BUILD: Buffer-free Incremental Learning with OOD Detection for the Wild

Srishti Gupta [1 2] **Daniele Angioni** [1 2] **Lea Schönherr** [3] **Ambra Demontis** [1] **Battista Biggio** [1]

Abstract

Having a model that can dynamically learn new classes while detecting Out-of-Distribution (OOD) samples is a desirable property for most applications operating in the wild. While there is limited work in this direction, some works have attempted to achieve both by combining Incremental Learning (IL) and OOD detection, showing promising results for both tasks. Most of the works use a buffer containing some samples to either replay past samples while learning or to detect outliers at testing, which can cause potential issues: 1) it does not scale well with a growing number of samples, 2) it causes privacy issues as storing samples may not always be a compliant option, 3) it limits the outlier detection to the distribution in the buffer, and 4) it is computationally and memory expensive. In this work, we tackle this issue with a very simple yet effective framework: BUILD which performs both IL and OOD detection in a buffer-free manner with the capability to work in the wild. BUILD integrates a pre-trained vision transformer that is fine-tuned with hard attention masks, along with post-hoc OOD detectors applied during testing. We show that BUILD when combined with activation-based post-hoc OOD technique, can give not just competitive but better performance than the state-of-the-art baselines. To support our claims, we evaluate the proposed framework on the CIFAR-10 classification benchmark and the results show that BUILD gives superior and stabler performance in detecting OOD samples in computationally much cheaper way.

[1]University of Cagliari, Italy [2]Sapienza University of Rome, Italy [3]CISPA Helmholtz Center for Information Security, Germany. Correspondence to: Srishti Gupta <srishti.gupta@uniroma1.it>.

Published at ICML 2024 Workshop on Foundation Models in the Wild. Copyright 2024 by the author(s).

1. Introduction

To deploy a machine learning application in the wild, the two most desired properties are (i) the ability to safely learn new information without forgetting past ones, and (ii) to detect when the model operates outside of the training distribution, e.g. when the model receives a sample from a new class. These two properties clash with the assumption on which machine learning is founded: the independent and identical distribution (IID) of the inputs for both training and runtime data. Both are expected to be sampled independently from the same underlying distribution a.k.a. closed-set assumption. This assumption does not hold in real-world applications where the data is much more complex and diverse than the data available for training. To address the practical limitations of real-world applications, researchers incorporate strategies such as *Incremental Learning* (IL) (De Lange et al., 2021) (a.k.a. Continual or Lifelong learning) and *Out-of-Distribution* (OOD) detection (Salehi et al., 2022).

IL aims to develop models that can be updated with new information (e.g. add new classes) without forgetting past data, known as *catastrophic forgetting* (CF). The most promising results in IL are achieved by *replay-based* methods (Rebuffi et al., 2017; Rolnick et al., 2019) where some samples from past data are saved in a buffer to train future data. Instead, OOD detection deals with the problem of detecting *unknown* samples, such as from unknown distributions. Recent works (Kim et al., 2023a; Weyssow et al., 2023; Kim et al., 2022) have tried to bridge the two paradigms to achieve the ability to learn incrementally while detecting OOD at test time. All of these works rely on buffering strategies to combine In-Distribution (ID) and OOD detection.

Buffers are widely used to save a subset of old data that can be used i) in IL, to train future classes along with current data to prevent CF or ii) in OOD, so the model can learn what *potentially* OOD data may look like at test time. However, the use of buffers is tied to certain limitations: (i) as the number of classes in IL grows, a fixed-size buffer cannot scale its representative power, (ii) many applications cannot maintain past data for privacy constraints, (iii) the use of buffer restricts OOD detection to a limited distribution of samples, making it unusable for the wide distribution of OOD samples that can be found in the wild, (iv) buffers require additional computations and a large memory footprint.

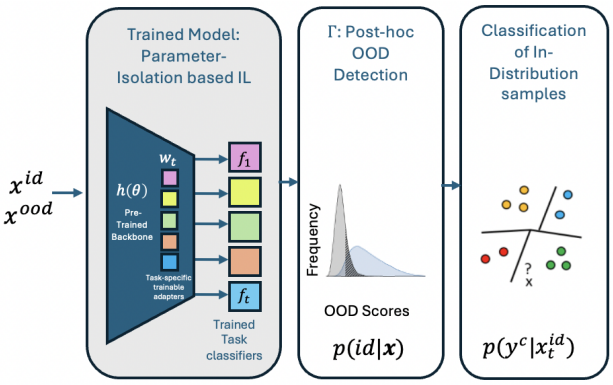

Figure 1. Conceptual representation of BUILD

In this work, we propose a modular and lightweight buffer-free IL model with OOD functionalities capable of functioning in the wild: BUILD. Our pipeline consists of combining the parameter-isolation-based IL technique with recent activation-based post-hoc OOD detection techniques. To the best of our knowledge, we are the first to propose a buffer-free incremental learning approach that can also perform OOD detection. We support our claims by evaluating BUILD using the CIFAR10 classification benchmark. We evaluate each task classifier in the open-world setting while classifying in a task-aware setting. We compare our results with the current state-of-the-art IL+OOD model MORE (Kim et al., 2022).

2. Background and Related Works

While the two domains—IL and OOD—have made tremendous progress lately, we discuss the most relevant techniques in this section with emphasis on bufferless methods.

2.1. OOD Detection

OOD detection aims to detect samples that are semantically different from the training distribution. Categorized according to the life cycle steps, it can be applied at the time of training (Hendrycks et al., 2019b;a) or post-hoc inference (Djurisic et al., 2023; Sun et al., 2021). Outlier Exposure (OE) (Hendrycks et al., 2019a) is a popular training method in which evaluation data is exposed at the time of training (Perera & Patel, 2019; Liang et al., 2018). However, this method has recently lost popularity, as it is difficult to reconcile with the principles of ML and might give overoptimistic results (Zhang et al., 2023). In post-hoc methods, the techniques are applied only after training. Baseline techniques like Maximum Softmax Probability (Hendrycks & Gimpel, 2017) distinguish OOD from ID simply based on the softmax confidence score. Whereas, in the activation-based method, OOD activations are identified and modified to push the confidence scores away from the ID activations. For example, in Rectified Activation (ReAct) (Sun et al., 2021) technique where the d-dimensional input $\boldsymbol{x} \in \mathbb{R}^d$ when encoded by the feature extractor to say, m-dimensional feature space, such that $\boldsymbol{z} \in \mathbb{R}^m$. These activations (or feature vectors) are extracted from the penultimate layer and modified using the rectification parameter c such that the new activations: $\overline{\boldsymbol{z}} = ReAct(\boldsymbol{z}; c)$ are then used for classification: $f^{ReAct}(\boldsymbol{x}; \theta) = \mathbf{W}^T\overline{\boldsymbol{z}} + \mathbf{b}$ where $\mathbf{W}^T$ and $\mathbf{b}$ are derived from the classification layer. These techniques are compatible with different scorings: softmax confidence (Hendrycks & Gimpel, 2017) and energy scoring (Liu et al., 2021). In our experiments, we show results with both scorings i.e. ReAct+MSP and ReAct+Energy respectively.

2.2. Incremental Learning

IL is tackling a major challenge in neural networks: The *stability-plasticity dilemma* where plasticity refers to the ability of the model to learn new experiences while having the *stability* to retain previous knowledge while encoding it. One prominent way to tackle this issue is to somehow save representations of old experiences without the availability of all past data. Various proposed techniques are regularization-based (Kirkpatrick et al., 2017; Li & Hoiem, 2017), replay-based (Rebuffi et al., 2017; Rolnick et al., 2019), or parameter-isolation (Serra et al., 2018; Rusu et al., 2022). IL can be performed in two settings: Class Incremental Learning (CIL) and Task Incremental Learning (TIL). The difference between the two settings is that for TIL the model incrementally learns a set of clearly distinguishable tasks, while for CIL the model must incrementally learn to distinguish between a growing number of classes. Replay-based techniques use a memory buffer to store a set of exemplars per class, which are later replayed while learning new tasks to alleviate CF. These techniques have shown the best performance in the CIL setting. They have shown a reduction in forgetting, but as the size of classes grows, forgetting cannot be prevented. Whereas parameter-isolation methods dedicate different model parameters to each task, notably used in the TIL setting. In recent years, Hard Attention to the Task (HAT) (Serra et al., 2018) applied to a multi-head model have shown the best performances in TIL settings. HAT (i) learns a set of almost binary task-specific masks to protect network parameters important for previous tasks, (ii) sets an objective to promote parameter sharing and sparsity, to exploit the maximum capacity of the network, and (iii) trains a task-specific classification head on top of the underlying feature representation given by the deep network. Since each task is learned on different parameters, forgetting is negligible. The con side of HAT is that it is designed for the TIL setting.

However, recent work in MORE (Kim et al., 2022) applied

HAT in an adapter module of the transformer layer combining it with OE, to give CIL functionality to a multi-head model. This is an important step towards non-forgetting CIL models. However, the use of buffer and outlier exposure in their implementation makes it a less attractive approach. By removing the buffer, we make the pipeline lightweight, and by removing OE, the model can operate in the wild.

3. Methodology

We introduce BUILD and explain here how our framework can be used to easily adapt a foundation model with the ability to detect OOD samples from unknown classes in an efficient and modular manner.

Model architecture. An overview at a high level is shown in Fig. 1 in the gray box on the left (Trained Model). The model is composed of (i) a fixed pre-trained backbone h with parameters $\boldsymbol{\theta}$, (ii) a set of task-specific trainable adapters with parameters $\boldsymbol{w}_t$ inserted at each transformer layer, and (iii) a set of classification heads f_t on top of the backbone, with parameters $\boldsymbol{\phi}_t$ for a given task t. For a given sample $\boldsymbol{x}$, the training of a task classification head t can be defined as: $\hat{f}_t = f_t(\boldsymbol{z}_t; \boldsymbol{\phi}_t)$, where $\boldsymbol{z}_t = h(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{w}_t)$ is the feature representation for task t.

Training Time. Once the architecture is defined we follow a multi-head training using the HAT approach Serra et al. (2018). To train the task classifier t, the training set $\mathcal{D}_t^{id}$ with c in-distribution classes: $\forall(\boldsymbol{x}, y) \in \mathcal{D}_t^{id}$ and $\{y_1, y_2, ..y_c\} \in \mathcal{Y}_t^{id}$ is used to update the parameters $\boldsymbol{w}_t$ and $\boldsymbol{\phi}_t$ while leaving the original backbone parameters $\boldsymbol{\theta}$ unchanged. We also note that BUILD does not expose outliers during training as done in MORE and other previous works (Kim et al., 2022; 2023b), nor does it use time-consuming solutions such as computing distance metrics on training samples and back-update on the classification heads.

Testing Time. At test time, the model can be subjected to either ID or OOD classes: $\boldsymbol{x} \in \{\boldsymbol{x}_{id}, \boldsymbol{x}_{ood}\}$ such that $\boldsymbol{x}_{ood} \in \mathcal{D}_k^{ood}$ where $k \neq t$, as expected in the wild. BUILD allows one to select any post-hoc OOD detector Γ mounted on top of the trained model $\hat{f}$ as shown in Fig.1 (Post-hoc OOD Detection). Therefore, at the detection time, the sample does not have task information and Γ detects for each head t, whether the sample is in-distribution to that task or not via $p(id|\boldsymbol{x})$.

Following our framework, we can choose Γ as an activation-based OOD detector, such as ReAct. In this case, Γ wraps the last part of the network starting from the penultimate layer of the task classifier t to extract activations $\boldsymbol{z}_t$. The activations are then processed by the task classifier to obtain the class prediction within that task: $p(y^c|\boldsymbol{x}_t^{id})$ and a confidence score.

4. Experimental Analysis

To show the validity of BUILD, we report an analysis of both its OOD detection capability and task-aware classification performance.

4.1. Experimental Settings

Baseline. We consider MORE as our baseline. Therefore, we replicate different parts of the original paper for better comparison.

Model. We use the following architecture: DeiT-S/16 (Touvron et al., 2021) with 2-layers adapter module (Houlsby et al., 2019) with 64-dimensional latent space at each transformer layer. We borrow the checkpoints of the backbone transformer from the baseline work where Vision Transformer(ViT) is pre-trained on ImageNet classes after removing 389 classes that are similar to the classes in CIFAR datasets to avoid data leaking at train time (Kim et al., 2022).

Dataset. We run our experiments on the CIFAR-10 image classification dataset composed of $50,000$ and $10,000$ training and testing samples, respectively, with $1,000$ samples per class. We consider a sequence of 5 tasks composed of 2 classes each without changing the original class order, e.g. t_0 comprise samples from class $0, 1$; t_1 samples with labels $2, 3$, etc.

Training details. We train the task-specific adapters, normalization layers, and classification heads using the stochastic gradient descent (SGD) optimizer for 20 epochs, setting the learning rate to 0.005 and the batch size to 64 for both settings. Additionally, for MORE, we set a buffer size of 200 and 10 epochs for the back-update procedure.

OOD scores. To compute the OOD scores we first use two non-post-hoc methods from the original baseline, i.e. using the Mahalanobis distance with and without the combination with the softmax output (which we respectively refer to as SMMD and MD). We also use three other methods more aligned with our framework BUILD, in particular (i) MSP, (ii) React with MSP (ReAct-MSP), and (iii) with the energy scores (ReAct-Energy).

Evaluation protocol and metrics. To evaluate the OOD detection performances, we consider for each task t all test samples belonging to tasks different than t as OOD and samples from t as ID. We then evaluate each task-specific classifier $\hat{f}_k$ at test time using the detector mentioned above. We measure the discriminative power of the OOD detector by evaluating the Area Under the ROC Curve (*AUROC*) and the false positive rate when the true positive rate is 95% (*FPR@95*). To test the TIL classification performances, we measure the In-Distribution accuracy (*ID_acc*) of each task-specific classifier on samples belonging to its task of specialization.

4.2. Results and Discussion

We expected BUILD to provide competitive performance compared to MORE, however, the experimental results showed that BUILD outperformed its baseline.

		MD	SMMD	MSP	ReAct-MSP	ReAct-Energy
FPR@95	MORE	46.01	**26.30**	65.52	59.30	58.13
	BUILD	**36.53**	34.72	**19.92**	**20.21**	**23.10**
AUROC	MORE	90.7	**95.19**	78.13	79.43	80.68
	BUILD	**92.33**	93.29	**95.13**	**95.23**	**95.54**
ID Acc.	MORE	98.54	98.54	97.31	97.17	97.17
	BUILD	**99.24**	**99.24**	**99.24**	**99.23**	**99.23**

Table 1. Performances of MORE and BUILD averaged over the five tasks classifiers.

Average performance. In Table 1, we show the AUROC, FPR@95, and ID accuracy, respectively, averaged over all five task classifiers. Notably, BUILD combined with post-hoc detectors gives lowest FPR@95 of 19.92 (MSP) outperforming the best MORE score of 26.3. BUILD also achieves a higher AUROC of 95.54 with the energy score in the ReAct setting, surpassing MORE's highest AUROC of 95.19. Only for SMMD MORE archives slightly better results for both, AUROC and FPR@95, however, in terms of ID accuracy, BUILD consistently exceeds MORE across all detectors.

Task-specific performance. In Fig. 2 we show the evaluations of each metric (y-axis) for each task-specific classifier (x-axis). We can see that BUILD can obtain strong detection and classification performance for all tasks, independently of the OOD detection method used. On the other hand, MORE presents highly imbalanced results (i) across different tasks, e.g., the AUROC of task 1 is much lower than the one measured on task 0, and (ii) across different OOD detectors, e.g., where all detectors that are not included in the initial design of MORE cannot compete with MD and SMMD.

Discussion. Our framework BUILD is not only lightweight and modular but also outperforms the state-of-the-art baseline. We observe that, in general, activation-based detectors can be the most promising approach to be used with BUILD, as both versions of ReAct show strong detection performance while maintaining good in-distribution accuracy. This is achieved thanks to its independent and modular design, while MORE is case-specific and works better with MD and SMMD, which are part of its design from the training to the testing phase. In particular, although using state-of-the-art activation-based OOD detectors together with MORE, their detection performance is damaged by its overall design. This behavior may be due to the activation vectors being influenced by the knowledge obtained by the OOD samples exposed during training, which promotes overfitting to a specific type of OOD distribution, while BUILD is not affected by this bias, as it does not use outlier exposure during training, hence the detectors are only fitted to the ID data. (ii) training without buffer Moreover, the modular design of BUILD allows to obtain more uniform results among different tasks, while the highly imbalanced results among different tasks obtained by MORE are probably caused by the random selection of samples stored in the buffer. We also observed that MORE presents a higher computational overhead and takes almost 20 times longer than BUILD. This is primarily due to the cost of inverting the covariance matrix to compute the Mahalanobis distance. MORE is also memory expensive, as large memory consumption comes from the memory buffer storing raw images of size $32 \times 32 \times 3$.

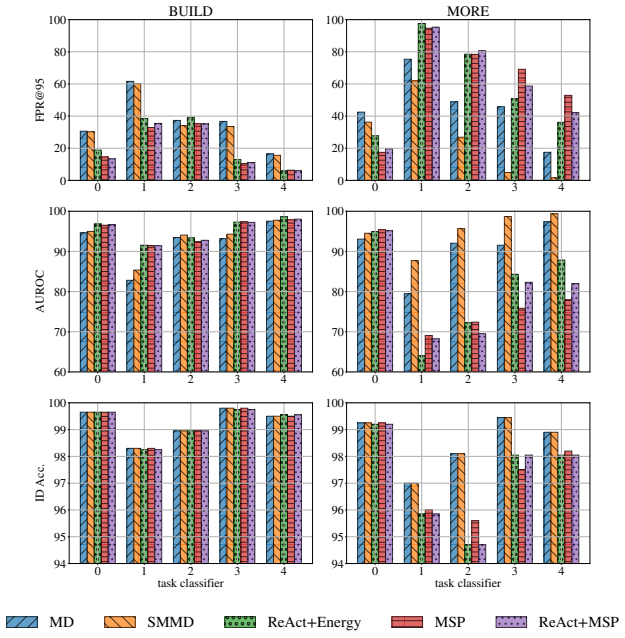

Figure 2. Individual performance of each task-specific classifier. In the columns the results from BUILD (right) and MORE (left). In the rows, from top to bottom: (i) the Area Under the ROC curve (AUROC) ↑, (ii) the False Positive Rate at 95% True Positive Rate (FPR@95) ↓ and (iii) the In-Distribution Accuracy (ID Acc.) ↑.

5. Conclusions, Limitations and Future Works

This paper proposes BUILD, a novel framework that allows ML models to work in the wild due to the combination of IL and OOD detection methods in a buffer-free fashion. BUILD learns new tasks by leveraging hard attention masking for training the adapter modules of a pre-trained network, but differently from previous works which make use of outlier exposure to provide OOD detection capability to the network, it exploits post hoc activation-based OOD detectors. This removes any dependence on the buffer

making the framework lightweight, modular, and computationally inexpensive. The resulting network outperforms the state-of-the-art baseline in the TIL setting.

Limitations Although our method is modular, efficient, presents competitive performances, and virtually prevents catastrophic forgetting due to the implementation of HAT, forgetting still occurs because classification becomes challenging with more classes.

Future works We envision an in-depth analysis of a wider range of OOD detectors, the inclusion of near- and far-OOD datasets, and a rejection mechanism for never-seen-before classes in future work. This would better highlight the modularity and flexibility of BUILD, demonstrating to the community the potential of a buffer-free approach.

Acknowledgments

This work has been conducted while Srishti Gupta and Daniele Angioni were enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with the University of Cagliari.

References

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Djurisic, A., Bozanic, N., Ashok, A., and Liu, R. Extremely simple activation shaping for out-of-distribution detection. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ndYXTEL6cZz.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hkg4TI9xl.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=HyxCxhRcY7.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty, 2019b.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp, 2019.

Kim, G., Liu, B., and Ke, Z. A multi-head model for continual learning via out-of-distribution replay. In *Conference on Lifelong Learning Agents*, pp. 548–563. PMLR, 2022.

Kim, G., Xiao, C., Konishi, T., Ke, Z., and Liu, B. Open-world continual learning: Unifying novelty detection and continual learning, 2023a.

Kim, G., Xiao, C., Konishi, T., and Liu, B. Learnability and algorithm for continual learning. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 16877–16896. PMLR, 2023b.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114 (13):3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Li, Z. and Hoiem, D. Learning without forgetting, 2017.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1VGkIxRZ.

Liu, W., Wang, X., Owens, J. D., and Li, Y. Energy-based out-of-distribution detection, 2021.

Perera, P. and Patel, V. M. Deep transfer learning for multiple class novelty detection, 2019.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning, 2017.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T. P., and Wayne, G. Experience replay for continual learning, 2019.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks, 2022.

Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M. H., and Sabokrou, M. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges, 2022.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Sun, Y., Guo, C., and Li, Y. React: Out-of-distribution detection with rectified activations, 2021.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 2021.

Weyssow, M., Zhou, X., Kim, K., Lo, D., and Sahraoui, H. On the usage of continual learning for out-of-distribution generalization in pre-trained language models of code. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE '23. ACM, November 2023. doi: 10.1145/3611643.3616244. URL http://dx.doi.org/10.1145/3611643.3616244.

Zhang, J., Yang, J., Wang, P., Wang, H., Lin, Y., Zhang, H., Sun, Y., Du, X., Zhou, K., Zhang, W., Li, Y., Liu, Z., Chen, Y., and Li, H. Openood v1.5: Enhanced benchmark for out-of-distribution detection, 2023.