

SHELs: EXCLUSIVE FEATURE SETS FOR NOVELTY DETECTION AND CONTINUAL LEARNING WITHOUT CLASS BOUNDARIES

Meghna Gummedi, David Kent, Jorge A. Mendez, and Eric Eaton

Department of Computer and Information Science

University of Pennsylvania

{meghnag, dekent, mendezme, eeaton}@seas.upenn.edu

ABSTRACT

While deep neural networks (DNNs) have achieved impressive classification performance in closed-world learning scenarios, they typically fail to generalize to unseen categories in dynamic open-world environments, in which the number of concepts is unbounded. In contrast, human and animal learners have the ability to incrementally update their knowledge by recognizing and adapting to novel observations. In particular, humans characterize concepts via exclusive (unique) *sets* of essential features, which are used for both recognizing known classes and identifying novelty. Inspired by natural learners, we introduce a Sparse High-level-Exclusive, Low-level-Shared feature representation (SHELs) that simultaneously encourages learning exclusive sets of high-level features and essential, shared low-level features. The exclusivity of the high-level features enables the DNN to automatically detect out-of-distribution (OOD) data, while the efficient use of capacity via sparse low-level features permits accommodating new knowledge. The resulting approach uses OOD detection to perform class-incremental continual learning without known class boundaries. We show that using SHELs for novelty detection results in statistically significant improvements over state-of-the-art OOD detection approaches over a variety of benchmark datasets. Further, we demonstrate that the SHELs model mitigates catastrophic forgetting in a class-incremental learning setting, enabling a combined novelty detection and accommodation framework that supports learning in open-world settings.

1 INTRODUCTION

The successes of deep neural networks (DNNs) have occurred mainly in closed-world learning paradigms, where models are trained to handle fixed distributions of learning problems. To operate in realistic open-world environments, DNNs must be extended to support learning after deployment to adapt to new information and requirements (Liu, 2020). This necessitates both the abilities to *detect* novelty, as addressed by out-of-distribution (OOD) detection, and *accommodate* this novelty into the learnt model, as addressed by continual or lifelong learning. However, these two problems have rarely been examined together. The work we present here bridges these two fields by developing a novel OOD detector that automatically triggers accommodation of new knowledge in a natural continual learning loop.

Work in continual learning focuses on learning tasks (or classes) consecutively, incrementally acquiring new behaviors over changing data distributions (Kirkpatrick et al., 2017; Jung et al., 2020; Yang & Hospedales, 2017; Lee et al., 2019; 2021; Shin et al., 2017; Van de Ven et al., 2020). However, most existing approaches must be explicitly told when new tasks are introduced. In contrast, humans automatically detect when current percepts do not match known patterns (Weizmann et al., 1971), spurring the refinement or introduction of new knowledge without explicit task or class boundaries (Colombo & Bundy, 1983; Hunter et al., 1983). In this way, novelty detection and accommodation are inherently intertwined in natural learners, with the learnt representations both characterizing known concepts and identifying their boundaries. Tversky (1977) describes this process as a feature matching problem, whereby humans represent objects as a collection of essential features, enabling them to recognize novel instances via 1) the presence of a new feature, 2) the absence of known features, or 3) a new combination of known and unknown features. Motivated by this idea, we explore how we can incorporate such representations into open-world continual learning.

The key insight of our work is to represent each class as an *exclusive set* of high-level features within a DNN, while lower level features can be shared. The SHELs formulation realizes the exclusivity at the high levels by leveraging cosine normalization, and encourages sharing at the lower levels via group sparsity, leading to representations that consist of only essential features. Such structures identify combinations of features that encapsulate a unique *signature* for each class. Comparing the features triggered by OOD percepts with known class signatures enables the learner to detect novel classes and accommodate them, endowing it with the ability to learn continually without class boundaries.

The key contributions of this work include:

- We describe a novel mechanism for detecting OOD data within DNNs by representing concepts as exclusive combinations of reusable lower level sparse features. We call this representation SHELS.
- We develop a continual learning algorithm that leverages these exclusive feature sets in tandem with sparsity to incrementally learn concepts, mitigating catastrophic forgetting of previously learnt knowledge.
- We demonstrate a combined framework that supports class-incremental continual learning without the specification of explicit class boundaries, enabling the agent to operate effectively in an open-world setting through the integration of novelty detection and accommodation.

2 REPRESENTING CONCEPTS AS EXCLUSIVE SETS OF HIGHER LEVEL FEATURES

It is well-established in developmental psychology that children characterize concepts as collections of essential features, known as schemas (Piaget, 1926). As an example, a child living with a dalmatian dog may learn to characterize it by its four-legs, fur, tail, black and white spots, and medium size. These features constitute the child’s dog schema. Critically, schemas can share features (enabling transfer between concepts), but are differentiated by their unique sets of essential features (Tversky, 1977). The child may also have a horse schema that shares the four-legs, fur, and tail features, but is larger. When the child first encounters a cow, it may differentiate it as a new animal, since it has never before seen a large animal with four-legs, spots, and a tail. It is the exclusive set of essential features that enables the child to both recognize concepts and detect that the cow is OOD. The child may mistakenly call the cow a horse, using their closest matching schema. With their parent’s correction, the child can create a new cow schema to accommodate the new information. Thus, both features and schemas are continually acquired and refined over time.

From a mathematical standpoint, schemas represent a collection of exclusive sets of features. Precisely, we can state:

Definition 1 (Exclusive Sets). *Two sets S, T are exclusive if and only if they each contain unique elements: $\text{exclusive}(S, T) \iff S - T \neq \emptyset \wedge T - S \neq \emptyset$. (Note that this definition inherently precludes subset relationships and prevents S and T from being empty.)*

Definition 2 (Collection of Exclusive Sets). *A collection of sets \mathcal{C} is exclusive if and only if every pair of sets within that collection is exclusive: $\text{exclusive}(\mathcal{C}) \iff \forall S, T \in \mathcal{C} S \neq T \implies \text{exclusive}(S, T)$.*

To manifest the idea of schemas in a representational space for machine learning, we embed each data instance $x \in \mathbb{R}^d$ as a set of derived features $f = g(x) \in \mathbb{R}^k$. Our goal is for the non-zero entries of the embeddings f of each class to form a collection of exclusive sets. Orthogonality between the vectors in the resulting vector space captures this notion of exclusive feature sets. In particular, a set of orthogonal feature embeddings $f_1, \dots, f_C \in \mathbb{R}^k$ for each of C classes forms a collection of exclusive sets (see Appendix A for the proof). In fact, orthogonality is more stringent than exclusivity, since non-orthogonal features can still constitute exclusive sets. However, our work shows empirically that orthogonal feature embeddings are sufficient for both characterizing classes and enabling OOD detection.

To impose orthogonality, we employ cosine normalization (Luo et al., 2018) and use cosine scores in the loss function when learning. This embeds the in-distribution (ID) classes along the surface of the unit ball, spacing them apart. Intuitively, this maximizes the chance that a new OOD class would have an embedding $g(x)$ with low cosine similarity with the other classes, facilitating OOD detection. A continual learner utilizing this technique can then adapt its embedding to preserve orthogonality when accommodating the new class, permitting future OOD detection of other new classes. Figure 1 illustrates this process, which we develop into a complete algorithm in Section 4.

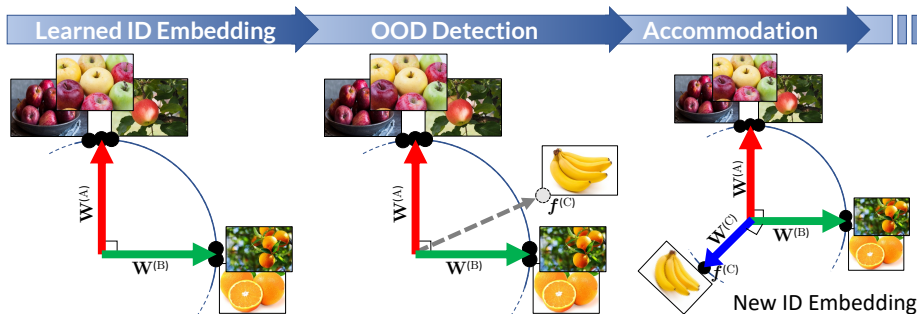


Figure 1: ID data is represented by exclusive sets of features, corresponding to orthogonal embeddings along the unit ball with broad separation of the classes. An OOD instance in the same embedding lies in a different region, enabling detection of the novelty and triggering accommodation of the embedding.

3 RELATED WORK

The notions of orthogonality and sparsity discussed above connect to work in novelty detection and continual learning. We explore those connections in the following paragraphs, and survey other (less) related literature in Appendix B.

Novelty Detection Novelty detection has been studied in various forms, including open-world recognition (Bendale & Boulton, 2015), uncertainty estimation (Lakshminarayanan et al., 2017), and anomaly detection (Andrews et al., 2016). A popular baseline proposed by Hendrycks & Gimpel (2017) thresholds the softmax scores output by a network to detect OOD inputs. Similarly, OpenMax (Bendale & Boulton, 2016) compares the distance of the penultimate layer activation patterns of a sample to those of ID data. Such distance-based methods fall short when ID and OOD data are similar, which is a challenging case that we explicitly consider for SHELS. ODIN (Liang et al., 2018) complements thresholding with input pre-processing and temperature scaling to compute more-effective softmax scores. The method by Lee et al. (2018) models every activation layer learnt using ID data as a class-conditional Gaussian distribution, and defines confidence scores using the Mahalanobis distance with respect to the nearest class-conditional distribution. However, both of these approaches assume access to sample OOD data to tune their hyperparameters, which is unrealistic in open-world settings. Techapanurak et al. (2020) propose to compute the softmax scores over the temperature-scaled cosine similarity, outperforming both ODIN and Mahalanobis while making use of only ID data for hyperparameter tuning. Our SHELS approach also uses cosine similarity, but unlike existing formulations, SHELS encourages the embeddings of ID classes to be orthogonal (and therefore exclusive), which we demonstrate improves OOD detection.

Continual Learning The major challenge in continual learning is avoiding catastrophic forgetting, whereby learning new knowledge interferes with previously known concepts (McCloskey & Cohen, 1989). A variety of approaches have been proposed to mitigate this problem, from replay-based methods that use samples of previous task data to retain performance (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019), to regularization-based strategies that selectively update weights to avoid forgetting (Kirkpatrick et al., 2017; Zenke et al., 2017). The regularization-based method by Jung et al. (2020) uses a combination of sparsity and weight update penalties based on the importance of nodes in the network. We take a similar approach, but use a more structured form of sparsity along with high-level exclusivity to enable both continual learning and OOD detection. Several recent continual learning methods use orthogonality to avoid interference between tasks, embedding each task model on an orthogonal subspace (Chaudhry et al., 2020; Saha et al., 2021; Deng et al., 2021; Duncker et al., 2020). We explore a complementary use of orthogonality, to create exclusive feature sets that enable OOD detection during continual learning. Notably, the only prior work studying both OOD detection and continual learning reuses previously learnt representations to accommodate new classes (Lee et al., 2018), limiting its applicability to settings where the ID representations are sufficient to handle OOD data. Moreover, they do not demonstrate the use of novelty detection and accommodation as an integrated system as we do.

4 FRAMEWORK

Our approach trains DNNs that have 1) high-level feature sets that are exclusive to each class and 2) low-level features that are shared among classes. We call this the *SHELS* representation—a Sparse High-level-Exclusive Low-level-Shared representation. SHELS representations promote effective class discrimination and form capacity-efficient models, enabling both novelty detection and accommodation. Learning features this way differs substantially from typical DNN learning, which does not optimize for network efficiency and therefore exploits the full representational capacity, leading to representations that are redundant, correlated, and sensitive to the input distribution. Our approach learns SHELS representations using a combination of sparsity regularization and cosine normalization, enabling DNNs to learn features that are non-correlated, essential, and (at higher levels) exclusive to each class. We exploit this representation to detect new classes as novel activation patterns of high-level features, enabling OOD detection. We then accommodate the necessary information for new classes into the DNN by identifying and updating only those nodes that are unimportant to previously learnt classes, thereby mitigating catastrophic forgetting for continual learning.

4.1 EXCLUSIVE HIGH-LEVEL FEATURE SETS WITH SHARED LOW-LEVEL FEATURES

We next develop the regularization terms and overall loss function of our approach.

Notation For a neural network of L hidden layers, let \mathbf{W}^l denote the weight matrix for layer $l \in 1, \dots, L$, with all weight matrices collected together into a tensor of parameters \mathbf{W} . The vector of outgoing weights from node n_g in layer l , which we denote a *group*, is given by $\mathbf{W}^l_{[g,:]}$, while the vector of incoming weights to node n_i is given by $\mathbf{W}^l_{[:,i]}$. $\Omega(\mathbf{W}^l)$ is a regularization term on the network weights at layer l and $\rho(n)$ is the average activation of a node n over a dataset. The features produced by layer l of the network on input \mathbf{x} are represented by $\mathbf{f}^l(\mathbf{x})$ and $f_n(\mathbf{x})$ denotes the feature activation at node n . We use $c \in 1, \dots, C$ to denote seen ID classes and $D_{1:C}$ to denote the dataset over all seen classes. In continual learning formulations, we use $\mathbf{W}^{(c)}$ to denote the weight tensor trained up to the c -th class.

4.1.1 EXCLUSIVE HIGH-LEVEL FEATURES

To impose exclusivity in the high-level features (Figure 2a), we employ cosine normalization in the last layer of the neural network, in place of the usual dot product. Typical classification networks predict class scores for a given input \mathbf{x} by applying the softmax function to the output of the last linear layer: $\text{softmax}(\mathbf{W}^L \mathbf{f}^{L-1}(\mathbf{x}) + \mathbf{b}^L)$. We replace this linear transformation with the cosine of the angle between the weights $\mathbf{W}_{[:,c]}^L$ for class c and the features $\mathbf{f}^{L-1}(\mathbf{x})$ (i.e., the cosine similarity) to compute class scores:

$$\cos^{(c)}(\mathbf{W}, \mathbf{x}) = \frac{\mathbf{W}_{[:,c]}^{L\top} \mathbf{f}^{L-1}(\mathbf{x})}{\|\mathbf{W}_{[:,c]}^L\| \|\mathbf{f}^{L-1}(\mathbf{x})\|}. \quad (1)$$

Using the raw cosine similarity scores as the class scores encourages the weights of the last layer to be sensitive to *which* features are activated instead of the degree of the feature activations. Therefore, training the network to optimize for the raw cosine scores results in orthogonal—and consequently exclusive—high-level features per class. We empirically demonstrate that cosine normalization leads to exclusive features sets, and compare the exclusivity in our method to two baseline methods: the standard CNNs used by most other novelty detection approaches, and the method of [Techapanurak et al. \(2020\)](#), which uses a form of cosine similarity. See Appendix C for this analysis.

As an alternative, we explored the exclusive sparsity regularizer of [Zhou et al. \(2010\)](#), which encourages competition among features selected for a class. We found it to be less effective than cosine normalization (see Appendix D), since it encourages *individual* features to be exclusive to a class, in contrast to *sets* of features being exclusive to a class.

4.1.2 SHARED LOWER LEVEL FEATURES

Sparse regularizers are primarily used to reduce the complexity of DNNs by zeroing out certain weights. However, we can also use sparse regularizers to enable continual learning. For class-incremental continual learning, we need the ability to accommodate new information into the network without disrupting any knowledge learnt for previous classes. We accomplish this by creating sparsity in the network during training, allowing us to maintain unused nodes for learning future classes. We impose such sparsity in a structured fashion using the group Lasso regularizer defined by [Yuan & Lin \(2006\)](#) and adapt it to DNNs ([Yoon & Hwang, 2017](#); [Wen et al., 2016](#)), promoting the removal of redundancies by eliminating correlated features while also encouraging the sharing of features:

$$\Omega_G(\mathbf{W}^l) = \sum_g \|\mathbf{w}_g^l\|_2 = \sum_g \sqrt{\sum_i (\mathbf{w}_{[g,i]}^l)^2}. \quad (2)$$

Group Lasso promotes inter-group sparsity; that is, it eliminates groups (nodes) in the lower level that are redundant and are not shared among multiple high-level groups, as illustrated in Figure 2b.

4.1.3 LAYERED SPARSE REPRESENTATION LEARNING

To learn SHELS representations, we combine group sparsity at different layers with cosine normalization at the last layer of the network by optimizing for the following loss:

$$\mathcal{L}(\mathbf{W}) = \underbrace{\frac{1}{|D_{1:C}|} \sum_{c=1}^C \sum_{k=1}^{|D_c|} -\log(\cos^{(c)}(\mathbf{W}, \mathbf{x}_k))}_{\mathcal{L}_{CE}} + \alpha \sum_l (1 - \mu^{(l)}) \underbrace{\left(\sum_g \|\mathbf{W}_{[g,:]}^l\|_2 \right)}_{\Omega_G}, \quad (3)$$

where \mathcal{L}_{CE} is the standard cross-entropy loss applied to the cosine similarity scores of Equation 1 and Ω_G is the group sparsity regularizer. The parameter α controls the amount of regularization, and by setting $\mu^{(l)} = \frac{l-1}{L-1}$ we can interpolate between more sharing at the lower layers and less sharing (and therefore more exclusivity) at the higher layers, with $\mu^{(1)} = 0$ and $\mu^{(L)} = 1$ at the outermost layers. We provide an empirical analysis of the effects of group sparsity regularization on OOD detection by varying the α and $\mu^{(l)}$ hyperparameters in Appendix E.

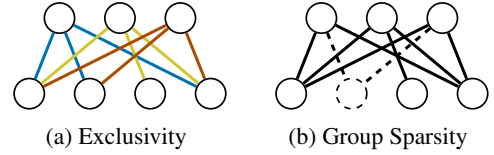


Figure 2: (a) Exclusivity encourages each upper-level node to choose an exclusive set of lower level nodes (the matching colored edges). (b) Group Sparsity eliminates neurons that are redundant or not shared by multiple upper-level nodes (the dashed-edge neuron).

4.2 NOVELTY DETECTION

With the learnt model composed of SHELS features, we exploit the exclusivity in the high-level feature sets with respect to each known class to detect novel classes. Since each class is represented by an exclusive set of high-level features (their *signature*), a novel class is detected when the high-level feature activations differ from the signatures of known (ID) classes. Concretely, we define the signature of class c as the mean high-level features over all seen data for that class.

Our novelty detector (Algorithm 1) first trains a model \mathcal{M} for T epochs to learn SHELS features using the ID training data $D_{1:C}$ (lines 2-5). For an input \mathbf{x} , we compute a class similarity score s_c for each ID class $c \in 1, \dots, C$ via:

$$s_c(\mathbf{x}) = \mathbf{W}_{[:,c]}^L \mathbf{f}^{L-1}(\mathbf{x}), \quad (4)$$

which measures similarity between the high-level features of the sample and the signature of known ID class c . We then compute thresholds \mathbf{th} for each ID class c as the mean of s_c minus one standard deviation, computed over the training data of each class that the model correctly classifies (line 6). At inference time (lines 8-15), we compute the similarity score $s(\mathbf{x})$ with respect to each class c and find the maximum over the classes' scores $s^*(\mathbf{x}) = \max s(\mathbf{x})$ and the corresponding class $c^* = \arg \max s(\mathbf{x})$. Then, \mathbf{x} is predicted as belonging to class c^* if its similarity score $s^*(\mathbf{x})$ is greater than the threshold for the class, \mathbf{th}_{c^*} ; otherwise, it is identified as *novel*. Note that the method of [Techapanurak et al. \(2020\)](#) uses normalized cosine similarities when computing the class scores which can amplify arbitrarily small feature activations, resulting in spurious detections. Our approach of computing class scores using Equation 4 addresses this shortcoming, resulting in better novelty detection performance as shown by our results in Section 5.1.

4.3 NOVELTY ACCOMMODATION

Following the detection of a novel class, our framework updates the SHELS representation to accommodate it, which involves learning new knowledge relevant to the novel class while preserving previously learnt knowledge to prevent catastrophic forgetting ([McCloskey & Cohen, 1989](#)). Since group sparsity has the effect of eliminating certain nodes from the network, these unimportant nodes remain available for incorporating new knowledge. We measure a node's importance as the average activation of the node over the entire training data; [Jung et al. \(2020\)](#) use a similar method to compute node importance. Precisely, we measure the importance of a node n up to class C as:

$$\rho^{(C)}(n) = \frac{1}{|D_{1:C}|} \sum_{k=1}^{|D_{1:C}|} \mathbf{f}_n(\mathbf{x}_k). \quad (5)$$

Our approach focuses the learning of the new class on unimportant nodes, while ensuring that important nodes are not modified. In particular, we target two types of modifications that could affect important nodes: *model drift* and *negative knowledge transfer*. Model drift (illustrated in Appendix F, Figure 12a) occurs when the incoming weights of important nodes are changed during the learning of the new class. To avoid model drift, we penalize updates to these incoming weights. On the other hand, negative knowledge transfer occurs when an unimportant node is updated and used as input to an important node (see Appendix F, Figure 12b). To avoid negative knowledge transfer, we set the weights connecting unimportant nodes (i.e., $\rho^{(C)}(n) = 0$) to important nodes to zero and restrict their updates, thereby stabilizing important nodes. The combined approach for avoiding forgetting is illustrated in Figure 3.

To selectively update weights, we add a weight penalty regularizer Ω_{WP} to Equation 3 to penalize the incoming weights $\mathbf{W}_{[:,i]}$ of node n_i from being updated, weighted by the node's importance $\rho^{(C)}(n_i)$. This gives us the loss:

$$\mathcal{L}(\mathbf{W}^{(C+1)}) = \mathcal{L}_{CE} + \alpha \sum_l \sum_g \left((1 - \mu) \left\| \mathbf{W}_{[g,:]}^{(C+1),l} \right\|_2 \right) + \beta \sum_l \sum_i \underbrace{\rho^{(C)}(n_i) \left\| \mathbf{W}_{[:,i]}^{(C+1),l} - \mathbf{W}_{[:,i]}^{(C),l} \right\|_2}_{\Omega_{WP}}. \quad (6)$$

Algorithm 1 Novelty Detection Algorithm

```

1: INITTRAINING( $D_{1:C}, \mu, \alpha, T$ )
2:   Randomly initialize model  $\mathcal{M}$  parameters,  $\mathbf{W}$ 
3:   Define loss  $\mathcal{L}(\mathbf{W})$  as in Equation 3
4:   for  $t = 1, \dots, T$  do ▷ Adam optimizer
5:      $\mathbf{W}_t \leftarrow \mathbf{W}_{t-1} - \nabla \mathcal{L}(\mathbf{W}_{t-1})$ 
6:    $\mathbf{th} \leftarrow \text{computeTh}(\mathcal{M}, D_{1:C})$ 
7:   Output:  $\mathcal{M}, \mathbf{th}$  ▷ Trained model
8: DETECT( $X, \mathcal{M}$ )
9:    $\mathbf{s} \leftarrow \text{computeScore}(X, \mathcal{M})$ 
10:   $s^* \leftarrow \max(\mathbf{s})$ 
11:   $c^* \leftarrow \arg \max(\mathbf{s})$ 
12:  if  $s^* > \mathbf{th}_{c^*}$  then
13:    Output: class  $c^*$ 
14:  else
15:    Output: novel

```

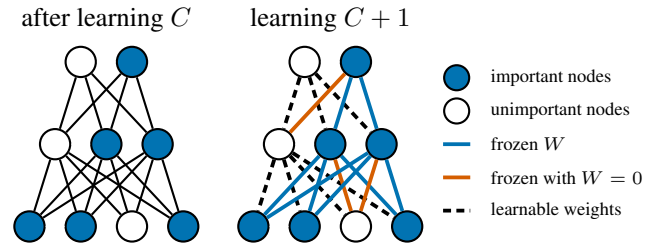


Figure 3: Novelty accommodation via selective weight updates. Important nodes (blue) for classes $1 \dots C$ are identified by $\rho^{(C)}(n) > 0$. Incoming weights to important nodes (blue/orange lines) are frozen (penalized from updates by Ω_{WP}) when learning class $C+1$. To avoid negative transfer, orange weights from unimportant nodes are frozen to 0.

This ensures that the new class is primarily learnt in unimportant nodes, without modifying important nodes. The group sparsity and cosine normalization terms further ensure the ability to detect and accommodate future novel classes.

Our novelty accommodation method (Algorithm 2) first computes the node importance $\rho^{(C)}(n)$ for ID classes of the SHELS model \mathcal{M} trained on dataset $D_{1:C}$ (line 1). Then, `accomm()` uses $\rho^{(C)}(n)$ to accommodate the novel class by updating \mathcal{M} over T epochs, optimizing Equation 6 on data D_{C+1} (lines 3-6).

Algorithm 2 Novelty Accommodation Algorithm

```

1:  $\rho^{(C)}(n) \leftarrow \text{computeNodeImp}(D_{1:C}, \mathcal{M})$ 
2: ACCOMM( $D_{C+1}, \mathcal{M}, \alpha, \beta, \mu, T, \rho^{(C)}(n)$ )
3:   Define loss  $\mathcal{L}(\mathbf{W}^{(C+1)}, \mathbf{W}^{(C)})$  as in Equation 6
4:   for  $t = 1, \dots, T$  do ▷ Adam optimizer
5:      $\mathbf{W}_t^{(C+1)} \leftarrow \mathbf{W}_{t-1}^{(C+1)} - \nabla \mathcal{L}(\mathbf{W}_{t-1}^{(C+1)}, \mathbf{W}^{(C)})$ 
6:   Output:  $\mathcal{M}$  ▷ Updated model
  
```

4.4 NOVELTY DETECTION AND ACCOMMODATION FOR CONTINUAL LEARNING

We combine the novelty detection from Section 4.2 with the novelty accommodation from Section 4.3, resulting in a joint framework that supports class-incremental novelty accommodation without the specification of class boundaries by performing novelty detection. Figure 4 describes the framework and its components.

The continual learning agent first learns the SHELS model \mathcal{M} over the ID data $D_{1:C}$, and computes the thresholds th and the node importance values $\rho(n)$. The model is then deployed to a detection phase, using the approach from Section 4.2 to perform inference on unseen data D_{test} in a dynamic environment, which may contain ID or OOD data. If it detects D_{test} as OOD with a confidence greater than a preset threshold, the agent switches to novelty accommodation. During accommodation, the agent receives additional data of the newly detected class D_{C+1} and accommodates the novel class by updating previously-unimportant nodes using the approach from Section 4.3. Critically, the agent expands the representation learnt for the previous classes $1, \dots, C$ to incorporate the new class $C+1$ while maintaining orthogonality, ensuring that future classes can still be detected as OOD, as illustrated in Figure 1. Once the new class has been accommodated, the agent switches back to deployment mode, and the process repeats.

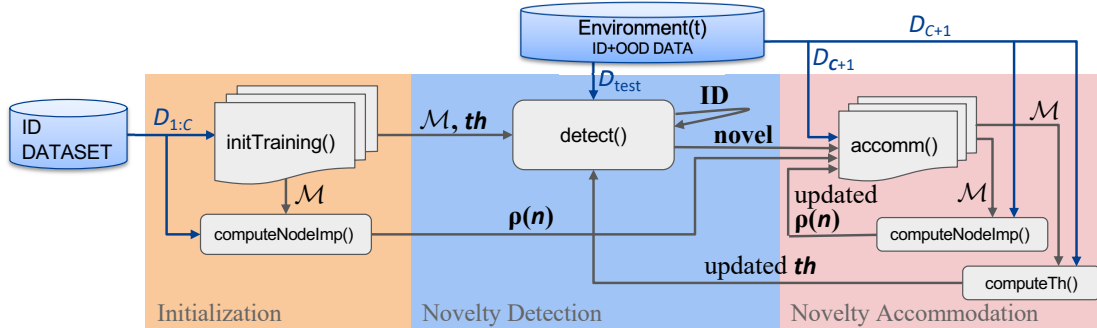


Figure 4: Our framework for novelty detection and accommodation without class boundaries.

5 EXPERIMENTS AND RESULTS

This section evaluates the effectiveness of our framework separately for both novelty detection and novelty accommodation. We then demonstrate how our SHELS representation and approach enable an iterative novelty detection and accommodation process in a class-incremental continual learning scenario without labeled class boundaries.¹

Datasets We evaluate our framework using MNIST (LeCun et al., 1998), FashionMNIST (FMNIST; Xiao et al., 2017), CIFAR10 (Krizhevsky & Hinton, 2009), SVHN (Netzer et al., 2011), and GTSRB (Stallkamp et al., 2012). MNIST, FMNIST, CIFAR10, and SVHN each consist of 10 classes, while GTSRB consists of 43 classes.

Networks and training details All experiments use DNN architectures that consist of ReLU convolutional layers followed by ReLU fully-connected layers and cosine normalization. For MNIST, FMNIST, CIFAR10 (within-dataset), SVHN, and GTSRB, the architecture consists of six convolutional layers followed by one or more linear layers and a final cosine normalization layer. Across-datasets experiments with CIFAR10 as the ID classes use VGG16 pretrained on ImageNet, replacing the final fully-connected layer with a cosine normalization layer. We use the Adam optimizer for all datasets except for CIFAR10 across datasets, for which we use SGD as it achieves higher ID performance.

¹Source code supporting re-creation of all experiments can be found at <https://github.com/Lifelong-ML/SHELS>.

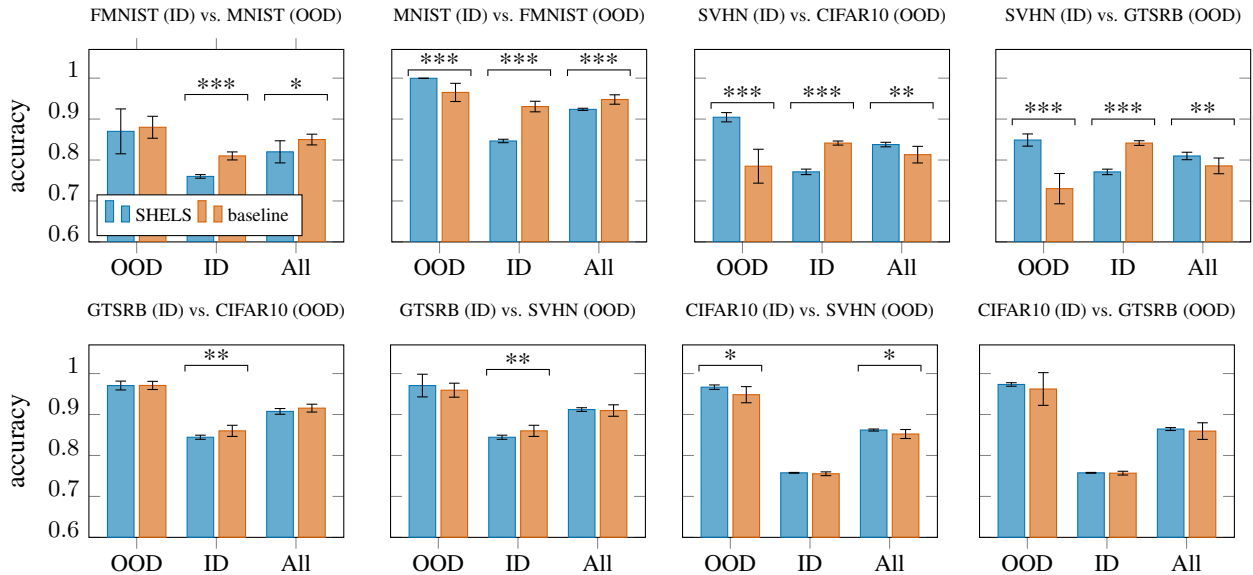


Figure 5: Novelty detection across datasets, showing OOD detection accuracy (OOD), ID classification accuracy (ID), and combined ID and OOD classification accuracy (ALL) with mean \pm standard deviation. Statistically significant differences (after Holm-Bonferroni adjustment) are denoted for $p < .05$ (*), $p < .01$ (**), and $p < .001$ (***)

Implementation details, architectures, and hyperparameters can be found in Appendix G. Additionally, in all experiments, we compute thresholds and tune hyperparameters using only ID data.

5.1 NOVELTY DETECTION

We consider two experimental setups of varying difficulties to evaluate our method. The standard evaluation for OOD detection in prior work considers novelty detection *across datasets*, where ID data differs significantly from OOD data. In this case, one entire dataset is considered ID, and an entirely different dataset is selected as OOD. Additionally, we propose evaluating novelty detection *within datasets*, where ID and OOD data share similar characteristics, requiring novelty detection approaches to learn more fine-grained differences than in the across-datasets experiments. For this case, we randomly sample C classes from a single dataset as ID, and use the remaining classes from the dataset as OOD. For MNIST, FMNIST, CIFAR10, and SVHN, we sample $C = 5$ ID classes and use the remaining 5 as novel. For GTSRB, we sample $C = 23$ and use the remaining 20 classes as OOD data.

In all experiments, we train a classification model over an ID train and validation set. We then evaluate the model on both ID test data and OOD test data. Using this procedure, we compare our OOD detection approach (Section 4.2) to a baseline of the best performing state-of-the-art OOD detector (Techapanurak et al., 2020), which outperforms alternative methods including ODIN and Mahalanobis. We evaluate performance using the following measures:

- OOD detection accuracy—the ability to correctly classify an OOD data point as novel
- ID classification accuracy—the ability to correctly determine the class of an ID data point
- Combined ID and OOD accuracy—the ability to classify a data point as the correct ID class or as OOD
- AUROC—the ability to discriminate between ID and OOD data over the combined ID and OOD datasets.

To aid generalization of results, all experiments are repeated over 10 trials with random seeds, within-dataset experiments use different sets of ID classes for each seed, and we perform statistical significance analyses over the trials.

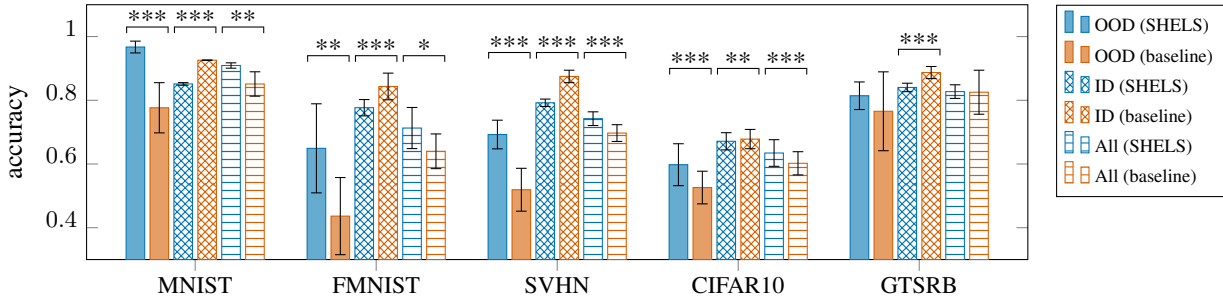
Detection and classification accuracies for across-datasets experiments are shown in Figure 5, with corresponding AUROC measures given in Table 1. The results show a trade-off between the learnt models’ abilities to perform OOD detection and maintain ID classification accuracy. To analyze this trade-off, we performed two-tailed unpaired Student’s t-tests between the baseline and our SHELs approach, using a Holm-Bonferroni adjustment to account for testing multiple measures. All test statistics, p-values, and corrections are reported in Appendix H. The t-tests show that SHELs significantly improves OOD performance with respect to the baseline in half of the cases—MNIST-FMNIST, SVHN-CIFAR10, SVHN-GTSRB, CIFAR10-SVHN—with comparable performance otherwise. This improvement in OOD detection is further supported by the AUROCs (Table 1), which shows that SHELs outperforms the baseline over a majority of experiments. In contrast, t-tests show the baseline outperforms our method in ID accuracy in all cases except when CIFAR10 is used as the ID data, where performance is comparable. Note that the combined metric

Table 1: AUROC of OOD detection across datasets (mean \pm standard deviation over 10 trials).

ID Data	OOD Data	SHELS	Baseline	Sig.
MNIST	FMNIST	0.99 \pm 0.002	0.94 \pm 0.03	***
FMNIST	MNIST	0.88 \pm 0.063	0.77 \pm 0.064	**
GTSRB	CIFAR10	0.96 \pm 0.01	0.90 \pm 0.032	***
GTSRB	SVHN	0.98 \pm 0.005	0.94 \pm 0.018	***
SVHN	CIFAR10	0.90 \pm 0.007	0.90 \pm 0.01	
SVHN	GTSRB	0.88 \pm 0.009	0.89 \pm 0.008	*
CIFAR10	GTSRB	0.93 \pm 0.002	0.92 \pm 0.025	
CIFAR10	SVHN	0.97 \pm 0.003	0.95 \pm 0.05	

Table 2: AUROC of OOD detection within datasets (mean \pm std. dev. over 10 trials).

Dataset	SHELS	Baseline	Sig.
MNIST	0.96 \pm 0.017	0.91 \pm 0.062	*
FMNIST	0.79 \pm 0.063	0.74 \pm 0.084	
GTSRB	0.88 \pm 0.055	0.81 \pm 0.061	*
CIFAR10	0.70 \pm 0.059	0.69 \pm 0.067	
SVHN	0.82 \pm 0.023	0.80 \pm 0.018	

Figure 6: Novelty detection within datasets, showing OOD detection accuracy (OOD), ID classification accuracy (ID), and combined ID and OOD classification accuracy (ALL) with mean \pm standard deviation. Statistically significant differences (after Holm-Bonferroni adjustment) are denoted for $p < .05$ (*), $p < .01$ (**), and $p < .001$ (***).

generally shows comparable performance between the methods, exemplifying the nature of the OOD vs. ID tradeoff. As such, while SHELS significantly improves OOD detection at the cost of some ID accuracy, it has comparable overall performance to the state-of-the-art OOD detection baseline on the relatively easier across-datasets experiments.

Detection and classification accuracies for the more challenging within-dataset experiments are shown in Figure 6, with corresponding AUROC measures given in Table 2. The results show a similar OOD vs. ID accuracy trade-off as in the across-datasets experiments, although the differences between SHELS and the baseline become more apparent due to the more difficult within-dataset case. Paired t-tests showed significant improvements in OOD detection using SHELS for four of the five datasets, with no significant difference found for GTSRB. The AUROC measure further supports SHELS’s improved OOD detection performance, outperforming the baseline by large margins for many of the datasets. In contrast, paired t-tests showed significant improvements in ID classification accuracy using the baseline approach for all datasets, although SHELS exhibits significant improvements in combined accuracy for four of the five datasets, with no significant difference found for GTSRB. We conclude that our approach’s improvements in OOD detection accuracy result in comparable or significant improvements in overall performance compared to the state of the art.

5.2 NOVELTY ACCOMMODATION

We evaluate SHELS’s ability to accommodate novel classes through class-incremental continual learning experiments on the MNIST, FMNIST (see Appendix I), and GTSRB (see Appendix I) datasets and compare it with AGS-CL (Jung et al., 2020), a similar sparsity and weight freezing approach that has shown robust performance, and GDUMB (Prabhu et al., 2020), a replay-based method. We also demonstrate the importance of key components of SHELS that enable incremental novelty accommodation, including encouraging group sparsity and penalizing weight updates for novelty accommodation, via ablation studies. We begin by training a classification model to learn SHELS features using the ID train set. For MNIST, the ID data consists of 7 randomly sampled classes and the remaining 3 classes are OOD. The OOD classes are introduced incrementally in the typical class-incremental continual learning fashion, and our algorithm accommodates them in the SHELS representation as described in Section 4.3.

The class-incremental learning performance of SHELS is shown in the orange performance curves of Figure 7. When a new class is presented to the model, for instance class 8, it experiences a drop in test accuracy depicted by the solid orange curve at epoch 10, as the test set now includes data from the previously unseen class 8. The model then learns class 8, as shown by the dashed novel class test accuracy curve, while preserving performance on the previous seven classes, as evidenced by the upward trend of the overall test accuracy curve. This trend repeats for subsequently introduced novel classes, showing that SHELS learns and accommodates novel classes while mitigating catastrophic forgetting. SHELS

significantly outperforms the state-of-the-art AGS-CL method. AGS-CL’s performance is closer to SHELS without group sparsity (Ablation 1 below) than to the full SHELS implementation, indicating that SHELS induces sparsity for incorporating new knowledge better than AGS-CL. We also compare our approach to GDUMB, which uses K replay samples balanced across the ID classes. For our experiments, we set K to 1% of the total training data. While GDUMB outperforms SHELS, it is important to note that SHELS does not need to store replay data, which may not be possible in memory-constrained or privacy-sensitive applications. As K increases, GDUMB serves as an upper bound on performance, representing the non-continual learning setting where all data is available to the training algorithm at all times. As the fundamental mechanism of SHELS is complementary to replay, we show that the ideas of SHELS and GDUMB are compatible in Appendix I, combining them to yield a new SHELS-GDUMB algorithm.

Ablation 1: Group Sparsity To show the importance of encouraging group sparsity for novelty accommodation, we evaluate our method by repeating the same experiment without group sparsity regularization, (*no sparsity* curves in Figure 7). The regular drop in test accuracy when new classes are introduced and the new class performance staying at zero indicates that the network has no capacity available to accommodate new class information.

Ablation 2: Penalizing Weight Updates To show the effectiveness of penalizing weight updates for novelty accommodation, we evaluate our method without any penalties on weight updates (*no wt. penalty* curves in Figure 7). While new classes can be learnt successfully, as shown by the new class performance, there is no mechanism to maintain performance on previous tasks, catastrophic forgetting occurs, and test accuracy drops significantly.

The ablations demonstrate the roles of group sparsity and weight update penalization in our method, enabling SHELS to accommodate novel classes in a class-incremental continual learning setting. We evaluate our continual learning approach over additional datasets and longer class sequences in Appendix I, showing qualitatively similar results.

5.3 NOVELTY DETECTION AND ACCOMMODATION

In this section, we demonstrate the performance of SHELS to both detect and accommodate OOD data using the pipeline presented in Section 4.4. To this end, we conducted a class-incremental continual learning experiment where *class boundaries are not given to the learning agent*. The experiment is designed such that the agent must alternate between novelty detection and accommodation phases, and the decision to switch from detection to accommodation is made by the agent. At each epoch in the detection phase, the agent encounters a random sample of either ID or OOD data. In our experiments, OOD data is always provided at the fourth detection epoch for consistent visualization—note that this consistency has no effect on the agent’s ability to detect OOD data. If the agent detects a sample as OOD with a confidence score greater than a preset threshold, it triggers an accommodation phase where it is provided with more of the data from the last detection epoch. The model accommodates this data as a novel class, and once the new class is learnt, it switches back to detection, and the detection/accommodation cycle continues. To the best of our knowledge, this is a novel evaluation paradigm for continual learning without labeled class boundaries. With no methods for this setting in the literature, we provide results from our approach as a case study and compare against a combination of the OOD detection and continual learning components of Lee et al. (2018), which we denote MAHA.

We use seven randomly sampled classes from the MNIST dataset as ID and the remaining as OOD, and four randomly sampled classes from FMNIST dataset as ID and three out of the remaining six classes as OOD. A performance comparison of our approach with MAHA is shown in Figure 8. For MNIST, we begin by learning a classification model consisting of SHELS features using the ID train set. The model is then deployed and it enters the detection phase. For the first three epochs, it encounters a random sample of test data from the previously seen seven classes, which it correctly identifies as ID as shown by the low OOD confidence scores. At the fourth epoch, the model encounters a sample of OOD class data, correctly identifies it, and switches to the accommodation phase. The newly detected class is accommodated (as shown by the novel class test accuracy curve) while maintaining performance on the previous classes (as shown by the overall test accuracy curve). The model switches back to the detection phase, and the process repeats until the final detection phase, where all ten classes have been accommodated and all data is correctly detected as ID. We repeat the same procedure for FMNIST. For both datasets, SHELS correctly identifies and accommodates all novel classes. In comparison, while MAHA is able to detect and accommodate all novel MNIST classes, it fails to detect two of the novel FMNIST classes, resulting in significantly lower performance when compared to SHELS.

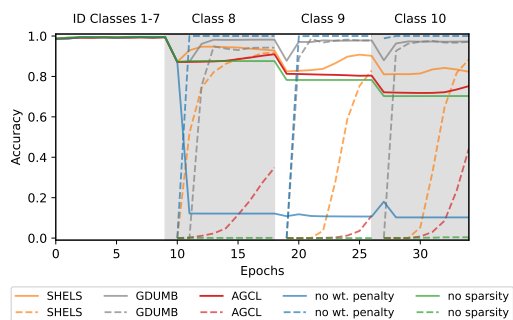


Figure 7: Class incremental learning on MNIST. Solid lines show overall test accuracy, dashes show test accuracy of the novel class only.

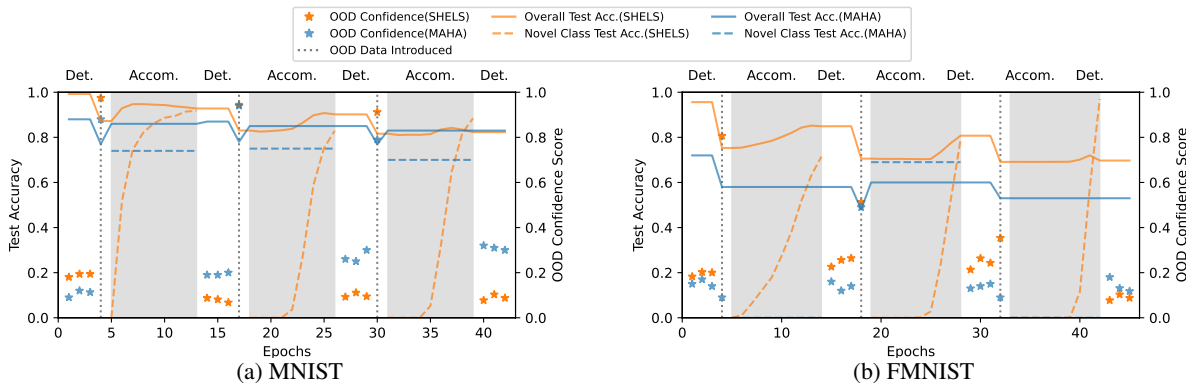


Figure 8: Class incremental learning without boundaries, alternating between detection and accommodation phases.

We offer the following explanation for why SHELS outperforms MAHA. For novelty detection, MAHA computes a confidence score using the Mahalanobis distance with respect to the nearest ID class-conditional distribution. While this approach identifies novel classes with features that significantly differ from the ID classes, it fails to identify novel classes with some features similar to the ID classes, which is a frequent case within the FMNIST dataset. SHELS overcomes this drawback by computing the similarity between the high-level activation patterns of the novel class and the exclusive feature sets for each ID class. Since each ID class is represented by a unique set of high-level features, then even if a novel class shares some features with the ID classes, comparing the similarity between these high-level exclusive feature sets still results in effective novelty detection. As for novelty accommodation, MAHA reuses the previously learnt representation, and thus lacks the ability to learn new knowledge. For both the MNIST and FMNIST datasets, MAHA performs poorly when compared to SHELS, since new knowledge needs to be learnt when accommodating new classes. In contrast, due to the sparsity, SHELS is able to learn new knowledge and accommodate new classes more effectively.

The above experiments are analogous to an agent deployed in the open world, where it encounters objects from known classes in its environment. Eventually, the agent will encounter an object of an unknown class, detect that the object is novel, gather additional data to accommodate the new object in its model, and continue to operate in its environment with the updated set of ID classes. Our results validate that the SHELS representation enables integrated novelty detection and accommodation within a single model, and can operate effectively in such an open-world setting.

6 CONCLUSION

In this work, we show the potential of combining novelty detection with continual learning to develop open-world learners. We demonstrated a system capable of class-incremental continual learning without class boundaries, which automatically detects the presence of new classes via a novel OOD detection method that leverages our human-inspired SHELS representation and significantly outperforms the state of the art. In particular, SHELS leverages the *exclusivity* of high-level feature sets to detect OOD data, and the *sharing* of low-level features to free up nodes for accommodating future classes. Our experimental results illustrate the trade-off between ID classification and OOD detection accuracies. To mitigate this trade-off, one promising avenue of future work is to develop alternative methods for encouraging exclusive feature sets. Our method can also be extended by integrating complementary continual learning approaches to increase learning robustness. In particular, dynamic network expansion would endow agents with the capability to automatically grow the capacity of the SHELS network if it runs out of unused nodes to accommodate large numbers of new classes, enabling potentially unbounded learning. With a suitably robust system, we plan to evaluate our approach in physical open-world environments through situated robot deployments as future work.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their helpful feedback on the earlier draft. The research presented in this paper was partially supported by the DARPA Lifelong Learning Machines program under grant FA8750-18-2-0117, the DARPA SAIL-ON program under contract HR001120C0040, the DARPA ShELL program under agreement HR00112190133, and the Army Research Office under MURI grant W911NF20-1-0080.

REFERENCES

- Jerone Andrews, Thomas Tanay, Edward J. Morton, and Lewis D. Griffin. Transfer representation-learning for anomaly detection. In *ICML 2016 Anomaly Detection Workshop*, 2016.
- Antreas Antoniou, Harri Edwards, and Amos Storkey. How to train your MAML. In *7th International Conference on Learning Representations (ICLR-19)*, 2019.
- Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR-15)*, pp. 1893–1902, 2015.
- Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR-16)*, pp. 1563–1572, 2016.
- Arslan Chaudhry, Marc’ Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations (ICLR-19)*, 2019.
- Arslan Chaudhry, Naeemullah Khan, Puneet Dokania, and Philip Torr. Continual learning in low-rank orthogonal subspaces. In *Advances in Neural Information Processing Systems 33 (NeurIPS-20)*, pp. 9900–9911, 2020.
- Hung-Jen Chen, An-Chieh Cheng, Da-Cheng Juan, Wei Wei, and Min Sun. Mitigating forgetting in online continual learning via instance-aware parameterization. In *Advances in Neural Information Processing Systems 33 (NeurIPS-20)*, pp. 17466–17477, 2020.
- John Colombo and Robert S. Bundy. Infant response to auditory familiarity and novelty. *Infant Behavior and Development*, 6(2-3):305–311, 1983.
- Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening sharpness for dynamic gradient projection memory benefits in continual learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS-21)*, 2021.
- Lea Duncker, Laura Driscoll, Krishna V. Shenoy, Maneesh Sahani, and David Sussillo. Organizing recurrent network dynamics by task-computation to enable continual learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS-20)*, pp. 14387–14397, 2020.
- Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. PathNet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML-17)*, pp. 1126–1135, 2017.
- Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations (ICLR-17)*, 2017.
- Michael A. Hunter, Elinor W. Ames, and Raymond Koopman. Effects of stimulus complexity and familiarization time on infant preferences for novel and familiar stimuli. *Developmental Psychology*, 19(3):338, 1983.
- Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. Continual learning with node-importance based adaptive group sparse regularization. In *Advances in Neural Information Processing Systems 33 (NeurIPS-20)*, pp. 3647–3658, 2020.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 114(13):3521–3526, 2017.
- Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30 (NeurIPS-17)*, 2017.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

- Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31 (NeurIPS-18)*, 2018.
- Seungwon Lee, James Stokes, and Eric Eaton. Learning shared knowledge for deep lifelong learning using deconvolutional networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, pp. 2837–2844, 2019.
- Seungwon Lee, Sima Behpour, and Eric Eaton. Sharing less is more: Lifelong learning in deep networks with selective layer transfer. In *Proceedings of the 38th International Conference on Machine Learning (ICML-21)*, pp. 6065–6075, 2021.
- Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *Proceedings of the 36th International Conference on Machine Learning (ICML-19)*, pp. 3925–3934, 2019.
- Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations (ICLR-18)*, 2018.
- Bing Liu. Learning on the job: Online lifelong and continual learning. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-20)*, pp. 13544–13549, 2020.
- David Lopez-Paz and Marc’ Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30 (NeurIPS-17)*, pp. 6467–6476, 2017.
- Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN-18)*, pp. 382–391, 2018.
- 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.
- Jorge A. Mendez and Eric Eaton. Lifelong learning of compositional structures. In *9th International Conference on Learning Representations (ICLR-21)*, 2021.
- Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.
- Oleksiy Ostapenko, Pau Rodriguez, Massimo Caccia, and Laurent Charlin. Continual learning via local module composition. In *Advances in Neural Information Processing Systems 34 (NeurIPS-21)*, 2021.
- Jean Piaget. *The Language and Thought of the Child*. Harcourt, Brace, 1926.
- Ameya Prabhu, Philip H. S. Torr, and Puneet K. Dokania. GDumb: A simple approach that questions our progress in continual learning. In *Proceedings of the European Conference on Computer Vision (ECCV-20)*, pp. 524–540, 2020.
- Qi Qin, Wenpeng Hu, Han Peng, Dongyan Zhao, and Bing Liu. BNS: Building network structures dynamically for continual learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS-21)*, 2021.
- Jathushan Rajasegaran, Munawar Hayat, Salman Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for incremental learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS-19)*, pp. 12669–12679, 2019.
- Scott Reed and Nando de Freitas. Neural programmers-interpreters. In *4th International Conference on Learning Representations (ICLR-16)*, 2016.
- Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *9th International Conference on Learning Representations (ICLR-21)*, 2021.
- Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems 30 (NeurIPS-17)*, pp. 2990–2999, 2017.
- Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30 (NeurIPS-17)*, 2017.

- Johannes Stalldkamp, Marc Schlipfing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012.
- Engkarat Techapanurak, Masanori Sukanuma, and Takayuki Okatani. Hyperparameter-free out-of-distribution detection using cosine similarity. In *Proceedings of the Asian Conference on Computer Vision (ACCV-20)*, 2020.
- Amos Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.
- Lazar Valkov, Dipak Chaudhari, Akash Srivastava, Charles Sutton, and Swarat Chaudhuri. Houdini: Lifelong learning as program synthesis. In *Advances in Neural Information Processing Systems 31 (NeurIPS-18)*, pp. 8687–8698, 2018.
- Gido M. van de Ven, Hava T. Siegelmann, and Andreas S. Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020.
- Tom Veniat, Ludovic Denoyer, and Marc’Aurelio Ranzato. Efficient continual learning with modular networks and task-driven priors. In *9th International Conference on Learning Representations (ICLR-21)*, 2021.
- Frederic Weizmann, Leslie B. Cohen, and R. Jeanene Pratt. Novelty, familiarity, and the development of infant attention. *Developmental Psychology*, 4(2):149, 1971.
- Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems 29 (NeurIPS-16)*, 2016.
- Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. In *5th International Conference on Learning Representations (ICLR-17)*, 2017.
- Jaehong Yoon and Sung Ju Hwang. Combined group and exclusive sparsity for deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML-17)*, pp. 3958–3966, 2017.
- Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.
- Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning (ICML-17)*, pp. 3987–3995, 2017.
- Yang Zhou, Rong Jin, and Steven Chu–Hong Hoi. Exclusive Lasso for multi-task feature selection. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-10)*, pp. 988–995, 2010.

A ORTHOGONAL FEATURES ARE EXCLUSIVE SETS

This section formalizes the notation that orthogonal features are exclusive sets. Let each data instance $\mathbf{x} \in \mathbb{R}^d$ be embedded as set of derived features $\mathbf{f} = g(\mathbf{x}) \in \mathbb{R}^k$. Recall that our goal is for each class to have an exclusive embedding of the non-zero entries of \mathbf{f} .

Proposition 1. *Let $\mathbf{f}_1, \dots, \mathbf{f}_C \in \mathbb{R}^k$ be the orthogonal embeddings for each of C classes, such that $\forall i \neq j \mathbf{f}_i \perp \mathbf{f}_j$, with $k \geq C$. Then, there exists an orthogonal transformation T such that the transformed embeddings $T(\mathbf{f}_1), \dots, T(\mathbf{f}_C)$ form a collection of exclusive sets.*

Proof. Since $\mathbf{f}_1, \dots, \mathbf{f}_C \in \mathbb{R}^k$ are orthogonal, they are linearly independent and form a basis for a subspace $\mathbb{R}^C \subseteq \mathbb{R}^k$. Therefore, we can define a change-of-basis matrix T to transform from the basis defined by $\mathbf{f}_1, \dots, \mathbf{f}_C$ onto the standard basis \mathcal{B} for \mathbb{R}^C . By definition, T must be an orthogonal transformation. Each vector of the standard basis has a single index, unique with respect to all other vectors of the standard basis, that contains a non-zero element. We can interpret each vector $v \in \mathcal{B} \in \{0, 1\}^C$ as encoding an equivalent corresponding set $\hat{\mathcal{V}}$ formed by $\bigcup_{i=1}^C \{i \times v_i\} - \{0\}$ (the set of all non-zero indices of that vector). Under this construction, each pair of vectors $u \neq v \in \mathcal{B}$ has equivalent exclusive sets $\hat{\mathcal{U}}$ and $\hat{\mathcal{V}}$, since the indices of their non-zero elements must differ, and therefore the collection must also be exclusive. \square

B CONNECTIONS TO OTHER RELATED AREAS

Since our approach spans multiple subjects that are usually studied separately (novelty detection and continual learning), we draw connections to a wide variety of prior work. Some of the connections worth considering, but outside the scope of our [Related work](#) section, are discussed below.

Few-shot Learning. Similar ideas to the high-level exclusive and low-level shared feature representation of SHELS have been previously explored in the few-shot transfer learning literature. Networks trained with few-shot learning approaches based on MAML (Finn et al., 2017; Antoniou et al., 2019) result mainly in changes at the later layers in a network (i.e., changes in higher level features), creating a mostly immutable shared set of low-level features. Protonets use a learnt feature embedding over which to make classification decisions, which can be interpreted as a shared low-level feature representation (Snell et al., 2017). Few-shot learning approaches have not, however, addressed the notion of high-level *set-based exclusivity* discussed in our work. Such a concept is less relevant to the few-shot learning paradigm, which is more concerned with learning models that can be quickly adjusted to new tasks when new data is presented. In contrast, OOD detection approaches must differentiate novel OOD data *before* any model updates, and so this notion of maintaining high-level exclusivity (among both previously seen and OOD classes) becomes very important in the OOD detection paradigm. Few-shot learning approaches such as MAML address a transfer learning problem where they do not need to maintain performance on previous tasks, and are thus more suited to fast adaptation rather than novelty detection. Further, the prior work in few-shot learning doesn't explicitly discuss or evaluate the low-level shared or high-level exclusive feature representations that our approach explicitly encourages in the loss function, or that we explicitly evaluate with an exclusivity metric.

Modular Architectures. One tangentially related line of work that has studied sparsity in continual learning is the discovery of modular and compositional architectures. In this case, sparsity is manually embedded into the design of the network via modularity, instead of discovered by the agent as in our approach. The majority of such methods freeze the weights of the modules immediately after training them on a task (Reed & de Freitas, 2016; Fernando et al., 2017; Valkov et al., 2018; Li et al., 2019; Veniat et al., 2021), unlike our approach that automatically detects which weights to freeze based on their importance to previous tasks. Other methods instead continually update the weights of the modules without freezing them (Rajasegaran et al., 2019; Chen et al., 2020; Mendez & Eaton, 2021; Qin et al., 2021; Ostapenko et al., 2021).

C COSINE NORMALIZATION INDUCES EXCLUSIVE FEATURE SETS

To empirically verify the ability of cosine normalization to lead to exclusive feature sets, we trained a DNN on five randomly sampled classes from the FMNIST dataset and measured the exclusivity of the high-level feature activations of the classes. Given two sets of feature activations, we measured their exclusivity as the ratio of the total number of dissimilar activations to the total number of activation at the highest-level representation:

$$exclusivity(\mathbf{x}_1, \mathbf{x}_2) = \frac{\hat{\mathbf{f}}^{L-1}(\mathbf{x}_1) \oplus \hat{\mathbf{f}}^{L-1}(\mathbf{x}_2)}{\hat{\mathbf{f}}^{L-1}(\mathbf{x}_1) \vee \hat{\mathbf{f}}^{L-1}(\mathbf{x}_2)}, \quad (7)$$

Table 3: Exclusivity metric across 5 randomly sampled classes.

(a) Standard linear layer method						(b) Scaled cosine similarity method (Baseline)						(c) Cosine normalization with raw cosine similarity method (ours)					
classes	pullover	coat	bag	dress	boot	classes	pullover	coat	bag	dress	boot	classes	pullover	coat	bag	dress	boot
pullover	0	0.73	0.68	0.58	0.74	pullover	0	0.74	0.98	0.95	1.0	pullover	0	1.0	1.0	1.0	1.0
coat	0.72	0	0.75	0.76	0.73	coat	0.74	0	0.98	0.76	1.0	coat	1.0	0	1.0	1.0	1.0
bag	0.68	0.75	0	0.69	0.68	bag	0.98	0.98	0	1.0	1.0	bag	1.0	1.0	0	1.0	1.0
dress	0.57	0.76	0.69	0	0.71	dress	0.95	0.76	1.0	0	1.0	dress	1.0	1.0	1.0	0	1.0
boot	0.74	0.73	0.68	0.71	0	boot	1.0	0.97	1.0	1.0	0	boot	1.0	1.0	1.0	1.0	0

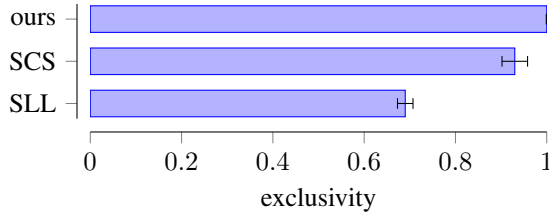


Figure 9: Exclusivity averaged across 10 experiments of 5 randomly sampled classes from the FMNIST dataset comparing standard linear layer method, scaled cosine similarity method and our method. SLL represents the standard linear layer method and SCS is the baseline scaled cosine similarity method. Error bars denote standard errors.

where $\hat{\mathbf{f}}^{L-1}(\mathbf{x})$ is the unit vector of the final high-level features. A score of 1.0 indicates perfect exclusivity. We used three methods for training the DNN: a standard linear layer, the method of [Techapanurak et al. \(2020\)](#), and the cosine similarity formulation of Equation 1. Table 3 shows the resulting exclusivity for a sample of five classes, while Figure 9 shows aggregated results over ten trials, clearly demonstrating that our proposed method is the only one that finds a collection of exclusive feature sets. [Techapanurak et al. \(2020\)](#) train their model using the softmax of the scaled cosine similarity scores in the loss function, while our approach uses the raw cosine similarity scores. This choice to use the raw scores, although seemingly minor, makes a fundamental difference in learning exclusive feature sets.

The architecture and hyperparameters used for the experiment are the same as the ones used for the novelty detection experiments with FMNIST (see Appendix G) and are kept consistent across the three different methods. We also ensure that the randomly sampled classes for the 10 trials are the same for all three methods to ensure a fair comparison.

D USING ZHOU ET AL.’S EXCLUSIVE SPARSITY AS AN ALTERNATIVE

Here, we compare the degree of exclusivity in the feature sets when the model is trained using the exclusive sparsity regularizer of [Zhou et al. \(2010\)](#) in comparison with our cosine normalization method. Notably, Zhou et al.’s regularizer has a very different interpretation of exclusivity to what we use in this work: it encourages *individual* features to be exclusive to a class (limiting transfer), where as our approach encourages *sets* of features to be exclusive (unique) to a class. We use the exclusive sparsity regularizer as defined by [Zhou et al. \(2010\)](#):

$$\Omega_E(\mathbf{W}^l) = \frac{1}{2} \sum_g \|\mathbf{w}_g^l\|_1^2 = \frac{1}{2} \sum_g \left(\sum_i \|\mathbf{w}_{[g,i]}^l\| \right)^2. \quad (8)$$

To learn the model we combine the exclusive sparsity regularizer with the standard cross entropy loss:

$$\mathcal{L}(\mathbf{W}) = \mathcal{L}_{CE} + \gamma \sum_l \left(\mu^{(l)} \Omega_E(\mathbf{W}^l) \right). \quad (9)$$

The parameter γ controls the effect of regularization and is set to 0.001. We set $\mu^{(l)} = \frac{l-1}{L-1}$ to encourage less exclusivity at the lower layers and more exclusivity at the higher layers, with $\mu^{(1)} = 0$ to $\mu^{(L)} = 1$ at the outermost layers. We trained the DNN on five randomly sampled classes from the FMNIST dataset and measured the exclusivity of the high-level feature activations using Equation 7, similar to the experiment in Appendix C. The architecture and hyperparameters are same as for the novelty detection experiments with FMNIST (see Appendix G). Figure 10 shows aggregated results over ten random seeds, comparing the Zhou et al.’s exclusive sparsity method to our method, clearly demonstrating that our method results in more exclusive high-level feature sets.

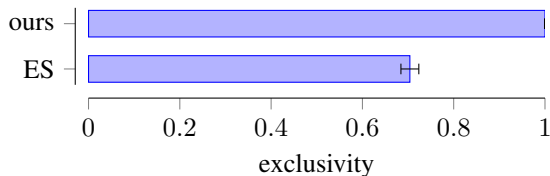


Figure 10: Exclusivity averaged across 10 experiments of 5 randomly sampled classes from the FMNIST dataset comparing Zhou et al.’s Exclusive Sparsity (ES) method and cosine normalization (ours) method for generating exclusive feature sets. Error bars denote standard errors.

E EXPERIMENTAL ANALYSIS OF THE EFFECTS OF GROUP SPARSITY

We ran additional novelty detection experiments to analyze the effects of group sparsity on OOD detection by varying both the group sparsity regularization weight α and the layer-specific mixing weight $\mu^{(l)}$ in Equation 3. These experiments followed the same procedure and used the same hyperparameters as the within-datasets experiments of Section 5.1, with our approach divided into two conditions: (1) SHELs with decreasing group sparsity as the layers increase by setting $\mu^{(l)} = \frac{l-1}{L-1}$ (layer-wise), and (2) SHELs with a constant amount of group sparsity at each layer by setting $\mu^{(l)} = 0$ (constant). Additionally, we varied α on the interval $[0, 0.02]$ in increments of 0.001 to examine the effects of increasing the contribution of the group sparsity term in the loss function. The results are shown in Figure 11.

The results show a consistent trend that increasing the weight of group sparsity regularization has a detrimental effect on both ID classification and OOD detection accuracy. This is an important consideration for combining OOD detection and continual learning, as our continual learning ablation studies presented in Section 5.2 show that group sparsity regularization is essential to SHELs’s effective continual learning performance. As such, finding a way to

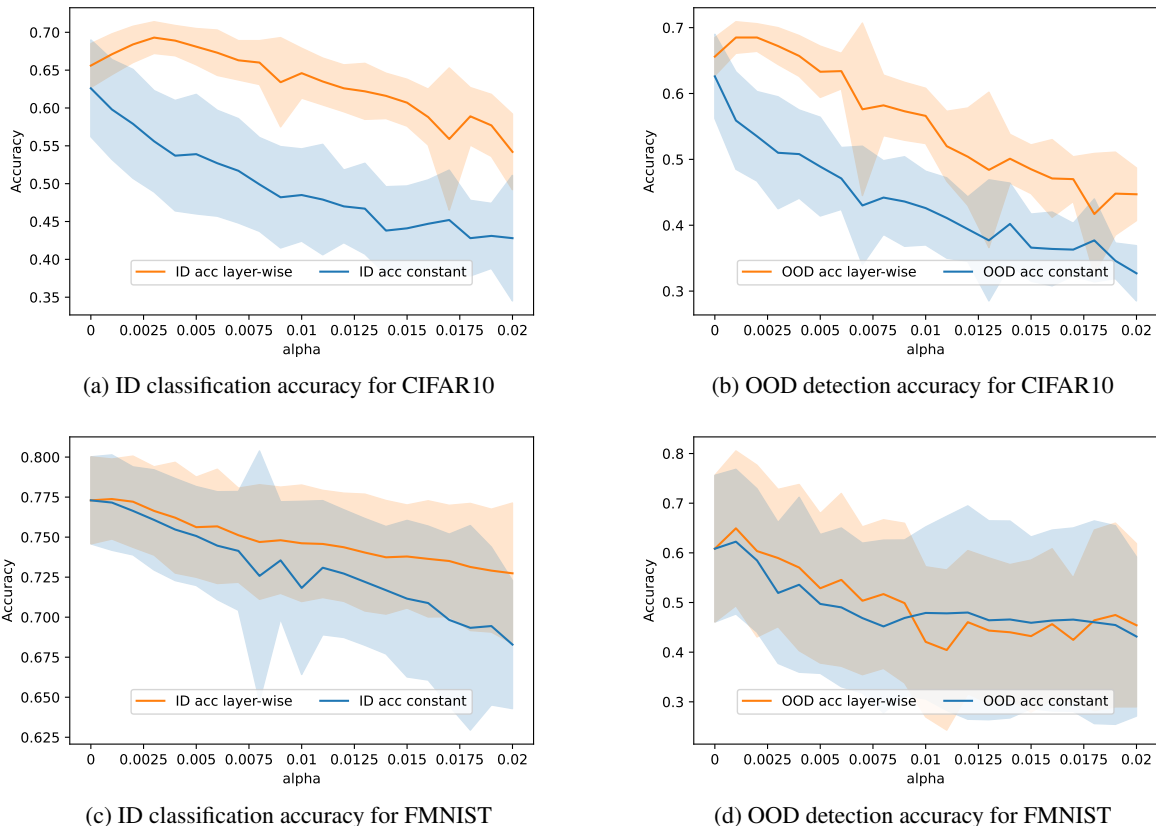


Figure 11: Analysis of the effects of group sparsity, showing mean \pm standard deviation over ten trials.

incorporate group sparsity regularization into novelty detection models without negatively impacting their performance is a key contribution of our novelty detection approach. Based on the above experiments, we use the decreasing layer-wise implementation for our method, which is responsible for the low-level shared feature learning of the SHELS representation, and a value of $\alpha = 0.001$ for all additional experiments.

F SUPPLEMENTAL ILLUSTRATIONS OF OUR APPROACH

Figure 12 illustrates forgetting based on model drift and negative transfer, and its interactions with node importance.

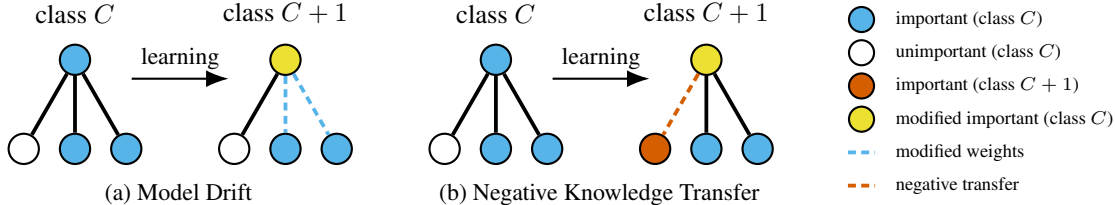


Figure 12: Catastrophic forgetting caused by (a) Model Drift and (b) Negative Knowledge Transfer.

G IMPLEMENTATION DETAILS

G.1 ARCHITECTURE DETAILS

The network architecture details for MNIST, FMNIST, SVHN, GTSRB and CIFAR10 (within-dataset) are given in Tables 4 and 5. The architecture consists of six convolutional layers followed by linear and cosine layers. The convolutional layers for each dataset are the same as shown in Table 4, while the linear layers differ across datasets and are detailed for each of the datasets in Table 5. For across dataset experiments with CIFAR10 as ID, we use the PyTorch model for VGG16, pretrained on ImageNet, with the last linear layer replaced with cosine normalization.

Table 4: Convolutional(Conv) layers for MNIST, FMNIST, SVHN and GTSRB network architectures; $n_1 = 1$ for MNIST and FMNIST and $n_1 = 3$ for SVHN, GTSRB, and CIFAR10 (within-dataset).

Layer	Channel(In)	Channel(Out)	Kernel	Stride	Padding
Conv-1	n_1	32	3×3	1	1
Conv-2	32	32	3×3	1	1
Maxpool	32	32	2×2	2	
Conv-3	32	64	3×3	1	1
Conv-4	64	64	3×3	1	1
Maxpool	32	32	2×2	2	
Conv-5	64	128	3×3	1	1
Conv-6	128	128	3×3	1	1
Maxpool	32	32	2×2	2	
Flatten()					

Table 5: Linear and Cosine layers for MNIST, FMNIST, SVHN, GTSRB and CIFAR10 (within-dataset) network architectures; C is the total number of ID classes for the experiment.

(a) MNIST and FMNIST		(b) GTSRB and CIFAR10 (within-dataset)		(c) SVHN	
Layer	Out-layers	Layer	Out-layers	Layer	Out-layers
Linear-1	256	Linear-1	2048	Linear-1	1024
Cosine	C	Linear-2	512	Linear-2	512
		Cosine	C	Cosine	C

G.2 HYPERPARAMETERS

We train models for all the experiments consisting of the datasets MNIST, FMNIST, GTSRB, and SVHN using the Adam optimizer with a learning rate of 0.0001. For across-datasets experiments with the CIFAR10 dataset, we train the models using the SGD optimizer with an initial learning rate of 0.001, decaying it by a factor of 2 after every four epochs, momentum of 0.9, and weight decay of $5e-4$. For within-dataset experiments with CIFAR10 we use the Adam optimizer with a learning rate of 0.0001. For the novelty detection experiments in Section 5.1, for all datasets, we use a batch size of 32 and set α to 0.001. The total number of training epochs varies for the different datasets: for MNIST and GTSRB we use 10 epochs, for FMNIST and SVHN we use 20 epochs, and for CIFAR10 we use 30 epochs. For the novelty accommodation experiment and the combined novelty detection and accommodation experiment in Sections 5.2 and 5.3 with the MNIST dataset, we use the same hyperparameters from the novelty detection experiment for MNIST. Additionally, we set β to 2000, decrease the learning rate by a factor of 10, and use an OOD confidence score threshold of 0.75. For the novelty accommodation experiment and the combined novelty detection and accommodation experiment using FMNIST in Appendix I and Section 5.3, we use the same hyperparameters from the novelty detection experiment for FMNIST in addition to setting β to 3000, decreasing the learning rate by a factor of 2, and using an OOD confidence score threshold of 0.3. For the novelty accommodation experiments using GTSRB in Appendix I, we use the same hyperparameters from the novelty detection experiment for GTSRB in addition to setting β to 50000, and keep the same learning rate.

To ensure a fair comparison with the AGS-CL baseline, for the AGS-CL experiments we set μ to be the same as α from the corresponding SHELS experiment to ensure the same amount of sparsity regularization. We also set λ to be the same as β from the corresponding SHELS experiment to ensure the same amount of weight freezing. However, we did find it necessary to tune the learning rate for AGS-CL: for MNIST we reduced the learning rate by a factor of 10, for FMNIST we reduced the learning rate by a factor of 8, and for GTSRB we reduced the learning rate by a factor of 2. All AGS-CL models were trained using the Adam optimizer with an initial learning rate of 0.0001 and a fixed batch size of 32. For GDUMB experiments we use $K = 1\%$ of the training data, SGD optimizer, fixed batch size of 32, learning rates [0.05,0.0005] and SGDR schedule with $T_0 = 1$, $T_{\text{mult}} = 2$. For SHELS-GDUMB experiments we also set $K = 1\%$ of the training data and use the Adam optimizer with a learning rate of 0.0001 and a fixed batch size of 32. For both GDUMB and SHELS-GDUMB experiments, we use all the ID training data to warm start the model and use the replay buffer when we begin to incrementally learn OOD classes. For the MAHA baseline, we set the threshold scores for each ID class to one standard deviation minus the mean of the scores of all the correctly identified samples from the training data of the corresponding class. Also, we set the same OOD confidence score threshold for MAHA as that used by the corresponding SHELS experiment.

G.3 DATASET DETAILS

We detail the dataset splits for novelty detection experiments from Section 5.1 in Tables 6 and 7. For the within-dataset experiments, the OOD set comprises of a subset of default test set of that dataset corresponding to the OOD classes, while for the across-datasets experiments, the OOD set consists of the entire default test set of the OOD dataset.

The input image sizes for each dataset compatible with the architectures described in Appendix G.1 are given in Table 8. For the across-datasets novelty detection experiments, we resize the OOD data image size to the ID data image size. For example, for the SVHN(ID) vs. CIFAR10(OOD) experiment, we resize the CIFAR10 images to be of the same size as SVHN, i.e., 32×32 to be compatible with the network architecture.

H STATISTICAL TESTING DETAILS

For all across-datasets novelty detection experiments, we performed unpaired Student’s t-tests to determine whether the OOD detection method had a statistically significant effect on OOD detection accuracy, ID classification accuracy, combined accuracy, and OOD detection AUROC. This was evaluated on both our SHELS approach and the state-of-

Table 6: Dataset sizes for novelty detection within-dataset experiments.

	MNIST	FMNIST	SVHN	GTSRB	CIFAR10
train set	~ 26000	26400	~ 31000	~ 19000	25000
val set	~ 3500	3600	~ 4000	~ 2600	3000
test set	~ 4800	5000	~ 12000	~ 6900	5000
OOD set	~ 5000	5000	~ 13000	~ 5500	5000

Table 7: Dataset details for novelty detection across-datasets experiment.

	MNIST	FMNIST	SVHN	GTSRB	CIFAR10
train set	52800	52800	64466	34502	50000
val set	7200	7200	8791	4707	6000
test set	10000	10000	26032	12569	10000

Table 8: Input image size in pixels for each dataset.

Dataset	Input Size
MNIST	32×32
FMNIST	32×32
SVHN	32×32
GTSRB	112×112
CIFAR10 (within-dataset)	112×112
CIFAR10 (across-datasets)	224×224

the-art OOD detection baseline (Techapanurak et al., 2020). We performed two-tailed tests to test for effects in either direction, making no prior assumption that either method would outperform the other. We applied Holm-Bonferroni adjustments to protect against type I error from multiple comparisons, since we are testing four measures in each experiment, and use the adjusted p-values to test for statistical significance at a significance level of $\alpha = 0.05$. We use unpaired samples in these experiments, since each trial for each condition was trained independently with a different random seed. Summary and test statistics for all across-datasets experiments are reported below in Table 9.

For all within-dataset novelty detection experiments, we performed paired Student’s t-tests to determine whether OOD detection method had a statistically significant effect on OOD detection accuracy, ID classification accuracy, combined accuracy, and OOD detection AUROC, evaluated between our SHELS approach and the state-of-the-art OOD detection baseline. We again performed two-tailed tests to test for effects in either direction, making no prior assumption that either method would outperform the other. We applied Holm-Bonferroni adjustments to protect against type I error from multiple comparisons since we are again testing four measures per experiment, and use the adjusted p-values to test for statistical significance at a significance level of $\alpha = 0.05$. Unlike in the across-datasets experiments, we performed this statistical analysis using paired samples, by pairing samples with the same permutations of ID vs. OOD classes. For example, trial 1 for the MNIST dataset produced paired samples across both approaches which each consisted of ID classes $\{1, 0, 8, 6, 3\}$, trial 2 consisted of ID classes $\{5, 7, 8, 3, 9\}$ for both approaches, etc. Summary and test statistics for all within dataset experiments are reported below in Table 10.

I ADDITIONAL CONTINUAL LEARNING EXPERIMENTS

We ran additional novelty accommodation experiments on the FMNIST and GTSRB datasets. We use four randomly sampled classes from FMNIST as ID and learn three out of the remaining six OOD classes. We learn a subset of the OOD classes for these experiments due to network capacity limitations of the architectures described in Section G.1. In order to accommodate the full set of FMNIST classes, one could dynamically add nodes to the SHELS model, which we discuss briefly in Section 6 as future work. Figure 13 details the class-incremental learning performance with known class boundaries, following the approach described in Section 4.3. We include the same baselines and ablation studies. Similar to the trends seen with MNIST, SHELS outperforms AGS-CL. The AGS-CL baseline performs similar to SHELS without group sparsity, further reinforcing that SHELS is more effective at inducing sparsity enabling the model to learn new features for future classes. As expected, GDUMB still serves as an upper performance bound for SHELS due to its use of replay data. The ablations show significant catastrophic forgetting without weight penalization (blue curves), and an inability to accommodate novel data due to the lack of network capacity without group sparsity (green curves).

For GSTRB we use 23 randomly sampled classes as ID and the remaining 20 classes as OOD. We learn nine out of the 20 classes incrementally due to network capacity limitations as described before for the FMNIST experiments. We have also found that our model performs best when trained on relatively diverse in-distribution data. As such, our continual learning experiments train on an ID dataset containing multiple classes. In cases where this is not feasible, the model can be pre-trained with other available datasets and deployed for continual learning.

We demonstrate the performance of SHELS in GTSRB following the approach described in Section 4.3 and also include the baselines AGS-CL and GDUMB for comparison as shown in Figure 14. We observe similar trends as seen with MNIST and FMNIST: SHELS outperforms AGS-CL while GDUMB serves as an upper performance bound.

To create a more fair comparison between SHELS, which does not retain previous class data, and replay-based methods, such as GDUMB, we combined the two approaches into a new SHELS-GDUMB algorithm. This new method incorporates a size K replay buffer, as described in the original GDUMB paper (Prabhu et al., 2020), into SHELS. Figure 14 shows that SHELS-GDUMB outperforms SHELS, since it has access to previous ID data while learning

Table 9: Summary statistics (mean \pm std. dev.) and Student’s t-test results for across-datasets novelty detection.

Dataset	Measure	Baseline	SHELS	p-value	Adjustment
FMNIST vs. MNIST	OOD detection acc	88.94 \pm 2.68	87.45 \pm 5.48	$t(18) = 0.77, p = .45$	$\tilde{p} = .45$
	ID classification acc	81.04 \pm 0.98	76.75 \pm 0.44	$t(18) = 12.62, p < .0001$	$\tilde{p} < .001$
	Combined acc	84.99 \pm 1.31	82.12 \pm 2.7	$t(18) = 3.02, p = .007$	$\tilde{p} = .021$
	AUROC	0.88 \pm 0.063	0.77 \pm 0.064	$t(18) = 3.87, p = .0011$	$\tilde{p} = .0022$
MNIST vs. FMNIST	OOD detection acc	96.51 \pm 2.22	99.98 \pm 0.03	$t(18) = 4.94, p = .0001$	$\tilde{p} < .001$
	ID classification acc	93.06 \pm 1.29	84.65 \pm 0.42	$t(18) = 19.6, p < .0001$	$\tilde{p} < .001$
	Combined acc	94.78 \pm 1.15	92.37 \pm 0.27	$t(18) = 6.45, p < .0001$	$\tilde{p} < .001$
	AUROC	0.99 \pm 0.002	0.94 \pm 0.03	$t(18) = 5.26, p < .0001$	$\tilde{p} < .001$
SVHN vs. CIFAR10	OOD detection acc	78.49 \pm 4.14	90.45 \pm 1.13	$t(18) = 8.81, p < .0001$	$\tilde{p} < .001$
	ID classification acc	84.14 \pm 0.57	77.11 \pm 0.67	$t(18) = 25.27, p < .0001$	$\tilde{p} < .001$
	Combined acc	81.31 \pm 2.03	83.78 \pm 0.55	$t(18) = 3.71, p = .002$	$\tilde{p} = .004$
	AUROC	0.90 \pm 0.007	0.90 \pm 0.01	$t(18) = 0, p = 1$	$\tilde{p} = 1$
SVNH vs. GTSRB	OOD detection acc	73.01 \pm 3.70	84.88 \pm 1.49	$t(18) = 9.41, p < .0001$	$\tilde{p} < .001$
	ID classification acc	84.14 \pm 0.57	77.10 \pm 0.67	$t(18) = 25.31, p < .0001$	$\tilde{p} < .001$
	Combined acc	78.57 \pm 1.92	80.99 \pm 0.91	$t(18) = 3.6, p = .002$	$\tilde{p} = .004$
	AUROC	0.88 \pm 0.009	0.89 \pm 0.008	$t(18) = 2.63, p = .017$	$\tilde{p} = .017$
GTSRB vs. CIFAR10	OOD detection acc	97.10 \pm 1.00	97.08 \pm 1.08	$t(18) = 0.04, p = .966$	$\tilde{p} = 1$
	ID classification acc	86.01 \pm 1.36	84.45 \pm 0.50	$t(18) = 3.4, p = .003$	$\tilde{p} = .009$
	Combined acc	91.56 \pm 0.96	90.76 \pm 0.70	$t(18) = 2.13, p = .047$	$\tilde{p} = .094$
	AUROC	0.96 \pm 0.01	0.90 \pm 0.032	$t(18) = 5.66, p < .0001$	$\tilde{p} < .001$
GTSRB vs. SVHN	OOD detection acc	95.94 \pm 1.71	97.07 \pm 2.77	$t(18) = 1.1, p = .287$	$\tilde{p} = .574$
	ID classification acc	86.01 \pm 1.36	84.45 \pm 0.50	$t(18) = 3.4, p = .003$	$\tilde{p} = .009$
	Combined acc	90.97 \pm 1.40	91.21 \pm 0.45	$t(18) = 0.52, p = .612$	$\tilde{p} = 1$
	AUROC	0.98 \pm 0.005	0.94 \pm 0.018	$t(18) = 6.77, p < .0001$	$\tilde{p} < .001$
CIFAR10 vs. SVHN	OOD detection acc	94.84 \pm 1.97	96.68 \pm 0.54	$t(18) = 2.85, p = .011$	$\tilde{p} = .044$
	ID classification acc	75.53 \pm 0.46	75.76 \pm 0.10	$t(18) = 1.55, p = .14$	$\tilde{p} = .28$
	Combined acc	85.24 \pm 1.10	86.21 \pm 0.25	$t(18) = 2.72, p = .014$	$\tilde{p} = .042$
	AUROC	0.97 \pm 0.003	0.95 \pm 0.05	$t(18) = 1.26, p = .223$	$\tilde{p} = .223$
CIFAR vs. GTSRB	OOD detection acc	96.23 \pm 4.00	97.34 \pm 0.43	$t(18) = 0.87, p = .394$	$\tilde{p} = 1$
	ID classification acc	75.68 \pm 0.46	75.76 \pm 0.10	$t(18) = 0.54, p = .598$	$\tilde{p} = 1$
	Combined acc	85.95 \pm 2.03	86.45 \pm 0.36	$t(18) = 0.77, p = .453$	$\tilde{p} = 1$
	AUROC	0.93 \pm 0.002	0.92 \pm 0.025	$t(18) = 1.26, p = .223$	$\tilde{p} = .892$

Table 10: Summary statistics (mean \pm std. dev.) and Student’s t-test results for within-dataset novelty detection.

Dataset	Measure	Baseline	SHELS	p-value	Adjustment
MNIST	OOD detection acc	77.65 \pm 8.31	97.95 \pm 1.26	$t(9) = 8.14, p < .0001$	$\tilde{p} < .001$
	ID classification acc	92.58 \pm 0.84	85.07 \pm 0.69	$t(9) = 27.21, p < .0001$	$\tilde{p} < .001$
	Combined acc	85.11 \pm 4.04	91.51 \pm 0.65	$t(9) = 5.90, p = .0005$	$\tilde{p} = .001$
	AUROC	0.91 \pm 0.06	0.96 \pm 0.02	$t(9) = 2.80, p = .0207$	$\tilde{p} = .0207$
FMNIST	OOD detection acc	43.64 \pm 12.76	64.81 \pm 14.91	$t(9) = 5.06, p = .0007$	$\tilde{p} = .002$
	ID classification acc	84.35 \pm 4.41	77.68 \pm 2.70	$t(9) = 11.60, p < .0001$	$\tilde{p} < .001$
	Combined acc	83.99 \pm 5.75	71.25 \pm 6.88	$t(9) = 3.48, p = .0069$	$\tilde{p} = .014$
	AUROC	0.74 \pm 0.09	0.79 \pm 0.07	$t(9) = 2.04, p = .072$	$\tilde{p} = .072$
SVHN	OOD detection acc	51.92 \pm 7.10	69.24 \pm 4.75	$t(9) = 15.40, p < .0001$	$\tilde{p} < .001$
	ID classification acc	87.49 \pm 2.06	79.23 \pm 1.23	$t(9) = 14.75, p < .0001$	$\tilde{p} < .001$
	Combined acc	69.70 \pm 2.78	74.23 \pm 2.27	$t(9) = 7.13, p < .0001$	$\tilde{p} < .001$
	AUROC	0.81 \pm 0.02	0.82 \pm 0.03	$t(9) = 1.31, p = .224$	$\tilde{p} = .224$
CIFAR10	OOD detection acc	52.62 \pm 5.41	59.78 \pm 6.94	$t(9) = 7.95, p < .0001$	$\tilde{p} < .001$
	ID classification acc	67.83 \pm 3.20	67.12 \pm 2.85	$t(9) = 3.98, p = .0032$	$\tilde{p} = .006$
	Combined acc	60.22 \pm 3.85	63.45 \pm 4.40	$t(9) = 7.23, p = .0001$	$\tilde{p} < .001$
	AUROC	0.69 \pm 0.07	0.70 \pm 0.06	$t(9) = 0.33, p = .3305$	$\tilde{p} = .331$
GTSRB	OOD detection acc	76.35 \pm 13.06	81.43 \pm 4.57	$t(9) = 1.21, p = .257$	$\tilde{p} = .514$
	ID classification acc	88.71 \pm 2.01	83.94 \pm 1.26	$t(9) = 6.32, p = .0001$	$\tilde{p} < .001$
	Combined acc	82.53 \pm 7.28	82.68 \pm 2.24	$t(9) = 0.07, p = .949$	$\tilde{p} = 1$
	AUROC	0.82 \pm 0.06	0.88 \pm 0.06	$t(9) = 3.36, p = .008$	$\tilde{p} = .024$

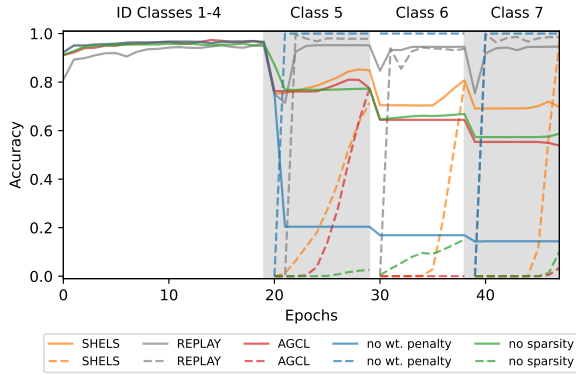


Figure 13: Class incremental learning on FMNIST, pre-trained on four classes, with three more classes learnt incrementally. Solid lines show overall test accuracy, dashes show test accuracy of the novel class only.

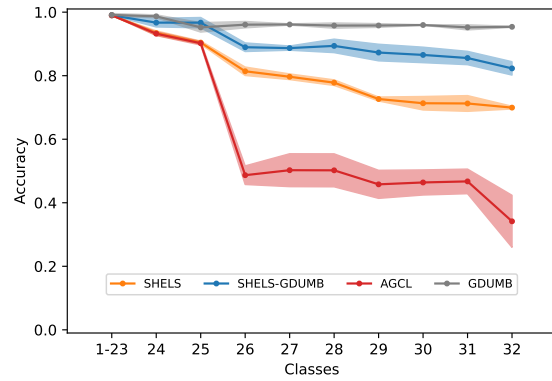


Figure 14: Class incremental learning on GTSRB, pre-trained on 23 classes with nine classes learnt incrementally. Curves show overall test accuracy, showing mean \pm standard deviation across three random seeds.

the new class, making it significantly easier to avoid catastrophic forgetting. This result demonstrates the SHELs representation’s compatibility with replay. We observe that GDUMB still performs slightly better than SHELs-GDUMB, due to the weight penalty feature of SHELs, which penalizes updates to previously important nodes, consequently leading to network saturation. However, note that the GDUMB algorithm on its own (i.e., without SHELs) cannot perform OOD detection, and is thus unsuitable for the combined novelty detection and accommodation paradigm.