Contrastive Consolidation of Top-Down Modulations Achieves Sparsely Supervised Continual Learning

Viet Anh Khoa Tran
Peter Grünberg Institute
Forschungszentrum Jülich & RWTH Aachen
v.tran@fz-juelich.de

Emre Neftci
Peter Grünberg Institute
Forschungszentrum Jülich & RWTH Aachen
e.neftci@fz-juelich.de

Willem A. M. Wybo
Peter Grünberg Institute
Forschungszentrum Jülich
w.wybo@fz-juelich.de

Abstract

Biological brains learn continually from a stream of unlabeled data, while integrating specialized information from sparsely labeled examples without compromising their ability to generalize. Meanwhile, machine learning methods are susceptible to catastrophic forgetting in this natural learning setting, as supervised specialist fine-tuning degrades performance on the original task. We introduce **task-modulated contrastive learning (TMCL)**, which takes inspiration from the biophysical machinery in the neocortex, using predictive coding principles to integrate top-down information continually and without supervision. We follow the idea that these principles build a view-invariant representation space, and that this can be implemented using a contrastive loss. Then, whenever labeled samples of a new class occur, new affine modulations are learned that improve separation of the new class from all others, without affecting feedforward weights. By co-opting the view-invariance learning mechanism, we then train feedforward weights to match the unmodulated representation of a data sample to its modulated counterparts. This introduces modulation invariance into the representation space, and, by also using past modulations, stabilizes it. Our experiments show improvements in both class-incremental and transfer learning over state-of-the-art unsupervised approaches, as well as over comparable supervised approaches, using as few as 1% of available labels. Taken together, our work suggests that top-down modulations play a crucial role in balancing stability and plasticity.

1 Introduction

Input data streams encountered by animals or humans during development differ markedly from those commonly used in machine learning. In contemporary machine learning (e.g. foundation models), data streams typically consist of unlabeled data, augmented with some degree of supervised fine-tuning in the final training stages [1]. Such an approach is difficult to translate to the continual learning setting encountered in natural data streams, as the naive introduction of fine-tuning stages often leads to catastrophic forgetting [2–4]. Meanwhile, animals and humans receive mostly unsupervised inputs, interspersed with sparse supervised data, which could, for instance, be provided through an external teacher (e.g. a parent telling their child that the object is called an 'apple'). Compared to unsupervised data streams, such sparse supervisory episodes are infrequent. Therefore, the learning dilemma that

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-5030

arises is how continual learning algorithms can benefit from sparse supervisory episodes without negatively affecting representations learned in an unsupervised manner.

Here, we draw inspiration from the circuitry in biological brains to solve this learning dilemma. We leverage the fact that cortical neurons can – broadly speaking – be subdivided into a proximal, perisomatic zone, receiving feedforward inputs [5–7], and a distal, apical region, receiving top-down modulatory inputs [6–11] (Figure 1, left). Through their physical separation, these zones are functionally distinct [12], and implement different plasticity principles [13, 14]. Learning of the perisomatic, feedforward connections is believed to follow a form of predictive coding [15, 16] that is the biological analogue of self-supervised learning such as VICReg [17] or CPC [18]. At the same time, top-down modulations to distal dendrites provide a contextual, modulatory signal to the feedforward network [19–26]. We hypothesize that this signal is learned during the supervised learning episodes. In machine learning, this concept has been explored in the context of parameter-efficient fine-tuning by training task-specific scaling and/or shifting terms [27–32]. This provides a straightforward solution to continual learning problems for which the task identity is known during both training and evaluation (i.e. task-incremental learning [33]), as modulations for new tasks can be learned without affecting core feedforward weights [34–36]. However, when the task identity is not known at evaluation (i.e. class-incremental learning), the modulated representations for each task need to be consolidated in a shared representation space.

To achieve class-incremental learning, we hypothesize, based on the spatial and functional segregation of distal dendrites, that the top-down signal is not affected by the predictive plasticity of feedforward weights in the perisomatic region. As such, it leaves a permanent imprint on the network that, through occasional reactivation, integrates the new percept in the neural representation space, while also providing a form of functional regularization that limits forgetting. We demonstrate that these effects are achieved by standard predictive coding principles that proceed over modulated representations. In our task-modulated contrastive learning (TMCL) algorithm, we use the currently available labeled examples for each new class to learn modulations that *orthogonalize* their representations from all others. These modulations are then frozen and applied to the network during the feedforward weight learning, using only currently available unlabeled samples, and effectively *consolidate* the task-specific knowledge encoded in the modulations into the feedforward weights (Figure 1, middle). This departs from the conventional pretraining-finetuning paradigm, where naive reintegration of specialized models into the general one causes catastrophic forgetting [1, 4] (Figure 1, right).

We evaluate the performance of TMCL on the standard class-incremental CIFAR-100 benchmark, outperforming state-of-the-art purely unsupervised, purely supervised, and hybrid approaches in label-scarce scenarios. Furthermore, we evaluate its transfer learning capabilities across a diverse set of downstream tasks, demonstrating its effectiveness in learning generalizable representations that extend beyond adaptation to CIFAR-100. Finally, we show that our method dynamically navigates the stability-plasticity dilemma through adaptation of the consolidation term.

2 Related Work

Biological representation learning. Several authors have explored the idea that the cortex learns in a self-supervised manner [15, 16, 37–40]. Although complementary approaches based on adversarial samples have also been proposed [37], most theories focus on some form of predictive coding, where the cortex learns to predict the next inputs given the current neural representation. Mikulasch et al. [38, 39] take a classic view on this, where a loss function to the next layer reconstructs the input, while Kermani Nejad et al. [40] theorize that the architecture of the cortical microcircuit is well-suited for predictive coding. Finally, Illing et al. [15] and Halvagal and Zenke [16] propose local plasticity rules based on, respectively, CPC [18] and VICReg [17] that, as they argue, in a natural setting could proceed by comparing neural representations at subsequent time steps. We extend this idea with an explanation of how top-down modulations could be incorporated into the learning process.

Learning with modulations. The expressivity of learning modulations was initially demonstrated by Perez et al. [27] to solve visual reasoning problems. Frankle et al. [28] subsequently showed that a surprisingly high performance can be achieved with ResNets [41] while just training BatchNorm layers, which — if performed per-task — is equivalent to learning affine modulations. Finally, it was shown simultaneously in language and vision that fine-tuning through modulations in transformer

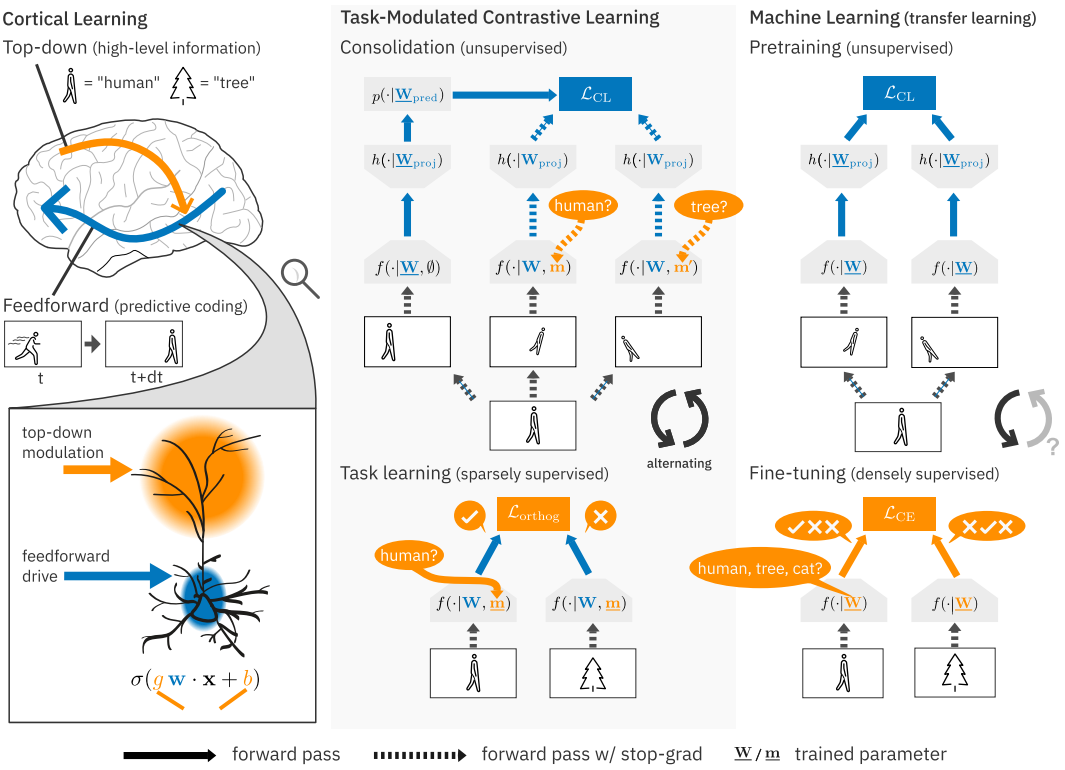

Figure 1: **Biologically inspired consolidation of high-level modulations into feedforward weights.** Cortical learning (**left**) is characterized by the interplay between top-down (orange) and feedforward (blue) processing, where top-down connections impart high-level information on the feedforward sensory processing pathway (top). The feedforward pathway, on the other hand, learns to predict neural representations of future inputs (predictive coding). Notably, top-down and feedforward information arrives at spatially segregated loci on sensory neurons (bottom), suggesting distinct roles in shaping the neuronal input-output relation (cf. [19]) as well as distinct plasticity processes governing weight changes. Translating this view to a machine learning algorithm (**middle**), we *(i)* train modulations to implement high-level object identification tasks as the analogue of top-down inputs (bottom, solid arrows, but not dashed ones, indicate that gradients backpropagate in the opposite direction, and underlined parameters are trained), while we *(ii)* train for view invariance over modulated representations – and thus also for modulation invariance – as the analogue of predictive coding (top). As a consequence, high-level information continually permeates into the sensory processing pathway, which can be contrasted with the traditional machine learning (**right**) approach of unsupervised pretraining for view invariance (top) followed by supervised fine-tuning (bottom). In this case, it is unclear how high-level information can be incorporated into the sensory processing pathway to improve subsequent learning.

models is particularly powerful, as it reaches the same performance as using all parameters [29, 30, 32].

Modulations are an attractive way to implement task-incremental learning, as task-specific modulations can be learned for each new task without affecting feedforward weights. Masse et al. [34] propose gating random subsets of neurons, whereas Iyer et al. [36] provide a biological interpretation. Fine-tuning through modulations [30, 32] can also be considered as a form of continual learning, as it can be applied in sequence to any new dataset.

Continual representation learning. Traditionally, continual learning has focused on purely supervised methods [42–67]. These methods can be categorized into replay-based approaches [42–50], regularization-based approaches [50–62, 67] and approaches introducing new parameters [63–66]. TMCL can be considered a regularization-based approach, but it also introduces new parameters. However, these parameters are not used during inference. Recently, purely self-supervised con-

https://doi.org/10.52202/085713-5030

tinual learning algorithms have been proposed [68–72], where state-of-the-art algorithms [69, 72] predict past representations from a stored model copy without exemplar replay. Very recently, semi-supervised continual learning approaches have emerged [73–76], which consolidate by distilling from expert models [76, 77] or for which the labels are only used for readout learning [74].

We highlight SIESTA [49], CLS-ER [78] and DualNet [73], which are inspired by the complementary learning systems theory (CLS) [79, 80], the idea that learning occurs at *fast* (task-learning) and *slow* (consolidation) timeframes. However, all these methods interpret CLS to assume sample replay provided as episodic memory via the hippocampus. Instead, we suggest functional replay of task modulations ('How did we solve the task?'), and do not investigate methods with exemplar replay. Still, we point out similarities to the replay-based DualNet, which introduces a fast supervised network generating modulations on top of a slow self-supervised network. However, DualNet consolidates only as new labels arrive. Consolidation in TMCL requires no labels, instead exploiting previously learned task modulations.

3 Modulation-Invariant Continual Representation Learning

We follow the idea from Iyer et al. [36] that cortical networks learn to interpret novel information by learning new top-down modulations, and propose that consolidation of these modulations is a crucial component of learning in biological brains. This motivates our task-modulated contrastive learning (TMCL) algorithm as the machine learning analogue of this consolidation, tackling continual representation learning. We consider the parameters of conventional machine learning models as *task-agnostic* feedforward weights $\mathbf{W}$. On top of these weights, we introduce per-task affine transformation parameters as *task-specific* modulations $\mathbf{m}_t$, the analogue of biological top-down modulations. We denote the modulated network as $f(\mathbf{x}|\mathbf{W}, \mathbf{m})$ with feedforward weights $\mathbf{W}$ and modulations $\mathbf{m}$, while $f(\mathbf{x}|\mathbf{W}, \emptyset)$ represents the unmodulated network (i.e. where the modulations are identity operations).

In TMCL, the overall objective is to arrive at an unmodulated representation space where all classes $c \in \mathcal{C}$ in the dataset $\mathcal{D}$ have compact representations clustered around mutually orthogonal class centers, i.e.

$$\gamma^c \perp \gamma^{c'}, \forall c, c' \in \mathcal{C}, \tag{1}$$

with $\gamma^c = \mathbb{E}_{\mathbf{x} \in \mathbf{X}^{(c)}}[f(\mathbf{x}|W, \emptyset)]$, where $\mathbf{X}^{(c)}$ is the set of samples from class c. Because we assume a continual learning setting, where we do not have all class samples at our disposal, we do not optimize for (1) directly. Rather, we achieve this by breaking the optimisation procedure down into two distinct learning objectives. The first objective (Figure 2, bottom left) is to orthogonalize any given class c from the others in a *modulated* representation space, i.e. we learn a modulation $\mathbf{m}^c$ so that the class center of class c becomes orthogonal to all other classes in the modulated space:

$$\gamma^c_{\mathbf{m}^c} \perp \{\gamma^{c'}_{\mathbf{m}^c} : c' \in \mathcal{C} \setminus \{c\}\}, \tag{2}$$

where $\gamma^c_{\mathbf{m}}$ denotes the representation of the class center under modulation $\mathbf{m}$, i.e. $\gamma^c_{\mathbf{m}} = \mathbb{E}_{\mathbf{x} \in \mathbf{X}^{(c)}}(f(\mathbf{x}|W, \mathbf{m}))$. The second objective (Figure 2, bottom right) is entirely unsupervised and trains network weights to become *modulation-invariant*, so that

$$\gamma^c = \gamma^c_{\mathbf{m}^{c'}} \forall c' \in \mathcal{C}. \tag{3}$$

It can be seen that a representation space that satisfies *both* (2) and (3), also satisfies (1).

In our continual setting, which we adapt from Fini et al. [69], training is partitioned into $s \in 1, \ldots, S$ sessions, so that (1) can only be achieved approximately. In each session s, we only observe unlabeled samples $x \in \mathcal{D}^{(s)} \subset \mathcal{D}$ belonging to the session-specific partition of classes $\mathcal{C}^{(s)} \subset \mathcal{C}$. Additionally, a fraction of labeled samples $(x, y) \in \mathcal{D}^{(s)}_{\text{sup}} \subset \mathcal{D}^{(s)}$ is made available to (3). As a consequence, during each session (Figure 2), we learn objective (2) restricted to $\mathcal{D}^{(s)}_{\text{sup}}$ in a first phase, and then learn objective (3) using unlabeled samples from $\mathcal{D}^{(s)}$. We explain the implementation of both phases in detail below.

Learning modulations that orthogonalize new class representations. Whenever a new class label is observed, we learn class modulations on top of frozen feedforward weights to implement objective (2), which improves separation of the new class representations from all others currently

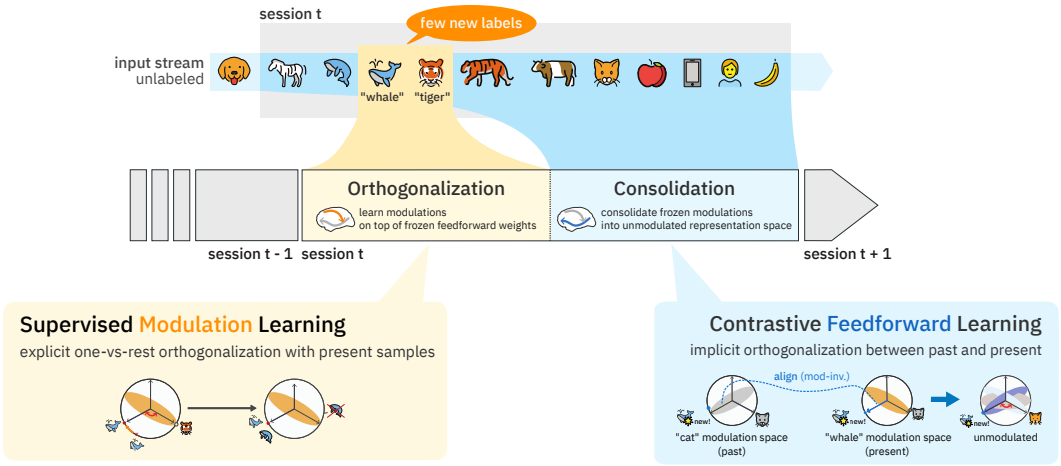

Figure 2: **Sparsely labeled class-incremental representation learning.** We implement continual learning over mostly unlabeled data streams, where only a few labeled samples are provided (**top**). To give an intuition of our algorithm (**bottom**), we consider that after successfully incorporating the data seen thus far, sufficiently collapsed neural representations exist for the already seen data classes after session $t-1$ (here dog, cat). For a new data class in session t (e.g. whale), such a collapsed representation may not yet exist. We then learn a *new* set of modulations to collapse "whale" representations in the modulated representation space, *orthogonalizing* them from all other available labeled examples, thus obtaining an orthogonal subspace for everything that is non-whale. Then, occasional reactivation of the "whale" modulation in $\mathcal{L}_{\text{CL}}$ draws unmodulated "whale" representations towards this collapsed representation (cf. Figure 1, middle), while drawing other samples to the orthogonal subspace, thus *consolidating* "whale" into the unmodulated representation space.

available. We assume that explicit class labels for these other classes are not available, therefore, the conventional machine learning approach of all-vs-all classification is not applicable. Instead, we learn one-vs-rest class modulations, only using currently available samples as negatives (Figure 2, bottom left). Note that if negative samples were to collapse to a single representation, (2) and (3) could not hold simultaneously and therefore (1) could not be achieved either. For this reason, we apply a variation of the **orthogonal projection loss** (OPL) [81] instead of binary cross-entropy (BCE) [82]. We define $s_{\mathbf{m}}(\mathbf{u},\mathbf{v}) = \text{sim}(f(\mathbf{u}|\mathbf{W},\mathbf{m}), f(\mathbf{v}|\mathbf{W},\mathbf{m}))$ as the cosine similarity between samples $\mathbf{u}$ and $\mathbf{v}$ under modulation $\mathbf{m}$, i.e. $\text{sim}(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}^T\mathbf{v}}{\|u\|\|v\|}$. Then, given a batch $\mathbf{X}^{(c)}$ of c-class examples and a batch $\mathbf{X}^{(\neg c)}$ of non-c examples, we define

$$\mathcal{L}_{\text{OPL}}^{(c)} := \sum_{\mathbf{p},\mathbf{p}'\in\mathbf{X}^{(c)}} \underbrace{(1 - s_{\mathbf{m}}(\mathbf{p},\mathbf{p}'))}_{\uparrow \text{collapse}} + \sum_{\substack{\mathbf{p}\in\mathbf{X}^{(c)} \\ \mathbf{n}\in\mathbf{X}^{(\neg c)}}} \underbrace{|s_{\mathbf{m}}(\mathbf{p},\mathbf{n})|}_{\uparrow \text{orthogonalization}} . \tag{4}$$

$\mathbf{m}^{(\mathbf{c})}$ is then found as $\min_{\mathbf{m}} \mathcal{L}_{\text{OPL}}^{(c)}$. The second term draws the cosine similarities between class c and non-class c representations to zero, leading to an orthogonalization of class c representations from all others, therefore approximating objective (2).

Consolidation of modulations into a view- and modulation-invariant representation space. To implement objective (3), we co-opt self-supervised contrastive learning, which is considered a biological analogue of predictive learning principles of the feedforward connections [15, 16]. Contrastive learning objectives train for view-invariance (VI), as they attract representations of views of the same source sample under different view augmentations to each other, while repelling representations of other samples [17, 18, 83–92]. These view augmentations α, forming representations colloquially referred to as 'positives', are sampled randomly from a set of augmentations $\mathcal{A}$ (i.e. $\alpha \sim \mathcal{A}$), which includes combinations of e.g. random crops, color jitter and horizontal flips. We note that modifying the set of positives, while using the same contrastive learning objective, results in different invariances being learned. We generalize the contrastive loss $\mathcal{L}_{\text{CL}}(\{\mathbf{z}_1,\ldots,\mathbf{z}_K\})$ as a generic learning rule

https://doi.org/10.52202/085713-5030

operating on a set of K positives $\mathbf{z}_1, \ldots, \mathbf{z}_K$. Then, we formalize the view-invariant loss as

$$\texttt{VI} := \mathcal{L}_{\text{CL}}(\{\varphi_{\texttt{VI}}(\alpha_k(\mathbf{x})|\mathbf{W}, \emptyset) \mid k = 1, \ldots, K,\ \alpha_k \sim \mathcal{A}\}), \tag{5}$$

where $\varphi_{\texttt{VI}} = h_{\texttt{VI}} \circ f$ and $h_{\texttt{VI}}(\cdot)$ is an MLP used exclusively for the view-invariant objective.

To consolidate the orthogonalizing modulations into the unmodulated representation space, we propose to employ *differently modulated* views of the source sample as positives, hence constructing a modulation-invariant representation space (Figure 1, top center). Concretely, we uniformly sample $\mathbf{m}_2, \ldots, \mathbf{m}_V \sim \mathcal{M}^{(s)}$ from the set of trained modulations at session s. We formalize the modulation-invariant loss, which implements objective (3), as

$$\begin{aligned}\texttt{MI} := \mathcal{L}_{\text{CL}}\Big(&\{p_{\texttt{MI}}(\varphi_{\texttt{MI}}(\alpha_1(\mathbf{x})|\mathbf{W}, \emptyset))\} \\ &\cup \Big\{ \text{sg}(\varphi_{\texttt{MI}}(\alpha_k(\mathbf{x})|\mathbf{W}, \mathbf{m}_k)) \;\Big|\; k = 2, \ldots, K,\ \alpha_k \sim \mathcal{A},\ \mathbf{m}_k \sim \mathcal{M}^{(s)} \Big\}\Big),\end{aligned} \tag{6}$$

where $\text{sg}(\cdot)$ denotes a stop-gradient, $\varphi_{\texttt{MI}} = h_{\texttt{MI}} \circ f$, and $p_{\texttt{MI}}(\cdot), h_{\texttt{MI}}(\cdot)$ are MLPs used exclusively for the modulation-invariant objective. Therefore, a compact representation of each new class is *consolidated* in the unmodulated representation space. During the consolidation phase of TMCL, the combination of VI and MI is optimized jointly.

Mapping prior work to the canonical contrastive loss and comparing to TMCL. Notably, in supervised contrastive learning (SupCon), samples of the same class are used as positives [93], thus implementing invariance to features not predictive of class identity. SupCon is a straightforward method to additionally leverage the few available labels in semi-supervised continual learning setups. However, in contrast to our MI objective, SupCon only separates classes for which labels are available in the current session, while MI implicitly separates current samples from past class centers.

State-of-the-art unsupervised continual representation learning algorithms such as CaSSLe [69] and PNR [72] train the network representations to be invariant to the model state (state invariance or SI), by using as positives one representation obtained from the current network state and one representation obtained from a stored, past network state. In TMCL, because the modulations for any given class are *not* updated after the initial learning, they effectively constitute an imprint of neural activities that separates that class from all others. Therefore, training the unmodulated representations to maintain similarity with this imprint also stabilizes continual learning, and can be understood as a form of SI that does not require storing the full network state twice (cf. Table A1, right).

4 Experiments

Experimental protocol. We adopt a standard class-incremental continual learning protocol on both CIFAR-100 and ImageNet-100, dividing the dataset into five sessions, each containing 20 disjoint classes. We additionally introduce a supervised cross-entropy (CE) baseline with a projection head [94], which has been reported to outperform self-supervised methods. For each session, we train the model for 100 orthogonalization epochs, where we train modulations for the new classes, and 200 consolidation epochs. We further start the first session with a pretraining phase of 250 epochs, where the feedforward weights are updated via VI, or analogously with SupCon and CE for the respective supervised methods. For all loss terms except CaSSLe and PNR, we use $K = 4$ positives. The backbone f is a modified ConViT architecture [95] as introduced in DyTox [65]. We emphasize that this architecture is equivalent to ResNet-18 in both memory and compute (Table A1, b). We evaluate the representations via linear probing of the last four layers of f as suggested by Caron et al. [88]. Further details are provided in supplementary materials A.

Semi-supervised continual representation learning. In Table 1, we present the all-vs-all linear readout accuracies on the CIFAR-100 and ImageNet-100 datasets after class-incremental learning. As previously reported [94], continual supervised learning outperforms self-supervised learning if all labels are observable, while purely supervised methods significantly degrade with only 10% of labels or fewer. Clearly, a combination of both supervision and self-supervision proves most performant in this setup, as VI + SupCon significantly outperforms the state-of-the-art unsupervised algorithm (VI + SI (PNR)). While *state invariance* is helpful, most of the improvement stems from the additional supervision (VI + SupCon). In the fully labeled scenario, VI + MI improves

Table 1: **Semi-supervised continual representation learning.** Final linear readout accuracy (all-vs-all) on class-incremental ImageNet-100 and CIFAR-100 with 5 sessions (averaged over three and four seeds respectively, ± denotes the standard deviation).

Method	CIFAR-100/5			ImageNet-100/5		
	100%	10%	1%	100%	10%	1%
Supervised methods						
SupCon	58.4±0.7	47.7±0.9	39.1±1.4	64.0±0.7	48.7±1.2	33.6±0.5
CE	60.1±0.5	50.7±0.3	41.6±0.5	66.0±0.4	50.1±1.2	35.2±0.2
Self-supervised methods						
VI		59.3±0.2			59.7±0.3	
. . . integrating class labels (semi-supervised)						
+ CE	60.6±0.6	59.5±0.3	59.8±0.2	64.5±0.7	61.4±0.9	60.3±0.5
+ SupCon	62.2±0.2	59.6±0.4	59.3±0.2	**66.6**±0.4	61.4±0.6	60.2±0.3
+ MI (**TMCL**)	60.7±0.4	**61.1**±0.3	60.7±0.3	64.5±0.3	**63.5**±0.4	**62.0**±0.2
. . . introducing state invariance (and class labels)						
+ SI (PNR)		60.2±0.2			59.6±1.0	
+ CE	61.2±0.3	60.3±0.6	60.1±0.3	64.5±0.5	60.3±0.8	59.4±0.9
+ SupCon	**62.7**±0.2	60.7±0.3	60.1±0.3	67.0±0.2	60.2±0.3	58.5±0.4
+ MI	60.9±0.2	60.7±0.2	**60.9**±0.2	63.8±0.9	62.7±0.6	61.7±0.4

Table 2: **Transfer learning.** Final all-vs-all kNN accuracy on diverse downstream tasks after five incremental CIFAR-100 sessions (averaged over four seeds, ± denotes the standard deviation).

	Method	Aircraft	CIFAR-10	CUBirds	DTD	EuroSAT	GTSRB	STL-10	SVHN	VGGFlower
1% CIFAR labels	SupCon	8.3±1.1	52.0±2.0	3.3±0.3	15.5±0.6	64.1±3.3	38.5±3.5	44.9±1.4	46.6±2.1	18.9±2.0
	+ SI (CaSSLe)	11.6±3.1	51.9±1.3	3.4±0.3	16.3±1.4	66.4±4.3	38.3±4.9	44.8±1.8	46.3±1.1	21.9±5.3
	VI	27.5±0.7	77.0±0.3	10.0±0.2	27.6±0.8	86.0±0.4	67.8±0.2	65.4±0.5	48.3±0.4	58.5±0.6
	+ SupCon	27.4±0.6	77.3±0.3	9.8±0.2	27.9±0.5	85.8±0.1	68.2±1.3	64.9±0.6	49.8±0.4	58.4±0.6
	+ MI (**TMCL**)	28.0±0.5	78.0±0.3	10.7±0.2	29.4±0.8	87.1±0.2	68.2±0.9	66.3±0.3	49.0±0.7	61.7±0.5
	+ SI (PNR)	28.5±0.5	78.3±0.1	11.1±0.1	28.6±0.7	87.0±0.2	69.4±0.4	67.0±0.4	49.1±0.8	64.7±0.5
	+ SupCon	29.1±0.2	78.3±0.2	10.5±0.4	28.2±0.4	87.0±0.5	**70.2**±0.2	66.8±0.3	**49.9**±0.6	64.4±0.3
	+ MI	**29.9**±0.8	**79.0**±0.2	**11.8**±0.3	**29.5**±0.3	**87.5**±0.2	69.8±0.9	**67.6**±0.3	49.7±0.7	**66.2**±0.4
100% C. l.	CE [94]	**29.0**±0.6	78.4±0.5	10.0±0.1	28.5±0.8	83.4±0.4	64.0±0.4	66.2±0.5	**53.2**±0.6	58.2±0.8
	VI	27.5±0.7	77.0±0.3	10.0±0.2	27.6±0.8	86.0±0.4	67.8±0.2	65.4±0.5	48.3±0.4	58.5±0.6
	+ SupCon	28.1±0.7	**79.1**±0.3	10.4±0.5	28.9±0.7	86.6±0.1	68.6±0.6	**66.8**±0.2	49.8±0.8	58.2±0.4
	+ MI	28.9±0.3	78.2±0.2	**10.7**±0.2	**29.2**±0.2	**87.0**±0.1	**71.0**±0.8	66.7±0.3	50.7±0.6	**62.8**±0.5

over VI + SI, yet underperforms VI + SupCon. This changes as we move towards label-sparse scenarios, where MI consistently outperforms SupCon. Interestingly, MI is not orthogonal to SI, as their combination results in the strongest performances in label-sparse scenarios.

Representational quality for transfer learning. In Table 2, we report k-nearest neighbors (kNN) performance of different tasks on top of the CIFAR-100 continually pretrained models, without any further fine-tuning or training. First, on those models that only observe 1% of the training labels (or no labels at all, i.e. VI, VI + SI), we demonstrate that modulation invariance improves representational quality beyond solely adapting to CIFAR-100 classes, as performances across almost all probed datasets improve. Here, MI is not orthogonal to SI, and the combination of VI+SI+MI results in the most transferable representations. Furthermore, amongst methods exploiting CIFAR-100 labels, MI outperforms SupCon, supporting the hypothesis that direct supervision results in representations that are 'greedily' invariant to features which are not relevant for the current task. Strikingly, this cannot be solely attributed to the sparse supervision setup; conducting the same study on models trained with 100% labels, we observe that MI outperforms SupCon, except for CIFAR-10 and STL-10, which are semantically similar to CIFAR-100. Furthermore, semi-supervised VI + MI outperforms fully supervised CE on most tasks, further underlining the lack of generalization of class-based invariance

https://doi.org/10.52202/085713-5030

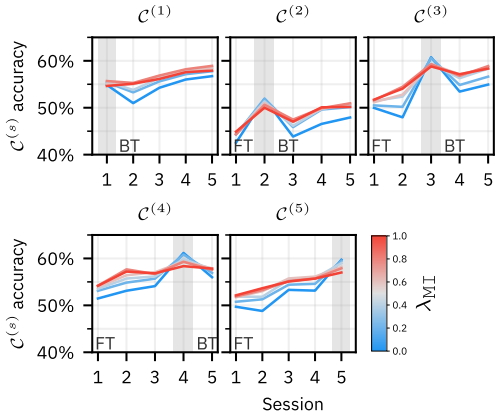

Figure 3: **Controlling the stability-plasticity trade-off.** Accuracy on CIFAR-100 test samples associated with $\mathcal{C}^{(s)}$, measuring forward and backward transfer (averaged over four seeds). Color scale indicates the strength of MI.

Table 4: **Backwards and forwards transfer performances,** using 1% of labels for SupCon and MI (averaged over four seeds, ± denotes the standard deviation).

Methods	BT	FT
SupCon	-18.5±0.4	-27.7±0.4
+ SI (CaSSLe)	-18.1±0.2	-27.4±0.4
VI	-1.7±0.4	-6.5±0.4
+ MI	-0.3±0.2	-3.7±0.4
+ SupCon	-1.9±0.1	-5.9±0.3
+ SI (CaSSLe)	1.0±0.1	-4.4±0.4
+ MI	**1.3**±0.2	**-3.0**±0.3
+ SupCon	1.0±0.1	-3.7±0.1

learning. In contrast to the CIFAR-100 results (Table 1), we observe that as more labels are provided to MI, representational transfer quality improves for most tasks.

Robustness to noisy labels. Erroneous labels are commonly encountered in natural learning environments. While in that case, misclassification of samples would be expected, the quality of the representations should not deteriorate. However, introducing label noise severely degrades performance in fully supervised methods (Table 3, red cells denoting performance inferior to the self-supervised baseline), and naive integration via CE also significantly underperforms the self-supervised baseline. While VI + SupCon only degrades with as much as 90% label noise, TMCL is robust to label noise up to 99% and does not underperform the self-supervised baseline. We hypothesize that this robustness arises because in TMCL, the erroneous labels do not directly affect weight learning. Rather, they lead to nonsensical modulations, which represent a noise contribution to which the contrastive learning machinery learns to become invariant.

Table 3: **Label noise.** Final linear readout accuracy on continual CIFAR-100, replacing a fraction of labels with random labels. (averaged over four seeds, ± denotes the standard deviation).

Method	Label noise			
	30%	**50%**	**90%**	**99%**
Fully supervised methods				
CE	57.3±0.2	54.4±0.2	8.2±1.2	6.4±2.2
SupCon	58.0±0.4	56.3±0.3	48.6±1.0	47.0±0.6
Self-supervised baseline				
VI	59.3±0.2			
. . . integrating (noisy) labels				
+ CE	59.2±0.6	59.2±0.2	58.3±0.4	58.1±0.5
+ SupCon	**60.6**±0.3	60.0±0.2	58.6±0.5	58.7±0.3
+ MI (**TMCL**)	60.4±0.4	**60.3**±0.1	**60.0**±0.5	**59.4**±0.4

The strength of MI controls the stability-plasticity trade-off. We first investigate how different methods perform before and after observing task samples. Therefore, we introduce backwards and forward transfer metrics that measure the difference in task performance pre- and post-session compared to a model that is trained solely on that particular task using VI (definitions provided in the supplementary material). We observe that MI strongly improves both backwards and forward transfer compared to pure VI, while SupCon only provides mediocre improvements in forward transfer and even degrades backwards transfer (Table 4). Introducing a model state invariance term significantly enhances backward transfer, demonstrating positive knowledge transfer from previously learned tasks. Remarkably, the combination of MI and SI on average achieves performance within 3 percentage points of the task-specific baseline model, demonstrating significant forward transfer. To further investigate the effect of MI, we vary its strength $\lambda_{\texttt{MI}} \in [0, 1]$ in the combination $\texttt{VI} + \lambda_{\texttt{MI}}\texttt{MI}$. In Figure 3, for each session s, we observe the accuracy of test samples associated with classes from $\mathcal{C}^{(s)}$ during the course of training. We observe that increasing $\lambda_{\texttt{MI}}$ enhances both forward transfer (left of the gray region) and backward transfer (right of the gray region), albeit at the expense of current task performance (gray region).

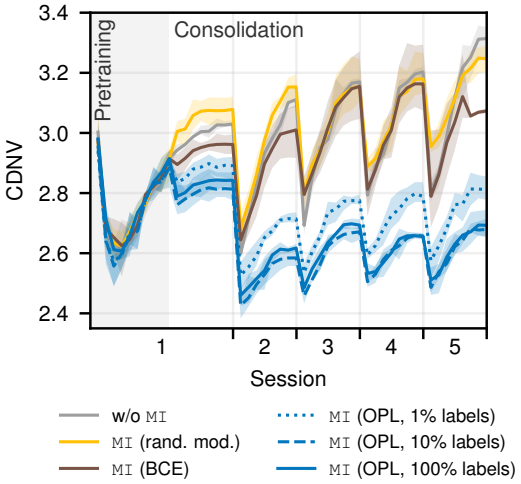

Figure 4: **CDNV during consolidation.** Measured on the CIFAR-100 test split on runs observing all labels (averaged over four seeds, shading indicates min. and max.)

Table 5: **Ablations.**

(a) **Ablating the role of orthogonalizing modulations.** Linear readout accuracy on continual CIFAR-100 and ImageNet-100 observing all labels (average, ± denotes the standard deviation).

Method	CIFAR-100	ImageNet-100
VI + MI (**TMCL**)	60.7±0.4	63.5±0.4
untrained mods.	60.1±0.5	60.6±1.5
random mods.	59.7±0.5	60.9±0.7
OPL → BCE	59.5±0.5	63.8±0.6
w/o MI	59.3±0.2	59.7±0.3

(b) **Ablating the role of architectural choices.** Linear readout accuracy on continual CIFAR-100 (averaged over four seeds, ± denotes the standard deviation).

Method	CIFAR-100
VI + MI (**TMCL**)	**60.7**±0.4
w/o pred.	55.1±0.1
w/o stop-grad	60.3±0.3
w/o pred. & stop-grad	60.3±0.1

Orthogonalized modulations improve class-separation. We hypothesized in Section 3 that BCE is suboptimal, since negatives are collapsed towards a single vector. As we replace the orthogonal projection loss with binary cross-entropy, we observe degraded performance on CIFAR-100 (Table 5b). On ImageNet-100, performance is similar, which we attribute to the fact that the BCE training does not fully converge to an antiparallel configuration: On CIFAR-100, the average minimum BCE loss over all classes is 0.005±0.039, while on ImageNet, it is 0.053±0.035. Interestingly, while untrained modulations (i.e. frozen after normal initialization) as well as random modulations (i.e. redrawn each time) degrade performance further, they still provide a moderate improvement over pure VI. This suggests that predicting representations based on a slightly perturbed model state already provides a regulatory effect. We further measure the class-distance normalized variance (CDNV, appendix E) of representations [97], which is the ratio between intra-class variance and class-mean distances. A lower CDNV thus indicates a combination of higher intra-class collapse and/or higher distances between classes. We observe that modulation invariance improves class-separation as the CDNV significantly decreases once training enters the consolidation stage, but only in combination with orthogonalizing modulations (Figure 4). Moreover, the effect is still clearly visible while using 1% labels.

Table 6: **Results on ResNet-18.** Reporting linear evaluation performances on continual CIFAR-100 (averaged over four seeds, ± denotes the standard deviation).

Method	Labeled frac.	
	100%	10%
VI	53.4±0.1	
+ MI (**TMCL**)	58.1±0.1	56.7±1.1
+ SupCon	54.5±0.2	54.6±0.5
+ SI (**PNR**)	59.9±0.3	
+ SI (**CaSSLe**)[a]	60.1±0.4	
+ SI (**PNR**)[a]	60.3±0.4	
+ ER[b] [61]	54.6	
+ DER[b] [48]	55.3	
+ LUMP[b] [96]	57.8	
+ Less-Forget[b] [51]	56.4	
+ POD[b] [56]	55.9	
CLS-ER [78]	56.7±0.2	49.8±0.2

[a] adapted from Cha et al. [72]
[b] adapted from Fini et al. [69]

Architectural ablations. The predictor network appears to be essential for MI, but only if a stop-gradient is imposed upon the modulated branches (Table 5b). Removing the stop-gradient while keeping the modulations frozen results in a moderate performance degradation, regardless of the presence of a predictor. Still, the stop-gradient is useful from both a computational perspective – reducing memory footprint and gradient computations – and from a biological perspective, as the

https://doi.org/10.52202/085713-5030

predictor with stop-gradient could be implemented by a hypothesized cortical predictive coding pathway [40].

ResNet architecture. As we replace ConViT with ResNet-18 (Table 6), we observe that `SI` significantly improves performance over standard `VI`, underlining the susceptibility of ResNets to forgetting compared to ViT-based architectures [98]. Introducing `MI` also significantly improves performance over `VI` as well as other label-integrating methods (`SupCon`, CLS-ER), but slightly underperforms `SI`. We hypothesize that the ability to directly modulate spatial relationships in ViTs lends more expressivity to the modulations in these models. Furthermore, the reliance of ResNet architectures on BatchNorms hinders stable training of different modulations, as it requires freezing the batch statistics. This is observable especially in the high standard deviation in accuracy of TMCL given 10% labels. As the underlying data statistics change with each session, the BatchNorms will adapt accordingly, and we hypothesize that this interferes with previously learned modulations. This is not the case for LayerNorm — as used by transformer architectures – which normalizes across features and does not depend on stored normalization statistics.

5 Limitations

Our work focuses on developing an understanding of algorithms that leverage the cortical circuitry, which is characterized by top-down and bottom-up information streams [99], to learn in naturalistic scenarios. Our analysis demonstrates this on a standard CIFAR-100 class-incremental setup, but omits data- and domain-incrementality. TMCL relies on static modulations, resulting in memory requirements that scale with the number of classes. For 100 classes, this amounts to around 4.1M parameters, while CaSSLe and PNR store a copy of the network (10.7M, Table A1b). Still, alternatives such as pruning simple classes or, better yet, a network that generates such modulations, should be explored. Finally, we observe that representational quality on the trained dataset (CIFAR-100) only moderately improves as more labels are presented, underperforming `SupCon` in the fully labeled regime. Still, given the improved transfer performance of TMCL, we hypothesize that this is indeed biologically plausible as the unmodulated network is not primed to solve CIFAR-100 specifically, but rather driven towards generalizable representations. Top-down priming could be implemented via a separate set of task-specific modulations, as has been explored in previous work [34–36].

6 Discussion

In this work, we have proposed a novel, brain-inspired algorithm for continual learning. It has been proposed that modulations provide a powerful framework for task-incremental learning, as they allow a general feature detector to learn new tasks by adapting modulations only [34, 36]. We extend this framework to class-incremental learning, and show that modulations can be consolidated into a shared representation space, sharpening percepts from data classes observed asynchronously. This consolidation co-opts the general machinery for view invariance learning, which in the brain is thought to be available anyway for predictive coding [15, 16]. Furthermore, there appear to be measurable parallels between our algorithm and cortical learning, as large-scale brain imaging indicates that representations of new stimuli orthogonalize from all others throughout learning, and unsupervised pretraining affects task learning [100]. Elucidating the drivers of this orthogonalization will provide further insight into how the brain leverages its biophysical machinery to achieve continual learning.

As we have shown that training for modulation invariance imparts task-specific information on the unmodulated network, one tantalizing possibility is that the effective learning objectives for networks, or parts of networks, could be tuned by incorporating sets of modulations that solve specific tasks. In a mixed language-vision approach [27], this could afford a fine-grained control over the eventual representation learning beyond what is currently possible with e.g. multi-task datasets [101]. In neurobiology, this represents a new view on intra-cortical and thalamocortical interactions. As cortical regions are targeted by modulations originating from distinct sets of brain areas with specific roles [102], the precise configuration of modulatory afferents could provide an understanding of the effective area-specific learning objectives. In turn, as these connectivity patterns are driven by genetically determined cues, this theory may afford insight into the distinct roles of genetics and plasticity in developing functional brains [103].

Acknowledgements

This work was funded by the Federal Ministry of Education and Research (BMBF) under grant no. 01IS22094E WEST-AI as well as by Helmholtz Association's project-oriented funding programme (PoF 2, Topic 3). Furthermore, the authors gratefully acknowledge computing time on the supercomputers JURECA [104] at Forschungszentrum Jülich under grant no. jinm60. The authors also gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS [105] at Jülich Supercomputing Centre (JSC). The authors also extend their gratitude to Sven Krauße for providing thoughtful comments on the manuscript.

References

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, and others. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, december 4-9, 2017, long beach, CA, USA*, pages 506–516, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/e7b24b112a44fdd9ee93bdf998c6ca0e-Abstract.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 07 May 2024 20:04:34 +0200.

[3] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. ISSN 18792782. doi: 10.1016/j.neunet.2019.01.012. Publisher: Elsevier Ltd.

[4] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=CZZFRxb0LC.

[5] Rodney J. Douglas and Kevan A.C. Martin. Neuronal Circuits of the Neocortex. *Annual Review of Neuroscience*, 27(1):419–451, July 2004. ISSN 0147-006X, 1545-4126. doi: 10.1146/annurev.neuro.27.070203.144152. URL https://www.annualreviews.org/doi/10.1146/annurev.neuro.27.070203.144152.

[6] Satoshi Manita, Takayuki Suzuki, Chihiro Homma, Takashi Matsumoto, Maya Odagawa, Kazuyuki Yamada, Keisuke Ota, Chie Matsubara, Ayumu Inutsuka, Masaaki Sato, Masamichi Ohkura, Akihiro Yamanaka, Yuchio Yanagawa, Junichi Nakai, Yasunori Hayashi, Matthew E. Larkum, and Masanori Murayama. A Top-Down Cortical Circuit for Accurate Sensory Perception. *Neuron*, 86(5):1304–1316, 2015. ISSN 10974199. doi: 10.1016/j.neuron.2015.05.006. URL http://dx.doi.org/10.1016/j.neuron.2015.05.006. Publisher: Elsevier Inc.

[7] Mathieu Lafourcade, Marie-Sophie H. Van Der Goes, Dimitra Vardalaki, Norma J. Brown, Jakob Voigts, Dae Hee Yun, Minyoung E. Kim, Taeyun Ku, and Mark T. Harnett. Differential dendritic integration of long-range inputs in association cortex via subcellular changes in synaptic AMPA-to-NMDA receptor ratio. *Neuron*, 110(9):1532–1546.e4, May 2022. ISSN 08966273. doi: 10.1016/j.neuron.2022.01.025. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627322000642.

[8] Morgane M Roth, Johannes C Dahmen, Dylan R Muir, Fabia Imhof, Francisco J Martini, and Sonja B Hofer. Thalamic nuclei convey diverse contextual information to layer 1 of visual cortex. *Nature Neuroscience*, 19(2):299–307, February 2016. ISSN 1097-6256, 1546-1726. doi: 10.1038/nn.4197. URL http://www.nature.com/articles/nn.4197.

[9] Naoya Takahashi, Thomas G Oertner, Peter Hegemann, and Matthew E Larkum. Active cortical dendrites modulate perception. *Science*, 354(6319):1587–90, 2016. ISSN 0036-8075. doi: 10.1126/science.aah6066.

[10] Guy Doron, Jiyun N. Shin, Naoya Takahashi, Moritz Drüke, Christina Bocklisch, Salina Skenderi, Lisa de Mont, Maria Toumazou, Julia Ledderose, Michael Brecht, Richard Naud, and Matthew E. Larkum. Perirhinal input to neocortical layer 1 controls learning. *Science*, 370(6523):eaaz3136, December 2020. doi: 10.1126/science.aaz3136. URL https://www.science.org/doi/10.1126/science.aaz3136. Publisher: American Association for the Advancement of Science.

[11] Benjamin Schuman, Schlomo Dellal, Alvar Prönneke, Robert Machold, and Bernardo Rudy. Neocortical Layer 1: An Elegant Solution to Top-Down and Bottom-Up Integration. *Annual Review of Neuroscience*, 44(March):221–252, March 2021. doi: 10.1146/annurev-neuro-100520-012117. URL https://www.annualreviews.org/doi/epdf/10.1146/annurev-neuro-100520-012117.

[12] Matthew E. Larkum, Thomas Nevian, Maya Sandler, Alon Polsky, and Jackie Schiller. Synaptic Integration in Tuft Dendrites of Layer 5 Pyramidal Neurons: A New Unifying Principle. *Science*, 325(5941):756–760, August 2009. ISSN 0036-8075. doi: 10.1126/science.1171958. URL http://www.sciencemag.org/cgi/doi/10.1126/science.1171958.

[13] Björn M. Kampa, Johannes J. Letzkus, and Greg J. Stuart. Requirement of dendritic calcium spikes for induction of spike-timing-dependent synaptic plasticity. *The Journal of Physiology*, 574(1):283–290, July 2006. ISSN 0022-3751, 1469-7793. doi: 10.1113/jphysiol.2006.111062. URL https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.2006.111062.

[14] Björn M. Kampa, Johannes J. Letzkus, and Greg J. Stuart. Dendritic mechanisms controlling spike-timing-dependent synaptic plasticity. *Trends in Neurosciences*, 30(9):456–463, September 2007. ISSN 01662236. doi: 10.1016/j.tins.2007.06.010. URL https://linkinghub.elsevier.com/retrieve/pii/S0166223607001798.

[15] Bernd Illing, Jean Ventura, Guillaume Bellec, and Wulfram Gerstner. Local plasticity rules can learn deep representations using self-supervised contrastive predictions. In *Advances in Neural Information Processing Systems*, volume 34, pages 30365–30379. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/feade1d2047977cd0cefdafc40175a99-Abstract.html.

[16] Manu Srinath Halvagal and Friedemann Zenke. The combination of Hebbian and predictive plasticity learns invariant object representations in deep sensory networks. *Nature Neuroscience*, 26(11):1906–1915, November 2023. ISSN 1097-6256, 1546-1726. doi: 10.1038/s41593-023-01460-y. URL https://www.nature.com/articles/s41593-023-01460-y. Publisher: Springer Science and Business Media LLC.

[17] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. October 2021. URL https://openreview.net/forum?id=xm6YD62D1Ub.

[18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, January 2019. URL http://arxiv.org/abs/1807.03748. arXiv:1807.03748 [cs].

[19] Willem A. M. Wybo, Matthias C. Tsai, Viet Anh Khoa Tran, Bernd Illing, Jakob Jordan, Abigail Morrison, and Walter Senn. NMDA-driven dendritic modulation enables multitask representation learning in hierarchical sensory processing pathways. *Proceedings of the National Academy of Sciences*, 120(32):e2300558120, August 2023. doi: 10.1073/pnas.2300558120. URL https://www.pnas.org/doi/10.1073/pnas.2300558120.

[20] Samantha R. Debes and Valentin Dragoi. Suppressing feedback signals to visual cortex abolishes attentional modulation. *Science*, 379(6631):468–473, February 2023. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.ade1855. URL https://www.science.org/doi/10.1126/science.ade1855.

[21] Dina V. Popovkina and Anitha Pasupathy. Task Context Modulates Feature-Selective Responses in Area V4. *The Journal of Neuroscience*, 42(33):6408–6423, August 2022. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.1386-21.2022. URL https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.1386-21.2022.

[22] Sanne Rutten, Roberta Santoro, Alexis Hervais-Adelman, Elia Formisano, and Narly Golestani. Cortical encoding of speech enhances task-relevant acoustic information. *Nature Human Behaviour*, 3(9):974–987, September 2019. ISSN 2397-3374. doi: 10.1038/s41562-019-0648-9. URL https://www.nature.com/articles/s41562-019-0648-9. Publisher: Nature Publishing Group.

[23] Patrick J. Mineault, Elaine Tring, Joshua T. Trachtenberg, and Dario L. Ringach. Enhanced Spatial Resolution During Locomotion and Heightened Attention in Mouse Primary Visual Cortex. *Journal of Neuroscience*, 36(24):6382–6392, June 2016. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0430-16.2016. URL https://www.jneurosci.org/content/36/24/6382. Publisher: Society for Neuroscience Section: Articles.

[24] Laura Busse, Jessica A. Cardin, M. Eugenia Chiappe, Michael M. Halassa, Matthew J. McGinley, Takayuki Yamashita, and Aman B. Saleem. Sensation during Active Behaviors. *The Journal of Neuroscience*, 37(45):10826–10834, November 2017. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.1828-17.2017. URL https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.1828-17.2017.

[25] Serin Atiani, Mounya Elhilali, Stephen V. David, Jonathan B. Fritz, and Shihab A. Shamma. Task Difficulty and Performance Induce Diverse Adaptive Patterns in Gain and Shape of Primary Auditory Cortical Receptive Fields. *Neuron*, 61(3):467–480, February 2009. ISSN 0896-6273. doi: 10.1016/j.neuron.2008.12.027. URL https://www.cell.com/neuron/abstract/S0896-6273(09)00006-3. Publisher: Elsevier.

[26] Morgane M. Roth, Johannes C. Dahmen, Dylan R. Muir, Fabia Imhof, Francisco J. Martini, and Sonja B. Hofer. Thalamic nuclei convey diverse contextual information to layer 1 of visual cortex. *Nature Neuroscience*, 19(2):299–307, February 2016. ISSN 1546-1726. doi: 10.1038/nn.4197. URL https://www.nature.com/articles/nn.4197. Publisher: Nature Publishing Group.

[27] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer, December 2017. URL http://arxiv.org/abs/1709.07871. arXiv:1709.07871 [cs].

[28] Jonathan Frankle, David J. Schwab, and Ari S. Morcos. Training BatchNorm and Only BatchNorm: On the Expressive Power of Random Features in CNNs, March 2021. URL http://arxiv.org/abs/2003.00152. arXiv:2003.00152 [cs].

[29] Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. TinyTL: Reduce Activations, Not Trainable Parameters for Efficient On-Device Learning, June 2021. URL http://arxiv.org/abs/2007.11622. arXiv:2007.11622 [cs].

[30] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning, August 2022. URL http://arxiv.org/abs/2205.05638. arXiv:2205.05638.

[31] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & Shifting Your Features: A New Baseline for Efficient Model Tuning. *Advances in Neural Information Processing Systems*, 35:109–123, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/00bb4e415ef117f2dee2fc3b778d806d-Abstract-Conference.html.

[32] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

[33] Gido M. Van De Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, December 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00568-3. URL https://www.nature.com/articles/s42256-022-00568-3.

[34] Nicolas Y. Masse, Gregory D. Grant, and David J. Freedman. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *Proceedings of the National Academy of Sciences*, 115(44), October 2018. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1803839115. URL https://pnas.org/doi/full/10.1073/pnas.1803839115.

[35] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020.

[36] Abhiram Iyer, Karan Grewal, Akash Velu, Lucas Oliveira Souza, Jeremy Forest, and Subutai Ahmad. Avoiding Catastrophe: Active Dendrites Enable Multi-Task Learning in Dynamic Environments. *Frontiers in Neurorobotics*, 16:846219, April 2022. ISSN 1662-5218. doi: 10.3389/fnbot.2022.846219. URL http://arxiv.org/abs/2201.00042. arXiv:2201.00042 [cs].

[37] Nicolas Deperrois, Mihai A. Petrovici, Walter Senn, and Jakob Jordan. Learning beyond sensations: How dreams organize neuronal representations. *Neuroscience & Biobehavioral Reviews*, 157:105508, February 2024. ISSN 0149-7634. doi: 10.1016/j.neubiorev.2023.105508. URL https://www.sciencedirect.com/science/article/pii/S0149763423004773.

[38] Fabian A Mikulasch, Lucas Rudelt, and Viola Priesemann. Local dendritic balance enables learning of efficient representations in networks of spiking neurons. *Proceedings of the National Academy of Sciences*, 118(50):e2021925118, July 2021. doi: 10.1073/pnas.2021925118.

[39] Fabian A. Mikulasch, Lucas Rudelt, Michael Wibral, and Viola Priesemann. Where is the error? Hierarchical predictive coding through dendritic error computation. *Trends in Neurosciences*, 46(1): 45–59, January 2023. ISSN 01662236. doi: 10.1016/j.tins.2022.09.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0166223622001862.

[40] Kevin Kermani Nejad, Paul Anastasiades, Loreen Hertäg, and Rui Ponte Costa. Self-supervised predictive learning accounts for cortical layer-specificity, April 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.04.24.590916.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[42] Anthony V. Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995. doi: 10.1080/09540099550039318. URL https://doi.org/10.1080/09540099550039318. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Fri, 26 May 2017 22:53:37 +0200.

[43] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *2017 IEEE conference on computer vision and pattern recognition, CVPR 2017, honolulu, HI, USA, july 21-26, 2017*, pages 5533–5542. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.587. URL https://doi.org/10.1109/CVPR.2017.587. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Fri, 24 Mar 2023 00:02:54 +0100.

[44] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, december 4-9, 2017, long beach, CA, USA*, pages 6467–6476, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Thu, 21 Jan 2021 15:15:21 +0100.

[45] Oleksiy Ostapenko, Mihai Marian Puscas, Tassilo Klein, Patrick Jähnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *IEEE conference on computer vision and pattern recognition, CVPR 2019, long beach, CA, USA, june 16-20, 2019*, pages 11321–11329. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.01158. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Ostapenko_Learning_to_Remember_A_Synaptic_Plasticity_Driven_Framework_for_Continual_CVPR_2019_paper.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Wed, 07 Dec 2022 23:06:27 +0100.

[46] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *7th international conference on learning representations, ICLR 2019, new orleans, LA, USA, may 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Thu, 25 Jul 2019 14:25:58 +0200.

[47] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

[48] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, NeurIPS 2020, december 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b704ea2c39778f07c617f6b7ce480e9e-Abstract.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 19 Jan 2021 15:57:21 +0100.

[49] Md Yousuf Harun, Jhair Gallardo, Tyler L. Hayes, Ronald Kemker, and Christopher Kanan. SIESTA: efficient online continual learning with sleep. 2023, 2023. URL https://openreview.net/forum?id=MqDVlBWRRV. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Thu, 01 Aug 2024 15:37:25 +0200.

[50] Yibo Yang, Haobo Yuan, Xiangtai Li, Jianlong Wu, Lefei Zhang, Zhouchen Lin, Philip Torr, Dacheng Tao, and Bernard Ghanem. Neural Collapse Terminus: A Unified Solution for Class Incremental Learning and Its Variants, August 2023. URL http://arxiv.org/abs/2308.01746. arXiv:2308.01746.

[51] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE conference on computer vision and pattern recognition, CVPR 2019, long beach, CA, USA, june 16-20, 2019*, pages 831–839. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00092. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Hou_Learning_a_Unified_Classifier_Incrementally_via_Rebalancing_CVPR_2019_paper.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Mon, 03 Mar 2025 21:01:29 +0100.

[52] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE conference on computer vision and pattern recognition, CVPR 2019, long beach, CA, USA, june 16-20, 2019*, pages 374–382. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00046. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Wu_Large_Scale_Incremental_Learning_CVPR_2019_paper.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Mon, 30 Aug 2021 17:01:14 +0200.

[53] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer vision - ECCV 2018 - 15th european conference, munich, germany, september 8-14, 2018, proceedings, part III*, volume 11207 of *Lecture notes in computer science*, pages 144–161. Springer, 2018. doi: 10.1007/978-3-030-01219-9_9. URL https://doi.org/10.1007/978-3-030-01219-9_9. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Mon, 30 Dec 2024 20:27:58 +0100.

[54] Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer vision - ECCV 2018 - 15th european conference, munich, germany, september 8-14, 2018, proceedings, part XII*, volume 11216 of *Lecture notes in computer science*, pages 241–257. Springer, 2018. doi: 10.1007/978-3-030-01258-8_15. URL https://doi.org/10.1007/978-3-030-01258-8_15. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 01 Apr 2025 19:06:38 +0200.

[55] Arslan Chaudhry, Puneet Kumar Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer vision - ECCV 2018 - 15th european conference, munich, germany, september 8-14, 2018, proceedings, part XI*, volume 11215 of *Lecture notes in computer science*, pages 556–572. Springer, 2018. doi: 10.1007/978-3-030-01252-6_33. URL https://doi.org/10.1007/978-3-030-01252-6_33. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 14 May 2019 10:00:45 +0200.

[56] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled outputs distillation for small-tasks incremental learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer vision - ECCV 2020 - 16th european conference, glasgow, UK, august 23-28, 2020, proceedings, part XX*, volume 12365 of *Lecture notes in computer science*, pages 86–102. Springer, 2020. doi: 10.1007/978-3-030-58565-5_6. URL https://doi.org/10.1007/978-3-030-58565-5_6. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Sun, 02 Oct 2022 15:59:31 +0200.

[57] Enrico Fini, Stéphane Lathuilière, Enver Sangineto, Moin Nabi, and Elisa Ricci. Online continual learning under extreme memory constraints. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer vision - ECCV 2020 - 16th european conference, glasgow, UK, august 23-28, 2020, proceedings, part XXVIII*, volume 12373 of *Lecture notes in computer science*, pages 720–735. Springer, 2020. doi: 10.1007/978-3-030-58604-1_43. URL https://doi.org/10.1007/978-3-030-58604-1_43. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Sun, 04 Aug 2024 19:40:12 +0200.

[58] Zhizhong Li and Derek Hoiem. Learning without forgetting. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer vision - ECCV 2016 - 14th european conference, amsterdam, the netherlands, october 11-14, 2016, proceedings, part IV*, volume 9908 of *Lecture notes in computer science*, pages 614–629. Springer, 2016. doi: 10.1007/978-3-319-46493-0_37. URL https://doi.org/10.1007/978-3-319-46493-0_37. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 21 Mar 2023 20:52:16 +0100.

[59] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co²L: Contrastive continual learning. In *2021 IEEE/CVF international conference on computer vision, ICCV 2021, montreal, QC, canada, october 10-17, 2021*, pages 9496–9505. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00938. URL https://doi.org/10.1109/ICCV48922.2021.00938. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Wed, 27 Sep 2023 21:14:23 +0200.

[60] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, december 4-9, 2017, long beach, CA, USA*, pages 2990–2999, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/0efbe98067c6c73dba1250d2beaa81f9-Abstract.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Fri, 16 Dec 2022 08:18:46 +0100.

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

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

[63] Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th international conference on machine learning, ICML 2018, stockholmsmässan, stockholm, sweden, july 10-15, 2018*, volume 80 of *Proceedings of machine learning research*, pages 4555–4564. PMLR, 2018. URL http://proceedings.mlr.press/v80/serra18a.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Wed, 03 Apr 2019 18:17:30 +0200.

[64] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016. URL http://arxiv.org/abs/1606.04671. arXiv: 1606.04671 tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Mon, 13 Aug 2018 16:46:11 +0200.

[65] Arthur Douillard, Alexandre Rame, Guillaume Couairon, and Matthieu Cord. DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9275–9285, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.00907. URL https://ieeexplore.ieee.org/document/9880199/.

[66] Shipeng Yan, Jiangwei Xie, and Xuming He. DER: dynamically expandable representation for class incremental learning. In *IEEE conference on computer vision and pattern recognition, CVPR 2021, virtual, june 19-25, 2021*, pages 3014–3023. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00303. URL https://openaccess.thecvf.com/content/CVPR2021/html/Yan_DER_Dynamically_Expandable_Representation_for_Class_Incremental_Learning_CVPR_2021_paper.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Mon, 18 Jul 2022 16:47:40 +0200.

[67] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, seattle, WA, USA, june 13-19, 2020*, pages 13205–13214. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01322. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Zhao_Maintaining_Discrimination_and_Fairness_in_Class_Incremental_Learning_CVPR_2020_paper.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Mon, 16 Sep 2024 10:52:02 +0200.

[68] 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.

[69] Enrico Fini, Victor G. Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-Supervised Models are Continual Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9611–9620, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.00940. URL https://ieeexplore.ieee.org/document/9878593/.

[70] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *The tenth international conference on learning representations, ICLR 2022, virtual event, april 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=9Hrka5PA7LW. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Sat, 20 Aug 2022 01:15:42 +0200.

[71] Dapeng Hu, Shipeng Yan, Qizhengqiu Lu, Lanqing Hong, Hailin Hu, Yifan Zhang, Zhenguo Li, Xinchao Wang, and Jiashi Feng. How Well Does Self-Supervised Pre-Training Perform with Streaming Data?, July 2022. URL http://arxiv.org/abs/2104.12081. arXiv:2104.12081 [cs].

[72] Sungmin Cha, Kyunghyun Cho, and Taesup Moon. Regularizing with pseudo-negatives for continual self-supervised learning. In *Forty-first international conference on machine learning*, 2024. URL https://openreview.net/forum?id=9jXSO7TIBH.

[73] Quang Pham, Chenghao Liu, and Steven C. H. Hoi. DualNet: Continual learning, fast and slow. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021, NeurIPS 2021, december 6-14, 2021, virtual*, pages 16131–16144, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/86a1fa88adb5c33bd7a68ac2f9f3f96b-Abstract.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 03 May 2022 16:20:48 +0200.

[74] 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 *IEEE/CVF winter conference on applications of computer vision, WACV 2024, waikoloa, HI, USA, january 3-8, 2024*, pages 2829–2838. IEEE, 2024. doi: 10.1109/WACV57701.2024.00282. URL https://doi.org/10.1109/WACV57701.2024.00282. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Wed, 17 Apr 2024 11:21:26 +0200.

[75] Alex Gomez-Villa, Bartlomiej Twardowski, Kai Wang, and Joost Van de Weijer. Plasticity-optimized complementary networks for unsupervised continual learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1690–1700, 2024.

[76] Xiaofan Yu, Tajana Rosing, and Yunhui Guo. Evolve: Enhancing unsupervised continual learning with multiple experts. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2366–2377, 2024.

[77] 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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3866–3876, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-8739-9. doi: 10.1109/CVPRW56347.2022.00432. URL https://ieeexplore.ieee.org/document/9857174/.

[78] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *The tenth international conference on learning representations, ICLR 2022, virtual event, april 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=uxxFrDwrE7Y. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Sat, 20 Aug 2022 01:15:42 +0200.

[79] James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, July 1995. ISSN 0033-295X. doi: 10.1037/0033-295X.102.3.419.

[80] Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated. *Trends in Cognitive Sciences*, 20(7):512–534, July 2016. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2016.05.004. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(16)30043-2. Publisher: Elsevier.

[81] Kanchana Ranasinghe, Muzammal Naseer, Munawar Hayat, Salman Khan, and Fahad Shahbaz Khan. Orthogonal Projection Loss, March 2021. URL http://arxiv.org/abs/2103.14021. arXiv:2103.14021.

[82] Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of Neural Collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, October 2020. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2015509117. URL http://arxiv.org/abs/2008.08186. arXiv:2008.08186 [cs].

[83] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, NeurIPS 2020, december 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/70feb62b69f16e0238f741fab228fec2-Abstract.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 19 Jan 2021 15:57:31 +0100.

[84] 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.

[85] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, NeurIPS 2020, december 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 19 Jan 2021 15:57:04 +0100.

[86] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, seattle, WA, USA, june 13-19, 2020*, pages 9726–9735. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00975. URL https://doi.org/10.1109/CVPR42600.2020.00975. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Tue, 31 Aug 2021 14:00:04 +0200.

[87] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer vision - ECCV 2020 - 16th european conference, glasgow, UK, august 23-28, 2020, proceedings, part XI*, volume 12356 of *Lecture notes in computer science*, pages 776–794. Springer, 2020. doi: 10.1007/978-3-030-58621-8_45. URL https://doi.org/10.1007/978-3-030-58621-8_45. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Fri, 27 Nov 2020 15:04:44 +0100.

[88] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision, ICCV 2021, montreal, QC, canada, october 10-17, 2021*, pages 9630–9640. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00951. URL https://doi.org/10.1109/ICCV48922.2021.00951. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Fri, 11 Mar 2022 10:01:59 +0100.

[89] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE conference on computer vision and pattern recognition, CVPR 2021, virtual, june 19-25, 2021*, pages 15750–15758. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01549. URL https://openaccess.thecvf.com/content/CVPR2021/html/Chen_Exploring_Simple_Siamese_Representation_Learning_CVPR_2021_paper.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Mon, 18 Jul 2022 16:47:41 +0200.

[90] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th international conference on machine learning, ICML 2021, 18-24 july 2021, virtual event*, volume 139 of *Proceedings of machine learning research*, pages 12310–12320. PMLR, 2021. URL http://proceedings.mlr.press/v139/zbontar21a.html. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Thu, 01 Jun 2023 15:27:03 +0200.

[91] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled Contrastive Learning. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 668–684, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19809-0. doi: 10.1007/978-3-031-19809-0_38.

[92] Thomas Yerxa, Yilun Kuang, Eero Simoncelli, and SueYeon Chung. Learning efficient coding of natural images with maximum manifold capacity representations. *Advances in Neural Information Processing Systems*, 36:24103–24128, 2023.

[93] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[94] Daniel Marczak, Sebastian Cygert, Tomasz Trzciński, and Bartłomiej Twardowski. Revisiting supervision for continual representation learning. In *European conference on computer vision*, pages 181–197. Springer, July 2024. doi: 10.48550/arXiv.2311.13321.

[95] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases. *International conference on machine learning*, 2022:2286–2296, 2021. ISSN 1742-5468. URL http://arxiv.org/abs/2103.10697. arXiv:2103.10697 [cs].

[96] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International conference on learning representations*, 2022. URL https://openreview.net/forum?id=9Hrka5PA7LW.

[97] Tomer Galanti, András György, and Marcus Hutter. On the role of neural collapse in transfer learning. In *The tenth international conference on learning representations, ICLR 2022, virtual event, april 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=SwIp410B6aQ. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Sat, 20 Aug 2022 01:15:42 +0200.

[98] Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture Matters in Continual Learning, February 2022. URL http://arxiv.org/abs/2202.00275. arXiv:2202.00275.

[99] Charles D. Gilbert and Wu Li. Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14(5):350–363, 2013. ISSN 1471003X. doi: 10.1038/nrn3476.

[100] Lin Zhong, Scott Baptista, Rachel Gattoni, Jon Arnold, Daniel Flickinger, Carsen Stringer, and Marius Pachitariu. Unsupervised pretraining in biological neural networks. *Nature*, pages 1–8, 2025.

[101] Amir R. Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling Task Transfer Learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, Salt Lake City, UT, June 2018. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00391. URL https://ieeexplore.ieee.org/document/8578489/.

[102] Steven E. Petersen, Benjamin A. Seitzman, Steven M. Nelson, Gagan S. Wig, and Evan M. Gordon. Principles of cortical areas and their implications for neuroimaging. *Neuron*, 112(17):2837–2853, September 2024. ISSN 08966273. doi: 10.1016/j.neuron.2024.05.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627324003556.

[103] Anthony M. Zador. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nature Communications*, 10(1):3770, August 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-11786-6. URL https://www.nature.com/articles/s41467-019-11786-6.

[104] Jülich Supercomputing Centre. JURECA: Data Centric and Booster Modules implementing the Modular Supercomputing Architecture at Jülich Supercomputing Centre. *Journal of large-scale research facilities*, 7(A182), 2021. doi: 10.17815/jlsrf-7-182. URL http://dx.doi.org/10.17815/jlsrf-7-182.

[105] Jülich Supercomputing Centre. JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre. *Journal of large-scale research facilities*, 7(A183), 2021. doi: 10.17815/jlsrf-7-183. URL http://dx.doi.org/10.17815/jlsrf-7-183.

[106] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th international conference on learning representations, ICLR 2019, new orleans, LA, USA, may 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Thu, 25 Jul 2019 14:26:04 +0200.

[107] 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.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim TMCL is effective in learning representations, improving over pure self-supervised approaches with sparse supervision. We underline this claim mainly by showing improvements on the main task (CIFAR-100, Table 1) and most importantly as we evaluate our representations on different tasks (Table 2) in Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the main limitations in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We focus on the biological correlates in this work, there are not theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All implementation details required to re-implement our methods – as well as pseudo-code – are contained in the supplementary material. We use publicly available datasets and standard benchmarks from literature.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

https://doi.org/10.52202/085713-5030

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use publicly available standard datasets and submit our code. Once submitted, we will publish our code to GitHub.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and test details are included in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All results are obtained over four different pseudo-random number generator seeds and we provide the standard deviation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Training infrastructure, approximate execution time and memory usage are reported in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors acknowledge and have ensured that the conducted research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper describes a continual learning method, highlighting potential biological correlates and therefore does not pose any striking societal impacts beyond the ones posed by fundamental machine learning and neuroscientific research.

https://doi.org/10.52202/085713-5030

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release any data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide proper acknowledgements for the datasets, libraries and figure assets used in the supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

https://doi.org/10.52202/085713-5030

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs have solely been used to speed up manual implementations of the methods by per-line autocompletion. Functionally, all parts have been implemented by the authors.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Implementation Details

Table A1: **Methodological differences.**

(a) **Learning protocol.** Number of epochs are given per session.

	Ours	[69, 72]
n_{views}	4	2
Pretraining epochs	250 (CIFAR-100), 200 (ImageNet-100)	-
Consolidation epochs	200	500
Orthogonalization epochs	100	-

(b) **Architectures.** Architecture numbers assuming 32×32 image inputs and 100 classes.

	ConViT (DyTox) [65, 95]	**ResNet-18** [41]
Parameters	10.7M	11.2M
+ modulations	4.1M	-
FLOPS	1.4G	1.4G

For all CIFAR-100 experiments, we use the same class split as in CaSSLe, i.e. the same across all seeds. For the ImageNet-100 as well as the 10 session experiments (Section F), we use different class splits per seed.

For ConViT experiments, we use the AdamW optimizer [106] with a batch size of 256 and feedforward weight decay of 0.0001 for all experiments, using a per-session cosine learning rate decay with 10 warmup epochs. ConViT experiments are trained with a feedforward learning rate of 0.001 and a modulation learning rate of 0.01. The ResNet experiments are trained with a feedforward learning rate of 1.0 and a modulation learning rate of 0.3 using the LARS optimizer ($\eta = 0.02$). The Barlow Twins losses are scaled down by a factor of 0.1 for ConViT experiments, and by 0.025 for ResNet experiments. We pick the redundancy-reduction weighting factor $\lambda_{\text{BT}} = 0.005$ for all experiments.

The ConViT backbone has 5 'local' self-attention blocks, replacing the self-attention layers with gated-positional self-attention layers, followed by a 'global' self-attention block with standard self-attention. We use 12 attention heads and a model dimension of 384. All images are resized to input size 32 and we use a patch size of 4. The ResNet backbone conforms to the original ResNet-18, except that the first convolution layer has a kernel size of 3 and padding 2, the first MaxPool is removed, and we remove the final MLP layers.

The code for our experiments is available at `https://github.com/tran-khoa/tmcl`.

Modulations. The gain and bias modulations are applied to the query, key, value and output projections of all multi-head attention modules, and to both layers of the feedforward MLPs that follow the attention modules. Additionally, we modulate the positional prior of the ConViT-specific gated-positional self-attention (GPSA) layers, i.e. we modulate the operation $\mathbf{v}_{\text{pos}}^{h\,\text{T}}\mathbf{r}_{ij}$ (cf. Equation 7 from d'Ascoli et al. [95]). Gain and bias modulations are initialized from a Gaussian distribution, respectively from $\mathcal{N}(1, 0.02)$ and $\mathcal{N}(0, 0.02)$. We impose weight decay on the modulations with decreasing strength for deeper layers,

$$\texttt{weight-decay}(l) := 0.4 - 0.36(\cos(\pi \cdot l/L)) + 1)/2 \tag{7}$$

for modulations of the l-th ViT layer (out of $L = 6$ layers). Only random horizontal flips are used as augmentations for the modulations.

Orthogonalization We virtually constrain the number of batches to the actual number of samples divided by the batch size. Let $\mathcal{C}_t$ be the set of classes available at session t. For each batch, we sample uniformly (with replacement) $c \sim \mathcal{C}_t$. Let X_c be the set of training samples of class c and $X_{\neg c}$ be all other samples available at session t. Then, with probability 0.5, each class is sampled uniformly from X_c, or class $X_{\neg c}$ otherwise.

$$P(x_t = x) = 0.5 \cdot P(x \sim \text{Uniform}(X_c)) + 0.5 \cdot P(x \sim \text{Uniform}(X_{\neg c})). \tag{8}$$

Consolidation The projector h is a three-layer MLP (dimensions `2028, 2048, 2048`) with ReLU activation and BatchNorm in the hidden layers. The predictor p for `MI`, CaSSLe and PNR is a two-layer MLP (dimensions `2048, 2048`) with ReLU activation and BatchNorm in the input layer. All projectors and predictors are reset at the end of each session.

For SupCon, we use a two-layer MLP (dimensions `2048, 128`) with ReLU activation and BatchNorm in the input layer, and we use a temperature of 0.1 in the softmax.

As augmentations, we use

https://doi.org/10.52202/085713-5030

```
RandomResizedCrop(
    size=(32),
    scale=(0.08, 1.0),
    ratio=(3.0 / 4.0, 4.0 / 3.0),
    resample=Resample.BICUBIC,
),
ColorJitter(
    brightness=0.4,
    contrast=0.4,
    saturation=0.2,
    hue=0.1,
    p=0.8,
),
RandomGrayscale(p=0.2),
RandomHorizontalFlip(p=0.5),
RandomSolarize(p=0.2, thresholds=0.0, additions=0.0),
```

although Solarize is only applied for even views ($v \bmod 2 = 0$).

Linear probing and k-nearest neighbors For linear probing, we follow standard methodology for self-supervised learning with vision transformers [107], i.e. we train a linear classifier on top of the `[CLS]` tokens from the four last layers on all training samples, regardless of the labels available during continual representation learning. For ResNets, we use the output of the last layer. We train for 100 epochs using stochastic gradient descent with momentum (batch size 1024, base learning rate 0.1 with cosine decay). We do not use any augmentations except for random horizontal flips. For k-nearest neighbors (kNN), we obtain the representations of the last layer of the backbone instead. No augmentations are used. The prediction is obtained by considering $k = 20$ nearest neighbors, weighted by distance with temperature $t = 0.07$.

Compute We run our experiments on the JUWELS-Booster [115] and JURECA [116] clusters at Forschungszentrum Jülich. For both systems, we use a single NVIDIA A100 GPU per experiment. We observe empirically that SupCon based methods have the highest GPU memory consumption (`22 GB`), while modulation invariance methods use `17 GB`. View and state invariance require `14 GB` of GPU memory. Augmentations are run on GPU and the datasets do not require image decoding, therefore CPU and RAM requirements are negligible. All runs take up to 10 hours to finish on the five session scenario.

B Forward and Backward Transfer

B.1 Metrics

Let task i be the classification problem on the classes from session i. We then define $A_{t,i}$ as the evaluation accuracy of task i at the end of training session t. We evaluate this accuracy in the task-agnostic setting, i.e. the classifier (linear or kNN) is unaware that the input data is limited to the task under consideration. In our FT and BT metrics, $\hat{A}_i$ is the task-agnostic kNN evaluation accuracy of task i on a model trained from scratch using Barlow Twins on data from task i. Then, we define:

Backward Transfer

$$\mathrm{BT} = \frac{1}{T-1} \sum_{i=1}^{T-1} \frac{1}{T-i} \sum_{t=i+1}^{T} (A_{t,i} - \hat{A}_i) \tag{9}$$

Forward Transfer

$$\mathrm{FT} = \frac{1}{T-1} \sum_{i=2}^{T} \frac{1}{i-1} \sum_{t=1}^{i-1} (A_{t,i} - \hat{A}_i) \tag{10}$$

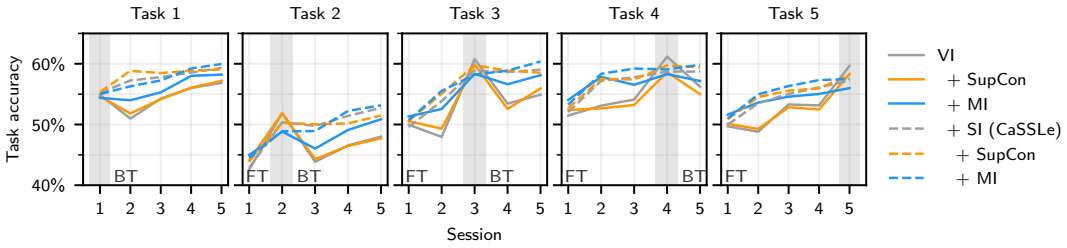

Figure A1: **Forward and backward transfer of different methods.** Accuracies on class-incremental CIFAR-100 (5 sessions) given either 1% of labels or completely unsupervised (averaged over four seeds).

B.2 Forward and Backward Transfer of Different Methods

Previously, we investigated the stability-plasticity trade-off as we modify the strength of the modulation invariance term (Figure 3). We further demonstrate in the sparsely labeled learning scenario (1% labels), that modulation invariance also provides improved forward and backward transfer compared to `SupCon`, while `SupCon` surprisingly shows lower plasticity than `VI` (Figure A1). The introduction of state invariance shows further improvements, as the combination of `SI` and `MI` on top of `VI` yields the highest forward and backward transfer on most tasks except for the first task.

C Python-style Pseudocode for TMCL

```
# f(xs, t): forward pass of backbone with inputs xs and modulations t
# C_1, ..., C_t: list of classes of session 1, ..., t
def orthogonalization(xs, ys):  # sparse labeled samples at session t
  for step in range(orthog_steps):
    c = random.sample(C_t, k=1)
    positives = random.sample(xs[ys == c], k=batch_size // 2)
    negatives = random.sample(xs[ys != c], k=batch_size // 2)

    pos_f, neg_f = f(positives, t=c), f(negatives, t=c)
    loss = opl_loss(pos_f, neg_f)
    loss.backward()
    update(f.modulations[c])

def consolidation(xs, is_pretrain=False): # unlabeled samples at session t
    for step in range(cons_steps):
        batch = random.sample(xs, k=batch_size)
        views = [aug(batch) for _ in range(num_views)]

        # view invariance
        # h_vi: view-inv. projector
        vi_projs = [h_vi(f(v, t=None)) for v in views]
        vi_loss = contrastive_loss(*vi_projs)  # Multi-view Barlow Twins

        if is_pretrain:
            vi_loss.backward()
            update(f.feedforward_weights)
            continue

        # if enabled: model state invariance
        # f_past: frozen backbone from prev. session
        # h_vi_past: frozen view-inv. projector from prev. session
        # p_si: state-inv. predictor
        with torch.no_grad():
            si_projs_past = [h_vi_past(f_past(v, t=None))]
```


https://doi.org/10.52202/085713-5030

```
        si_preds_curr = p_si(vi_projs)
        si_loss = mean(
            distill_loss(curr, past)  # CaSSLe/PNR loss function
            for curr, past in zip(si_preds_curr, i_projs_past)
        )

        # modulation invariance
        # h_mi: mod-inv. projector
        # p_mi: mod-inv. predictor
        mi_tasks = [[None] * batch_size] # unmodulated first view
        mi_tasks += [random.sample(C_1 + ... + C_t, k=batch_size) for _ in views[1:]]
        mi_projs = [h_mi(f(v, t=t)) for v, t in zip(views, mi_tasks)]
        mi_pred = p_mi(mi_projs[0])
        mi_loss = contrastive_loss(mi_pred, *mi_projs[1:])  # Multi-view Barlow Twins

        loss = vi_loss + si_loss + mi_loss
        loss.backward()
        update(f.feedforward_weights)

def train(sessions):
    for session_idx, (xs_unlabeled, xs_labeled, ys_labeled) in sessions:
        if session_idx == 0:
            consolidation(xs_unlabeled, is_pretrain=True)
        orthogonalization(xs_labeled, ys_labeled)
        consolidation(xs_unlabeled)
```

D Implicit Orthogonalization via Modulation Invariance

This section seeks to walk through the intuition behind the implicit orthogonalization via modulation invariance. To do so, we assume a non-incremental fashion with four classes A, B, C, D. For conceptual clarity, we focus the explanation here on the class centers of these respective classes.

In the orthogonalization phase of TMCL, we train modulations m_A to achieve $A|_{m_A} \perp \{B|_{m_A}, C|_{m_A}, D|_{m_A}\}$, m_B to achieve $B|_{m_B} \perp \{A|_{m_B}, C|_{m_B}, D|_{m_B}\}$, m_C to achieve $C|_{m_C} \perp \{A|_{m_C}, B|_{m_C}, D|_{m_C}\}$, and m_D to achieve $D|_{m_D} \perp \{A|_{m_D}, B|_{m_D}, C|_{m_D}\}$. Here, $A|_{m_A}$ denotes the representational vector of the class center of class A under modulation m_A, and similar for all others.

In the consolidation phase, our goal is to arrive at a representation space where $A|_\emptyset \perp B|_\emptyset \wedge A|_\emptyset \perp C|_\emptyset \wedge A|_\emptyset \perp D|_\emptyset \wedge B|_\emptyset \perp C|_\emptyset \wedge B|_\emptyset \perp D|_\emptyset \wedge C|_\emptyset \perp D|_\emptyset$ (where $A|_\emptyset, B|_\emptyset, C|_\emptyset$ and $D|_\emptyset$ denote the class centers of A, B, C, D in the unmodulated network). It can be seen that this will be the case, if we simultaneously achieve the orthogonality relations of point 1 and $A|_\emptyset = A|_{m_A} = A|_{m_B} = A|_{m_C} = A|_{m_D}$ (where $A|_{m_{A,B,C,D}}$ denotes the class center of class A respectively under modulations m_A, m_B, m_C, and m_D) and similar for classes B, C, D. The contrastive objective maximizes similarity between a given data sample without modulation and under modulations m_A, m_B, m_C, and m_D, and therefore implicitly drives the network to a state for which $A|_\emptyset = A|_{m_A} = A|_{m_B} = A|_{m_C} = A|_{m_D}$ and similar for B, C, D.

We note that achieving the full orthogonality relation $A|_\emptyset \perp B|_\emptyset \wedge A|_\emptyset \perp C|_\emptyset \wedge A|_\emptyset \perp D|_\emptyset \wedge B|_\emptyset \perp C|_\emptyset \wedge B|_\emptyset \perp D|_\emptyset \wedge C|_\emptyset \perp D|_\emptyset$ would likely require iterating steps 1 and 2. However, we did not seek such an iterating implementation, as we focused on the continual learning setting where we identified modulation learning (step 1) with a single phase of fast learning that occurs whenever a new class is observed, and step 2 with a slower consolidation phase. Under these conditions, the full orthogonality relation cannot be expected to be achieved rigorously, but as is demonstrated by our CDNV results, our TMCL algorithm still leads to a representation space where the clustering of the individual classes is improved.

E Class-Distance Normalized Variance

The **class-distance normalized variance** (CDNV) is defined as

$$\text{CDNV} := \frac{1}{|\mathcal{C}|^2 - |\mathcal{C}|} \sum_{\substack{c,c' \in \mathcal{C} \\ c \neq c'}} \frac{\overbrace{\text{Var}(\mathbf{Z}^{(c)}) + \text{Var}(\mathbf{Z}^{(c')})}^{\text{intra-class collapse}}}{\underbrace{2 \left\| \mu(\mathbf{Z}^{(c)}) - \mu(\mathbf{Z}^{(c')}) \right\|}_{\text{inter-class distance}}}, \tag{11}$$

where $\mathbf{Z}^{(c)} = [\mathbf{z}_1^{(c)}, \ldots, \mathbf{z}_N^{(c)}] \in \mathbb{R}^{N \times D}$ is a batch of backbone representations (i.e. $\mathbf{z}_i^{(c)} = f(\mathbf{x}_i^{(c)} | \mathbf{W}, \emptyset)$) of all class-$c$ samples in the CIFAR-100 test split, and $\mathcal{C}$ is the set of CIFAR-100 classes (lower is better).

F Split-CIFAR-100 with 10 sessions

We demonstrate that TMCL is also effective in the scenario of sparsely supervised continual learning with 10 sessions, i.e. 10 classes per session (Table A2).

Table A2: **Semi-supervised continual representation learning.** Final all-vs-all accuracy on class-incremental CIFAR-100 (10 sessions) given either 1% of labels or completely unsupervised, averaged over four seeds ($\pm$ denotes the standard deviation).

Method	kNN	linear
SupCon + SI (PNR)	34.7$\pm$0.6	20.9$\pm$1.4
CE	39.2$\pm$0.8	28.2$\pm$0.9
VI	55.5$\pm$0.5	50.3$\pm$0.4
+ MI (**TMCL**)	57.7$\pm$0.2	52.2$\pm$0.3
+ SupCon	55.1$\pm$0.3	49.9$\pm$0.4
+ SI (PNR)	57.4$\pm$0.5	54.0$\pm$0.9
+ SupCon	57.5$\pm$0.2	54.0$\pm$0.6
+ CE	57.4$\pm$0.3	54.2$\pm$0.8
+ MI	**58.7**$\pm$0.3	**54.6**$\pm$0.5

G Datasets

Our analysis focuses on the CIFAR-100 dataset [117] and the ImageNet-100 dataset [111]. For the transfer learning experiments, we perform kNN evaluation on `Aircraft` [118], `CIFAR-10` [117], `CUBirds` [122], `DTD` [109], `EuroSAT` [113], `GTSRB` [114], `STL-10` [110], `SVHN` [119], and `VGGFlower` [120].

H Further acknowledgements

We implement our methods based on PyTorch [108] with the Lightning framework [112]. For the augmentations on CIFAR-100, we used kornia [121]. Backbone implementations are adapted from the timm library [123]. Data loading and augmentations for the ImageNet-100 experiments are implemented via NVIDIA DALI (https://github.com/NVIDIA/DALI). For the figures, we relied on icons designed by OpenMoji (License: CC BY-SA 4.0).

https://doi.org/10.52202/085713-5030

Supplementary References

[108] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, April 2024.

[109] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[110] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[111] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[112] William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019.

[113] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[114] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

[115] Stefan Kesselheim, Andreas Herten, Kai Krajsek, Jan Ebert, Jenia Jitsev, Mehdi Cherti, Michael Langguth, Bing Gong, Scarlet Stadtler, Amirpasha Mozaffari, et al. Juwels booster–a supercomputer for large-scale ai research. In *International Conference on High Performance Computing*, pages 453–468. Springer, 2021.

[116] Dorian Krause and Philipp Thörnig. Jureca: modular supercomputer at jülich supercomputing centre. *Journal of large-scale research facilities JLSRF*, 4:A132–A132, 2018.

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

[118] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[119] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[120] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[121] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*, 2020.

[122] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[123] Ross Wightman. PyTorch Image Models.

 https://doi.org/10.52202/085713-5030