a geometric regularizer on the embedding space

6.4 Gradient Alignment Analysis

Figure 5 summarizes the gradient alignment of each method's SSL loss with respect to the prior task's covariance structure, measured during each task's training. Specifically, the projection of the entire gradient value onto the different covariance terms, conveying how learning new tasks directly affects representation space covariance measures.

Covariance Gradient Projections Figure 5 presents the α_{inter} and α_{intra} values jointly across methods for both standard FT and ER. After training on a given task, we calculate the projection of our loss gradients onto each covariance component independently, and plot these values. Figure 5 demonstrates that across all SSL methods and regardless of whether a replay buffer is used or not, gradient updates tend to affect the Σ_{inter} more heavily than Σ_{intra}. A larger α_{inter} gradient norm indicates that parameter updates at that boundary are disproportionately reshaping the inter-sample geometry of the embedding space, which directly corresponds to the structure that supports class discrimination and that is most at risk of forgetting.

Across all three methods and both training conditions, Σ_{inter} gradient norms consistently and substantially exceed Σ_{intra} norms at every task boundary. For VICReg under FT, the inter-class gradient norm begins at approximately 5.3 at the T1 $\rightarrow$ T2 boundary and declines to approximately 1.2 by T4 $\rightarrow$ T5, while the intra-class norm remains below 0.8 throughout. A similar pattern holds for SimCLR, where the inter-class norm begins near 2.0 and the intra-class norm stays below 0.35 across all boundaries. Barlow Twins shows the same qualitative pattern at a smaller absolute scale. This asymmetry directly implies that the SSL objectives studied here are geometrically biased toward modifying inter-class structure at task boundaries, even though it is precisely this structure that needs to be preserved for downstream discriminability. The declining magnitude of both norms across boundaries suggests that the encoder's sensitivity to new task data degrades over the task sequence, possibly reflecting a form of representational rigidity that develops as training progresses. Under ER, the pattern is qualitatively identical, inter-class gradients still dominate, but the absolute magnitudes are slightly reduced, consistent with the replay buffer providing a stabilizing signal that moderates the strength of updates at each boundary.

7 Discussion

The three diagnostic frameworks applied in this work reveal a geometrically specific account of catastrophic forgetting in continual SSL. Rather than a uniform collapse of representational structure, forgetting manifests through the degradation of certain geometric properties, each of which has implications for how future continual SSL methods might be designed.

The decline of the covariance trace ratio across tasks for Barlow Twins and VICReg under fine-tuning and ER confirms that inter-class discriminative structure degrades faster than within-sample invariance throughout continual training. This asymmetry matters because it implies that the encoder does not simply lose representational quality uniformly, rather it loses the geometry that downstream linear probes depend on most. The practical consequence is that broad accuracy metrics will tend to underestimate the extent of forgetting, since within-sample invariance can partially persist even as inter-class separability collapses. Taken together with the gradient results discussed below, this suggests that the degradation of the trace ratio is directly driven by the structure of the SSL update itself.

The Grassmannian drift analysis reveals that task subspaces drift substantially from their initial configurations as training progresses, regardless of method or training condition. This is unsurprising in principle since the SSL objective provides no explicit incentive to preserve earlier representations. Importantly, a model's earliest representations of a task

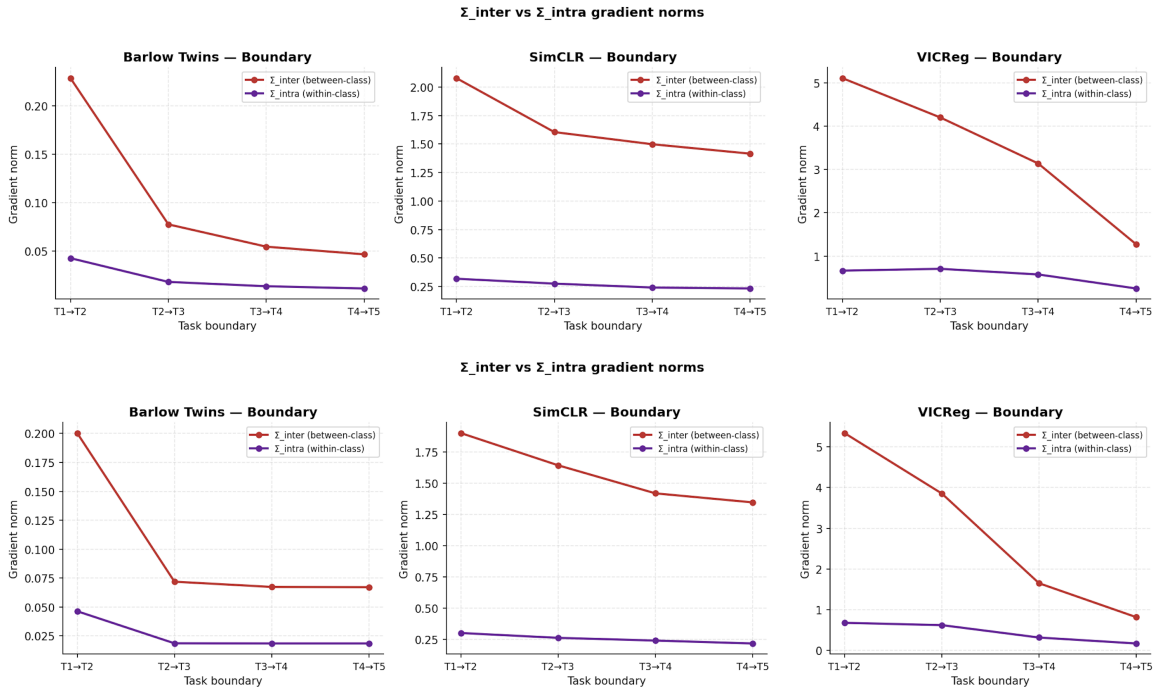


Figure 5: Gradient projections of losses for each method onto each covariance. Finetuning (Top) and experience replay (Bottom) gradient projection values are shown across all three methods. The magnitude of the projections onto Σ_{inter} are consistently higher than the magnitudes onto Σ_{intra}.

are arguably its richest since they are formed when the encoder devotes its full capacity to that task's data, without interference from other tasks. As training continues, this initial geometry is progressively eroded, and there is no mechanism in standard SSL to recover it.

This observation motivates a class of approaches that explicitly regularize the encoder to remain close to its geometry at the end of each task. For instance, this could be accomplished by penalizing the Grassmannian distance between the current encoder's subspace for prior task data and the subspace recorded at the corresponding checkpoint. Experience replay partially mitigates drift, but the effect is weak and inconsistent across methods. Random replay does not explicitly seek to minimize drift, so replaying raw data could be too indirect of a mechanism to meaningfully anchor subspace geometry. More targeted interventions that operate directly on the subspace structure may be necessary.

Perhaps the most actionable finding from the Grassmannian analyses is that experience replay consistently increases pairwise subspace distances between different tasks' data in the final encoder. The fine-tuning encoder allows task subspaces to collapse toward one another, reflecting a form of feature overwriting in which the encoder reuses the same representational directions across tasks rather than maintaining task-distinct structure. Replay partially prevents this collapse because replayed gradients from prior tasks happen to exert pressure against the encoder settling into the new dominant subspace.

This suggests that subspace separability is a meaningful geometric objective in its own right, and that methods which explicitly maximize pairwise Grassmannian distances between task subspaces during training could provide more principled protection against feature overwriting than replay alone. Such a regularizer would not require label information, since subspaces can be estimated from unlabeled task data, and would directly target the geometric failure mode that fine-tuning produces. The fact that replay achieves this effect only weakly indicates there is substantial room for improvement by making subspace separability an explicit training objective.

The gradient norm analysis reveals that SSL gradient updates at task boundaries project disproportionately onto Σ_{inter} relative to Σ_{intra}, consistently and across all three methods. This provides a mechanistic link to the covariance findings. Namely, the inter-class geometry observed to degrade faster in the trace ratio is precisely the geometry that SSL updates most aggressively modify at each task boundary.

It is less clear whether the asymmetric gradient pressure is an intrinsic property of self-supervised objectives in general, or whether it is specific to the redundancy-reduction and contrastive families studied here. It is possible that the SSL losses studied here happen to be particularly sensitive to inter-class geometry due to their reliance on batch-level statistics, which are dominated by between-sample relationships. Understanding whether this bias is universal or method-specific would clarify whether gradient-level interventions, such as projecting updates away from the principal directions of Σ_{inter} at each task boundary, could serve as an effective mitigation strategy.

Taken together, the three frameworks converge on a picture in which catastrophic forgetting in SSL is geometrically structured. First, the covariance and gradient results jointly imply that SSL objectives are self-undermining when trained in a non-stationary setting. They ultimately produce updates that disproportionately erode the inter-class geometry that is necessary for robust downstream representations. Second, the subspace drift results suggest that a model's initial representation of each task is its highest-quality one, and that preserving proximity to this geometry should be an objective in continual SSL. Third, the pairwise separability results suggest that preventing feature overwriting may be more tractable as a geometric objective (maximize task subspace distances) than as a data-level objective (select better replay samples), since the former directly targets the representational failure mode rather than addressing it indirectly through loss minimization.

Standard accuracy metrics would not have revealed any of these distinctions. Methods that achieve similar linear probe accuracy can differ substantially in their embedding geometry and these differences carry implications for how forgetting will compound over longer task sequences, at larger scale, or under more aggressive task shifts. The diagnostic framework introduced here is thus not only a lens for understanding existing methods, but a basis for designing continual SSL algorithms that target the geometric mechanisms of forgetting directly.

8 Conclusion and Future Work

This work has investigated catastrophic forgetting in continual self-supervised learning from an explicitly geometric perspective, using a host of diagnostic tools. Covariance decomposition, Grassmannian subspace analysis, and gradient alignment analysis all help characterize

the structural evolution of the embedding space across a sequence of tasks. Evaluating SimCLR, Barlow Twins, and VICReg in a Class-Incremental Learning setup on CIFAR-100, we find that forgetting in SSL is not a uniform representational collapse but a geometrically specific phenomenon. Inter-class discriminative structure degrades faster than within-sample invariance, SSL gradient updates at task boundaries are structurally biased toward overwriting inter-class geometry, and experience replay improves downstream accuracy and task subspace separability not by anchoring the encoder to its initial geometry but by acting as a geometric regularizer on the final representation space.

These findings open several directions for future work. First, the gradient alignment results suggest that explicitly regularizing the inter-class covariance directions at task boundaries, for instance, by projecting new-task gradients away from the principal directions of Σ_{inter} from the prior task's model, could provide a geometrically principled approach to mitigating forgetting in SSL without requiring label information. Second, the observation that experience replay does not reduce subspace drift but does increase final subspace separability motivates the design of replay strategies that are explicitly optimized for geometric objectives rather than purely for loss minimization on retained samples. Third, extending this geometric diagnostic framework to more expressive SSL architectures, such as DINO and JEPA-based models, and to larger-scale benchmarks would allow us to assess whether the structural patterns identified here are specific to the methods and settings studied or reflect broader properties of SSL under non-stationary data.

References

[1] Alessandro Achille, Tom Eccles, Loic Matthey, Chris Burgess, Nicholas Watters, Alexander Lerchner, and Irina Higgins. Life-long disentangled representation learning with cross-domain latent homologies. *Advances in Neural Information Processing Systems*, 31, 2018.

[2] Abhijeet Awasthi and Sunita Sarawagi. Continual learning with neural networks: A review. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 362–365, 2019.

[3] Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

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

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[11] Andrea Cossu, Antonio Carta, Vincenzo Lomonaco, and Davide Bacciu. Continual learning for recurrent neural networks: an empirical evaluation. *Neural Networks*, 143:607–627, 2021.

[12] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[13] Enrico Fini, Victor G Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9621–9630, 2022.

[14] Matteo Gamba. *On Implicit Smoothness Regularization in Deep Learning*. PhD thesis, KTH Royal Institute of Technology, 2024.

[15] Alex Gomez-Villa, Bartlomiej Twardowski, Lu Yu, Andrew D Bagdanov, and Joost Van de Weijer. Continually learning self-supervised representations with projected functional regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3867–3877, 2022.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

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

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

[20] Minqian Liu and Lifu Huang. Teamwork is not always good: An empirical study of classifier drift in class-incremental information extraction. In *Findings of the association for computational linguistics: ACL 2023*, pages 2241–2257, 2023.

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

[22] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. *arXiv preprint arXiv:2110.06976*, 2021.

[23] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[24] Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz Khan. Continual deep learning by functional regularisation of memorable past. *Advances in neural information processing systems*, 33:4453–4464, 2020.

[25] Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. *Advances in neural information processing systems*, 32, 2019.

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

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

[28] Chi Ian Tang, Lorena Qendro, Dimitris Spathis, Fahim Kawsar, Cecilia Mascolo, and Akhil Mathur. Kaizen: Practical self-supervised continual learning with continual fine-tuning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2841–2850, 2024.

[29] Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. *arXiv preprint arXiv:1901.11356*, 2019.

[30] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.

[31] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. Pmlr, 2017.

Appendix A. Hyperparameters

Table 1: Training hyperparameters for each SSL method. All methods share the same backbone, optimizer, and scheduler. Method-specific parameters are listed separately.

Hyperparameter	**Value**	**Applies To**
Shared		
Backbone	ResNet-18 (low-resolution)	All
Backbone output dim	512	All
Projector	3-layer MLP, 512→2048→2048	All
Epochs per task	300	All
Number of tasks	5	All
Optimizer	LARS	All
Learning rate	5	All
Weight decay	10^{-6}	All
LR schedule	Linear warmup + cosine annealing	All
Replay buffer size	0 (FT) / 200 (ER)	All
Barlow Twins		
Batch size	256	Barlow Twins
Projector output dim	2048	Barlow Twins
λ (off-diagonal weight)	0.1	Barlow Twins
SimCLR		
Batch size	512	SimCLR
Projector output dim	256	SimCLR
Temperature τ	0.7	SimCLR
VICReg		
Batch size	256	VICReg
Projector output dim	2048	VICReg
Invariance coeff λ	25	VICReg
Variance coeff μ	25	VICReg
Covariance coeff ν	1	VICReg