SEMANTIC-AWARE REPRESENTATION LEARNING FOR LIFELONG LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The human brain excels at lifelong learning by not only encoding information in sparse activation codes, but also leveraging rich semantic structures and relationships between newly encountered and previously learned objects. This ability to utilize semantic similarities is crucial for efficient learning and knowledge consolidation, yet is often underutilized in current continual learning approaches. To bridge this gap, we propose *Semantic-Aware Representation Learning* (SARL), which employs sparse activations and a principled approach to evaluate similarities between objects encountered across different tasks and subsequently uses them to guide representation learning. Using these relationships, SARL enhances the reusability of features and reduces interference between tasks. This approach empowers the model to adapt to new information while maintaining stability, significantly improving performance in complex incremental learning scenarios. Our analysis demonstrates that SARL achieves a superior balance between plasticity and stability by harnessing the underlying semantic structure. [1]

1 INTRODUCTION

Deep neural networks (DNNs) have been designed for static in-distribution batch learning, whereby the model assumes that each batch of training data is representative of the underlying joint distribution. However, this assumption fails when processing sequential data in continual learning (CL) scenarios where information is made available incrementally over time, and hence the training batch at a given point only contains samples from the current task. The inadequacy of standard training to handle such dynamic data distributions often results in catastrophic forgetting (McCloskey & Cohen, 1989), erasing previously acquired knowledge as the model learns the new data. To address these challenges, CL methods aim to retain previous knowledge while learning new tasks, enabling models to accumulate knowledge across tasks (Parisi et al., 2019). The key challenge in CL is maintaining an optimal balance between stability and plasticity (Mermillod et al., 2013): the model needs enough plasticity to quickly adapt to new tasks, but also enough stability to preserve previously acquired knowledge. Existing approaches often trade off between high plasticity (quick adaptation) and high stability (knowledge retention), without effectively balancing the two. This trade-off highlights the difficulty in consolidating knowledge efficiently while learning sequentially.

Many CL approaches tend to emphasize either stability or plasticity, often overlooking the complex relationship between these two essential components. Stability-focused methods aim to preserve important weights to maintain previously acquired knowledge (Zenke et al., 2017; Li et al., 2023), while plasticity-oriented approaches allow significant representation changes to adapt flexibly to new tasks (Caccia et al., 2022; Liang & Li, 2023). This dichotomy often treats stability and plasticity as separate objectives rather than interconnected factors in knowledge consolidation. We hypothesize that leveraging semantic relationships between objects can bridge this gap by enhancing knowledge transfer and integrating new information with existing knowledge.

The human brain exemplifies the power of semantic relationships in organizing and retaining knowledge. By forming associations between related concepts, the brain facilitates the retrieval, transfer, and consolidation of information. When encountering a new object similar to a previously learned one, the brain retrieves relevant information using semantic relationships (Saxena et al., 2022), aiding the learning of new objects and linking new information to existing knowledge. Furthermore,

---

[1] Code will be publicly available upon acceptance.

semantic relationships support knowledge consolidation and retention by reusing common concepts and adapting them in a cohesive manner. Moreover, semantic relationships play an important role in organizing memory (Artuso et al., 2022), necessitating that information be encoded in a way that captures these relationships. The brain employs sparse coding (Foldiak, 2003), where information is represented by the strong activation of a small subset of neurons. Notably, objects sharing concepts likely activate overlapping sets of neurons, efficiently encoding semantic relationships. We hypothesize that emulating these mechanisms can enable DNNs to better balance stability and plasticity by leveraging the rich semantic structure between objects.

To this end, Semantic-Aware Representation Learning (SARL) emulates sparse coding in the brain and employs a principled approach to capture the semantic relationships between objects and to use them to guide representation learning. SARL leverages activation sparsity to emulate sparse coding and represents each object through semantically rich object prototypes (Snell et al., 2017) derived from the mean sparse activations of the object samples. These prototypes effectively capture the semantic structure of object relationships, allowing the model to reuse and consolidate knowledge efficiently. By capturing the inherent similarities between objects, SARL leverages semantic relationships to guide the alignment of new object prototypes with those of similar classes from previously learned tasks, while promoting separation from dissimilar ones. This alignment encourages a structured representation of knowledge, facilitating knowledge transfer across tasks and reducing interference. Furthermore, SARL incorporates prototype regularization to ensure model stability, mitigating the risk of forgetting previously learned information, and fostering effective consolidation of knowledge in a cohesive manner. This mechanism enables the preservation of valuable representations while adapting to new tasks, ultimately enhancing the model's overall learning capabilities.

Our empirical analysis on challenging class-incremental learning scenarios across various datasets demonstrates that SARL significantly enhances lifelong learning performance by leveraging semantic structure for representation learning. By effectively aligning representations based on semantic relationships, SARL facilitates efficient knowledge transfer and consolidation, enabling the model to adapt seamlessly to new tasks while retaining previously acquired knowledge. These capabilities are further supported by our analysis, which reveals that SARL achieves a better balance between model stability and plasticity, mitigates forgetting, and reduces task recency bias.

2 RELATED WORK

Approaches to addressing catastrophic forgetting in CL can be broadly categorized into three groups. *Regularization-based* methods penalize changes in the model either in parameter space (Farajtabar et al., 2020; Ritter et al., 2018) or functional space (Benjamin et al., 2019; Li & Hoiem, 2017). These methods primarily focus on stability and generally fail in class-incremental learning settings where task information is unavailable. *Dynamic architecture* approaches (Rusu et al., 2016), expand the network to allocate distinct parameters for each task. While reducing forgetting, these methods lead to model size scaling linearly with the number of tasks and often require task identity at test time, limiting their applicability. *Rehearsal-based* approaches draw inspiration from the brain's experience replay for knowledge consolidation (Ólafsdóttir et al., 2018). They store a subset of data samples from previous tasks in a memory buffer, which are interleaved with new tasks samples to approximate the joint distribution. Rehearsal-based approaches have proven to be more general and effective for various continual learning scenarios (Farquhar & Gal, 2018).

The baseline Experience Replay (ER) (Riemer et al., 2018) interleaves the training of new task samples with previous task samples in memory. Several approaches have since been proposed to provide additional learning cues to the model from its previous state. Dark Experience Replay (DER++) (Buzzega et al., 2020) additionally stores the output logits and applies a consistency loss to the memory samples. ER-ACE (Caccia et al., 2022) uses an asymmetric cross-entropy loss, considering only new task logits, on the incoming samples to reduce representation drift. Gradient Coreset Replay (GCR)(Tiwari et al., 2022) selects and maintains a coreset based on gradient approximation.

Recently, there has been a shift towards multi-memory-based CL approaches, inspired by the complementary learning systems theory in the brain (McClelland et al., 1995). CLS-ER (Arani et al., 2022) simulates the interaction between fast and slow learning by using two semantic memories that aggregate model weights at different timescales with an exponential moving average. CO^{2}L

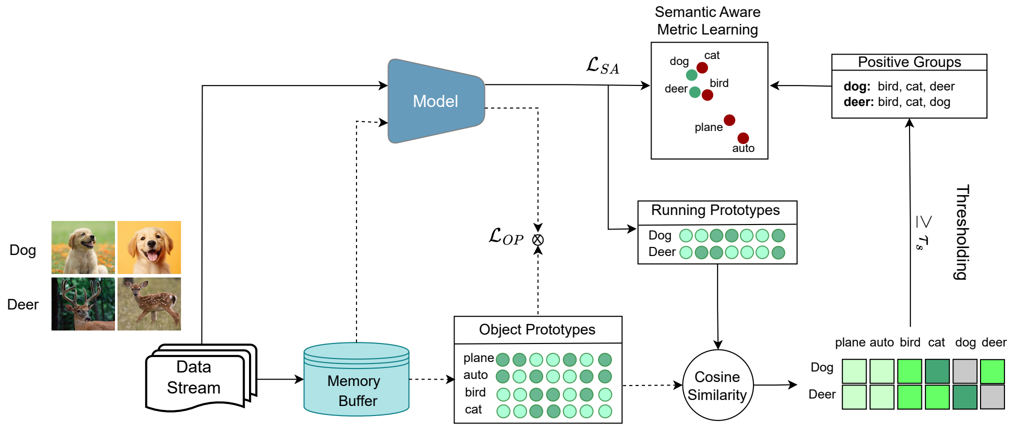


Figure 1: SARL employs activation sparsity to emulate brain-like sparse coding, representing each object with a class prototype derived from the mean representations of object samples. The semantic relationships are utilized to encourage new object prototypes to align with the class prototypes of similar objects and diverge from dissimilar ones. Additionally, SARL ensures model stability through prototype regularization, mitigating forgetting and enabling effective consolidation of information in lifelong learning. Darker shades represent higher values.

(Cha et al., 2021) initially learns representations with a modified SupCon loss and subsequently trains a classifier using samples from the last task and buffer data. Deep Retrieval and Imagination (DRI)(Wang et al., 2022a) leverages an embedding network and generative model to retrieve and generate imaginary data. SCoMMER (Sarfraz et al., 2023) enforces sparse coding for efficient representation learning and uses multiple memories, using the previous state of the model to reduce similarity drift. While these multi-memory approaches effectively reduce forgetting, they incur additional computational and memory costs, which can be prohibitive in some real-world settings. Therefore, we believe that it is crucial to improve CL performance in a single-model setting, which can later be extended to a dual-memory framework.

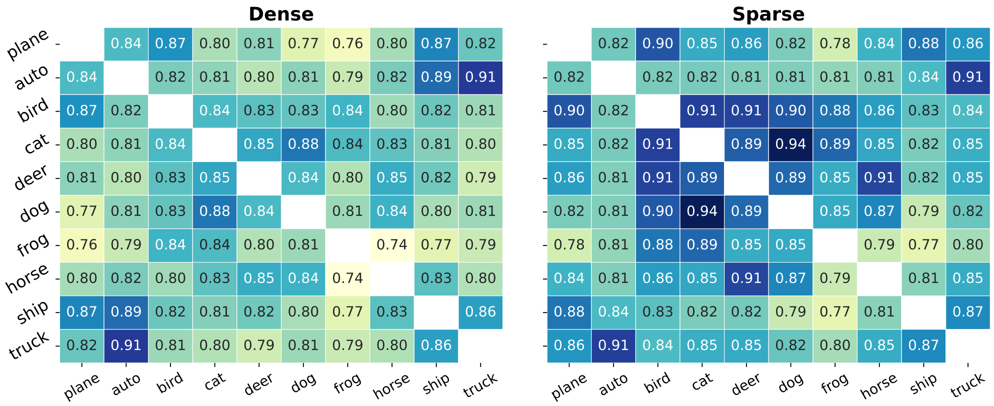
3 METHODOLOGY

We begin with an overview of the biological mechanisms that inspire our approach, followed by details on how we mimic these mechanisms in DNNs and an outline of our overall formulation.

3.1 BIOLOGICAL MOTIVATION

The human brain excels at adapting to dynamic environments by transferring knowledge from previously learned concepts to new ones, mitigating interference, and efficiently consolidating information. A critical component of this lifelong learning ability is the brain's capacity to exploit semantic relationships between objects, which guides both information encoding and memory formation (Binder & Desai, 2011). By establishing associations between related concepts, the brain organizes information into semantic networks in which interconnected concepts share common features, enabling efficient retrieval, transfer, and consolidation of knowledge (Saxena et al., 2022). When encountering a new objects, the brain rapidly integrates them into these networks, leveraging previously acquired knowledge to facilitate learning and reduce relearning efforts. This process is further supported by the brain's use of *sparse coding* (Foldiak, 2003), where only a small subset of neurons is activated. Objects with conceptual similarities activate overlapping neuron sets, hence efficiently encoding semantic relationships which can be leveraged for retrieval and learning.

Inspired by these cognitive mechanisms, we propose that incorporating sparsely activated neurons, which allow for the extraction of semantic structures, and leveraging these relationships for learning in DNNs, can significantly enhance their lifelong learning capabilities.



Dense

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck
plane		0.84	0.87	0.80	0.81	0.77	0.76	0.80	0.87	0.82
auto	0.84		0.82	0.81	0.80	0.81	0.79	0.82	0.89	0.91
bird	0.87	0.82		0.84	0.83	0.83	0.84	0.80	0.82	0.81
cat	0.80	0.81	0.84		0.85	0.88	0.84	0.83	0.81	0.80
deer	0.81	0.80	0.83	0.85		0.84	0.80	0.85	0.82	0.79
dog	0.77	0.81	0.83	0.88	0.84		0.81	0.84	0.80	0.81
frog	0.76	0.79	0.84	0.84	0.80	0.81		0.74	0.77	0.79
horse	0.80	0.82	0.80	0.83	0.85	0.84	0.74		0.83	0.80
ship	0.87	0.89	0.82	0.81	0.82	0.80	0.77	0.83		0.86
truck	0.82	0.91	0.81	0.80	0.79	0.81	0.79	0.80	0.86	

Sparse

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck
plane		0.82	0.90	0.85	0.86	0.82	0.78	0.84	0.88	0.86
auto	0.82		0.82	0.82	0.81	0.81	0.81	0.81	0.84	0.91
bird	0.90	0.82		0.91	0.91	0.90	0.88	0.86	0.83	0.84
cat	0.85	0.82	0.91		0.89	0.94	0.89	0.85	0.82	0.85
deer	0.86	0.81	0.91	0.89		0.89	0.85	0.91	0.82	0.85
dog	0.82	0.81	0.90	0.94	0.89		0.85	0.87	0.79	0.82
frog	0.78	0.81	0.88	0.89	0.85	0.85		0.79	0.77	0.80
horse	0.84	0.81	0.86	0.85	0.91	0.87	0.79		0.81	0.85
ship	0.88	0.84	0.83	0.82	0.82	0.79	0.77	0.81		0.87
truck	0.86	0.91	0.84	0.85	0.85	0.82	0.80	0.85	0.87	

Figure 2: Similarity matrices were computed using object representations from the model trained on the joint CIFAR-10 dataset, with dense and sparse activations, respectively.

3.2 SEMANTIC-AWARE REPRESENTATION LEARNING

Building on these cognitive insights, we introduce Semantic-Aware Representation Learning (SARL), which adopts a principled approach to extract the semantic relationships between objects encountered across sequential tasks and subsequently utilizes them to guide representation learning in a cohesive manner. SARL draws inspiration from the brain's sparse coding approach and employs activation sparsity to facilitate semantic information encoding in representations. Each object is represented by an object prototype, capturing the average features of its instances, which are saved in memory at the end of each task. By evaluating the similarity between these stored prototypes and the average features of new objects, SARL identifies sets of positive (similar) and negative (dissimilar) objects from both previous and current tasks. Leveraging these relationships, SARL applies a novel semantic-aware metric learning approach, guiding the model to bring representations of new objects closer to similar prototypes while distancing them from dissimilar ones. This mechanism enables the model to reuse knowledge from similar objects, enhancing learning efficiency and reducing interference. To further promote stability, SARL implements prototype-based regularization, encouraging the activations of previously learned objects to remain consistent with their established prototypes and thus preventing drift over time. By integrating new information in a structured manner, SARL fosters effective knowledge consolidation and maintains a balance between plasticity and stability in lifelong learning. The following sections introduce the different components of SARL.

3.3 SPARSE CODING

To mimic sparse coding in the brain, we employ activation sparsity using k-winners-take-all (k-WTA) (Maass, 2000), where only the K most active neurons in each layer are allowed to propagate to the next layer. For convolutional layers, we apply k-WTA locally across the channels at each spatial location rather than globally across the entire input and all channels. This localized approach ensures that the most relevant features are selected at each position, promoting specialization of filters. As a result, the network can focus on distinct patterns at different spatial locations, leading to more efficient feature extraction, better semantic encoding, and reduced interference in lifelong learning scenarios.

3.3.1 OBJECT PROTOTYPES

Building on the sparse coding mechanism through k-WTA, we extend this principle to our formulation of object prototypes. These prototypes are central to capturing and incorporating the intrinsic similarities between objects to guide representation learning. By capturing shared concepts through overlapping sparse activations, k-WTA offers an efficient and intuitive method to encode such relationships. Each object is represented by its object prototype, defined as the mean of all instance representations, a strategy commonly employed in metric learning (Kaya & Bilge, 2019). Specifically, we utilize the ℓ_2-normalized activations of the penultimate layer as object representations, with the prototype for each class c computed as:

$$\mathcal{O}_c = \frac{1}{N_c} \sum_i \mathbb{I}(y_i = c) a'_i \tag{1}$$

where a'_i represents the normalized activation of sample x_i with label y_i, and N_c is the number of instances in class c. This provides an efficient means of forming object prototypes that guide subsequent learning. At the end of each task, we calculate and store the object prototypes of all the new object categories. Hence, while learning task t, we have access to the object prototypes for all the object categories learned until task $t-1$.

3.3.2 SEMANTIC-AWARE METRIC LEARNING

Having developed an efficient method for representing objects using prototypes that capture the essence of each category, we can now evaluate the similarity between objects across tasks. We use cosine similarity, which measures the cosine of the angle between two class prototypes, providing a scale-invariant measure independent of magnitude. This approach is particularly advantageous when comparing object prototypes, as it focuses on directional alignment between vectors rather than their magnitudes, effectively capturing semantic similarity in high-dimensional spaces. The similarity between the object categories i and j, and the corresponding object prototypes $\mathcal{O}_i$ and $\mathcal{O}_j$ is given by:

$$sim(\mathcal{O}_i, \mathcal{O}_j) = \frac{\mathcal{O}_i \cdot \mathcal{O}_j}{\|\mathcal{O}_i\| \cdot \|\mathcal{O}_j\|} \tag{2}$$

This provides us with an efficient approach to evaluating the similarity between object categories encountered across different tasks, which is subsequently used to inform learning. Figure 2 shows that the proposed approach can capture semantic similarities, and sparse activations further enhance the model's ability to distinguish between similar and dissimilar objects. Section E in Appendix shows that sparse activations enhances the capacity of the model to capture semantic similarities between objects within animal and vehicle clusters as well as between the two clusters.

Specifically, for each new object category introduced in task t, we calculate intermediate object prototypes $\overline{\mathcal{O}}$ using Equation 1, and evaluate the similarity between each new object and other objects in the combined set of both newly introduced and previously learned object categories. We employ a warm-up stage for each new task to allow the model to learn meaningful representations before computing the intermediate object prototypes. For each new object category c, we use the degree of similarity between object prototypes to identify a set of similar object categories $\mathbb{S}_c$:

$$\mathbb{S}_c = \{ i \mid \mathcal{O}_i \geq \tau_s \wedge i \neq c \quad \forall i \in \mathbb{C}^{0:t} \} \tag{3}$$

where τ_s is the similarity threshold ranging from 0 to 1, and $\mathbb{C}^{0:t}$ represents the set of all object categories observed up to task t. In subsequent epochs following the warm-up stage, we employ our semantic-aware metric learning loss. For each new object category c, we encourage the mean ℓ_2-normalized representations of its object instances in the batch, denoted as $\overline{\boldsymbol{o}_c}$, to be closer to the object prototypes of objects in the set $\mathbb{S}_c$—indicating similarity. Simultaneously, they are pushed to be distant from dissimilar objects. Formally, the loss is given by:

$$\mathcal{L}_{SM} = \sum_{c \in \mathbb{C}^t} \frac{\sum_{i \in \mathbb{S}_c} \|\mathcal{O}_i - \bar{o}_c\|^2}{\sum_{i \notin \mathbb{S}_c} \|\mathcal{O}_i - \bar{o}_c\|^2} \tag{4}$$

where $\mathbb{C}^t$ is the set of new object categories at task t. The semantic-aware metric learning loss enables the model to leverage similarities between objects and acquire semantic-aware representations. This, in turn, facilitates effective knowledge sharing and reuse while mitigating interference. Such enhancements augment the model's capacity to consolidate knowledge more cohesively.

3.3.3 REGULARIZATION ON OBJECT PROTOTYPES

To enhance stability and preserve the semantic structure within the acquired representations, we introduce a regularization loss on the object prototypes. This regularization is applied to buffer samples, penalizing the divergence of mean representations of buffered samples in the batch from their originally learned object prototypes. Formally, in each training batch, we sample from the memory buffer, $\boldsymbol{M}_b$, calculate the ℓ_2-normalized mean representations for each object category, $\boldsymbol{o}_c^b$, and minimize the mean squared error between $\boldsymbol{o}_c^b$ and their corresponding object prototypes, $\mathcal{O}_c$, stored in the prototype memory, $\boldsymbol{M}_o$:

$$\mathcal{L}_{OP} = \sum_{c \in \mathbb{C}^{0:t-1}} \|\boldsymbol{o}_c^b - \mathcal{O}_c\|^2 \tag{5}$$

where $\mathbb{C}^{0:t-1}$ is the set of previously learned object categories. By imposing this regularization, we encourage the model to maintain the semantic integrity of learned object categories across tasks, minimizing deviations in the representations of buffered samples from their learned prototypes. This not only fosters stability in the model's knowledge but also mitigates the risk of forgetting important semantic relationships established in earlier learning phases.

3.3.4 FUNCTIONAL REGULARIZATION

Functional regularization plays a crucial role in helping models retain previous knowledge and effectively consolidate new information by directly regulating changes in the model's input-output behavior rather than just its parameters. When applied to new data, it encourages the model to adapt while remaining within the functional vicinity of its previous state, thereby facilitating smooth integration of new information. On old data, functional regularization helps maintain consistency with prior learned functions, minimizing forgetting by directly aligning the current model's outputs with those from earlier tasks. This approach stabilizes the learning process and allows the model to evolve while preserving essential knowledge.

Specifically, before training on a new task, we pass its samples through the model and save the output logits, denoted as $\boldsymbol{z}^t$, in memory. This allows us to enforce functional consistency during training on the new task without the need to maintain the previous model state in memory. To further promote consistency in model behavior and maintain semantic relationships, we also save the output logits along with the data samples in a memory buffer. During the replay of these samples, similar to Buzzega et al. (2020), we apply a consistency regularization loss on the model's output. At each training step with new task samples $\boldsymbol{x}^t$ and $\boldsymbol{y}^t$, along with their precomputed output logits $\boldsymbol{z}^t$ from previous model state, we sample $\boldsymbol{x}^b, \boldsymbol{y}^b,$ and $\boldsymbol{z}^b$ from the memory buffer and apply the following functional consistency loss:

$$\mathcal{L}_{FR} = \alpha \|F(x^b : \theta) - \boldsymbol{z}^b\|^2 + \beta |F(x^t : \theta) - \boldsymbol{z}^t\|^2 \tag{6}$$

where $F(.)$ is the model's output parameterized by weights θ. α and β controls the strength of functional consistency on buffer samples and new task samples, respectively. This combination of regularization on object prototypes and regularization of consistency improves the stability of the model and preserves the semantic structure learned in previous tasks.

3.4 OVERALL FORMULATION

SARL involves training a model $f(.;\theta)$ on a non static data stream, D containing a sequence of T i.i.d tasks $(D^1, D^2, .., D^T)$. To employ rehearsal, it maintains a small episodic memory, $\boldsymbol{M}_b$ using reservoir sampling (Vitter, 1985). Additionally, it maintains a memory of object prototypes, $\boldsymbol{M}_o$. During each training iteration of task t, the model is fed with the training batch $(x^t, y^t) \sim D^t$ and a random batch from episodic memory $(x^b, y^b, z^b) \sim \boldsymbol{M}_b$. Task 1 is trained using standard cross-entropy loss. For each subsequent task, we employ a warm-up stage where we train the model with the combination of cross-entropy loss, $\mathcal{L}_{CE}$ on the training batch and buffer samples and the regularization losses. At the end of the warm-up stage when the model has learned meaningful representations, we evaluate the intermediate object references for each object category, c, in the current task using Equation 1 and identify a set of corresponding similar object categories, $\mathbb{S}_c$ using Equation 3 to employ the semantic-aware metric learning loss in the subsequent training on the task.

Table 1: Comparison analysis of single- and dual-model CL methods across various CL settings. The baseline taken from the repective methods. We ran SCoMMER for Seq-TinyImageNet using the hyperparameter setting approach provided by the authors. We report the average accuracy and 1 std of 3 different seeds.

Buffer	Method	Seq-CIFAR10		Seq-CIFAR100		Seq-TinyImg	
		Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
–	JOINT	92.20±0.15	98.31±0.12	70.62±0.64	86.19±0.43	59.99±0.19	82.04±0.10
	SGD	19.62±0.05	61.02±3.33	17.58±0.04	40.46±0.99	7.92±0.26	18.31±0.68
200	ER	44.79±1.86	91.19±0.94	21.40±0.22	61.36±0.39	8.49±0.16	38.17±2.00
	FDR	30.91±2.74	91.01±2.74	22.02±0.008	61.72±1.02	8.70±0.19	40.36±0.68
	DER++	64.88±1.17	91.92±0.60	29.60±1.14	62.49±0.78	10.96±1.17	40.87±1.16
	ER-ACE	62.08±1.44	35.17±1.17	27.44±0.64	25.29±1.89	11.25±0.54	44.17±1.02
	GCR	64.84±1.63	90.8±1.05	33.69±1.40	64.24±0.83	13.05±0.91	42.11±1.01
	DRI	65.16±1.13	92.87±0.71	-	-	17.58±1.24	44.28±1.37
	CO^{2}L	65.57±1.37	93.43±0.78	31.90±0.38	55.02±0.3	13.88±0.40	42.37±0.74
	CLS-ER	66.19±0.75	93.90±0.60	43.80±1.89	73.49±1.04	23.47±0.80	49.60±0.72
	SCoMMER	69.19±0.61	93.20±0.10	40.25±0.05	69.39±0.43	16.30±1.54	49.37±0.81
	SARL	**70.97**±0.47	**95.72**±0.36	**48.96**±0.53	**78.91**±0.15	**28.95**±1.13	**71.56**±0.18
500	ER	57.74±0.27	93.61±0.27	28.02±0.31	68.23±0.16	9.99±0.29	48.64±0.46
	FDR	28.71±3.23	93.29±0.59	29.19±0.33	69.76±0.51	10.54±0.21	49.88±0.71
	DER++	72.70±1.36	93.88±0.50	41.40±0.96	70.61±0.11	19.38±1.41	51.91±0.68
	ER-ACE	68.45±1.78	40.67±0.06	30.14±1.11	24.81±0.63	17.73±0.56	49.99±1.51
	GCR	74.69±0.85	94.44±0.32	45.91±1.30	71.64±2.10	19.66±0.68	52.99±0.89
	DRI	72.78±1.44	93.85±0.46	-	-	22.63±0.81	52.89±0.60
	CO^{2}L	74.26±0.77	39.21±0.39	39.21±0.39	62.98±0.58	20.12±0.42	53.04±0.69
	CLS-ER	75.22±0.71	94.94±0.53	51.40±1.00	78.12±0.24	31.03±0.56	60.41±0.50
	SCoMMER	74.97±1.05	94.36±0.06	49.63±1.43	75.49±0.43	22.60±0.85	58.46±1.31
	SARL	**75.64**±0.36	**95.95**±0.24	**55.30**±0.61	**81.23**±0.24	**32.56**±1.23	**70.94**±0.43

The overall loss for training the model is given by:

$$\mathcal{L} = \mathcal{L}_{CE}(F(x^t : \theta), y^t) + \mathcal{L}_{CE}(F(x^b : \theta), y^b) + \lambda_{SM}\mathcal{L}_{SM} + \lambda_{OP}\mathcal{L}_{OP} + \mathcal{L}_{FR} \tag{7}$$

where λ_{SM} and λ_{OP} are the hyperparameters that control the weight of each loss.

At the end of each task, we evaluate the object prototypes of the object categories introduced in the task using Eqn 1. Additionally, to mitigate the biases of the batch norm statistics towards the current task (Pham et al., 2022), we employ a simple yet effective technique of passing the model through all buffer samples to shift the batch norm statistics towards the approximate joint distribution.

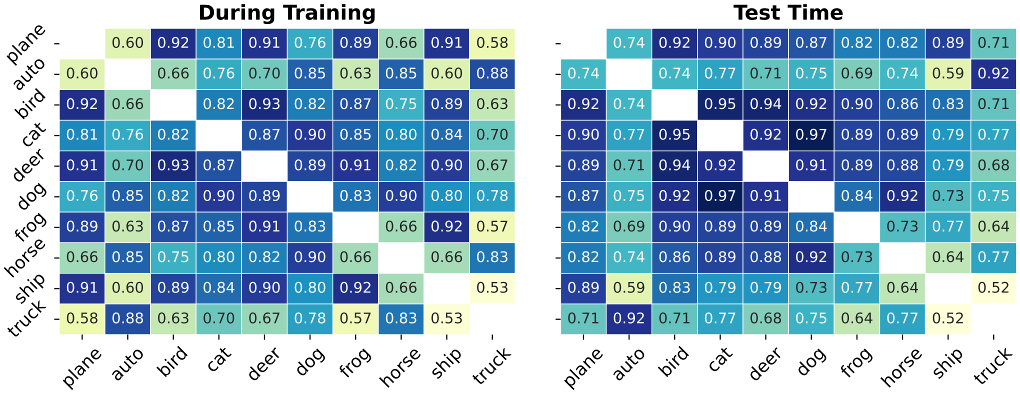
4 EMPIRICAL EVALUATION

Settings. To evaluate the effectiveness of SARL, we consider various CL scenarios that each present unique challenges for lifelong learning models. In the *class-incremental learning (Class-IL)* scenario, with each new task, the model is introduced to a new set of classes and must learn to distinguish not only among the current task's classes but also across all previously encountered classes. This setting tests the model's ability to build generalizable representations, consolidate knowledge, and transfer that knowledge to efficiently distinguish all classes seen so far. We only train our models for the CLass-IL setting; for completion, we also provide *task-incremental learning (Task-IL)* setting. In this context, the model has access to task labels during inference, enabling the utilization of task labels to selectively choose the subset of output logits.

While Class-IL focuses on knowledge accumulation across distinct tasks, it does not fully capture the complexity of real-world scenarios, where tasks often lack clear boundaries, and classes can reappear with varying distributions. To address this, we also evaluate SARL in the *generalized class-incremental learning (GCIL)* setting. GCIL introduces a more dynamic and realistic environment where classes reoccur across tasks, and the number of classes and their sample sizes can vary. This

Table 2: Comparison analysis of single- and dual-model CL methods on GCIL-CIFAR-100 dataset.

Dist.	Buffer	JOINT	SGD	ER	DER++	CLS-ER	SCoMMER	SARL
Uniform	200	58.36$_{\pm 1.02}$	12.67$_{\pm 0.24}$	16.40$_{\pm 0.37}$	18.84$_{\pm 0.60}$	25.06$_{\pm 0.81}$	30.84$_{\pm 0.80}$	**36.97**$_{\pm 0.54}$
	500			28.21$_{\pm 0.69}$	32.92$_{\pm 0.74}$	36.34$_{\pm 0.59}$	36.87$_{\pm 0.36}$	**39.03**$_{\pm 0.45}$
Longtail	200	56.94$_{\pm 1.56}$	22.88$_{\pm 0.53}$	19.27$_{\pm 0.77}$	26.94$_{\pm 1.27}$	28.54$_{\pm 0.87}$	29.08$_{\pm 0.31}$	**35.26**$_{\pm 0.75}$
	500			20.30$_{\pm 0.63}$	25.82$_{\pm 0.83}$	28.63$_{\pm 0.68}$	35.20$_{\pm 0.21}$	**35.47**$_{\pm 1.20}$



During Training

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck
plane		0.60	0.92	0.81	0.91	0.76	0.89	0.66	0.91	0.58
auto	0.60		0.66	0.76	0.70	0.85	0.63	0.85	0.60	0.88
bird	0.92	0.66		0.82	0.93	0.82	0.87	0.75	0.89	0.63
cat	0.81	0.76	0.82		0.87	0.90	0.85	0.80	0.84	0.70
deer	0.91	0.70	0.93	0.87		0.89	0.91	0.82	0.90	0.67
dog	0.76	0.85	0.82	0.90	0.89		0.83	0.90	0.80	0.78
frog	0.89	0.63	0.87	0.85	0.91	0.83		0.66	0.92	0.57
horse	0.66	0.85	0.75	0.80	0.82	0.90	0.66		0.66	0.83
ship	0.91	0.60	0.89	0.84	0.90	0.80	0.92	0.66		0.53
truck	0.58	0.88	0.63	0.70	0.67	0.78	0.57	0.83	0.53	

Test Time

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck
		0.74	0.92	0.90	0.89	0.87	0.82	0.82	0.89	0.71
	0.74		0.74	0.77	0.71	0.75	0.69	0.74	0.59	0.92
	0.92	0.74		0.95	0.94	0.92	0.90	0.86	0.83	0.71
	0.90	0.77	0.95		0.92	0.97	0.89	0.89	0.79	0.77
	0.89	0.71	0.94	0.92		0.91	0.89	0.88	0.79	0.68
	0.87	0.75	0.92	0.97	0.91		0.84	0.92	0.73	0.75
	0.82	0.69	0.90	0.89	0.89	0.84		0.73	0.77	0.64
	0.82	0.74	0.86	0.89	0.88	0.92	0.73		0.64	0.77
	0.89	0.59	0.83	0.79	0.79	0.73	0.77	0.64		0.52
	0.71	0.92	0.71	0.77	0.68	0.75	0.64	0.77	0.52	

Figure 3: Cosine similarities between the object prototypes computed at the end of each training task on Seq-CIFAR10 with 200 buffer size and the similarity structure of the final model on test set.

scenario tests the model's ability to deal with class imbalance, sample efficiency, and the continual integration of knowledge from overlapping or reappearing classes. Details of the experimental setup and datasets are provided in Appendix.

Results. We benchmarked SARL's performance against established rehearsal-based approaches across varying CL settings and dataset complexities. Table 1 demonstrates consistent performance improvements across all settings. Notably, under the low buffer regime, where the model has to retain knowledge while having access to very limited samples from previous tasks (approximately two samples per class for Seq-CIFAR-100 and one sample per class for Seq-Tiny-ImageNet), SARL effectively retains previously acquired knowledge, demonstrating its ability to learn new tasks while remaining in the functional vicinity of earlier tasks despite the limited buffer size and consolidating knowledge as it learns new tasks. Interestingly, as the complexity of the dataset increases with the number of classes with high semantic similarity, the performance gains are more pronounced, suggesting that SARL is able to exploit the semantic structure and utilize it to consolidate knowledge.

Table 2 highlights the advantage of SARL in the challenging GCIL setting, where the model must update its knowledge across multiple occurrences of the same object with varying sample sizes, while addressing class imbalance and task complexity. SARL's ability to update object prototypes when re-encountering a class ensures efficient use of new samples without penalizing for deviations from earlier suboptimal representations. SARL shows remarkable resilience to these challenges and provides additional credence to the benefits of semantic-aware representation learning.

These results are particularly impressive considering that SARL uses only a single model, while many state-of-the-art methods rely on multiple models. For instance, CLS-ER uses two additional models, and SCoMMER incorporates one. Although these methods claim to use a single model for inference, the distinction between training and inference in lifelong learning is blurred, as the model must continuously learn from the environment. This can make the overhead of maintaining multiple memories prohibitive in real-world applications, especially for resource-constrained systems like embedded devices. The ability of SARL to achieve superior results with a single model further underscores its practicality and efficiency for lifelong learning.

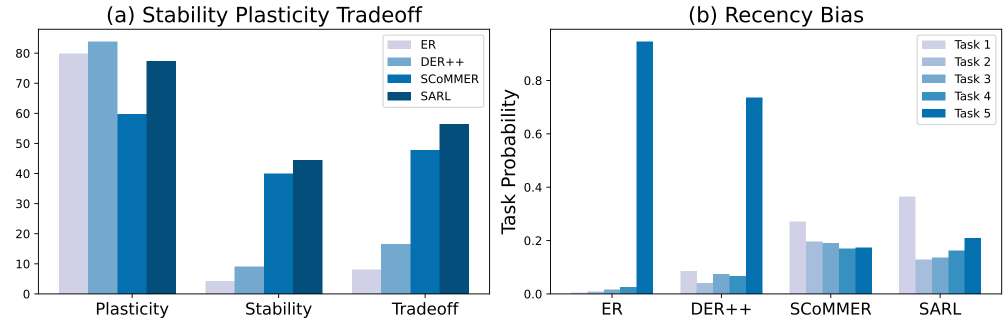


Figure 4: (a) Comparison of the stability and plasticity of the models; and (b) average probability of predicting each task at the end of training for models trained on Seq-CIFAR100 with 200 buffer size. Figure 5 in Appendix shows the task-wise performance of the different models.

5 ANALYSIS

Here, we delve deeper into what enables performance improvements in SARL. We compare with the baseline ER and the state-of-the-art in single-model (DER++) and multiple-memory (SCoMMER).

5.1 SEMANTIC STRUCTURE

At the core of SARL is semantic-aware representation learning. We investigate whether SARL can learn and retain a meaningful semantic structure during sequential task training. We evaluate the similarities between object prototypes calculated stored in memory during training that are evaluated at the end of each task for new object categories and compared them with the similarity structure of the trained model on test data. Figure 3 shows that SARL not only retains the similarity structure but also updates it by consolidating new knowledge. Notably, the class prototypes from early tasks, where the model was trained to distinguish only between two classes, show strong similarities even with other classes, indicating that the initial representations were not yet fully optimized for all classes. However, as the model encounters more tasks, the semantic relationships across classes become more refined and meaningful. For instance, classes such as ‘cat’, ‘dog’, and ‘horse’ exhibit higher similarity values post-training, reflecting their semantic coherence, while unrelated classes such as ’plane’ and ’truck’ maintain lower similarities. The semantic-aware metric learning loss, combined with object prototype regularization, guides training to enforce and maintain this semantic structure, allowing the model to incorporate new knowledge without losing the relationships between previously learned classes.

5.2 STABILITY-PLASTICITY TRADE-OFF

To assess how effectively SARL addresses the stability-plasticity dilemma inherent in CL, we employ the stability-plasticity trade-off metric of (Sarfraz et al., 2022). This metric defines plasticity (P) as the average performance of tasks when learned (average of the diagonal in the task-wise performance matrix) and stability (S) as the average performance of the previous t-1 tasks after task t is learned (mean of the last row in the task-wise performance matrix excluding the diagonal element). The trade-off is then calculated as (2xPxS)/(P+S).

Figure 4(a) illustrates that SARL significantly improves the stability of the model without overly compromising plasticity, thus providing a considerably better trade-off. SARL provides a remarkable 362% gain over ER and 127% gain over DER++. SCoMMER, which utilizes an additional exponentially averaged model to accumulate knowledge and uses it for inference, improves the stability of the model compared to ER and DER but it comes at the cost of significant degradation in the plasticity of the model, which becomes crucial as learns over longer task sequences. Figure 5 in Appendix shows that on the last task SCoMMER achieves only 47.7% accuracy compared to 73.60% for SARL. This supports our hypothesis that learning sparse representations in a semantically aware manner fosters synergy between plasticity and stability, leading to a more balanced relationship between the two.

5.3 TASK RECENCY BIAS

The process of sequentially learning tasks in CL introduces a recency bias in the model, leading to a higher likelihood of predicting newly learned objects (Hou et al., 2019). To assess the task recency bias of different methods, we compute the average probability of predicting each task at the end of training by averaging the softmax outputs of the classes associated with each task on the test set. Figure 4(b) shows that SARL is able to considerably reduce task recency bias compared to ER and DER++. The EMA model in SCoMMER also provides more uniform probabilities but we observe a decreasing trend in probability of predicting the last task which can effect the performance of the model in a longer sequence. Interestingly, SARL has higher probability of predicting the first task which suggest that the regularization losses effectively maintain the model's functional space closer to the initially learned function space.

These analyses collectively show that SARL is able to effectively evaluate semantic similarities between objects across tasks and utilize them to enforce a semantic structure in the representation space. This creates a synergy between plasticity and stability, leading to a better balance between the two and reduces forgetting and recency bias.

5.4 ABLATION STUDY

Table 3: Impact of adding distinct elements of SARL on Seq-CIFAR10 performance with 200 buffer size.

Sparsity	$\mathcal{L}_{OP}$	$\mathcal{L}_{SM}$	$\mathcal{L}_{FR}$	Accuracy
✗	✗	✗	✗	$44.79_{\pm 1.86}$
✓	✗	✗	✗	$58.23_{\pm 1.73}$
✓	✓	✗	✗	$62.53_{\pm 0.41}$
✓	✗	✓	✗	$61.60_{\pm 0.58}$
✓	✓	✓	✗	$63.83_{\pm 0.60}$
✓	✓	✓	✓	$70.97_{\pm 0.47}$

Here, we systematically assess the contribution of each key component of SARL by incrementally adding them on top of the baseline ER. Table 3 reveals the importance of each element in improving model performance. Firstly, the inclusion of sparse coding via activation sparsity alone yields a substantial performance boost. This underscores the importance of activation sparsity in mitigating task interference by allowing only the most salient neurons to remain active, effectively controlling task-specific noise and enhancing generalization. Next, we observe the separate effects of adding object prototype loss ($\mathcal{L}_{OP}$) and semantic-aware metric learning ($\mathcal{L}_{SM}$) on top of sparsity. Object prototype loss introduces regularization, stabilizing the model by maintaining the average representations of objects, which further improves accuracy to 63.01%. Semantic-aware metric learning, on the other hand, enhances forward transfer by leveraging the semantic similarities between objects while reducing interference. Notably, when both losses are applied together, the synergy between them becomes evident. The object prototype loss ensures stable representations of previously learned object, which in turn helps semantic-aware metric learning enforce a consistent semantic structure throughout training. Finally, adding forward regularization ($\mathcal{L}_{FR}$) to the mix encourages the model to stay within the functional vicinity of previously learned states as new tasks are introduced. This final combination of all components achieves the best performance, highlighting the cohesive benefits of SARL's components working together to reduce interference and improve knowledge retention.

6 CONCLUSION

Inspired by how the brain uses sparse coding and semantic relationships between objects to inform memory formation and retrieval, we proposed semantic-aware representation learning (SARL), which employs sparse activations to create semantically rich object prototypes which can effectively capture the inherent similarities between objects. The semantic structure is leveraged to align new object representations with similar objects from previous tasks while promoting separation from dissimilar ones. Our approach facilitates feature reuse while reducing interfering, creating a synergy between the plasticity and stability of the model. Our empirical evaluation in challenging CL scenarios demonstrates SARL's effectiveness. By enforcing a semantic structure on the representations of objects across tasks, SARL balances model plasticity and stability, reduces task recency bias, and mitigates forgetting. Our findings present a compelling case for incorporating semantic-aware representation learning similar to the brain's method.

REFERENCES

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=uxxFrDwrE7Y`.

Caterina Artuso, Francesco Bossi, Carmen Belacchi, and Paola Palladino. Effects of semantic relationship and preactivation on memory updating. *Cognitive Processing*, 23(3):407–422, 2022.

Ari Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SkMwpiR9Y7`.

Jeffrey R Binder and Rutvik H Desai. The neurobiology of semantic memory. *Trends in cognitive sciences*, 15(11):527–536, 2011.

Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):5497–5512, 2022.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=N8MaByOzUfb`.

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 9516–9525, 2021.

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

Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

Peter Foldiak. Sparse coding in the primate cortex. *The handbook of brain theory and neural networks*, 2003.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Haoran Li, Jingfeng Wu, and Vladimir Braverman. Fixed design analysis of regularization-based continual learning. In *Conference on Lifelong Learning Agents*, pp. 513–533. PMLR, 2023.

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

Yan-Shuo Liang and Wu-Jun Li. Adaptive plasticity improvement for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7816–7825, 2023.

Wolfgang Maass. On the computational power of winner-take-all. *Neural computation*, 12(11): 2519–2535, 2000.

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, 1995.

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

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504, 2013.

Fei Mi, Lingjing Kong, Tao Lin, Kaicheng Yu, and Boi Faltings. Generalized class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 240–241, 2020.

Xing Nie, Shixiong Xu, Xiyan Liu, Gaofeng Meng, Chunlei Huo, and Shiming Xiang. Bilateral memory consolidation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16026–16035, 2023.

H Freyja Ólafsdóttir, Daniel Bush, and Caswell Barry. The role of hippocampal replay in memory and planning. *Current Biology*, 28(1):R37–R50, 2018.

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.

Quang Pham, Chenghao Liu, and Steven HOI. Continual normalization: Rethinking batch normalization for online continual learning. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=vwLLQ-HwqhZ`.

Biqing Qi, Xinquan Chen, Junqi Gao, Dong Li, Jianxing Liu, Ligang Wu, and Bowen Zhou. Interactive continual learning: Fast and slow thinking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12882–12892, 2024.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems*, pp. 3738–3748, 2018.

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

Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Synergy between synaptic consolidation and experience replay for general continual learning. In *Conference on Lifelong Learning Agents*, pp. 920–936. PMLR, 2022.

Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Sparse coding in a dual memory system for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9714–9722, 2023.

Rajat Saxena, Justin L Shobe, and Bruce L McNaughton. Learning in deep neural networks and brains with similarity-weighted interleaved learning. *Proceedings of the National Academy of Sciences*, 119(27):e2115229119, 2022.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 99–108, 2022.

Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

Zhen Wang, Liu Liu, Yiqun Duan, and Dacheng Tao. Continual learning through retrieval and imagination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8594–8602, 2022a.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2204.04799*, 2022b.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022c.

Yichen Wu, Long-Kai Huang, Renzhen Wang, Deyu Meng, and Ying Wei. Meta continual learning revisited: Implicitly enhancing online hessian approximation via variance reduction. In *The Twelfth International Conference on Learning Representations*, 2024.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987, 2017.

A EXPERIMENTAL SETUP

To ensure fair comparisons between continual learning (CL) methods and isolate algorithmic improvements from training regimen variability, we adopt consistent experimental conditions across all models. In line with Buzzega et al. (2020), we use a ResNet-18 architecture trained with an SGD optimizer and a batch size of 32 for both task data and memory buffer. We use an initial learning rate (lr) of 0.03 with multistep decay scheduler with factor 0.1. Standard data augmentations—random cropping and horizontal flipping—are applied. For all tasks, we include a warm-up stage consisting of 3 epochs, during which the model trains solely on the task-specific data without utilizing memory buffer samples. This allows the model to initialize a strong representation for each task before integrating knowledge from past tasks. For Seq-CIFAR10 we trained for 20 epochs with decay steps For Seq-CIFAR10, the model is trained for 20 epochs with lr decay steps at 15th epoch, for Seq-CIFAR100 and GCIL-CIFAR100, we extend the training to 50 epochs with lr decay steps at 35 and 45. For the Tiny ImageNet dataset, we further increase the number of epochs to 100 with lr decay steps at 70 and 90.

A.1 SPARSE BACKBONE:

In all experiments, we incorporate activation sparsity by replacing ReLU activations with the k-Winner-Take-All (kWTA) activation function. In kWTA, for each layer, only the top k% of activations at each spatial location are allowed to propagate, while the rest are set to zero. This localized sparsity mechanism ensures that the most relevant features are selected at each spatial location, promoting filter specialization and reducing interference across tasks. The level of sparsity (i.e., k%) is kept fixed across all layers and experiments. Beyond this change to the activation function, no other modifications are made to the backbone architecture.

A.2 EVALUATION SETTINGS:

For the Class-IL setting, we follow established baselines and evaluate on three datasets: sequential CIFAR-10 (Seq-CIFAR10) (Krizhevsky et al., 2009), where 10 classes are split into 5 disjoint tasks with 2 classes per task; sequential CIFAR-100 (Seq-CIFAR100) (Krizhevsky et al., 2009), splitting 100 classes into 5 tasks with 20 classes each; and sequential TinyImageNet (Le & Yang, 2015) (Seq-TinyImageNet), where 200 classes are divided into 10 tasks of 20 classes each.

For the Generalized Class-IL (GCIL) setting (Mi et al., 2020), probabilistic modeling is used to randomly vary three task characteristics: the number of classes, the specific classes included, and the sample sizes. As in Sarfraz et al. (2023), we apply this GCIL setting to CIFAR-100, using 20 tasks with 1,000 samples per task and a maximum of 50 classes per task. To separate the effects of class imbalance from the model's ability to learn from recurring classes across tasks, we evaluate the model on both uniform and long-tailed data distributions. For consistency, the GCIL dataset seed is fixed at 1993 for all experiments, ensuring a fair comparison across methods.

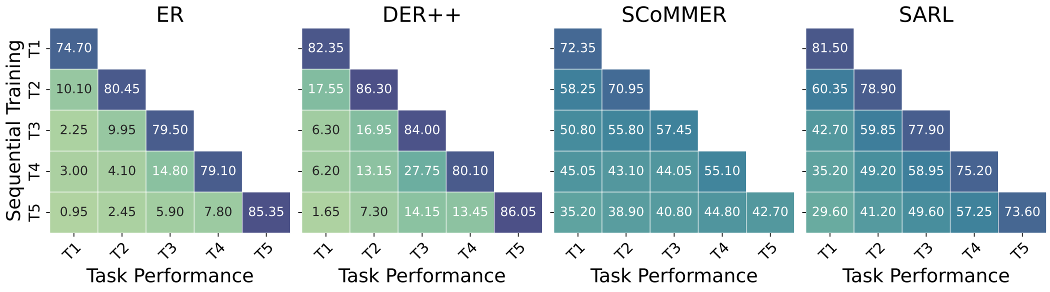


Figure 5: The task-wise performance (x-axis) of the different models assessed after training on each task (y-axis) in Seq-CIFAR100 with a buffer size of 200.

Table 4: Selected hyperparameters for SARL. For all our experiments, we use λ_{SM}=0.01, τ_s=0.8, and β=1.

Dataset	Buffer	kw%	λ_{OP}	α
Seq-CIFAR10	200	0.8	0.5	0.2
	500	0.8	0.5	0.2
Seq-CIFAR100	200	0.9	0.5	0.5
	500	0.9	0.5	0.2
Seq-TinyImageNet	200	0.9	0.2	0.2
	500	0.9	0.5	0.5
GCIL-Uniform	200	0.7	0.2	0.2
	500	0.7	0.2	0.2
GCIL-Longtail	200	0.7	0.2	0.5
	500	0.7	0.2	0.5

B ADDITIONAL RESULTS

B.1 ONLINE CONTINUAL LEARNING

Online continual learning (Online CL) is a highly challenging setting where the model is exposed to each data instance only once, requiring it to learn and adapt dynamically without revisiting previous data. To simulate this scenario, we follow previous works Caccia et al.; Wu et al. (2024) and evaluate our method under the single-head setting. Our evaluation includes two key metrics: the final *Averaged Accuracy* (**Acc**) across all tasks after sequential training and the *Averaged Anytime Accuracy* (**AAA**) Caccia et al., which measures the model's performance throughout the training stream. Let AA_j denote the test average accuracy after training on task T_j. Then, the metrics are defined as:

$$\text{AAA} = \frac{1}{N}\sum_{j=1}^{N}\text{AA}_j, \quad \text{Acc} = \text{AA}_N,$$

where N is the total number of tasks. Following (Wu et al., 2024), we evaluate our method on the Seq-CIFAR10 and Seq-CIFAR100,. Seq-CIFAR10 comprises 5 tasks, each containing 2 classes, Seq-CIFAR100 consists of 10 tasks with 10 classes each.

Although our method is not specifically designed for Online CL, we adapted it to this setting to evaluate its versatility and applicability. We maintain running feature sums and sample counts to calculate dynamic object prototypes during training, and use the first 10 iterations on a task as warmup before starting the accumulation of features for evaluating prototypes. Despite not being explicitly designed for this setting, SARL-Online demonstrates promising results. As shown in Table 5, SARL-Online achieves considerable improvements across all datasets, particularly on Seq-CIFAR100, where it achieves significant gains in both **AAA** and **Acc**.

B.2 RESULTS ON VIT BACKBONE

To further test the versatility of our approach, we evaluate SARL on the ViT Small backbone. It is well known that Vision Transformers (ViTs) struggle with smaller datasets due to their inherent architectural requirements, which demand pretraining on large-scale datasets to achieve performance comparable to convolutional networks like ResNet. This limitation is particularly evident in continual learning settings with smaller datasets, leading to a noticeable performance gap between ViT and ResNet backbones.

Despite this inherent challenge, Table 6 shows that SARL provides considerable performance gains over ER across all evaluation settings. These results highlight the robustness and adaptability of SARL, even in scenarios where the backbone architecture is less suited for the dataset size. Importantly, it should be noted that no hyperparameter tuning was performed specifically for this setting. Instead, we used the same hyperparameters optimized for the ResNet backbone. This suggests that

Table 5: Performance comparison across Seq-CIFAR10 and Seq-CIFAR100 in the online CL setting. Baseline results are taken from Wu et al. (2024). We report the mean and std of three runs.

	Seq-CIFAR10		Seq-CIFAR100	
Method	**AAA**	**Acc**	**AAA**	**Acc**
SGD	34.85±1.71	16.96±0.60	11.63±0.38	5.27±0.28
LWF	35.31±0.98	18.84±0.10	11.98±0.40	5.63±0.37
A-GEM	38.66±0.79	18.46±0.17	13.15±0.23	6.02±0.17
GEM	38.67±0.77	18.49±0.15	15.18±0.38	8.30±0.58
ER	55.53±2.58	43.83±4.84	23.19±0.38	16.07±0.88
DER	45.85±1.62	29.87±2.95	13.35±0.36	6.12±0.18
DER++	64.22±0.70	52.29±1.86	19.88±0.43	11.79±0.65
CLSER	63.02±1.54	52.80±1.66	25.46±0.57	17.88±0.69
OCM	66.14±0.95	53.39±1.00	22.54±0.79	14.40±0.82
ER-OBC	65.82±0.91	54.85±2.16	25.54±0.25	17.21±0.92
On-EWC	38.44±0.50	17.12±0.51	11.81±0.42	5.88±0.31
IS	37.33±0.23	17.39±0.19	12.32±0.22	5.20±0.18
MER	50.99±0.65	36.92±2.42	–	–
La-MAML	42.98±1.60	33.43±1.21	12.55±0.39	11.78±0.65
VR-MCL	**66.97**±1.58	56.48±1.79	27.01±0.48	19.49±0.69
SARL	66.85±1.15	**57.21**±0.27	**31.66**±1.49	**24.39**±1.44

further improvements in performance are possible by tailoring the hyperparameters to better suit the ViT architecture and adapting the functional regularization.

The significant performance improvements observed with SARL validate its capability to enhance learning effectiveness across diverse backbone architectures, emphasizing its potential as a versatile and robust continual learning method.

Table 6: Performance comparison on ViT-Small backbone. We report the mean and std of three runs.

			Seq-CIFAR10		Seq-CIFAR100	
Backbone	Buffer	Method	Class-IL	Task-IL	Class-IL	Task-IL
ViT-Small	200	ER	21.92±1.70	69.65±4.05	11.90±1.75	29.11±7.05
		SARL	**26.04**±0.45	**82.74**±1.39	**19.08**±3.00	**44.82**±4.46
	500	ER	24.41±1.46	72.30±1.79	12.51±1.47	31.08±4.25
		SARL	**32.94**±1.14	**83.25**±1.86	**22.00**±0.79	**47.68**±1.99

C ADDITIONAL METRICS

We provide additional metrics for all our experiments to comprehensively evaluate SARL's performance. Since the number of test samples per task is uniform across all tasks, the final accuracy, calculated as the total number of correctly classified samples across all tasks divided by the total number of samples, is equivalent to the average accuracy, which is the mean accuracy across individual tasks at the end of training. Table 7 provides additional metrics such as forgetting, stability, and plasticity, which provide deeper insights into model behavior. Forgetting measures the model's ability to retain knowledge from previous tasks by quantifying the average drop in accuracy of a task at the end of continual learning training compared to its accuracy when first learned, where lower forgetting indicates better knowledge preservation. Stability (S) reflects the average accuracy on previously learned tasks at the end of training, demonstrating how well the model performs on earlier tasks. Plasticity (P), on the other hand, evaluates the model's ability to effectively learn new tasks, measured as the average accuracy of tasks when they are initially trained. Trade-off is measured by (2 * S * P) / (P + S) and measure how well the method keeps a balance between the stability and plasticity of the model. Together, these metrics offer a holistic view of SARL's performance, capturing its strengths and trade-offs in diverse continual learning scenarios.

Note that the vast majority of baselines do not provide these metrics and hence we cannot compare the baselines across these metrics unfortunately. In addition to the comparison on Average Accuracy in Table 1, we compare the forgetting metrics for the baselines in Table 8 (where available). The forgetting metrics for baselines are from Buzzega et al. (2020) and Boschini et al. (2022). Note that DER++ is the strongest single model baseline and SARL considerable reduces forgetting by a considerable margin.

Table 7: Performance metrics across datasets and buffer sizes.

Dataset	**Buffer Size**	**Accuracy (%)**	**Forgetting (%)**	**Stability (%)**	**Plasticity (%)**	**Trade-off**
seq-cifar10	200	$70.97_{\pm 0.47}$	$14.83_{\pm 0.43}$	$67.72_{\pm 1.05}$	$82.64_{\pm 0.73}$	$74.44_{\pm 0.78}$
	500	$75.64_{\pm 0.36}$	$8.75_{\pm 6.89}$	$73.39_{\pm 1.00}$	$85.31_{\pm 0.92}$	$78.90_{\pm 0.88}$
seq-cifar100	200	$48.96_{\pm 0.53}$	$33.82_{\pm 0.56}$	$43.12_{\pm 0.91}$	$76.45_{\pm 1.01}$	$55.14_{\pm 0.98}$
	500	$55.30_{\pm 0.61}$	$26.78_{\pm 1.53}$	$50.21_{\pm 1.39}$	$76.73_{\pm 0.62}$	$60.69_{\pm 0.83}$
seq-tinyimg	200	$28.95_{\pm 1.13}$	$42.08_{\pm 2.94}$	$23.99_{\pm 1.41}$	$66.82_{\pm 1.52}$	$35.27_{\pm 1.29}$
	500	$32.56_{\pm 1.23}$	$27.34_{\pm 0.56}$	$29.88_{\pm 1.52}$	$57.16_{\pm 1.73}$	$39.24_{\pm 0.72}$

D HYPERPARAMETER TUNING

During hyperparameter tuning, we used a small validation set to adjust values for kw%, λ_{OP}, and the coefficient α for $\mathcal{L}_{FR}$. For all experiments, we set $\tau_s = 0.8$, $\beta = 1$ and $\lambda_{SM} = 0.01$. The value of kw% was selected from the set {0.7, 0.8, 0.9}, while λ_{OP} and α, were chosen from {0.2, 0.5, 1}. The final selected parameters are presented in Table 4. It is important to note that we did not perform an exhaustive hyperparameter search, so better configurations may exist. Furthermore, we observed a synergistic relationship among the parameters, and the model demonstrated robustness to a range of values, which significantly eased the tuning process.

To further illustrate this, the results in Table 9 demonstrate that SARL is not overly sensitive to specific hyperparameter values, as multiple configurations yield consistent performance improvements. For instance, across different values of α , λ_{OP} , and kw% , the method achieves robust Class-IL and Task-IL accuracies on Seq-CIFAR10, with minor variations in performance. While certain configurations (e.g., $\alpha = 0.2$, $\lambda_{OP} = 0.5$, kw% = 0.8) achieve the highest Class-IL accuracy (70.97%), other values (e.g., kw% = 0.9, $\alpha = 0.2$, $\lambda_{OP} = 0.5$) provide similar results (70.08%). We present results for all the grid values explored during hyperparameter tuning, demonstrating that SARL achieves consistent performance improvements across various configurations without relying heavily on precise hyperparameter adjustments. This reliability underscores SARL's practicality and suitability for a wide range of continual learning applications

Table 8: Forgetting analysis of CL methods across various CL settings.

Buffer	Method	Seq-CIFAR10	Seq-CIFAR100
200	ER	$61.24_{\pm 2.62}$	74.16
	FDR	$84.40_{\pm 2.67}$	-
	DER++	$32.59_{\pm 2.32}$	74.05
	SARL	$\mathbf{14.83}_{\pm 0.43}$	$\mathbf{33.82}_{\pm 0.56}$
500	ER	$45.35_{\pm 0.07}$	73.64
	FDR	$85.62_{\pm 0.36}$	-
	DER++	$22.38_{\pm 4.41}$	50.54
	SARL	$\mathbf{8.75}_{\pm 6.89}$	$\mathbf{26.78}_{\pm 1.53}$

Table 9: Sensitivity to hyperparameters on the performance of SARL. For all the experiments, we use learning rate $\eta = 0.1$, $\alpha = 0.999$, $\lambda_{OP} = 0.5$, $\lambda_{SA} = 0.01$, $\lambda_{CR} = 0.15$ and $\tau = 0.8$.

			seq-cifar10	
kw%	α	λ_{OP}	Class-IL	Task-IL
0.8	0.2	0.2	$70.73_{\pm 2.31}$	$95.77_{\pm 0.16}$
		0.5	$70.97_{\pm 0.47}$	$95.72_{\pm 0.36}$
		1	$70.57_{\pm 0.73}$	$95.71_{\pm 0.08}$
	0.5	0.2	$68.53_{\pm 1.14}$	$95.23_{\pm 0.07}$
		0.5	$69.40_{\pm 1.21}$	$95.21_{\pm 0.19}$
		1	$68.79_{\pm 0.84}$	$95.14_{\pm 0.09}$
	1	0.2	$65.30_{\pm 1.40}$	$93.73_{\pm 0.68}$
		0.5	$64.59_{\pm 0.62}$	$93.72_{\pm 0.42}$
		1	$63.86_{\pm 1.45}$	$93.19_{\pm 0.08}$
0.9	0.2	0.2	$70.05_{\pm 1.44}$	$95.49_{\pm 0.14}$
		0.5	$70.08_{\pm 1.48}$	$95.63_{\pm 0.22}$
		1	$69.07_{\pm 0.68}$	$95.52_{\pm 0.01}$
	0.5	0.2	$68.12_{\pm 2.90}$	$95.13_{\pm 0.41}$
		0.5	$66.66_{\pm 0.55}$	$94.92_{\pm 0.31}$
		1	$65.49_{\pm 0.08}$	$94.48_{\pm 0.38}$
	1	0.2	$63.99_{\pm 2.83}$	$93.34_{\pm 0.91}$
		0.5	$64.04_{\pm 0.06}$	$93.00_{\pm 1.05}$
		1	$63.80_{\pm 1.56}$	$92.87_{\pm 1.33}$

Table 10: Memory and performance comparison of SARL and baselines on Seq-CIFAR100.

Method	#Models	#FWD Passes (Tasks)	#FWD Passes (Buffer)	Buffer Size (MB)	Total Memory (MB)	Memory Ratio (vs ER)	Avg. Acc (%)	Avg. Acc / Memory Ratio
ER	1	1	1	5.86	48.52	1.0000	28.02	28.02
DER++	1	1	2	6.05	48.71	1.0039	41.40	41.24
CLS-ER	3	1	3	5.86	133.84	2.7584	51.40	18.63
SCoMMER	2	1	2	5.86	91.18	1.8792	49.63	26.41
SARL	1	1	1	8.15	50.09	1.0323	**55.30**	**53.57**

E SEMANTIC SIMILARITY BETWEEN CLUSTERS

To further strengthen our analysis in Figure 2 which provides similarity matrices for dense and sparse activations, we defined two clusters within the CIFAR-10 classes: the Animal Cluster, comprising ‘bird,’ ‘cat,’ ‘deer,’ ‘dog,’ ‘frog,’ and ‘horse,’ and the Vehicle Cluster, consisting of ‘airplane,’ ‘automobile,’ ‘ship,’ and ‘truck.’ The comparison of inter- and intra-cluster similarities is visualized in Figure 6, and the results indicate that sparse activations exhibit consistently higher average similarity within clusters compared to dense activations. For the Animal Cluster, sparse activations achieve an average similarity of 0.8873 compared to 0.8367 for dense activations. Similarly, for the Vehicle Cluster, sparse activations achieve 0.8694 compared to 0.8684 for dense activations. This suggests that sparse activations better group objects within the same semantic category, capturing semantic cohesion more effectively. By activating only the most relevant neurons, sparse representations reduce noise and focus on features critical for semantic grouping.

Sparse activations also demonstrate a slightly higher average similarity between clusters, with 0.8295 compared to 0.8068 for dense activations. However, this inter-cluster similarity is significantly lower than the intra-cluster similarity, aligning with the expected semantic distinction between clusters. Notably, sparse representations encode inter-cluster relationships more semantically rather than merely enforcing hard separations. For instance, airplane and bird, which share abstract features like wings and the ability to fly, exhibit a higher similarity of 0.90 with sparse activations compared to 0.87 with dense activations. In contrast, ship and horse, which are semantically dissimilar, show a similarity of 0.81 with sparse activations compared to 0.83 with dense activations. These findings reinforce our hypothesis that sparse activations can enhance the model’s capability to capture semantic structure across objects.

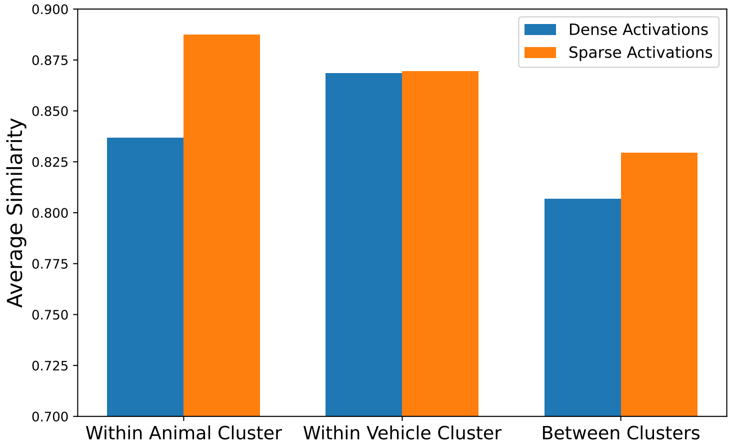


Figure 6: Inter- and Intra-Cluster Similarities Across Animal and Vehicle Clusters in CIFAR-10 for Dense and Sparse Activations

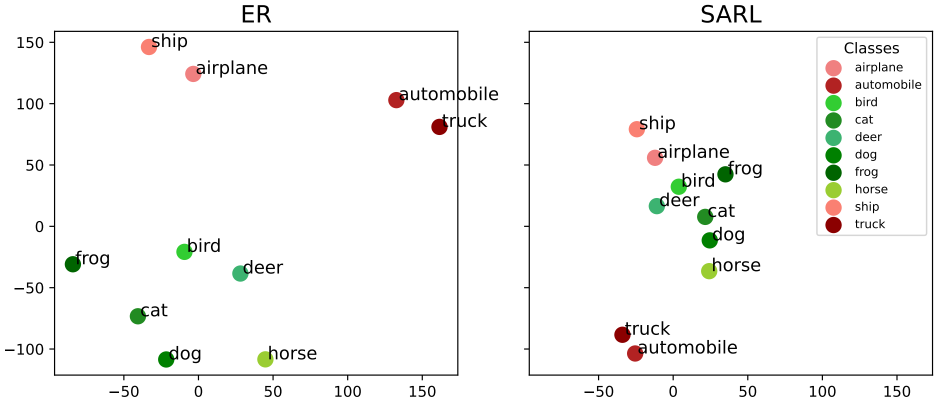


Figure 7: t-SNE plot of object prototypes using trained model with ER and SARL on Seq-CIFAR10 with 200 buffer size.

F VISUALIZATION OF REPRESENTATION SPACE

To further analyze the representations and evaluate how well SARL captures and enforces a semantic structure among objects, we project the object prototypes into a two-dimensional space using t-SNE and compare SARL with ER on Seq-CIFAR10 with a buffer size of 200 (Figure 7). The t-SNE visualizations reveal clear differences in the representational quality of SARL compared to ER. In the left panel (ER), object prototypes exhibit a lack of structured clustering, with semantically similar classes, such as animals ("bird," "frog," "deer"), being dispersed and overlapping with other categories. This indicates that ER struggles to effectively group related objects or separate distinct semantic categories in the representation space. In contrast, the right panel (SARL) demonstrates significant improvements, forming compact and cohesive clusters for semantically similar objects. For example, animals are tightly grouped together, and SARL effectively captures the semantic similarity between "bird" and "airplane," a relationship that ER fails to encode. This structured organization of prototypes highlights SARL's ability to leverage semantic relationships to guide representation learning, aligning similar classes while maintaining clear distinctions between unrelated ones. These findings underscore the effectiveness of SARL in creating a more meaningful, semantically aware, and structured representation space.

G EXTENDED RELATED WORKS

Continual learning has seen significant advancements in recent years, with diverse approaches targeting the challenges of catastrophic forgetting, model stability, and effective knowledge transfer across tasks. In this section, we extend our related works section and position our SARL in the broader context of these developments.

Meta-learning approaches have shown promise in continual learning by leveraging task-specific adaptations. VR-MCL (Wu et al., 2024) proposes an online CL approach that combines Meta-CL with regularization to efficiently approximating online Hessians, enhancing adaptability in real-time learning scenarios. While our work adopts a batch training setup, allowing multiple passes over sequential task data which we believe is more aligned with many real-world scenarios where datasets can be revisited periodically (e.g., staged data collection or iterative training), VR-MCL provides a focuses on online single-pass learning. The fundamental differences in training settings make direct comparisons challenging, but we recognize the potential of such meta-learning techniques to inspire future extensions of SARL in online CL. Interactive Continual Learning (ICL) (Qi et al., 2024) utilizes multimodal large language models (LLMs) to achieve state-of-the-art performance in continual learning. By leveraging extensive pretraining on diverse datasets (often a superset of classes in the baseline CL datasets) and significantly larger architectures, this approach gains a lot of advantage over the standard setting. It also makes them infeasible for many CL applications due to the magnitudes larger memory and computational costs. The reliance on pretraining introduces an inherent disparity when comparing with SARL, which operates with a randomly initialized model and focuses on both learning robust representations in the CL regime and retaining them as training progresses. Furthermore, Memory-based approaches are widely used in rehearsal-based CL to alleviate catastrophic forgetting. Bilateral Memory Consolidation (BiMeCo) (Nie et al., 2023) introduces a method that consolidates long-term and short-term memories to improve performance. Their approach is complementary to SARL and could potentially enhance SARL further if integrated. However, their evaluation setting of involves pretraining on a large number of base classes before sequential learning, whereas SARL and its baselines initiate continual learning with a randomly initialized model. This distinction in experimental setups underscores the importance of careful contextualization when comparing results across studies.

Prompt-based methods, such as L2P (Wang et al., 2022c) and DualPrompt (Wang et al., 2022b), utilize pre-trained backbones, like those trained on ImageNet, to achieve strong performance in class incremental learning setting. These pre-trained backbones inherently include representations for many classes encountered later in CL tasks or for classes that are semantically similar. This allows these methods to bypass one of the core challenges of CL: learning and preserving general representations of objects across sequential tasks. In contrast, SARL operates in a more challenging setting, starting from a randomly initialized state without leveraging pretraining. Extending SARL to integrate pre-trained models could be a promising direction for future research, but the current work focuses on enabling robust representation learning from a random start utilizing the semantic structure to learn and retain representations.

We would like to emphasize that evaluating CL methods across diverse baselines presents several challenges. Variations in architectures, the use of additional pretraining, access to implicit or explicit auxiliary information, and differences in task structures and metrics significantly influence reported results. Uniform evaluation settings are critical for meaningful assessments, as discrepancies in setup may reflect differences in methodology rather than genuine improvements.