LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT
WHERE INNOVATION BEGINS

SAND2021-11382R
LDRD PROJECT NUMBER: 224706
LDRD PROJECT TITLE: Continual Learning for Pattern Recognizers using Neurogenesis Deep Learning
PROJECT TEAM MEMBERS: James Z. Harris (PI), Dylan Fox, Yang Ho, Shannon Kinkead, Omar Garcia (PM)

ABSTRACT:

Deep neural networks have emerged as a leading set of algorithms to infer information from a variety of data sources such as images and time series data. In their most basic form, neural networks lack the ability to adapt to new classes of information. Continual learning is a field of study attempting to give previously trained deep learning models the ability to adapt to a changing environment. Previous work developed a CL method called Neurogenesis for Deep Learning (NDL). Here, we combine NDL with a specific neural network architecture (the Ladder Network) to produce a system capable of automatically adapting a classification neural network to new classes of data. The NDL Ladder Network was evaluated against other leading CL methods. While the NDL and Ladder Network system did not match the cutting edge performance achieved by other CL methods, in most cases it performed comparably and is the only system evaluated that can learn new classes of information with no human intervention.

I. INTRODUCTION AND EXECUTIVE SUMMARY OF RESULTS:

Fields including device inspection and proliferation detection require rapid response and adaptation to changes to their environment. Rare events in spare datasets must be recognized and accounted for. Deep learning is inherently neither adaptive nor versatile [1]. Many data points are required for model generalization and a human field expert is needed to be in-the-loop to fine-tune the model. Furthermore, deep learning models readily forget old tasks when learning new tasks, a process known as catastrophic forgetting [2]. Continual learning seeks to remedy these shortcomings.

The dual-memory model of mammalian memory has received recent attention for its benefits to brain-inspired CL. In this neurogenesis model, immature hippocampal neurons exhibit increased plasticity and are dynamic in their representations, whereas mature hippocampal neurons have reduced learning and maintain their representations. These structures allow mammals to learn

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

and create new memories while maintaining existing memories and knowledge, essentially a biological form of CL.

Prior work developed a CL method for deep neural networks (DNNs) called Neurogenesis for Deep Learning (NDL) and demonstrated how neurogenesis could improve image reconstruction of deep fully connected autoencoders. When confronted with a novel input, new "plastic" neurons are added and trained to incorporate the anomalous input and older "mature" neurons are preserved to maintain the model's original representations. By using the reconstruction error from within the autoencoder network, NDL detects novel inputs and then adapts to them. See [3] for a detailed look into this method, and its application on reconstruction problems using autoencoders. In this work, we constructed a ladder network out of convolutional autoencoders to achieve complex pattern recognition while leveraging previous work on NDL. We evaluated our ladder network and NDL pipeline (LN+NDL) against three other state-of-the-art adaptive CL methods on pattern recognition tasks; namely Elastic Weight Consolidation (EWC) [2], Learning Without Forgetting (LWF) [4], and Latent Replay [5].

EWC finds regions of the operational envelope that two disparate classes have in common. By preserving these overlapping experience regions, EWC is able to learn one task and preserve the shared experience of previous tasks. LWF places model weights into three categories: shared, task-specific, and new-task. Weights in the shared category are kept the same throughout training. The task-specific weights are refined for a given class. New-task weights are added when a novel input is detected and are trained to perform the new task. Latent replay saves the activations from earlier layers of a neural network. These activations can be considered a compressed representation of the original input and are reintroduced when training on new tasks to ensure the model maintains a representation of all previous tasks.

We can transform an autoencoder network intended for reconstruction problems into a pattern recognizer by creating a Ladder Network out of autoencoders [6]. Ladder networks train a decoder for each layer of the auto-encoding network. This is allows the NDL process to monitor reconstruction loss at each layer, detect anomalous inputs, and adapt the network by adding and training new neurons. While it is directly relevant to NDL, ladder networks are also capable of performing semi-supervised learning, so they may be trained using unlabeled data as well as labeled data. By applying NDL to a ladder network of autoencoders, we create a CL method capable of automatically detecting when an anomalous input occurs and simultaneously adapting to the input. Our goal is to achieve 10% higher classification accuracy than previous state-of-art methods with reasonable overhead and no loss in learning capacity over time. Upon completion, the codebase could be used by device inspection and proliferation detection programs to detect and respond to anomalies in dynamic environments.

We successfully constructed the Convolutional Ladder Network with NDL (LN+NDL) pipeline, analyzed its individual performance, and measured our success on accuracy, retention, and footprint. To support this evaluation, we designed an automated CL test and evaluation pipeline for all four methods. Addressing each metric in turn, the LN+NDL pipeline efficiently detected anomalies and adapted to them with comparable performance to the other three state-of-art methods. The model converged faster than the other methods and had comparable accuracy. However, the LN+NDL pipeline had lower retention compared to the other methods. This came as a surprise, since our method was expanding the architecture much more than the other methods. We determined the lower retention in our pipeline was caused by the ladder network architecture becoming unstable after multiple classes were added. Finally, the LN+NDL pipeline incurred a higher than expected footprint due to the duplicated layers in the ladder network architecture. We observed that the pipeline was capable of semi-supervised tasks as well as detecting the anomalous inputs presented to it. However, the LN+NDL pipeline overall fell below the bar held by the other state-of-art methods. If the ladder network instability issue could be resolved, we expect the pipeline would be an exceptional replacement to state-of-art methods due to its automated anomaly detection capabilities.

II. DETAILED DESCRIPTION OF RESEARCH AND DEVELOPMENT AND METHODOLOGY:

II.A Ladder Network architecture

Ladder networks are a specific type of neural network architecture developed in [7]. These networks are designed to take advantage of datasets with a large amount of unlabeled data. A ladder network is composed of three distinct neural networks; a “noisy” autoencoder, a decoder, and a “clean” autoencoder. Both the labeled and unlabeled inputs are input into the “noisy” encoder. This encoder seeks to encode the data into a small latent space like most autoencoders but has Gaussian random noise added to each layer. The data is also input into the clean encoder, which has no noise added to it. The “noisy” encoder encodes the data, which is then passed to the decoder. Each layer of the decoder takes the corresponding layer of the “noisy” encoder and the previous layer of the decoder as inputs. The decoder attempts to decode the noisy data from the noisy encoder into good representations similar to the “clean” encoder’s corresponding layer. For unlabeled data samples, the network is trained by back propagating the gradients of the loss between the decoded representation and the clean encoder. This reconstruction error also allows the NDL process to detect anomalous inputs. For labeled data examples, the network is trained by back propagating the gradients of the loss encountered by the final classification of the clean encoder. The clean encoder and the noisy encoder share weights such that the clean encoder is trained when unlabeled data is ran through the noisy encoder and the decoder. When then network is utilized to make a prediction, the inputs are ran through the clean encoder and the output of the clean encoder is utilized. See **Figure 1** for a diagram of the network.

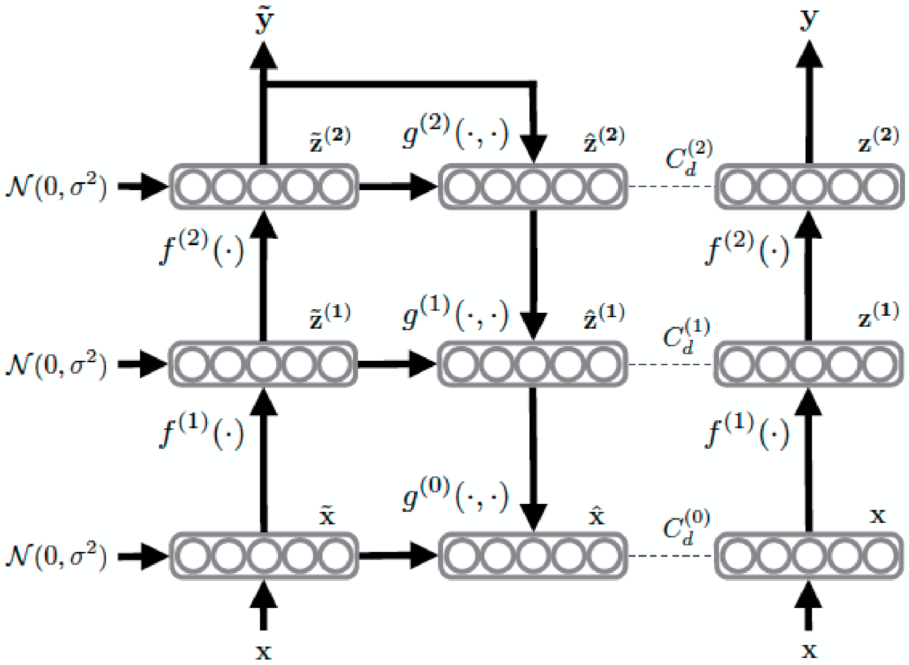


Figure 1 A diagram of the ladder network architecture from [7]. The noisy encoder is on the left and produces $\tilde{y}$. The clean encoder is on the right and produces y, the prediction utilized for inference. The reconstruction error (RE) is the difference between the decoder (center) and the clean encoder at a specific layer. A high RE during inference causes the NDL process to add neurons and adapt the network.

II.B Ladder Network with Neurogenesis

Combining neurogenesis with ladder networks allows the network to continually learn new classes of data. For example, a ladder network could be trained to recognize two classes of object, say cars vs. trucks. Then if a third class that is substantially different from the first two is introduced, e.g. tractors, the network may learn that a new class exists. For details of the NDL process applied to autoencoders, see [3]. The NDL process with ladder networks is as follows:

1. The neurogenesis takes two user set parameters, an acceptable reconstruction error (RE) and a specific number of samples acceptable over reconstruction error threshold (Thresh_bad).
2. For each layer in the network:

   a. The new data (containing data from existing classes in the network, possibly interspersed with new classes of data) is fed through this layer of the noisy encoder, followed by the corresponding layer of the decoder.
   b. The number of samples with an error between the output of the decoder and the original input greater than RE is calculated. Call this number N_bad.
   c. While N_bad exceeds Thresh_bad, channels are added to the current layer one by one. This enables the network to adapt to new data. Then new channels are briefly trained (with previous layers frozen), and the reconstruction errors reevaluated.

3. The samples are fed through the clean encoder in the network, which has possibly had channels added to some or possibly all layers in step 2. Note that the set of weights are shared between the noisy encoder and the clean encoder, so if channels are added to one, they are added to both.
4. The output of the clean encoder is used as the prediction. If channels have been added to the final layer, then a new class has been identified by the network.

This process allows the network to learn new classes of data.

II.C Metrics

Our success depends on whether we show advantage over the state-of-art methods based on the following three criteria:

1. Accuracy: Percentage of new classes detected and accounted for. The purpose of the accuracy metric is to show how efficiently we can detect anomalies. Based on past performance from NDL [3], we expect a 10% increase in accuracy over alternative methods.
2. Retention: Long-term retention of tasks as new tasks are added. We expect no drop in long-term retention or learning capacity which are present in the alternative methods.
3. Footprint: Memory and processor requirements. We expect our method to take up more memory and require more processing time than the other three methods, however we expect this overhead to be reasonable when accounting for the gains in accuracy and retention provided by this pipeline.

II.C.1 Neural Network Architecture

A simple neural network architecture we named ConvNet was used to collect data for our metrics. This network is complex enough to capture the MNIST dataset, however has a low enough learning capacity to strain the continual learning methods when all classes are added. The network architecture is show in **Table 1**. There are 2226 total weights in this network that can be trained.

Table 1 ConvNet architecture

LAYER	Output Dimension	Param count
INPUT	28x28x1	
CONV 1	28x28x2	52
BATCH NORMALIZATION	28x28x2	
RELU	28x28x2	
MAXPOOL	14x14x2	
CONV 2	14x14x4	204
BATCH NORMALIZATION	14x14x4	
RELU	14x14x4	
MAXPOOL	7x7x4	
LINEAR	10	1,970

II.D Datasets

We considered a set of datasets that would provide metrics relevant to real-world scenarios:

1. MNIST: We selected the handwritten number dataset called MNIST [8] to assess the performance of the ladder network and characterize greater aspects of the LN+NDL pipeline. This dataset would help us understand how the new pipeline is behaving, and provide an interpretable comparison between our pipeline and the state-of-art methods.
2. CORe50: The Continual Object Recognition, Detection and Segmentation Dataset called CORe50 [9] was considered to test how LN+NDL adapted to complex image data. We intended to run our model with this dataset, however we prioritized MNIST for its interpretability.
3. STEAD: The Stanford Earthquake Dataset [10] was considered to provide an example of real-world time-series data, which we would process as spectrograms. We were able to construct spectrograms from this dataset, however in the end we did not run a full continual learning analysis on the data because the dataset did not contain the class variety needed for continual learning.

II.E Continual Learning Test Bed

In order to gather data for the metrics mentioned above, a testbed was constructed for running the continual learning methods. The tests were separated into two suites, Suite 1 intended to provide data for the accuracy metric and Suite 2 intended to provide data for the retention metric. The footprint metric would use timing data from both test suites. We describe the two test suites below.

II.E.1 Test Suite 1

The first test suite focused on the accuracy metric. We pretrained the networks with a subset of N classes and then evaluated the ability for the networks to adapt to the next class. We selected *N*=2 for our tests. We trained the networks on two cases: "1,7" and "0,6", where the networks were pretrained with MNIST 1's and 7's in the first case and 0's and 6's in the second. We chose these cases by looking at the distribution of the MNIST data on a variational autoencoder (VAE) [11]. The VAE naturally grouped the classes in the following subsets: {1, 4, 7, 9}, {0, 6}, {5, 8}, {2, 3}. By training on {0, 6} and testing on a different class, we maximized the amount of adaptation required by the CL methods.

II.E.2 Test Suite 2

The second test suite focused on the retention metric. We pretrained the networks with the same approach as Suite 1, then we trained on a sequence of new classes. We chose the following two tests to run in this suite. The first test trains on two similar classes (4 and 9), then adapts on dissimilar classes. The second test has class similarity evenly distributed among the adapted classes:

1. Train on {1, 7} then adapt to 4, 9, 0, 6, 8, and 5.
2. Train on {0, 6} then adapt to 9, 8, 1, 3, 7, and 5.

III. RESULTS AND DISCUSSION:

III.A Ladder Network performance analysis

In order to assess the performance of the ladder network in handwritten digit recognition, the network was trained with 70% unlabeled data, and learning rates ranging between 0.01 and 0.02 with steps of 0.001 for 75 epochs. In each experiment, the maximum accuracy for the training and test epochs where tracked. In all cases, the ladder network showed increased performance throughout the training period, with all learning rates achieving at least an accuracy of 60%. However, when the learning rate was 0.01, the algorithm showed steady improvement throughout all epochs, and achieved a maximum accuracy of about 90%, indicating the smaller learning rate was most effective. The accuracy curve began to level off after 70 epochs, indicating that only small accuracy gains were likely to be achieved from further training.

III.B LN+NDL vs State-of-art

The test suites above were used to gather data for all three metrics, which we compare in this section. The ladder network code was able to automatically detect new classes of data on the

MNIST dataset and achieve reasonable accuracy on new classes of data. However, it did not perform as well as some of the current state of the art methods for continual learning. While the LN+NDL methodology did not perform as well as other state of the art methods, it does allow automatic detection of new classes of data, which none of the other methods allow. The other methods also must have human intervention to train on new data classes.

III.B.1 Accuracy Metric

Here we compare the accuracy of the LN+NDL pipeline to the other state-of-art methods. Each network was trained on a pair of classes, 1's and 7's or 0's and 6's. They were then tasked with adapting to a new class. To specify which classes were trained and added, the initial pretrained pair is noted as "1,7" or "0,6" and then any following classes are adapted by the network. **Table 2** shows the accuracies for each method on ConvNet. LN+NDL outperformed LWF in every case. LN+NDL also outperformed EWC and Latent Replay in some cases, however on average EWC and Latent Replay achieved higher accuracies.

Table 2 New-class training accuracies for models pretrained on classes "1,7" and "0,6"

ADDED CLASS	LN+NDL	EWC	Lat Replay	LWF
1,7 + 0	97.67	99.18	98.67	76.16
1,7 + 2	95.79	97.44	96.82	76.34
1,7 + 3	97.26	98.05	100	74.66
1,7 + 4	100	98.67	97.79	74.16
1,7 + 5	94.55	98.36	100	77.73
1,7 + 6	100	98.68	98.42	72.31
1,7 + 8	100	97.6	100	75.76
1,7 + 9	88.89	96.62	95.05	71.03
0,6 + 1	90.41	98.67	98.19	64.19
0,6 + 2	91.43	96.84	95.85	72.78
0,6 + 3	93.75	97.33	100	72.88
0,6 + 4	75	97.71	97.45	73.12
0,6 + 5	80	95.46	95.92	73.81
0,6 + 7	89.39	97.35	100	68.87
0,6 + 8	91.67	96.96	96.81	73.65
0,6 + 9	89.36	97.66	97.73	71.24
AVERAGE	**92.2**	**97.66**	**98.04**	**73.04**
STD DEV	**7**	**0.94**	**1.66**	**3.25**

Figure 2 shows an example training profile of all four methods. The LN+NDL network (red line) is considerably less stable while training. It does not converge to a local minima like the other methods, but instead it wavers around a minima without stabilizing. We attribute this behavior to the ladder network constantly agitating the weights during the denoising training step.

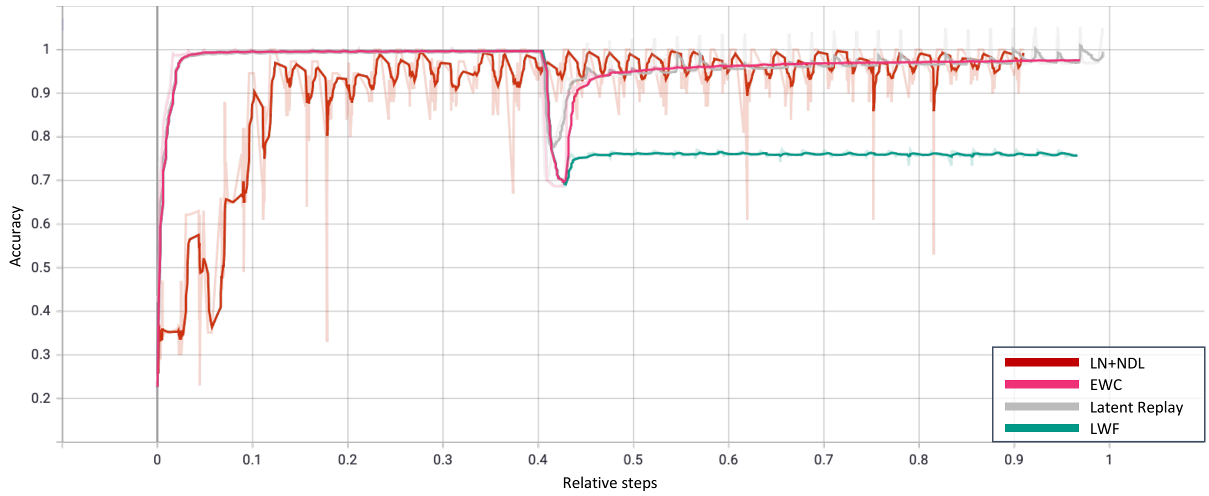


Figure 2 Accuracy of all methods, pretrained on 1,7 and adapted to 8

We now take a closer look at one of the "original" two classes that the networks were trained on before adapting. We expect to see little change in accuracy of this class before and after adaptation. **Table 3** shows the test accuracy of the 1's class before and after the models adapted to the other classes. The average test accuracy for LN+NDL decreased from 99.67% to 98.48%. This is higher compared to the other methods, however the decrease is within the standard deviation of noise inserted by the ladder network. **Table 4** shows the average test accuracy of the 0's class before and after the models adapted to the other classes. There was a 6% decrease of test accuracy for LN+NDL after adapting, which is a nonnegligible amount. **Figure 3** shows these accuracies during training. Evidently, the pipeline is capable of preserving the previous class in nearly all cases except when adapting to Class 4 and Class 5. We attribute the poor performance to the fact that those classes are very different from Class 0 [11].

Table 3 Class 1 Average Test Accuracies, 1+7 Pretrained

	LN+NDL	**EWC**	**Lat Replay**	**LWF**
BEFORE ADAPTING	99.67	99.9	100	99.91
AFTER ADAPTING	98.48	99.43	99.36	99.91
CHANGE [%]	**1.19**	**0.47**	**0.64**	**0**

Table 4 Class 0 Average Test Accuracies, 0+6 Pretrained

	LN+NDL	EWC	Lat Replay	LWF
BEFORE ADAPTING	98.42	98.89	98.74	98.89
AFTER ADAPTING	92.18	98.24	97.92	99.04
CHANGE [%]	**6.34**	**0.66**	**0.83**	**-0.15**

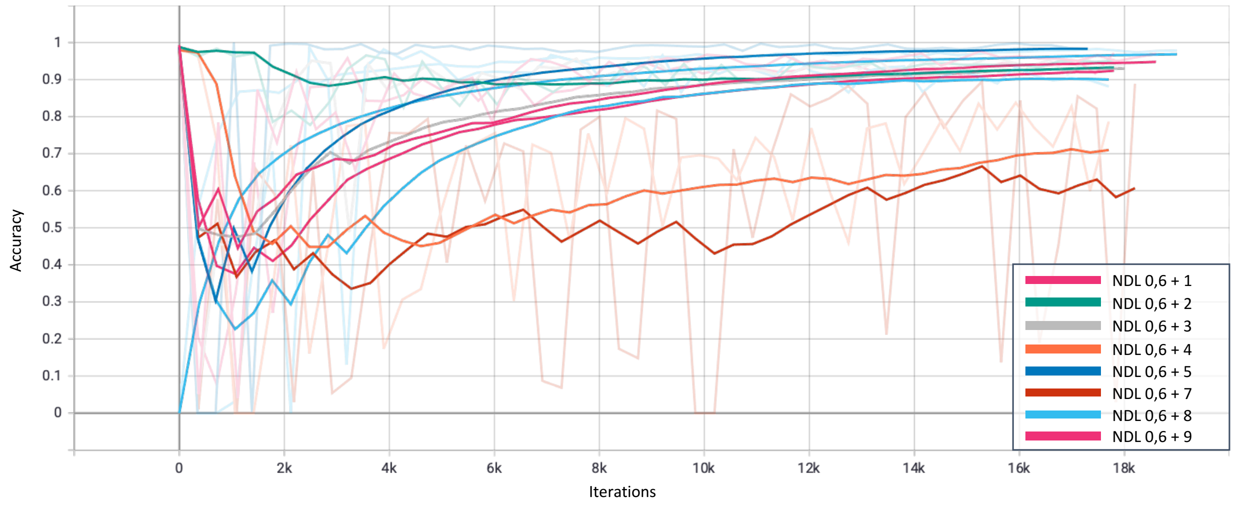


Figure 3 Class 0 Test Accuracy of LN+NDL as new classes are adapted

III.B.2 Retention Metric

We next compare LN+NDL retention to that of the other methods. Once a new class is added, all training samples of that class are included in the accuracy. If the model perfectly adapts to the new class, the accuracy will be greater than or equal to that of the original accuracy. We expected no loss in retention as more classes were added to the LN+NDL pipeline, however the pipeline experienced retention loss nearly every time a new class was added. **Figure 4** shows the accuracy of all classes while LN+NDL was trained. Arrows indicate when a new class was added to the mix. The network initially showed promising results, bouncing back to high accuracies after incorporating the first two new classes. However, the accuracy did not recover after the third class was incorporated.

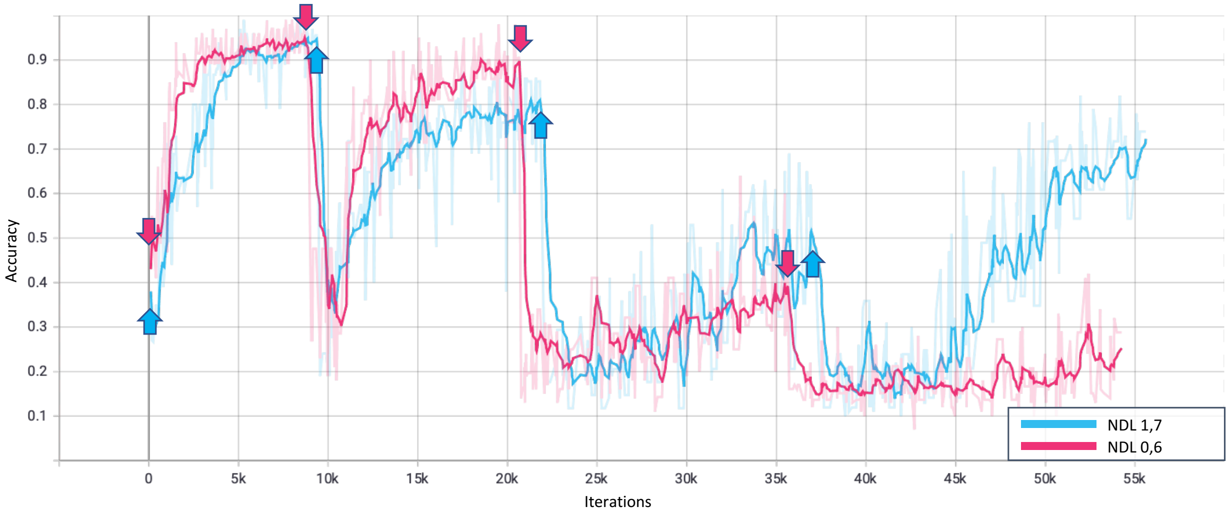


Figure 4 LN+NDL Train Accuracy of all classes as new classes are added. Arrows indicate when a new class was added. For the network pretrained on 1's and 7's (blue), the classes added were 4's, 9's, and 6's, and 0's. For the network pretrained on 0's and 6's (pink), the classes added were 9's, 8's, and 1's, and 3's.

Table 5 and **Table 6** show the train accuracy of all classes as more classes were adapted. All three state-of-art methods experienced a decrease in classification accuracy after the four new classes were added. This phenomenon was present in both the "1,7" and "0,6" configuration. LWF experienced the largest decrease of 21%. However, the loss in retention was much more pronounced for LN+NDL, which experienced a 74% decrease in the "0,6" scenario. This result suggests the new pipeline is experiencing catastrophic forgetting.

Table 5 Train Accuracy as new classes added, 1+7 Pretrained

ADDED CLASS	LN+NDL	EWC	Lat Replay	LWF
1,7	99.24	99.65	99.47	99.65
1,7 + 4	94.65	98.42	97.67	73.22
1,7 + 4 + 9	80.98	95.91	94.64	69.53
1,7 + 4 + 9 + 0	51.57	96.97	95.15	75.54
1,7 + 4 + 9 + 0 + 6	72.27	96.37	95.22	78.73
CHANGE [%]	**-27.18**	**-3.29**	**-4.27**	**-20.99**

Table 6 Train Accuracy as new classes added, 0+6 Pretrained

ADDED CLASS	LN+NDL	EWC	Lat Replay	LWF
0,6	98.71	99.65	98.51	99.65
0,6 + 9	95.01	98.4	97.69	73.34
0,6 + 9 + 8	89.77	95.78	96.18	69.57
0,6 + 9 + 8 + 1	39.19	96.96	96.2	75.53
0,6 + 9 + 8 + 1 + 3	25.23	96.35	94.66	78.79
CHANGE [%]	**-74.58**	**-3.31**	**-3.91**	**-20.93**

To determine whether catastrophic forgetting is occurring for LN+NDL, we analyzed the accuracy of the original classes as the model adapted. **Figure 5** shows the training accuracy of Class 1 as the 4, 9, 0, and 6 classes were added. The accuracy dropped almost immediately after Class 0 was added. The architecture changed drastically when incorporating Class 0, and it was unable to recover Class 1 until Class 6 was added and adapted. We hypothesize that the ladder network architecture became unstable after NDL added neurons for Class 0. The neurons that were added greatly changed the distribution of the weights within the ladder network architecture. **Figure 6** shows how the weights evolved throughout the training period. After Class 0 was added, i.e. the third histogram in **Figure 6**, the weights in both of the final encoder layers evolved significantly. This evolution placed the ladder network into an unstable state which made it difficult to recover from forgetting Class 1.

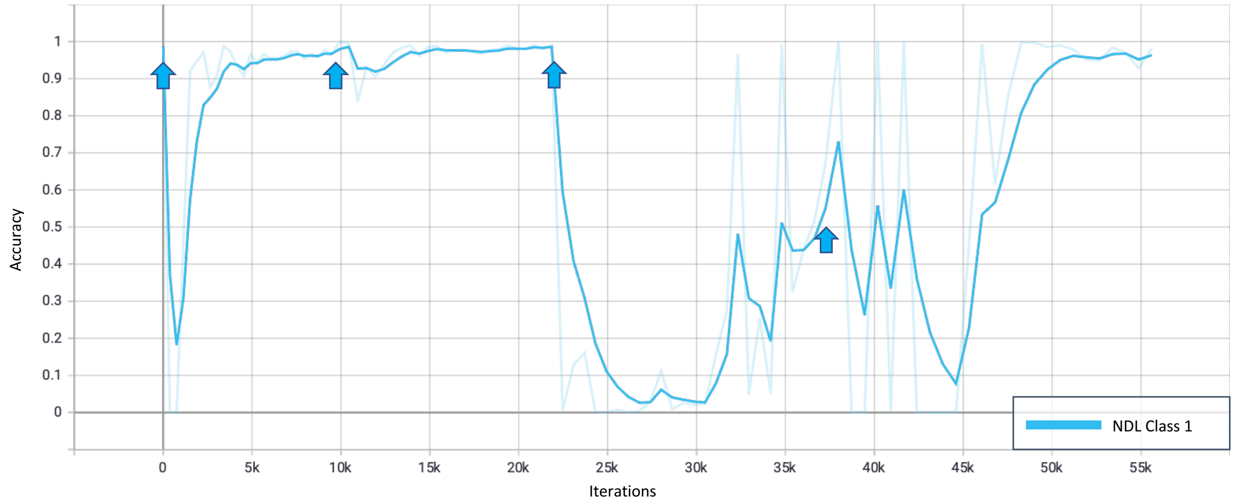


Figure 5 LN+NDL Train Accuracy of Class 1 as new classes are added to the "1,7" pretrained model. The blue arrows indicate when a new class was added. The third arrow indicates when Class 0 was added.

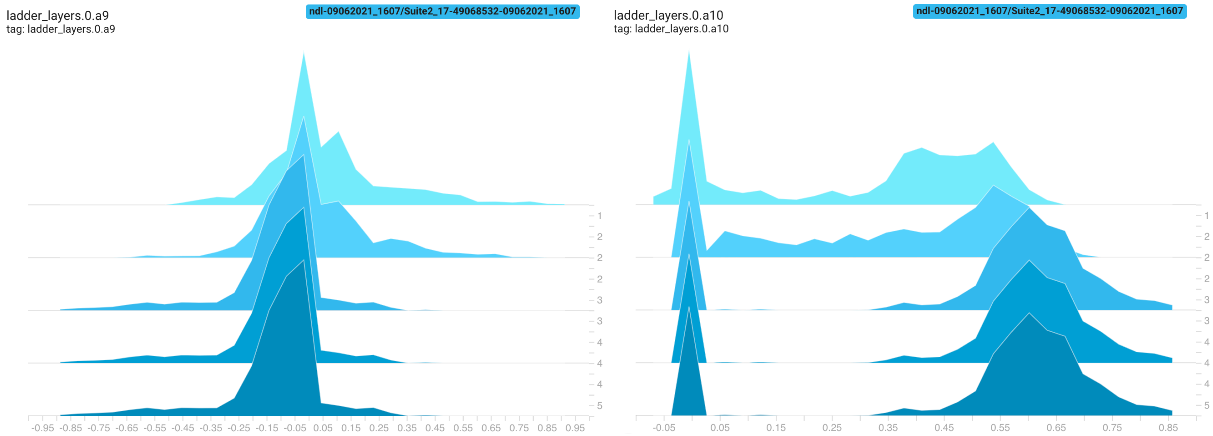


Figure 6 LN+NDL weight histogram overlays of the final two layers in the ladder network encoder architecture. The bottom of the stack is after adapting to Class 4. The top of the histogram stack is after adapting to Class 6. The third histogram from the back is after adapting to Class 0.

We observe a similar phenomenon for Class 6 of the “0,6” case, shown in **Figure 7**. After Class 1 is added to the training step, the training accuracy for Class 6 drops from 95% to 60%. The network was unable to recover the accuracy as new classes were added, and the accuracy bottoms out at 0% for Class 6. In other words, LN+NDL experienced total catastrophic forgetting for Class 6. There was a similar change in weight distribution after Class 1 was added to the architecture, shown in **Figure 8**.

The dramatic change in weight distributions again placed the ladder network in an unstable state that made it difficult to recover from catastrophic forgetting. If weights were added to the architecture in a way that wouldn’t cause instability, then the LN+NDL pipeline would have high retention. As it stands, however, the LN+NDL pipeline has lower retention than the state-of-art methods.

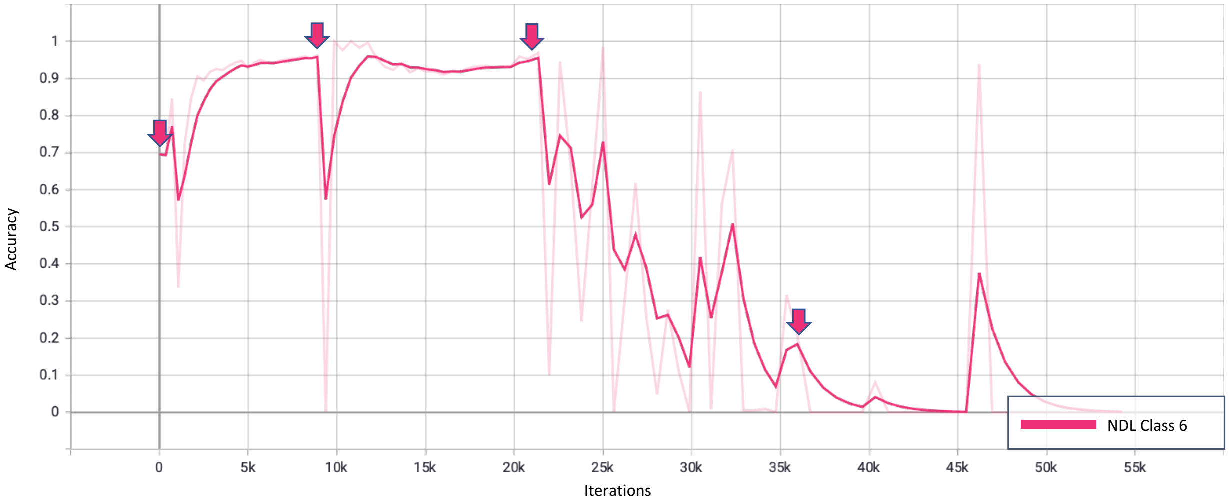


Figure 7 LN+NDL Train Accuracy of Class 6 as new classes are added to the "0,6" pretrained model. The pink arrows indicate when a new class was added. The third arrow indicates when Class 1 was added.

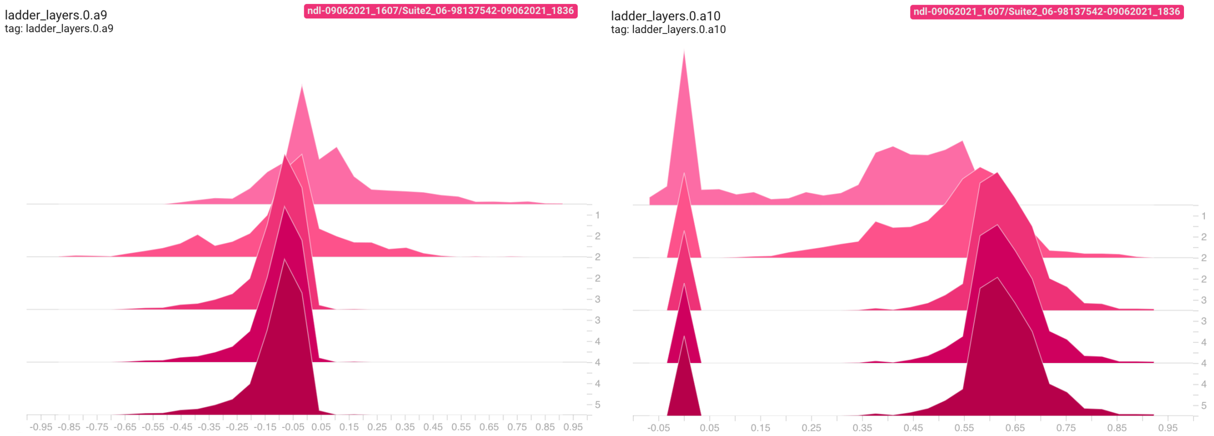


Figure 8 LN+NDL weight histogram overlays of the final two layers in the ladder network encoder architecture. The bottom of the stack are the weights after adapting to Class 9. The top of the stack are the weights after adapting to Class 3. The third histogram from the back is after adapting to Class 1.

III.B.3 Footprint Metric

The footprint of each method was characterized using timing and memory usage acquired during model training. The average memory usage is shown in **Table 7**. The ladder network architecture was expected to have a 3x overhead in memory usage, because the network used by the other methods is duplicated for the two encoders and one decoder in the architecture. The memory usage, however, was higher than 3x. On average, the usage was 7x higher than the other methods. We attribute the overhead to the unseen parameters and mechanisms used to train the LN+NDL pipeline. The unexpected increase made it impossible to train large CNN models such as VGG or MobileNet, because we were limited to a 32 GB GPU which the LN+NDL pipeline eventually surpassed while adapting to new classes.

Table 7 Method size requirements

	LN+NDL	**EWC**	**Lat Replay**	**LWF**
AVG. SIZE [MB]	5675	806	798	806

Although there was a higher than expected memory requirement for LN+NDL, the pipeline's computation speed remained competitive with the other models. **Table 8** shows the average total computation time of each continual learning methods The averages were calculated from the Suite 1 test runtimes.

Table 8 Method computation times

	LN+NDL	**EWC**	**Lat Replay**	**LWF**
TOTAL TIME [MIN]	71	73	70	71

IV. ANTICIPATED OUTCOMES AND IMPACTS:

IV.A Publications

In May 2022 we intend to publish to the NeurIPS 2022 conference on Brain-inspired AI for using ladder networks to apply the neurogenesis method to pattern recognition. Although this work showed an instability in the ladder network architecture, it still showed a valid method for using a previously proven continual learning method, neurogenesis, on task-based pattern recognition problems. Previous state-of-art continual learning methods perform well on task adaptation, however they cannot recognize when a new task is introduced. Furthermore, for EWC and Latent Replay the number of future tasks must be known before-hand. This work therefore makes significant contributions to the continual learning field by showing how a pattern recognizer can not only adapt to an anomalous input, but also determine whether an input is out-of-distribution. The ladder network can also accept unlabeled data, and this data can contain anomalous inputs as well. These results will further the field of continual learning and show that brain-inspired continual learning methods present benefits that prior methods are incapable of producing.

IV.B Tools and Capabilities

We developed many new tools and capabilities that will be applied to various fields in our team's departments. We constructed and optimized a LN+NDL pattern recognition pipeline capable of detecting and adapting to novel inputs. Even though the pipeline was found to have lower retention capabilities than the other state-of-art methods, the pipeline could be directly applied to anomalous input detection problems. By removing the NDL portion, the architecture also presents a semi-supervised pattern recognition solution for utilizing unlabeled data.

To support the evaluation of the new pipeline and comparison to the state-of-art methods, an automated deep learning evaluation pipeline was constructed. This pipeline performs automated data loading, training, and advanced metric collection capable of characterizing continual learning methods. The test-bed could be easily expanded to other continual learning methods.

Finally, along with our LN+NDL pipeline we constructed and vetted three continual learning methods: EWC, LWF, and Latent Replay. These methods are all state-of-art and capable of performing continual learning on real-world scenarios. The methods will be kept as capabilities we can use to accelerate development of other continual learning projects.

IV.C Staff development

Dylan Fox, Zach Harris, Yang Ho, and Shannon Kinkead were all early career employees at the time of this work. Dylan gained experience in deep learning architecture development and neurogenesis architecture development. Zach gained experience in Deep Learning architecture development, with a focus in ladder networks and state-of-art continual learning architecture design. Yang gained experience in deep learning benchmarking, state-of-art continual learning architecture design, and performance analysis of continual learning pipelines. Shannon, a PhD candidate, gained experience in deep learning training and performance analysis.

IV.D Impact and path forward

The progress made in this LDRD can be expanded upon in later works and future opportunities. If the ladder network growth can be regularized, then stable growth of the network will boost retention of the pipeline and offer a solution for rapid response to novel input. Such a system would greatly benefit NNSA nuclear threats missions by providing adaptable proliferation detection systems. These systems would reliably discover anomalies in a sea of data and rapidly adapt to the anomalies, a task imperative to the detection mission. This work would therefore be receptive to future calls in the proliferation detection domain. This work also impacts the Department of Homeland Security Transport Security Administration by offering solutions for device inspection software capable of detecting and adapting to anomalous devices in images.

Further work must be done in order to characterize and remedy the instability observed in the ladder network. However, if achieved, it would take a progressive step towards pioneering pattern recognition systems capable of adapting to rapid changes to their environment.

V. CONCLUSION:

Here, Continual Learning using NDL and Ladder Networks was shown to be capable of recognizing anomalous inputs and adapting to the new environment. The NDL plus Ladder network system displayed performance comparable with cutting edge methods for many added classes of data, but suffered from catastrophic forgetting and instability in the Ladder Network after some added classes. While the performance of the NDL plus Ladder network was generally slightly worse than other cutting edge CL methods, this system can automatically adapt to anomalous inputs, setting it apart from other CL methods.

REFERENCES:

[1] R. Kemker, M. McClure, A. Abitino, T. Hayes and C. Kanan, "Measuring catastrophic forgetting in neural networks.," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[2] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu and K. Milan, "Overcoming catastrophic forgetting in neural networks.," *Proceedings of the national academy of sciences,* vol. 114, no. 13, pp. 3521-3526, 2017.

[3] T. J. Draelos, N. E. Miner, C. C. Lamb, J. A. Cox, C. M. Vineyard, K. D. Carlson, W. M. Severa, C. D. James and J. B. Aimone, "Neurogenesis deep learning: Extending deep networks to accommodate new classes.," *2017 International Joint Conference on Neural Networks (IJCNN),* pp. 526-533, 2017.

[4] Z. Li and D. Hoiem, *IEEE transactions on pattern analysis and machine intelligence ,* vol. 40, no. 12, pp. 2935-2947, 2017.

[5] L. Pellegrini, G. Graffieti, V. Lomonaco and D. Maltoni, Latent Replay for Real-Time Continual Learning, arXiv, 2020.

[6] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby and O. Winther, "Ladder variational autoencoders.," *Advances in neural information processing systems,* vol. 29, pp. 3738-3746, 2016.

[7] M. Pezeshki, L. Fan, P. Brakel, A. Courville and Y. Bengio, "Deconstructing the ladder network architecture.," *International conference on machine learning,* vol. PMLR, pp. 2368-2376, 2016.

[8] L. Deng, *IEEE Signal Processing Magazine ,* vol. 29, no. 6, pp. 141-142, 2012.

[9] V. Lomonaco and D. Maltoni, "Conference on Robot Learning," in *PMLR*, 2017.

[10] S. M. Mousavi, in *IEEE Access 7*, 2019.

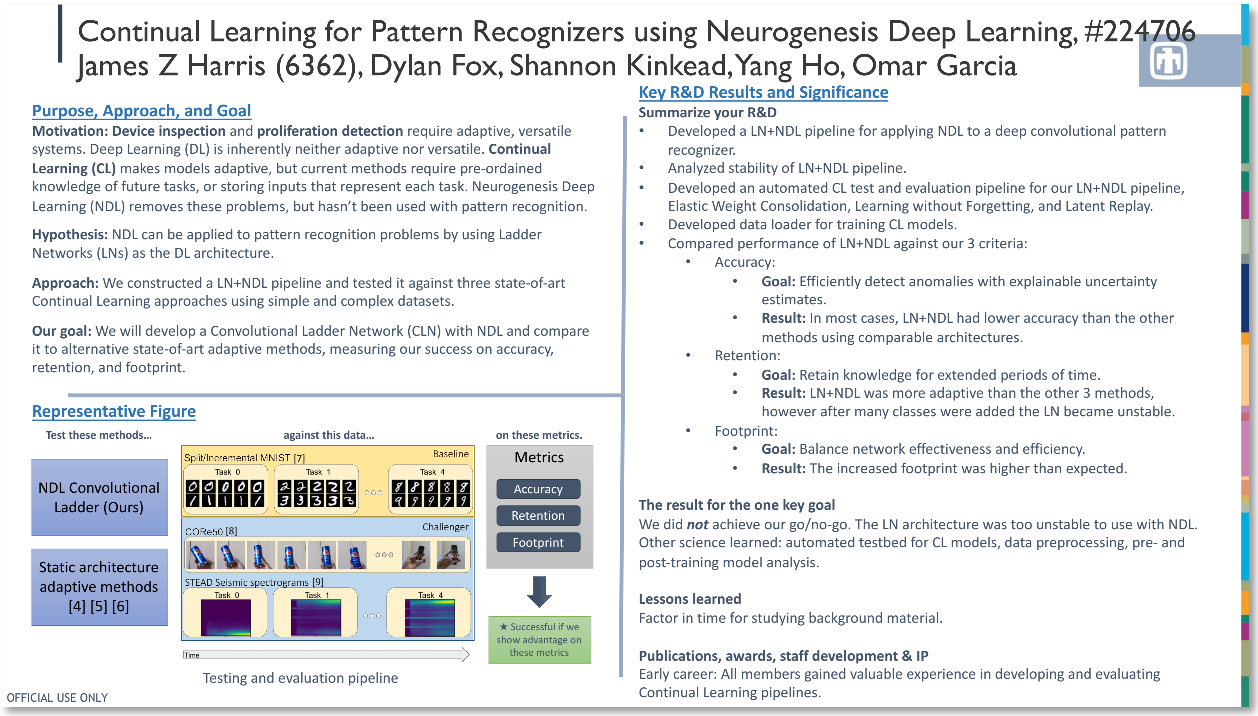
[11] "Convolutional Variational Autoencoder Tutorial," TensorFlow, 17 06 2021. [Online]. Available: https://www.tensorflow.org/tutorials/generative/cvae. [Accessed 10 09 2021].

[12] J. B. Aimone, Y. Li, S. W. Lee, G. D. Clemenson, W. Deng and F. H. Gage, "Regulation and function of adult neurogenesis: from genes to cognition.," *Physiological reviews,* vol. 94, no. 4, pp. 991-1026, 2014.

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT

WHERE INNOVATION BEGINS

ADDENDUM:



Continual Learning for Pattern Recognizers using Neurogenesis Deep Learning, #224706

James Z Harris (6362), Dylan Fox, Shannon Kinkead, Yang Ho, Omar Garcia

Purpose, Approach, and Goal

Motivation: Device inspection and **proliferation detection** require adaptive, versatile systems. Deep Learning (DL) is inherently neither adaptive nor versatile. **Continual Learning (CL)** makes models adaptive, but current methods require pre-ordained knowledge of future tasks, or storing inputs that represent each task. Neurogenesis Deep Learning (NDL) removes these problems, but hasn't been used with pattern recognition.

Hypothesis: NDL can be applied to pattern recognition problems by using Ladder Networks (LNs) as the DL architecture.

Approach: We constructed a LN+NDL pipeline and tested it against three state-of-art Continual Learning approaches using simple and complex datasets.

Our goal: We will develop a Convolutional Ladder Network (CLN) with NDL and compare it to alternative state-of-art adaptive methods, measuring our success on accuracy, retention, and footprint.

Representative Figure

Test these methods…

NDL Convolutional Ladder (Ours)

Static architecture adaptive methods [4] [5] [6]

against this data…

Split/Incremental MNIST [7] — Baseline

Task 0 · Task 1 · Task 4

CORe50 [8] — Challenger

STEAD Seismic spectrograms [9]

Task 0 · Task 1 · Task 4

Time

on these metrics.

Metrics

Accuracy

Retention

Footprint

★ Successful if we show advantage on these metrics

Testing and evaluation pipeline

Key R&D Results and Significance

Summarize your R&D

- Developed a LN+NDL pipeline for applying NDL to a deep convolutional pattern recognizer.
- Analyzed stability of LN+NDL pipeline.
- Developed an automated CL test and evaluation pipeline for our LN+NDL pipeline, Elastic Weight Consolidation, Learning without Forgetting, and Latent Replay.
- Developed data loader for training CL models.
- Compared performance of LN+NDL against our 3 criteria:
  - Accuracy:
    - **Goal:** Efficiently detect anomalies with explainable uncertainty estimates.
    - **Result:** In most cases, LN+NDL had lower accuracy than the other methods using comparable architectures.
  - Retention:
    - **Goal:** Retain knowledge for extended periods of time.
    - **Result:** LN+NDL was more adaptive than the other 3 methods, however after many classes were added the LN became unstable.
  - Footprint:
    - **Goal:** Balance network effectiveness and efficiency.
    - **Result:** The increased footprint was higher than expected.

The result for the one key goal

We did ***not*** achieve our go/no-go. The LN architecture was too unstable to use with NDL. Other science learned: automated testbed for CL models, data preprocessing, pre- and post-training model analysis.

Lessons learned

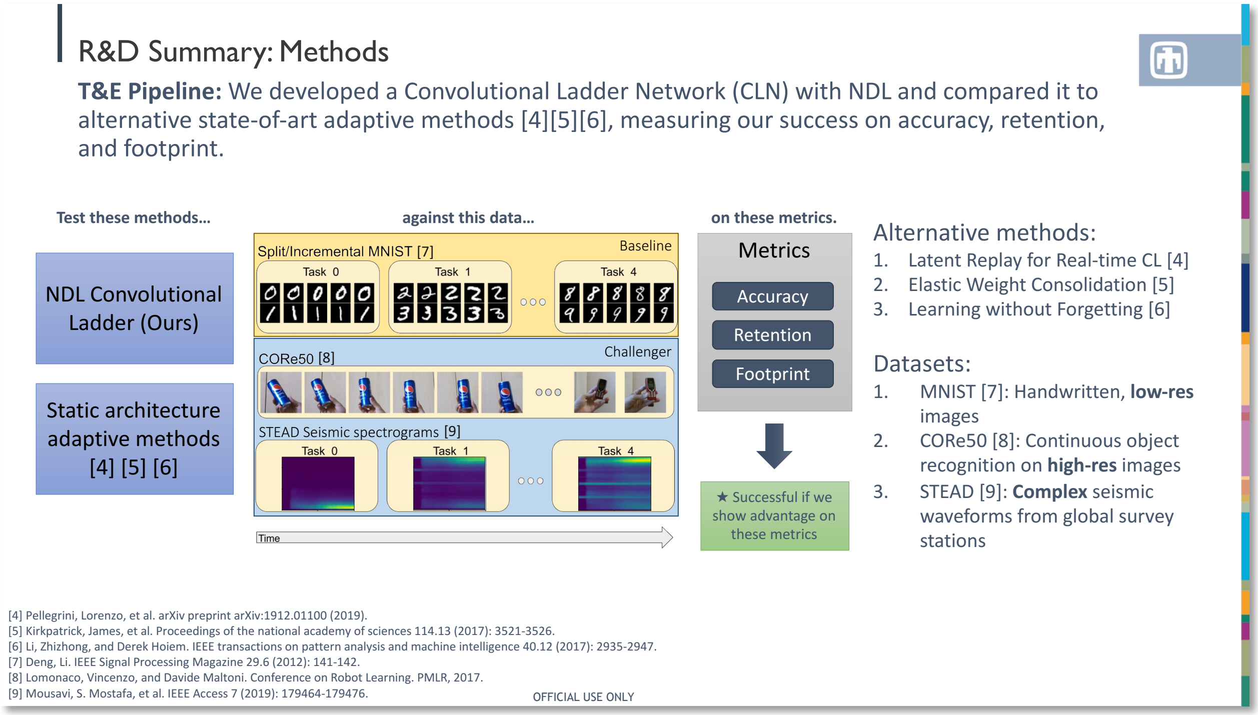
Factor in time for studying background material.

Publications, awards, staff development & IP

Early career: All members gained valuable experience in developing and evaluating Continual Learning pipelines.

OFFICIAL USE ONLY

LABORATORY DIRECTED
RESEARCH & DEVELOPMENT
WHERE INNOVATION BEGINS



R&D Summary: Methods

T&E Pipeline: We developed a Convolutional Ladder Network (CLN) with NDL and compared it to alternative state-of-art adaptive methods [4][5][6], measuring our success on accuracy, retention, and footprint.

Test these methods…

NDL Convolutional Ladder (Ours)

Static architecture adaptive methods [4] [5] [6]

against this data…

Split/Incremental MNIST [7] — Baseline

CORe50 [8] — Challenger

STEAD Seismic spectrograms [9]

on these metrics.

Metrics

Accuracy

Retention

Footprint

★ Successful if we show advantage on these metrics

Alternative methods:

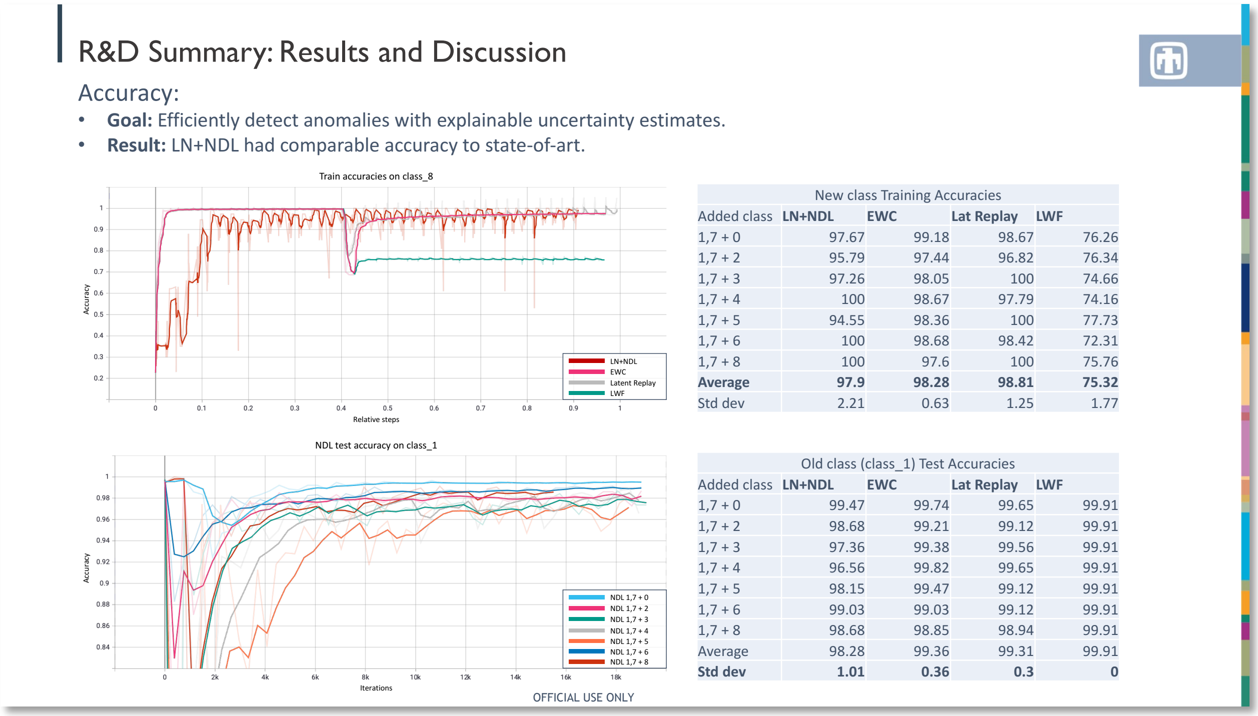
1. Latent Replay for Real-time CL [4]
2. Elastic Weight Consolidation [5]
3. Learning without Forgetting [6]

Datasets:

1. MNIST [7]: Handwritten, **low-res** images
2. CORe50 [8]: Continuous object recognition on **high-res** images
3. STEAD [9]: **Complex** seismic waveforms from global survey stations

[4] Pellegrini, Lorenzo, et al. arXiv preprint arXiv:1912.01100 (2019).
[5] Kirkpatrick, James, et al. Proceedings of the national academy of sciences 114.13 (2017): 3521-3526.
[6] Li, Zhizhong, and Derek Hoiem. IEEE transactions on pattern analysis and machine intelligence 40.12 (2017): 2935-2947.
[7] Deng, Li. IEEE Signal Processing Magazine 29.6 (2012): 141-142.
[8] Lomonaco, Vincenzo, and Davide Maltoni. Conference on Robot Learning. PMLR, 2017.
[9] Mousavi, S. Mostafa, et al. IEEE Access 7 (2019): 179464-179476.

OFFICIAL USE ONLY



R&D Summary: Results and Discussion

Accuracy:

- **Goal:** Efficiently detect anomalies with explainable uncertainty estimates.
- **Result:** LN+NDL had comparable accuracy to state-of-art.

Train accuracies on class_8

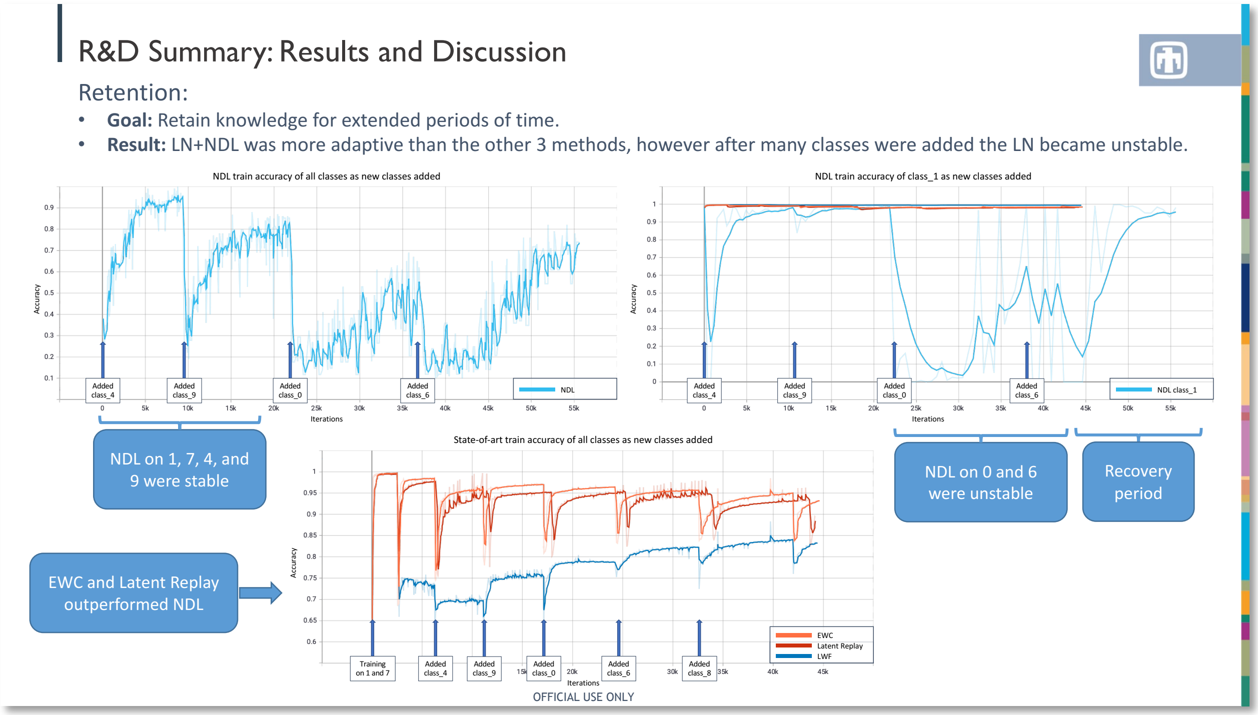
Accuracy
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Relative steps
LN+NDL
EWC
Latent Replay
LWF

New class Training Accuracies				
Added class	LN+NDL	EWC	Lat Replay	LWF
1,7 + 0	97.67	99.18	98.67	76.26
1,7 + 2	95.79	97.44	96.82	76.34
1,7 + 3	97.26	98.05	100	74.66
1,7 + 4	100	98.67	97.79	74.16
1,7 + 5	94.55	98.36	100	77.73
1,7 + 6	100	98.68	98.42	72.31
1,7 + 8	100	97.6	100	75.76
Average	**97.9**	**98.28**	**98.81**	**75.32**
Std dev	2.21	0.63	1.25	1.77

NDL test accuracy on class_1

Old class (class_1) Test Accuracies				
Added class	LN+NDL	EWC	Lat Replay	LWF
1,7 + 0	99.47	99.74	99.65	99.91
1,7 + 2	98.68	99.21	99.12	99.91
1,7 + 3	97.36	99.38	99.56	99.91
1,7 + 4	96.56	99.82	99.65	99.91
1,7 + 5	98.15	99.47	99.12	99.91
1,7 + 6	99.03	99.03	99.12	99.91
1,7 + 8	98.68	98.85	98.94	99.91
Average	98.28	99.36	99.31	99.91
Std dev	**1.01**	**0.36**	**0.3**	**0**

OFFICIAL USE ONLY



R&D Summary: Results and Discussion

Retention:

- **Goal:** Retain knowledge for extended periods of time.
- **Result:** LN+NDL was more adaptive than the other 3 methods, however after many classes were added the LN became unstable.

NDL on 1, 7, 4, and 9 were stable

NDL on 0 and 6 were unstable

Recovery period

EWC and Latent Replay outperformed NDL

LDRD Project Metrics

Presentations and Publications

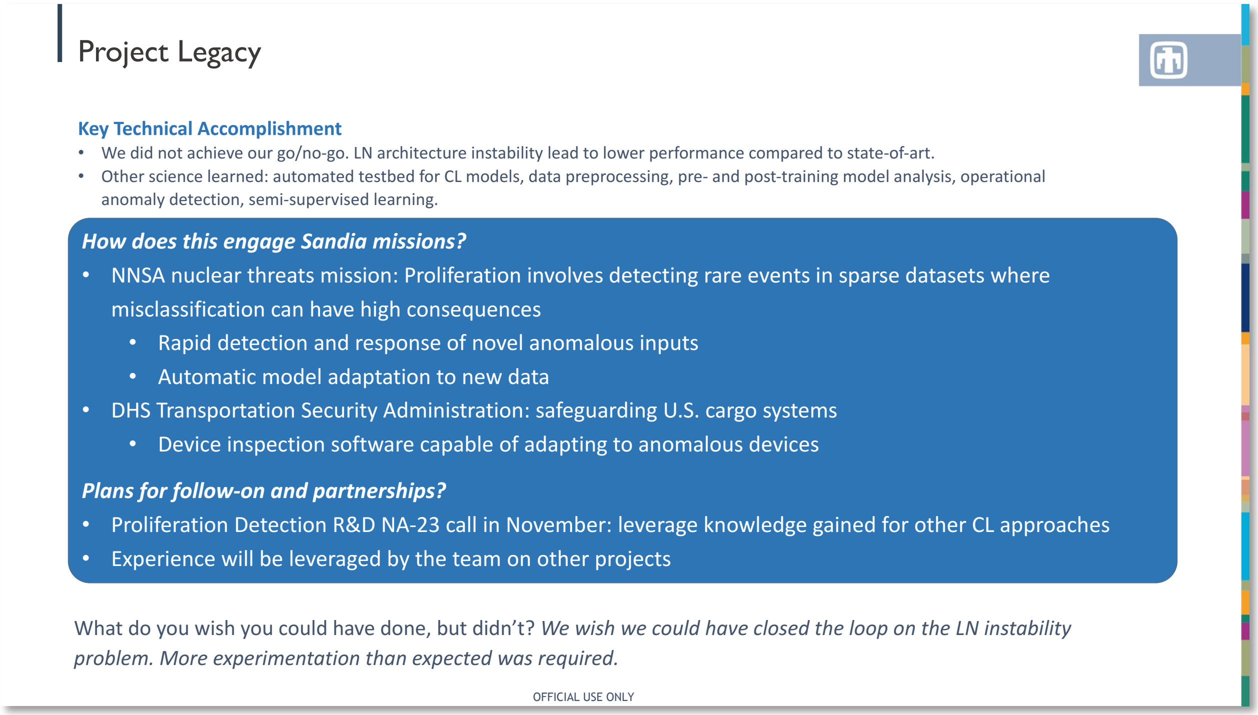
- SAND report.
- NeurIPS 2022 conference paper, submission in May 2022. Need to recompile SAND report.

Tools and Capabilities

- Optimized NDL + Ladder Network pattern recognition pipeline.
- Classifier capable of rapid detection and response of novel anomalous inputs.
- Classifier capable of both supervised and semi-supervised learning.
- Automated Deep Learning evaluation pipeline: Data loading, training, advanced metric collection.
- Operational state-of-art continual learning methods: EWC, LWF, Latent Replay.

Staff Development

- Yang Ho: DL benchmarking and performance analysis. State-of-art architecture development.
- Dylan Fox: LN and NDL architecture development.
- Shannon Kinkead: DL training and LN analysis.
- Zach Harris: DL architecture development. LN and state-of-art architecture development.



Project Legacy

Key Technical Accomplishment

- We did not achieve our go/no-go. LN architecture instability lead to lower performance compared to state-of-art.
- Other science learned: automated testbed for CL models, data preprocessing, pre- and post-training model analysis, operational anomaly detection, semi-supervised learning.

How does this engage Sandia missions?

- NNSA nuclear threats mission: Proliferation involves detecting rare events in sparse datasets where misclassification can have high consequences
  - Rapid detection and response of novel anomalous inputs
  - Automatic model adaptation to new data
- DHS Transportation Security Administration: safeguarding U.S. cargo systems
  - Device inspection software capable of adapting to anomalous devices

Plans for follow-on and partnerships?

- Proliferation Detection R&D NA-23 call in November: leverage knowledge gained for other CL approaches
- Experience will be leveraged by the team on other projects

What do you wish you could have done, but didn’t? *We wish we could have closed the loop on the LN instability problem. More experimentation than expected was required.*

OFFICIAL USE ONLY