BOWLL: A DECEPTIVELY SIMPLE OPEN WORLD LIFELONG LEARNER

Anonymous authors
Paper under double-blind review

ABSTRACT

The quest to improve scalar performance numbers on predetermined benchmarks seems to be deeply engraved in deep learning. However, the real world is seldom carefully curated and applications are seldom limited to excelling on test sets. A practical system is generally required to recognize novel concepts, refrain from actively including uninformative data, and retain previously acquired knowledge throughout its lifetime. Despite these key elements being rigorously researched individually, the study of their conjunction, open world lifelong learning, is only a recent trend. To accelerate this multifaceted field's exploration, we introduce its first monolithic and much-needed baseline. Leveraging the ubiquitous use of batch normalization across deep neural networks, we propose a deceptively simple yet highly effective way to repurpose standard models for open world lifelong learning. Through extensive empirical evaluation, we highlight why our approach should serve as a future standard for models that are able to effectively maintain their knowledge, selectively focus on informative data, and accelerate future learning.

1 INTRODUCTION

Modern deep learning are predominantly developed and assessed on carefully curated datasets, where information is accumulated from observed training inputs and a dedicated test set is meant to gauge so called "generalization" capabilities. Yet, one needs to only briefly imagine deploying such a pre-trained system into the real world to see that respective "unseen" data is unrealistic to fully capture through static methods. Take for instance a popular ImageNet model; which may struggle with numerous novel and previously uncaptured objects coming into sight, concepts' appearance differing drastically from the limited training data, or countless variability in conditions surrounding acquisition sensors and environmental conditions. Whereas our confined benchmark environments have thus historically enabled many of the initial algorithmic advances, the next stage of general system's life-cycles needs to account for both the transience and potential informativeness of future experiences in the world - both to guarantee robust deployment and ongoing efficient adaptation.

To satisfy requirements of real-world systems, recent works (Boult et al., 2019; Mundt et al., 2023) have postulated at least three essential criteria to take into account into machine learning model design: i) the model's ability to statistically differentiate between data-points that resemble the known training data and unexpected or unknown concepts; ii) the model's ability to actively query new informative data-points to involve in future training; iii) the model's ability to sequentially consolidate information obtained from newly queried data-points with past knowledge. In essence, each individual point is well known and has been extensively explored in the respective realms of: i) open-set recognition (Scheirer et al., 2013), ii) active learning (Settles, 2009), and iii) continual learning (Chen & Liu, 2018). Although their contributions seem entirely complementary, exploring their conjunction is a rather emerging avenue. Generally referred to as "open world learning", current works thus primarily portray a large focus on transcending traditional ML evaluation and establishing the setting's relevance. However, the latter is challenging due to the requirement of assessing three criteria in tandem and evaluating potential trade-offs between them in principled ways. Respectively, reaching consensus on evaluation and streamlining experimental comparison is largely still open.

With growing interests in open world applications, we posit that the field of open world learning is yet to develop its crucial baseline. There already exist promising approaches in the continual learning paradigm (De Lange et al., 2022), with GDUMB (Prabhu et al., 2020) being the most näive baseline that presents a worthy benchmark for non-stationary environments where tasks are introduced in temporal fashion. However, there does not exist a particular baseline that is common across open world learning works, not even within a specific application. Such a baseline is imperative to meaningfully measure performance and easy to implement in existing (neural) architectures and expand them to be open world learners. In this work, to remedy the lack of well suited baselines for evaluation, we introduce such a first monolithic baseline for open world learning. This baseline is derived from the insight that batch-normalization statistics (Ioffe & Szegedy, 2015), a commonly present component in most neural networks, can be repurposed to yield an intuitive open world learner. Our respective baseline BOWLL is designed with minimal assumptions and hence serves as a competitive experimental anchor when tested on a range of scenarios. As such, the statistics from the batch norm layer of a trained model serve as a tool that: distinguishes known unknowns from unknown unknowns, prioritizes acquisition of novel data, and minimizes catastrophic forgetting. We demonstrate that BOWLL can serve as the foundation for future work to build on, and quantify the contribution through several experiments. In summary, our contributions are as follows:

- Leveraging the ubiquitous use of batch-normalization, we introduce a first monolithic, easy to implement in existing models, baseline for open world lifelong learning: BOWLL.
- BOWLL consists of technically grounded mechanisms on the basis of the running mean and standard deviation of batch-norm. Using the latter as a simple Gaussian distribution, BOWLL enables robust prediction through measure of statistical deviation to flag unseen unknown data (the robust learner's perspective). It leverages batch-norm statistics to compute information density (Settles & Craven, 2008) to query novel data (the active learner's perspective). Finally, to balance the stability-plasticity (Jung et al., 2023), it also generates pseudo-inputs through deep inversion (Hsu et al., 2020) for rehearsal (Atkinson et al., 2018), alleviating catastrophic forgetting of past knowledge (the continual learner's perspective).
- Through a sequence of experiments we show that BOWLL learns tasks i) rapidly through few shots, ii) continually through diverse memory, iii) robustly through effective data choice.

2 Related Works

A closed world setting limits the performance of a model due to a constrained lifecycle that does not acknowledge the evolving nature of the world. In fact, the closed world approach adds adamant assumptions to the data available at training and testing, where task boundaries are well defined, data is free from distribution shifts, and minimal corruptions occur. This section discusses various approaches put forward to alleviate the problems of the closed world setting by exploring individual aspects of open world learning: out-of-distribution detection, active learning, and continual learning[1]. We also discuss the shortcomings that led to and necessitated the formulation of the BOWLL baseline. A comparison of important works that adopt individual or a subset of the three paradigms is given in table 1. In contrast to these techniques, BOWLL delivers a comprehensive baseline that is monolithically anchored in batch-norm statistics to allow models to learn in an open world.

Table 1: An overview of research works that contributed to resolving the challenges posed by closed world set-ups. The listed works take advantage of either of the three paradigms with configuration suitable for isolated training. GDUMB makes simple assumptions however it is not specifically designed for open world learning. BOWLL is a suitable baseline for open world application.

Technique	Open Set Recognition	Active Learning	Continual Learning
EWC (Kirkpatrick et al., 2017)			✓
GDUMB (Prabhu et al., 2020)		*	*
Towards Open World Object Detection (Joseph et al., 2021)	✓		✓
CLT (Arani et al., 2022)			✓
OpenVAE (Mundt et al., 2022b)	✓		✓
FoCAL (Ayub & Fendley, 2022)		✓	✓
BOWLL	✓	✓	✓

[1] We remark that the original definition of open world learning did not yet explicitly take into account catastrophic forgetting in its (accumulated) incremental learning step (Boult et al., 2019), instead accumulating data that is firs accepted and then successively queried. We thus add the word lifelong to further disambiguate terminology Mundt et al. (2023). We defer to the appendix A.1 for a detailed definition and further discussion.

Out of Distribution Detection: Out of Distribution Detection (OoD) (Yang et al., 2021), Mukhoti et al. (2021) is used to reject samples that are statistically different from the training distribution and make models predict more robustly. Open Set Recognition (OSR) (Scheirer et al., 2013; Geng et al., 2021) emerged as an idea to minimize the volume of unknown samples (unknown unknowns) outside the domain of known samples (known unknowns) by creating a "boundary" with the help of a measurable recognition function. Building on the idea of OSR, Bendale & Boult (2016); Mundt et al. (2019); Joseph et al. (2021) design algorithms to differentiate between known and unknown data instances. OoD is not an open-set classifier in itself, but can be incorporated as a detector to proportionately differentiate learnable labels. Such ideas are useful in open world applications, where a model faces corrupted and/or redundant data, minimally labelled data, data distribution shifts, and fluid task boundaries. In our framework, we design an outlier hypothesis based on the batch norm statistics to reject samples that largely deviate from the learned distribution. This hypothesis also enables to identify new classes without additional operations and let an oracle provide labels.

Active Learning: Active learning (Cohn et al., 1996) adapts an already trained model on new information by sampling novel data-points for future inclusion in the training process. One can determine the most informative points at the hand of an "acquisition function". That is, data-points that can reduce model uncertainty are desirable and/or points that can provide information about the pool of data i.e improve the generalization ability via mutual information (Guo & Greiner, 2007). The model is then trained from scratch by interleaving newly acquired data-points with the old instances, hence assuming the availability of the old dataset. One of the works that removes the dependency on availability of old data by synergizing an active learning strategy with continual learning includes Ayub & Fendley (2022). BOWLL formulates its *active query* on the basis of information density (Settles & Craven, 2008) to sample most informative points and then interleaves these with pseudo-points and (selective) past images of the previously trained dataset.

Continual Learning (Catastrophic Forgetting): A model undergoes catastrophic forgetting (McCloskey & Cohen, 1989) when acquiring new experiences from the data available at the current timestep, leading to a drastic performance drop on past experiences. Various research works on continual learning (CL) (Chen & Liu, 2018) have engineered three pathways to mitigate such forgetting, namely: i) regularization, ii) rehearsal, and iii) dynamic architectures. Farquhar & Gal (2019); De Lange et al. (2022); Wang et al. (2023) summarize these different methods. Another line of work harnesses complementary learning system (CLS) theory (Blakeman & Mareschal, 2020), which draws parallels between short-term/long-term memory and hippocampus/neocortex interactions to recreate the interplay between learned neural representations and a memory buffer (Arani et al., 2022). This helps to sustain the plasticity and stability (Jung et al., 2023) of the model. BOWLL employs the work by Yin et al. (2020) to generate pseudo-images of past data and further maintains a memory buffer of relevant data for training to enable stable performance over time.

3 Proposed BOWLL framework

Predominantly existing closed world training assumes the availability of past data and doesn't take into account factors such as any data distribution shifts, the existence of outliers, or potential redundancy of the data. Although the works of the related work section combat particular constraints of isolated setups, a lack of baselines to compare against and extract insights from poses a massive challenge to assess meaningful progress. Thus, we propose BOWLL as a cohesive formulation for open world lifelong learning that uses a single recurring component: batch normalization. §3.1 gives a brief overview of the interplay of different modules via the batch normalization statistics.

3.1 BOWLL at glance

Figure 1 depicts the information flow of BOWLL. We first wish to accept relevant data at the current timestep that feature resemblance to past training inputs of the model. This helps with minimizing unwanted interference and enforcing forward transfer while learning from data at the current timestep. The respective *OoD* module takes as input a stream of small batches of new data, and filters this stream via unsupervised out of distribution detection using population statistics maintained in the batch-norm layers. This is portrayed by the solid green line in the figure, which begins at the start of a new timestep. The accepted data-points then act as a "pool" for the *Active Query* module to procure data for the oracle to label. The purpose here is to acquire data-points that are dense in

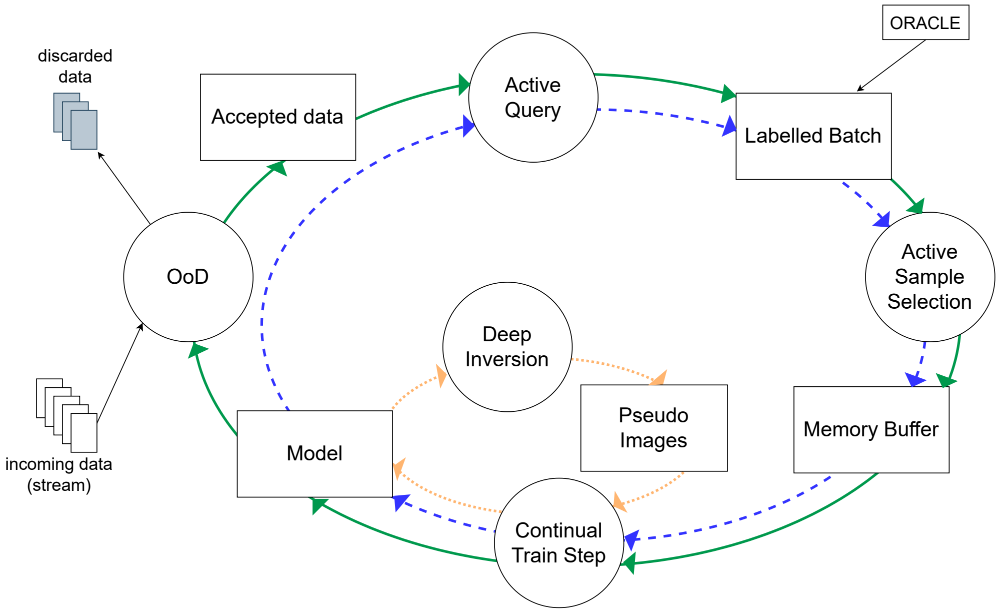

Figure 1: An illustration of the mechanics of BOWLL. The processes and objects are represented with circular and rectangular boxes respectively. The green line comes into play when the new dataset arrives and starts with discarding unimportant data. The OoD module(§ 3.3.1) rejects statistically extreme samples that interfere with already learned knowledge. The Active Query(§ 3.3.2) module intelligently selects informative data to be trained. The memory buffer makes mutual concession between past and new data-points using the same formulation in § 3.3.2. The Continual Train Step(§ 3.3.3) trains the Model on the data in the memory buffer and pseudo-images generated via Deep Inversion method. The blue dashed arrow depicts the BOWLL continual training regime with active querying on the "queryable pool" and populating the memory buffer with informative samples which runs until the pool is empty. The pseudo-data is generated at the beginning of the current timestep and is plugged into the continual train step. This is represented by the orange dotted arrow.

information and to enhance learning speed. The informativeness is quantified using entropy weighted with its similarity to other batches in the pool, to strike a trade-off between exploration and data relatedness. These samples are stored in a (short-term) memory buffer through an *Active Sample Selection* module, which replaces the least informative samples with the acquired labelled batch using the Active Query module's formulation. The goal here is to achieve stable performance on datasets in previous timesteps by a selective replacement strategy and maximize learning with novel data-points at the current timestep. To achieve competitive performance and retain representations learnt over a long time, we further generate class-conditioned pseudo-images using these representations. The pseudo-images act as a proxy to the already learned long-term features and are synthesized by Deep Inversion (Yin et al., 2020), i.e. using the running mean and running variance from the batch normalization layers, denoted by the orange dotted line. Finally, in the *Continual Train Step(s)*, the model is trained on data interleaved from the memory buffer and its generated images, before the buffer is once again updated by the *Active Query* module and the process is repeated until the queryable pool is exhausted. All individual modules are explained in detail in §3.3.1, 3.3.2 and 3.3.3.

3.2 Preliminaries: Batch Norm layers and Gaussian Uncertainty

A batch norm layer (Ioffe & Szegedy, 2015) uses an estimate of the mean μ_x and standard deviation σ_x of its input x to produce a shifted and scaled output y with mean β and standard deviation γ:

$$y = \gamma \frac{(x - \mu_x)}{\sqrt{\sigma_x^2 + \epsilon}} + \beta \tag{1}$$

where ϵ is some small constant, and γ and β are learnable parameters initialized to 1 and 0 respectively. The key property of the batch norm layer from our perspective is that, for every point in our

architecture with a batch norm layer, we automatically know the mean and standard deviation of the intermediate values at that point. This allows us to model these intermediate values x as being distributed according to a Gaussian distribution with mean μ_x and variance σ_x^2.
The batch normalization (BN) layer is an indispensable component in the majority of neural network architectures that improves learning and generalization by hypothesized internal covariate shift reduction (Ioffe & Szegedy, 2015). Although there may exist disagreement on the precise contribution of BN to optimization (Santurkar et al., 2018; Daneshmand et al., 2020; Schneider et al., 2020; Awais et al., 2021), the usefulness of the statistics derived from the BN layer remains unaffected.

3.3 The trifold nature of batch-norm for open world lifelong learning

3.3.1 Leveraging Batch-norm Statistics for Out of Distribution Detection

The *OoD* module is designed to reject extremely unusual data-points before they are placed into the queryable pool. This is useful because some of the data may be corrupted, or irrelevant, and we do not want to waste the capacity of our continual learner on trying to fit these data-points.
To determine if an input is unusual we can compare the intermediate values $\mathbf{x}^{(l)}$ at every batch norm layer l with their expected distribution $\mathcal{N}_{BN}^{(l)}$ according to the batch norm layer as described in § 3.2. The simplest sense in which the values $\mathbf{x}^{(l)}$ could be considered unusual is if they are low probability according to $\mathcal{N}_{BN}^{(l)}$. We could assign the activations a score η_0 given by

$$\eta_0 = \sum_l (\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\mu) \tag{2}$$

which is proportional to the negative log probability of $\mathbf{x}$ according to $\mathcal{N}_{BN}$ plus a constant. This is effectively a measure of whether or not the values $\mathbf{x}^{(l)}$ are larger in magnitude than expected.
This score is well motivated as a way of discarding outliers, and is effective at discarding outliers which produce very large intermediate values. Unfortunately, some data items we would like to discard produce intermediate values which are very small instead of very large. In order to address both cases simultaneously, we use a simple modification η_1 of η_0 given by

$$\eta_1 = \eta_0 - d \ln \eta_0 \tag{3}$$

where d is the dimension of $\mathbf{x}$. It can be seen that η_1 is large for both unusually large and unusually small intermediate values. We compute η_1 for each batch in the dataset and allow batches into the queryable pool if their corresponding value of η_1 is below some threshold τ.

In appendix A.2 we show how both η_0 and η_1 can be derived from the application of Bayes' theorem, but under different assumptions about the variance of intermediate values produced by outlying data. Specifically, η_0 corresponds to assuming that this variance is large with respect to that of the inlying data, whereas η_1 corresponds to assuming that it is simply unknown. We also briefly discuss on how we set the threshold τ that allows data into the queryable pool.

3.3.2 Leveraging Batch-Norm Statistics to Actively Query Informative Samples

The *Active Query* module queries the most informative samples from a candidate pool for future training of the model. We accept data-points from the pool that answer two questions: 1) How novel is the data-point? 2). How representative is the data-point of the pool? We design the acquisition function such that it respects the trade-off between exploration of novel data-points and exploitation through the use of similarity, using a combination of entropy of intermediate sample activations and similarity between the sample and the rest of the available data.

Given a queryable pool $D^Q = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{Q}$, for every data sample $\mathbf{x}_q$ we have the intermediate variance σ_q^2 from the BN layer. In the spirit of Settles & Craven (2008), we then calculate the novelty score γ_q as:

$$\gamma_q = \underbrace{\frac{1}{2}(1+\log(2\pi\sigma_q^2))}_{H(\mathbf{x}_q)} * \underbrace{\frac{1}{|Q|}\sum_{i=1,i\neq q}^{|Q|}\left[\frac{\mathbf{x}_i\cdot\mathbf{x}_q}{\|\mathbf{x}_i\|_2\cdot\|\mathbf{x}_q\|_2}\right]}_{\left[\cos(\mathbf{X}^{Q\setminus q},\mathbf{x}_q)\right]} \tag{4}$$

Here, γ_q is the score for sample $\mathbf{x}_q$, $H(\mathbf{x}_q)$ is the entropy, and the cosine term quantifies similarity between data-points. We thus have a strategy that leverages uncertainty via entropy and discourage the selection of unrelated data-points via cosine similarity. We provide an additional motivating example and detailed discussion to further highlight the necessity of both parts in appendix A.3.

3.3.3 Leveraging Batch-Norm Statistics to Alleviate Catastrophic Forgetting with a Complementary Learning System

The *Continual Train* module trains the model in a fashion that intends to avoid catastrophic forgetting while maximizing performance. Hence, we draw inspiration from complementary learning system (CLS) (Jung et al., 2023), where hippocampus and neocortex interact in the roles of short term memory, storing novel experiences, and long term memory, extracting representations that generalize over the consolidated data-points, respectively. In analogy to CLS, we adopt a dual memory recall system, where i) we maintain a memory buffer $\mathcal{F}$ to gather episodic information, ii) generate pseudo-images from stabilized past representations of the model.
For the former, i.e. the memory buffer at any timestep t, we have new points (X_t^q, Y_t^q) that are actively acquired from §3.3.2. Following the query's spirit, we sample high scoring points of size $|\mathcal{F}|$ and assign a score by modifying Eq. 4 as: $H(\mathbf{x}_f) * [\cos(\mathbf{X}^{(\mathcal{F}\cup X_t^q)\setminus f}, \mathbf{x}_f)]$. In this fashion, older, no longer necessary, parts of the memory may get replaced with novel information, whereas critical data points may prevail to maintain memory diversity and avoid completely flushing old tasks.
For the latter, i.e. the generated images to resemble already existing knowledge of the model, we use the trained model $\mathcal{M}$ to generate class-conditioned images as detailed in Deep Inversion (Yin et al., 2020). The auxiliary data is synthesized by optimizing on the loss l on initial random noise $\hat{x}$ while regularizing with respect to the batch normalization layers' stored running means and running variances of the pre-trained model (and thus past data):

$$\min_{\hat{x}} l(\hat{x}, y) + \mathcal{R}_{\text{prior}}(\hat{x}) + \mathcal{R}_{\text{feature}}(\hat{x}) \tag{5}$$

where:

$$\mathcal{R}_{\text{feature}}\,(\hat{x}) = \sum_l \|\mu_l(\hat{x}) - \mu_l\|_2 + \sum_l \|\sigma_l^2(\hat{x}) - \sigma_l^2\|_2 \tag{6}$$

and an (optional) prior $\mathcal{R}_{\text{prior}}$ can be employed to incorporate knowledge on the data modality:

$$\mathcal{R}_{\text{prior}}\,(\hat{x}) = \alpha_{\text{tv}}\mathcal{R}_{\text{TV}}(\hat{x}) + \alpha_{\ell_2}\mathcal{R}_{\ell_2}(\hat{x})\,. \tag{7}$$

Here μ_l and σ_l^2 represent the running mean and the running variance maintained by the batch normalization of the trained model. At the specific example of our later experiments' images, we adopt Yin et al. (2020)'s prior, composed of a total variation (TV) and L_2 distance term. In essence, these encourage adjacent pixels to be related in magnitude, discouraging noisy adversarial generation. The generated images thus contain previously extracted patterns that induce former contextual information while replaying during training, hence dampening forgetting. We refer to appendix A.4 for formal definitions of the priors and a discussion of conceivable future implementation alternatives.

Overall, the balanced combination of old and new data facilitates recall of past information while adapting the model to new information. We thus overcome the stability-plasticity dilemma by adopting a dual memory recall via a diverse memory and generated data.

3.4 Putting it all-together: BOWLL

In summary, in a traditional training regime, due to the sequential nature of the data, new and old information interfer, leading to forgetting of accrued knowledge. In addition, all new data, independently of its relevance and information content, is typically observed. In contrast, given a model previously trained on a dataset $D_0 = \{X_0^i, Y_0^i\}_{i=1}^N$ at timestep $t = 0$, the open world BOWLL baseline filters and actively selects important data from the incoming successive stream $\{X_t^i, Y_t^i\}_{i=1}^N$ at $t = 1, \ldots, T$ and proceeds to continually learn representations at overall competitive performance.

To this end, we have adopted continual learning strategies to preserve known knowledge while simultaneously actively grasping new semantics of recent data. Specifically, BOWLL's memory buffer $\mathcal{F}$ is populated with $\{X_0^i, Y_0^i\}_{i=0}^{|\mathcal{F}|}$ at random for $t = 0$ and gets updated with $\{X_t^q, Y_t^q\}$ after every active query step, where q is the index of the batch chosen by the *Active Query* module. At each training step, a batch of input X_t^i is then fetched from the memory buffer and interleaved with a batch of input $X_t^{\mathtt{syn}_j}$ from the generated synthetic images. To avoid unwarranted entanglement (Xie et al., 2020) between the generated and clean underlying distributions and balance training, the loss $\mathcal{L}$ is respectively composed of two cross-entropy (CE) losses: i) CE loss between one-hot encoding and predictions for $\mathcal{F}$ ii) CE loss between one-hot encoding and predictions for class-conditioned generated images. Overall, BOWLL thus actively queries novel related data, decides whether and which part of the memory buffer to replace to involve these instances in the learning step, and finally continually trains by mitigating forgetting on interleaved old and new instances: $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{CE}}(\sigma(f_\theta(X_t^i)), Y) + \lambda_2 \mathcal{L}_{\text{CE}}(\sigma(f_\theta(X_t^{\mathtt{syn}_j})), Y)$. For the sake of completeness, λ_1 and λ_2 are optional weights assigned to the individual loss terms, although their values have remained at a default of one in all of our experimentation. The complete algorithm is given in the appendix A.5.

4 Experimental Setup

We empirically demonstrate BOWLL's suitability as a baseline for open world learning. Importantly, we show that: i) BOWLL identifies and accommodates novel information rapidly, ii) BOWLL minimizes catastrophic forgetting when data distribution shift occurs, iii) BOWLL yields robust performance in realistic open world settings. To respectively highlight the latter, we investigate three set-ups: a) MNIST (Deng, 2012) → SVHN (Netzer et al., 2011) → DIGIT-5 (Peng et al., 2019a), each composed of differently appearing digits; b) split CIFAR-10 (Krizhevsky & Hinton, 2009), where five increments of disjoint class pairs are introduced sequentially; c) an extension of the latter to an open world, where datasets are now contaminated with corrupted instances (Hendrycks & Dietterich, 2019) and/or out-of-distributon data in the form of ImageNet (Deng et al., 2009).

Baselines and Metrics: We investigate five methods for comparisons against BOWLL, namely: 1) Joint (Hayes & Kanan, 2021): all datasets are available at the beginning and the model is trained from scratch, 2) Finetuned (Hayes & Kanan, 2021): datasets are introduced sequentially to mimick an incoming stream of data at different timesteps. Both of the above setups resemble traditional training. 3) GDUMB (Prabhu et al., 2020): although not particularly designed for open world learning, it is good contender against a plethora of continual learning approaches. Similar to BOWLL, GDUMB trains only on a memory buffer, but actively fills it at random. A suitable baseline to assess the quality of our active queries. 4) Experience Replay (ER) (Riemer et al., 2019; Rolnick et al., 2019): also mitigates forgetting through a random memory buffer, yet follows the conventional strategy to interleave it with all available current task data - a useful baseline to assess the focus on forgetting in our second experiment. 5) SoftO (Hendrycks & Gimpel, 2017): rejecting data on the basis of model Softmax confidence as a baseline for our third experiment with corrupted and OoD data.
We use several common evaluation metrics (Mundt et al., 2022a) to corroborate the performance of BOWLL: i) average accuracy - to assess the overall and final accuracy of the system, ii) learning curve area(LCA) - to determine the learning speed of the model after observing b mini-batches of data (and thus the quality of data queries) and its transferability (i.e. forward transfer in the case of small values of b), iii) number of (new) data points observed by the learner - to gauge the efficiency and data resourcefulness, iv) backward transfer (BWT) - to evaluate whether continued training harms prior tasks (negative BWT, synonymous to forgetting) or retrospectively improves them (positive BWT). We summarize mathematical definitions, datasets and training details in appendix A.6.

4.1 BOWLL learns rapidly with few shots

Table 2 shows the summary of results of BOWLL and other baselines on MNIST→ SVHN → USPS. Although the overall accuracy is not significantly higher, what should be taken into consideration is the learning pace and the number of samples the methods trains on. On the one hand, BOWLL delivers 88.54±1.44% accuracy with only 8508 samples placed into the memory buffer. The remaining data at disposition is excluded beforehand and thus never included in any model optimization. As such, only about 5% of the overall dataset is observed in contrast to regular training. On the other hand, the Bayesian-based *OoD* module ensures that the samples don't disrupt the already learned representations and facilitates smooth transfer of information for training at current timestep. This can be observed from the positive backward transfer (BWT) values, which signify the retrospective knowledge transfer from timestep $t = 1$ to $t = 0$, indicating that the information gained from future timesteps doesn't overwrite past information. Finally, the LCA_β demonstrates "β-shot" learning

Table 2: BOWLL achieves competitive accuracy on MNIST→ SVHN → USPS, at a more rapid learning speed (LCA) and significantly fewer observed training samples (total number of unique data points that have been placed in the 5k large memory buffer over time) than other settings.

Method	$LCA_{\beta}(\uparrow)$			BWT(↑)	# of new data points used in training	Accuracy(↑)
	β=1	β=5	β=10			
Joint	16.55±0.00	30.74±0.21	38.89±0.11	n/a	158257	97.74±0.05
Finetuned	42.24±0.13	69.16±0.12	77.93±0.10	-13.28±0.02	158257	89.66±0.07
GDUMB	40.60±0.11	68.56±0.10	77.32±0.09	-5.63±0.02	10000	87.33±0.15
BOWLL	67.82±1.24	75.67±0.45	77.70±2.62	0.86±0.00	8508±581	88.54±1.44

capability after training on only $\beta = 1, 5, 10$ mini-batches. Here, it can be seen that BOWLL delivers significantly higher learning speed even with few examples than for instance the popular GDUMB, which also only trains on a memory buffer but fills it at random. This can primarily be attributed to the active query of most informative data-points at early stages, but also learning from a set of pseudo-images that acts as proxy for past representations to enable high forward transfer.

Overall, the experiment demonstrates that BOWLL successfully mitigates forgetting and highlights BOWLL's active component to significantly speed up learning at a lower consumed amount of data.

4.2 BOWLL learns continually with equitable memory

Next, we evaluate BOWLL for incremental addition of new class labels on Split CIFAR-10 dataset with a ResNet-18 model. Figure 2 illustrates the demographics of the memory buffer for Split CIFAR-10 by GDUMB and BOWLL. GDUMB randomly select data-points with a balanced class distribution constraint. Hence, the population dynamic in the memory buffer remains same. This can put a limit on learning as the number of new samples is restricted, constrained to the past, and there is no notion of the informativeness of samples. In figure 3 we can see that the accuracy of GDUMB deteriorates more heavily as new class labels are added over-time. In case of BOWLL, the active query mechanism to acquire and replace informative data-points in the memory buffer yields population diversity and thus a less forgetful model. In fact, BOWLL is competitive with the ER baseline in terms of forgetfulness, despite the latter first training on all data before populating the memory buffer. We also study the effect of including generative images, where BOWLL* denotes training the model only on the memory buffer by discarding the second term in the loss. Figure 3 records a $\sim 3\%$ boost in performance when trained collectively on the memory buffer and generated

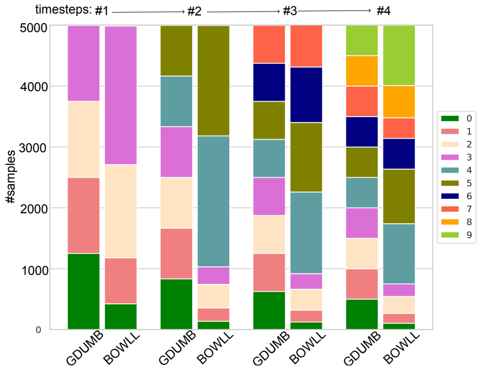

Figure 2: Demographics of $\mathcal{F}$ using BOWLL and GDUMB on Split CIFAR-10 dataset. BOWLL accrues and regulates diverse data-points, whereas GDUMB adopts a uniform policy to populate the memory buffer. Variants like ER or the follow-up baseline DER (Buzzega et al., 2020) (that includes logits) sample the memory equally randomly.

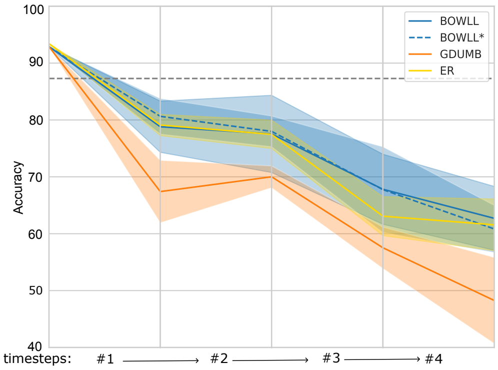

Figure 3: BOWLL obtains significantly better accuracy than GDUMB on Split CIFAR-10. It features large accuracy at any point in time, that is further improved when trained collectively on the diverse memory buffer and generated images. Although ER trains on all available data and maintains a similarly sized memory of old tasks, BOWLL is competitive by learning exclusively from data in the memory buffer.

images. We discuss the entailed trade-off between accuracy gain and computational effort in appendix A.4. In appendix A.8, we also demonstrate the contribution of each module via ablation.
Overall, the experiment highlights BOWLL's effective balance of stability-plasticity through dual memory recall of generated pseudo-images and diverse memory of informative samples.

4.3 BOWLL LEARNS ROBUSTLY WITH EFFECTIVE DATA CHOICE

To investigate BOWLL's performance in a full open world setting, we contaminate Split CIFAR-10 with corrupted images (Hendrycks & Dietterich, 2019) and additional out-of-distribution data from a subset of ImageNet (Deng et al., 2009). Figure 4 summarizes the final test accuracy on the clean Split CIFAR-10 after training on corrupted data. It is evident that BOWLL dominates the performance of other techniques. This is credited to the application of the trifold nature of the batch-norm statistics that enables a robust performance in the open world setting, further observable by the rapid inclusion of new information in figure 5. It shows that the learning trend of GDUMB is not stable, pertaining to its random memory buffer strategy that stores noisy, corrupted and irrelevant data. ER suffers even more, as it first learns from all data. Even though SoftO can reject some of the OoD data, it in turn suffers from forgetting in open world learning. In contrast, BOWLL's exhibits superior performance, as its *OoD* module rejects erroneous samples, the *Active Query* module prioritizes informative samples, and in conjunction with the continual step leads to an increasing learning curve. Further training details and complete comparison are given in appendices A.7 & A.9.
Overall, the experiment highlights that BOWLL is able to discard irrelevant data, selectively focus on important information only, and continually learn from it - making its monolithic batch-norm based nature the perfect contender for a well-rounded open world lifelong learning baseline.

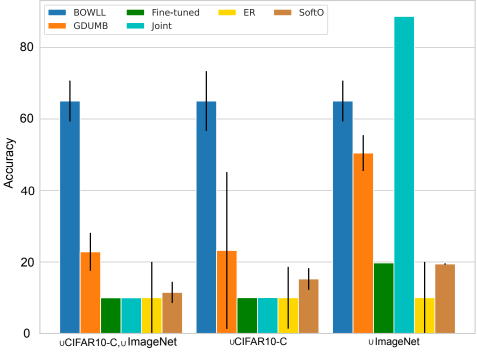

Figure 4: Final Accuracy on Split CIFAR-10 corrupted, tested on numerous corruptions types, and in the presence of ImageNet. BOWLL is considerably more qualified for these open world applications than other training setups that focus on single aspects alone.

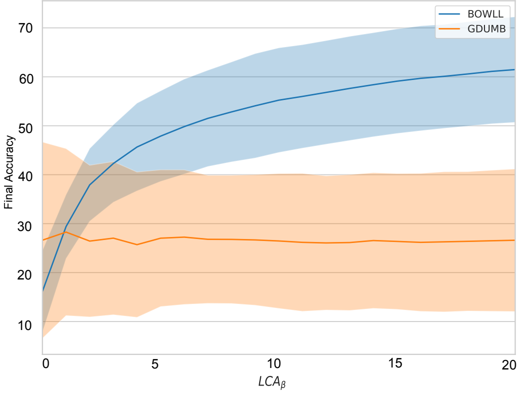

Figure 5: LCA_β trend for BOWLL and GDUMB on Split-CIFAR10 corrupted. GDUMB stagnates, whereas the components of BOWLL enforce better data choices and thus warrant robust learning. ER is not displayed, as it is even worse than GDUMB.

5 CONCLUSION

We have introduced a simple monolithic baseline for open world learning: BOWLL. BOWLL uses the statistics from the batch normalization layer to seamlessly tether the three paradigms of OoD detection, active learning and continual learning. It detects outliers to minimize disruption of already learned knowledge, achieves high learning speed by sampling data using active queries, and mitigates forgetting by adopting insights from complementary learning systems. We demonstrated that BOWLL is a few-shot, stable, and robust learner through empirical evaluations that include scenarios such as distribution shift, incremental class encounters, and a realistic open world. Owning to the use of a common component i.e the BN layer, BOWLL can thus be seen as an exceptional baseline with enticing prospects (listed in more detail in appendix A.10). In summary, we anticipate future works to both adopt more accurate implementations of the present batch-norm's simplified Gaussian, as well as broad application to other tasks, such as regression or clustering. Overall, we envision BOWLL to help the community make the first meaningful assessments in open world lifelong learning.

REFERENCES

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022.

Craig Atkinson, Brendan McCane, Lech Szymanski, and Anthony Robins. Pseudo-recursal: Solving the catastrophic forgetting problem in deep neural networks, 2018.

Muhammad Awais, Md. Tauhid Bin Iqbal, and Sung-Ho Bae. Revisiting internal covariate shift for batch normalization. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11): 5082–5092, 2021.

Ali Ayub and Carter Fendley. Few-shot continual active learning by a robot. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Sam Blakeman and Denis Mareschal. A complementary learning systems approach to temporal difference learning. *Neural Networks*, 122:218–230, 2020. ISSN 0893-6080.

T. E. Boult, S. Cruz, A.R. Dhamija, M. Gunther, J. Henrydoss, and W.J. Scheirer. Learning and the unknown: Surveying steps toward open world recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9801–9807, Jul. 2019.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15920–15930. Curran Associates, Inc., 2020.

Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3), 2018.

D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996. doi: 10.1613/jair.295.

Hadi Daneshmand, Jonas Kohler, Francis Bach, Thomas Hofmann, and Aurelien Lucchi. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18387–18398. Curran Associates, Inc., 2020.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning, 2019.

Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3614–3631, 2021.

Yuhong Guo and Russ Greiner. Optimistic active learning using mutual information. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pp. 823–829, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

Tyler L. Hayes and Christopher Kanan. Selective replay enhances learning in online continual analogical reasoning, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations (ICLR)*, 2017.

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*, 2015.

K J Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5830–5840, June 2021.

Dahuin Jung, Dongjin Lee, Sunwon Hong, Hyemi Jang, Ho Bae, and Sungroh Yoon. New insights for the stability-plasticity dilemma in online continual learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. *Computer Vision and Pattern Recognition (CVPR)*, 2015.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip HS Torr, and Yarin Gal. Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *arXiv preprint arXiv:2102.11582*, 2021.

Martin Mundt, Iuliia Pliushch, Sagnik Majumder, and Visvanathan Ramesh. Open set recognition through deep neural network uncertainty: Does out-of-distribution detection require generative classifiers? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

Martin Mundt, Steven Lang, Quentin Delfosse, and Kristian Kersting. CLEVA-compass: A continual learning evaluation assessment compass to promote research transparency and comparability. In *International Conference on Learning Representations*, 2022a.

Martin Mundt, Iuliia Pliushch, Sagnik Majumder, Yongwon Hong, and Visvanathan Ramesh. Unified probabilistic deep continual learning through generative replay and open set recognition. *Journal of Imaging*, 8(4), 2022b. ISSN 2313-433X.

Martin Mundt, Yongwon Hong, Iuliia Pliushch, and Visvanathan Ramesh. A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. *Neural Networks*, 160:306–336, 2023. ISSN 0893-6080.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality, 2019.

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 Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 2019.

Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. Latent replay for real-time continual learning. *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019a.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019b.

Ameya Prabhu, Philip H. S. Torr, and Puneet K. Dokania. Gdumb: A simple approach that questions our progress in continual learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 524–540, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58536-5.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *In International Conference on Learning Representations (ICLR)*, 2019.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, 2018.

Walter J. Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7): 1757–1772, 2013.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11539–11551. Curran Associates, Inc., 2020.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pp. 1070–1079, USA, 2008. Association for Computational Linguistics.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application, 2023.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *The IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2020.

A APPENDIX

The appendix complements the main body with additional details on:

- A.1 The required elements and a loose definition of the term open world learning
- A.2 A derivation of BOWLL's out of distribution score
- A.3 A detailed motivation of BOWLL's active query function
- A.4 Mathematical details of Deep Inversion priors and conceivable alternatives
- A.5 Pseudo-code algorithm of BOWLL
- A.6 Mathematical details of all evaluation measures
- A.7 Data sequences and training hyper-parameters used in experiments
- A.8 BOWLL ablation studies
- A.9 Additional results for experiment C's open world setting
- A.10 A discussion n limitations and prospects

A.1 THE ELEMENTS OF OPEN WORLD LIFELONG LEARNING

In this section, we provide a more detailed intuition behind the components of open world learning, and motivate our use of the term's extension to open world *lifelong* learning. Using natural language to ease understanding, Bendale & Boult (2016) define open world recognition (learning) as:

> "In open world recognition the system must be able to recognize objects and associate them with known classes while also being able to label classes as unknown. These "novel unknowns" must then be collected and labeled (e.g. by humans). When there are sufficient labeled unknowns for new class learning, the system must incrementally learn and extend the multi-class classifier, thereby making each new class "known" to the system. Open World recognition moves beyond just being robust to unknown classes and toward a scaleable system that is adapting itself and learning in an open world." - *Bendale and Boult, Towards Open World Recognition*

Without quoting the entire formal statement, (please refer to Bendale & Boult (2016) Definition 1), the mathematical description can intuitively be summarized as:

1. An *open set recognition function* that involves a novelty detector to determine whether any result from the recognition function is from an unknown class.
2. An *active labelling function*, typically a human oracle in supervised learning, to label any unknown data points and, if necessary, extend the set of existing known classes.
3. An *incremental learning function* that trains the system by adding the new data points and respectively continuing to train the recognition function.

BOWLL follows the general concept of open world learning and provides the first monolithic baseline to empower experimental progress and transparent comparison. As such, it further disambiguates above definition in places where the exact implementation allows for a vague interpretation.
In favor of generality, BOWLL thus encapsulates a broader interpretation of steps two and three, in the spirit of the desiderata of lifelong learning (Mundt et al., 2023).

Specifically, in the active labelling function of step 2, BOWLL includes an actual *active learning query*. In other words, rather than simply labelling all data that the out of distribution detector has detected as unknown, the active query step further gauges *informativeness* and *relatedness of the data*. This is important, because the lifelong open world learner should leverage examples that are remotely related to what the objectives are, rather than introducing any sorts of novel points. If we think of the novelty detector giving a high score to arbitrary unseen noise, it is important to gauge whether inclusion of this example is expected to reduce prospective loss. In turn, such inclusion of an active learning query further significantly reduces the amount of (redundant) data used for continuous training of open world learning system.

In addition, and most importantly, we place the strict requirement on the incremental learning step to avoid concatenation of data. In fact, existing algorithms that tackle some form of open world learning (Joseph et al., 2021) place a large emphasis on the novelty detector and active data inclusion, yet they continue extending the dataset in the spirit of traditional active learning. In the spirit of continual learning and overall real-world system plausibility, we avoid retention of all old data and thus need to combat any expected catastrophic forgetting (McCloskey & Cohen, 1989).

In the easiest sense, our three above points thus remain the same, but are augmented with the specific challenge of gauging data informativeness and mitigating common catastrophic interference. To respectively disambiguate our work and the BOWLL baseline from approaches that may simply concatenate queried unknown examples with existing known data, we have included the additional "lifelong" in the terminology *open world lifelong learning* in emphasis of the continual aspect.

A.2 Derivation of Out of Distribution Score

In this appendix we show how decision rules for outlier detection based on η_0 and η_1 The log probability density function of the multivariate t-distribution with mean μ, variance $\frac{\nu}{\nu-2}\mathbf{\Sigma}$, degrees of freedom ν, and dimensionality d is as follows:

$$\ln P(\mathbf{x}) = -\ln\Gamma\left(\frac{\nu+d}{2}\right) - \ln\Gamma\left(\frac{\nu}{2}\right) - \frac{1}{2}\ln\det(\nu\pi\mathbf{\Sigma}) - \frac{\nu+d}{2}\ln\left(1+\frac{1}{\nu}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right) \tag{8}$$

As a belief about future observations of $\mathbf{x}$ this corresponds to assuming that $\mathbf{x}$ is distributed according to a multivariate normal distribution, the mean and covariance of which were estimated from ν observations of previous $\mathbf{x}$s. In the limit $\nu \to \infty$ we recover the normal distribution

$$\ln P(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) + C \tag{9}$$

where C is a normalizing constant and does not depend on $\mathbf{x}$.

If we perform a Bayesian hypothesis comparison between the inlier hypothesis "the intermediate activations $\mathbf{x}$ are drawn from a normal distribution with mean μ and covariance $\mathbf{\Sigma}$" and the outlier hypothesis "the intermediate activations are drawn from a uniform distribution", we obtain a posterior log odds ratio of

$$\eta_0 = \frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) + C_0 \tag{10}$$

in favour of the outlier hypothesis, where C_0 includes both a normalizing constant and the prior log odds in favour of the outlier hypothesis.

The observed failure mode of using large values of η_0 in a decision rule for outlier detection is that outlying data which produces anomalously small intermediate activations is accepted as inlying. One way of interpreting this is to note that our inlier hypothesis corresponds to very high confidence that the covariance of activations is $\mathbf{\Sigma}$, (the limit $\nu \to \infty$ corresponding to having estimated $\mathbf{\Sigma}$ from arbitrarily many prior samples of $\mathbf{x}$). If we now ask what our outlier hypothesis corresponds to, we see that it corresponds to very high confidence that outliers produce activations with a very large covariance. (That is, the limit of equation 9 as $\mathbf{\Sigma}$ becomes large is the uniform distribution). From this point of view it is reasonable that our decision rule malfunctions for outliers which have anomalously small intermediate activations: we have implicitly assumed that such outliers do not exist.

Intuitively we want our outlier hypothesis to instead say something like "the intermediate activations are drawn from a normal distribution with a covariance about which we are highly uncertain". We can achieve this by taking the opposite limit of equation 8 to that which constructed equation 9, $\nu \to 0$. Doing so produces an unnormalized (and indeed, like the uniform distribution, unnormalizable) log probability density of

$$\ln P(\mathbf{x}) = -\frac{d}{2}\ln\left((\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right) \tag{11}$$

If we now perform Bayesian hypothesis comparison with equation 9 as the inlier hypothesis and 11 as the outlier hypothesis, we obtain a posterior log odds ratio of

$$\eta_1 = \frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) - \frac{d}{2}\ln\left((\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right) + C_1 \tag{12}$$

in favour of the outlier hypothesis, where C_1 is again the combination of a normalizing constant and the prior log odds ratio in favour of the outlier hypothesis. The reader will observe that, unlike η_0, η_1 is large for both large and small magnitude values of $(\mathbf{x} - \mu)$.

We set the threshold τ using the bootstrap method (Nalisnick et al., 2019). Although for the initial setting a held-out validation dataset is used, in our case we use the samples in the memory buffer. Similar to Nalisnick et al. (2019), we sample K 'new' data sets $X_{k=1}^{|\mathcal{F}|}$ of size M from the memory buffer $\mathcal{F}$ and then calculate $\eta_1 k$ for the kth bootstrap set. We then calculate α-quantile over the $\eta_1 k$ distribution to get the overall threshold estimate τ.

A.3 Motivation of the Query Function

The purpose of the query function is to select data from the pool to be labelled and used in training the model. We would like to select data whose labels will be most informative about the task(s) the model is being optimized to solve. There are two major reasons it might be less worthwhile to label a particular datum. Firstly, the datum might be too similar to other data for which we already have labels. Secondly, it might be unusual enough that it is mostly unrelated, not only to the data we already have labelled, but also to any data we expect to need to predict labels for in the future (e.g. at test time). We attempt to capture these two failure modes with two heuristics.

For the first heuristic α_q, we use the following quantity:

$$\alpha_q = \frac{1}{2}(1 + \log(2\pi\sigma_q^2)) \tag{13}$$

where σ_q is the standard deviation across pixels, channels and layers of the post-batch-norm intermediate activations of the network when given the input $\mathbf{x}_q$, i.e.

$$\sigma_q^2 = \frac{1}{|L|}\sum_{l\in L}\frac{1}{|C_l|}\sum_{c\in C_l}\frac{1}{|P_l|}\sum_{p\in P_l} a_{lcpq}^2 \tag{14}$$

where L is the set of layers and C_l and P_l are the sets of channels and pixels for layer l, and a_{lcpq} is the post-batch-norm activation at pixel p, channel c and layer l when the input to the network is $\mathbf{x}_q$. This is a measure of how spread out the intermediate activations are relative to their distribution on the existing labelled data, since the batch norm layer normalizes its inputs with respect to the statistics of that data. We use this measure of dissimilarity in intermediate activation space as a proxy for dissimilarity in terms of what information their labels would give us about the task.

For the second heuristic β_q we use the following normalized inner product in input space:

$$\beta_q = \frac{1}{|Q|}\sum_{i=1, i\neq q}^{|Q|}\left[\frac{\mathbf{x}_i \cdot \mathbf{x}_q}{\|\mathbf{x}_i\|_2 \cdot \|\mathbf{x}_q\|_2}\right] \tag{15}$$

where Q is the queryable pool and $\mathbf{x}_q$ is the datum we are evaluating. This quantity captures a notion of similarity averaged over the queryable pool. It can be seen to reach a value of 1 for the case where $\mathbf{x}_q$ is identical to every other $\mathbf{x}_i$, and a value of 0 for the case where $\mathbf{x}_q$ is orthogonal to them. Taking the pool to be a proxy for the kinds of data we want the model to perform well on, this addresses the second failure mode.

To combine these two heuristics, we multiply them, giving the query score formula from equation 4 of the main text, which we reproduce below:

$$\gamma_q = \alpha_q * \beta_q = \frac{1}{2}(1 + \log(2\pi\sigma_q^2)) * \frac{1}{|Q|}\sum_{i=1, i\neq q}^{|Q|}\left[\frac{\mathbf{x}_i \cdot \mathbf{x}_q}{\|\mathbf{x}_i\|_2 \cdot \|\mathbf{x}_q\|_2}\right] \tag{16}$$

where γ_q is the query score for datum $\mathbf{x}_q$. The choice of multiplying the heuristics does not immediately require us to assign a weight to the two heuristics, and we did not find it necessary to look for a way to introduce any such weighting.

A.4 Deep Inversion and Conceivable Alternative Implementations

We first provide the mathematical definitions of the priors used in the adopted Deep Inversion and their intuition, before discussing the nature and reasoning behind the employed replay of inputs. Finally, we also provide a brief comment on the computational trade-off and the gained accuracy.

Deep Inversion priors: Yin et al. (2020) adopt an optional data prior term in their Deep Inversion algorithm, repeated here for convenience from the main body:

$$\mathcal{R}_{\text{prior}}\left(\hat{x}\right) = \alpha_{\text{tv}}\mathcal{R}_{\text{TV}}(\hat{x}) + \alpha_{\ell_2}\mathcal{R}_{\ell_2}(\hat{x}) \tag{17}$$

In the specific context of images, they employ prior insights from Mahendran & Vedaldi (2015) to use a norm (generally an α norm, but here set to 2) on the image values and a natural additional regularizer to encourage piece-wise consistent image areas. Respectively, the former is to ensure that the range of the generated data point remains in the typical range of values expected in 0 to 1 normalized image:

$$\mathcal{R}_{\ell_2}(\hat{x}) = ||\hat{x}||_2^2 \tag{18}$$

In principle, this is to avoid that the inversion algorithm generates any arbitrary valued inputs to the system that simply satisfy batch-norm statistics.

In similar spirit, the latter total variation (TV) regularizer tries to avoid the creation of unmeaningful noisy and adversarial inputs that are tailored to fit batch-norm statistics. Mahendran & Vedaldi (2015) formulate this in a general case, but the variant employed by Yin et al. (2020) and BOWLL in its image experiments accounts for the discrete nature of images of fixed width i and height j. $\mathcal{R}_{\text{TV}}$ can then be expressed as another norm on the basis of finite differences between neighboring pixels:

$$\mathcal{R}_{\text{TV}}(\hat{x}) = \sum_{i,j} \left[(\hat{x}_{i,j+1} - \hat{x}_{i,j})^2 + (\hat{x}_{i+1,j} - \hat{x}_{i,j})^2)\right]^{\beta/2} \tag{19}$$

Following the prior authors' intuition to avoid "spikes" in the interpolation of values we adopt $\beta + 1$ and similarly do not search over the weighting factors α of the individual priors.

Rationale behind replaying inputs and feature-based alternatives: In BOWLL we have chosen to employ Deep Inversion in its original form by optimizing a synthetic input to have minimal divergence to expected batch norm statistics and then replaying the conceived generated data point. The rationale behind this choice is simple, BOWLL is intended to be not only a monolithic open world lifelong learning baseline, where each component leverages batch normalization, but also one that is highly competitive yet straightforward to implement. As such, we believe the success of a competitive baseline lies in beating various contenders despite its simplicity. Synthesizing generated inputs and rehearsing them in raw form, in turn allows concatenation with the data points that the complementary real memory buffer contains in the continual training steps.

However, we do acknowledge that the reader may at this point wonder if other variants, that perhaps directly employ the feature space are plausible. At present, we preserve BOWLL's competitive simple nature, yet point to works such as Pellegrini et al. (2020), that have shown the efficacy of "latent replay" variants, for completeness. Here, more involved mechanisms may enable potentially improved performance by rehearsing and regularizing various latent states. The respective insights are complementary to BOWLL's formulation and we anticipate future works to construct more complicated open world learning variants. For the latter, BOWLL now provides the first natural baseline for rigorous experimental comparison.

Accuracy-compute trade-offs: We remark that Deep Inversion is by far the most computationally expensive component of BOWLL. However, the main body's figure 3, where we evaluate a variant of BOWLL termed BOWLL* without the Deep Inversion part, also demonstrates that Deep Inversion provides the final 3% on top of already phenomenal performance. As even BOWLL* is an exceptional baseline, it is important to quantify the computational gain by stripping away Deep Inversion.

Overall, the OoD step is computationally almost negligible. Here, there is barely any overhead to a traditional forward-pass, as batch-normalization layers are calculated independently. The subsequent active learning component, operating already on a subset of the data is slightly more involved, yet still comparatively cheap. Again, the batch norm values are yielded by default, and as such calculating the entropy term is a negligible overhead. The cosine similarity term is slightly more involved, as in principle the current considered data points needs to be compared to a set of other data points. However, as data points are independent the respective involved matrix multiplication is fully parallelizable for any reasonably sized data inputs. In fact, we suspect extremely large input size to become a detriment to any neural network model before calculation of a cosine similarly becomes problematic. The population of the memory buffer adopts a variant of this active query equation, as described in the main body, however is computationally bounded by the fact that we typically desire a small memory and thus the amount of comparisons are strictly limited. In principle, again, if the memory buffer were to become extremely large, the computation time may be impacted, however, in that scenario the open world learning idea is essentially relaxed to the traditional isolated regime where all data is always present, defeating the original purpose of BOWLL.

This leaves Deep Inversion as the final element, as the actual training step on the memory buffer is synonymous to the conventional loop of a forward-backward pass over mini-batches in the memory buffer. In our experience the number of update iterations on the initially random $\hat{x}$ to align with batch-norm statistics has practically required hundreds, up to 2000 or 3000 steps (CIFAR) of updates in Deep Inversion to achieve meaningful convergence (orange loop in figure 1). However, we also note that we only generate as many synthetic examples as there are real examples in the memory buffer (in our cases 5000) in a one to one balance of components. In principle many more generated examples could be used and would likely improve the system further (at expense of compute). In fact, continual learning methods relying on generative rehearsal of data (Mundt et al., 2023; 2022b) typically conceive as many examples as are in the original dataset to achieve competitive performance. In light of the fact that even without Deep Inversion, BOWLL beats GDUMB significantly (figure 3) and retains its open world learning capability in its OoD detector and active query component, we leave the decision to the user on whether a mild gain in accuracy at significant expense in compute is required by the particular application.

A.5 BOWLL Algorithm

Algorithm 1 gives the pseudo-code for BOWLL. We executed all the experiments on a single Nvidia Tesla A100 or V100 GP. We use PyTorch (Paszke et al., 2019) for implementing the BOWLL framework. We also provide the source code in the supplementary material.

Algorithm 1 BOWLL pipeline

```
Input: Trained model $f_{\theta_{t-1}}$, Data $\{X_t^i, Y_t^i\}_{i=1}^N$ where $t = 1, 2, \dots, T$, acquisition batch size $\mathcal{B}$
Initialize: Memory Buffer: $\mathcal{F} \leftarrow \{X_0, Y_0\}$
for each timestep $t$, $\{X_t^i, Y_t^i\}_{i=1}^N$ do
    $X_t^{\text{syn}}, Y_t^{\text{syn}}$ = DeepInversion($f_{\theta_{t-1}}$, $|\mathcal{F}|$)
    Perform: $\{X_t^p, Y_t^p\}$ = OoD($f_{\theta_{t-1}}$, $X_t$, $\tau$)
    while pool $D^Q = \{X_t^p, Y_t^p\}_{p=1}^{|Q|}$ is not empty do
        $X_t^q, Y_t^q$ = ActiveQuery.runAcquisition($D^Q$, $\mathcal{B}$)
        Update memory buffer by replacing old samples with newly acquired data samples: $X_t, Y_t$ = ActiveQuery.runAcquisition($\mathcal{F} \cup \{X_t^q, Y_t^q\}$, $|\mathcal{F}|$)
        Train: ContinualLearning($f_{\theta_{t-1}}$, $\{X_t^{\text{syn}}, Y_t^{\text{syn}}\}$, $\{X_t^i, Y_t^i\}_{i=1}^{|\mathcal{F}|}$)
    end while
end for
```

A.6 Evaluation measures

We primarily use three metrics to evaluate and compare the performance of BOWLL, as sketched in the main body, in addition to reporting the overall number of data points used for training. We point to Mundt et al. (2022a) for an overview of relevant metrics considered in continual learning, and repeat the definitions for our respectively used ones here.

Average Classification Accuracy: We measure the performance of a model on the cumulative test dataset after learning on the train data at each incremental timestep. At each timestep t, after training the model on train dataset D_t^{train}, we calculate the accuracy a_t of the model on the test dataset D_t^{test}. The final average accuracy after T timesteps is given as:

$$A_T = \frac{1}{T} \sum_{t}^{T} a_t \qquad (20)$$

In context of BOWLL, average classification accuracy provides a good estimate of the overall system's balance between encoding new information while maintaining prior learned knowledge.

Learning Curve Area (LCA): LCA determines the learning speed of the model after being trained on batch b. At each timestep t, after training on batch b sampled from D_t^{train}, we record the accuracy $a_{t,b}$ on the test dataset D_t^{test}. Respectively, the learning curve at T under β is calculated as:

$$LCA_\beta = \frac{1}{\beta + 1} \sum_{b=0}^{\beta} \left(\frac{1}{T} \sum_{t=1}^{T} a_{t,b,t} \right) \tag{21}$$

Intuitively, the values of $\beta = 1, 5, 10$ highlight critical learning phases. $\beta = 1$ is synonymous with one-shot learning, directly assessing the system's capability for forward transfer and immediate adaptation. The values of 5 and 10 respectively highlight whether the chosen examples are informative and the system thus rapidly improves in quality. In context of BOWLL, LCA aids in assessing the quality of the active query, as prioritizing highly informative samples is expected to result in the most rapid reduction of the loss.

Backward Transfer (BWT): BWT measures the effect on performance of the model for previously learned data at timestep $< t$ when training on the data of the current timestep t:

$$BWT_t = \frac{1}{t - 1} \sum_{j=1}^{t-1} (a_{t,j} - a_{j,j}) \tag{22}$$

It is useful to differentiate observations of positive and negative BWT values. If a negative BWT value is experienced, the former performance deteriorates. Negative BWT is thus synonymous with forgetting. If the value is close to 0, then no adverse influence is experienced. In contrast, if a positive BWT value is observed, current training provides retrospective improvement to what has been learned before. The latter may be the case if subsequent data shares resemblance or tasks are related. We note that positive BWT seldom occurs in existing practical algorithms (Mundt et al., 2023), as many continual learning methods rely on various forms of constraints or regularization that aim to preserve existing representations. Although positive BWT still requires much future research, we believe effective memory management, such as in BOWLL, to be one key element.

A.7 Dataset Sequences and Training Hyper-parameters

A.7.1 MNIST→ SVHN → USPS

We use AlexNet (Krizhevsky et al., 2012) with batch normalization layers to train on MNIST (Deng, 2012) (60k tiny grayscale images resized to 32×32 split across 10 classes for the digits 0-9). We train using the Adam (Kingma & Ba, 2017) optimizer with a learning rate of 0.001 for 60 epochs. We first adapt the model to SVHN (Netzer et al., 2011) (a real-world analogue of MNIST with 32×32 resolution cropped digits in the form of house numbers) at timestep $t = 1$ and then DIGIT-5 (Peng et al., 2019b) (a collection of different appearing digits from 0-9 based on a combination of MNIST, MNIST-M, Synthetic Digits, SVHN, and the USPS datasets) at timestep $t = 2$. The memory buffer size is set to $|\mathcal{F}| = 5000$ and is initialized with random samples from the MNIST training. For OoD we use a mini-batch size of 4 and samples in memory buffer act as the validation dataset for setting the threshold τ via bootstrap sampling. After the OoD step, BOWLL delivers a queryable pool. BOWLL queries a mini-batch of size of 256, which then replaces least informative samples in the memory buffer. We use DeepInversion to generate pseudo-images of previously trained dataset. We generate $|\mathcal{F}|$ in total, with 500 images assigned to each class. We continually train on the memory buffer and the generated images for 1 epoch and repeat the process for all the samples in the "queryable pool". We report results wit standard deviations over 5 runs. In contrast to the original setup of GDUMB and ER, where a static model uses a masking strategy at the output nodes to deal with varying number of class labels, in our experiments, we modify the model's output nodes to grow when it encounters new class labels for the sake of fair comparison.

A.7.2 Split CIFAR-10

We divide CIFAR-10 (Krizhevsky & Hinton, 2009) (consisting of 50k color images equally balanced across 10 classes of 32×32 resolution) into 5 disjoint timesteps (classes $\{2, 5\} \to [0, 6] \to [1, 7] \to [3, 8] \to [4, 9]$) at timestep $t = 0 \to t = 1 \to t = 2 \to t = 3 \to t = 4$ i.e Split CIFAR-10. We use a ResNet-18 (He et al., 2016) to train on data with labels that come under $\{\ldots\}$ braces with two output nodes for classification at timestep $t = 0$. We use SGD with a learning rate of 0.1, a momentum value of 0.9 and weight decay of 0.0005 and train for 120 epochs with mini-batch size $b = 256$. This training is performed under traditional setup. For BOWLL, the memory buffer is filled with random samples from the training data (availability of old training data is assumed only at this initial point in time). The model first discards irrelevant data at timestep $t = 1$ with batch size $b = 8$. To enable open world learning of the model to the new data available at incremental timestep $t > 0$, we expand the number of output nodes of the classifier to the number of classes detected at timesteps t by the number of unique class labels detected in the queryable pool. The expanded model is then used for continual learning on the upcoming disjoint tasks. The accepted data is available to actively query and learn from. For one loop of active query, acquired batch of size 256, supersedes unimportant samples in the memory buffer. The model is then trained in continual fashion using the same configuration as stated earlier but for 2 epochs. The active query loop iterates until the queryable pool is vacant. We measure the test accuracy at the end of training for each timestep on classes seen thus far. We report results wit standard deviations over 5 runs.

A.7.3 Presence of OoD and corrupted data: CIFAR-10-C and ImageNet

CIFAR-10-C(Hendrycks & Dietterich, 2019): We apply 3 corruptions namely: impulse noise, Gaussian noise, shot noise to all the classes of CIFAR-10 (Krizhevsky & Hinton, 2009) dataset. Out of the 75 available noise types, we chose these 3 as they simulate common noise additions that occur more often that can affect model performance when trained on. Gaussian noise especially can be more harmful depending on their severity. We generate 50000 images for each noise type equally balanced across all classes. The image meta-information such as width, height and channel remain the same. **ImageNet**(Deng et al., 2009): We use a subset of the ImageNet dataset with labels $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ and about 1000 images per class. We discard remaining subset. These clean images from ImageNet act as the inputs which the model needs to discard in order focus on relevant "in-distribution" data.

We then interleave the corrupted data(CIFAR-10-C) and the subset of ImageNet dataset with the original CIFAR-10 dataset to simulate data for open world learning.

A.8 Ablation Study

We perform experiments to demonstrate the contribution of each module to the entire OWLL framework against employing each module in isolation under a closed world training environment. The ablation study is done on Split CIFAR-10 with the ResNet-18 model as detailed in previous section. We show the results of the ablation study in Table 3. We make the following observations: **i)** For, BOWLL without the OoD module the BWT is higher than that of BOWLL suggesting that OoD does help in discarding data-points that interfere with previously learned representations. **ii)** BOWLL without the Active Learning (AL) module, fails to deliver competitive performance over longer incremental timestep. **iii)** BOWLL without the Continual Learning (CL) module clearly performs the worst with lowest final accuracy and BWT. **iv)** Complete BOWLL achieves consistent accuracy and minimizes forgetting, as reflected in BWT.

In summary, the OoD module discards data that can interfere and potentially disrupt past learned representations, the active query procures data-points that are maximally informative about the incoming data stream and the dual memory backed continual training delivers an intricate balance between previously encoded knowledge and judicious maintenance of past and latest samples in the memory buffer. Above the conceptual contributions, our ablation study quantitatively demonstrates that each module contributes meaningfully to overall performance in open world learning.

A.9 Open world setting

Figure 6 demonstrates the LCA_β curve on Split CIFAR-10 combined with only CIFAR10-C data and figure 7 demonstrates the LCA_β curve on Split CIFAR-10 combined with subset of ImageNet. BOWLL outperforms GDUMB in both settings that simulate the open world learning environment. These additional individual experiments further support that BOWLL is effective when either corrupted related data or unrelated data is presence, nicely complimenting in supporting the main body's insight in the experiment with both kinds of data being present at once.

Table 3: Ablation study of BOWLL on Split CIFAR-10 reporting test accuracy at the end of every timestep and Backward Transfer(BWT) at the end of timestep i.e t=4. “x” indicates that the module is removed and “✓” indicates otherwise. Study(iv) represents the complete BOWLL framework.

Study	Modules			Accuracy				BWT(↑)
	OoD	AL	CL	t=1	t=2	t=3	t=4	
i)	x	✓	✓	37.19±0.54	50.18±0.75	56.14±0.96	58.29±7.04	-31.42±2.20
ii)	✓	x	✓	80.05±5.26	77.53±5.18	60.05±5.75	50.55±7.40	-40.98±3.05
iii)	✓	✓	x	65.15±6.93	31.54±6.77	28.91±5.49	19.52±7.85	-68.97±3.06
iv)	✓	✓	✓	78.80±4.50	77.54±6.78	67.81±6.15	62.70±5.61	-15.11±0.58

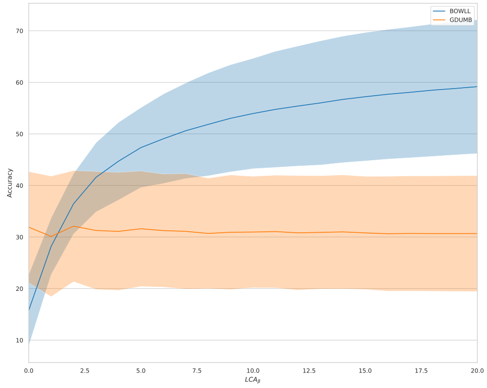

Figure 6: LCA_β trend of BOWLL on Split CIFAR-10 corrupted with noisy CIFAR10-C.

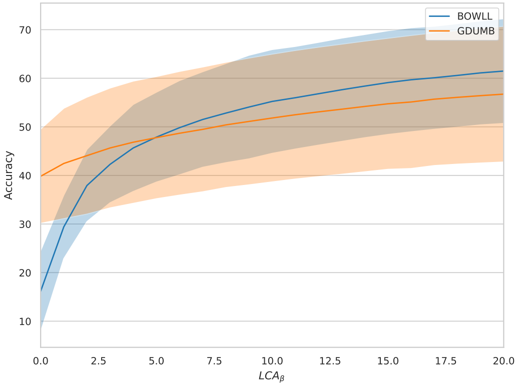

Figure 7: LCA_β trend of BOWLL on Split CIFAR-10 corrupted with subset of ImageNet.

A.10 Limitations and Prospects

We provide additional discussions on the limitations and future prospects for BOWLL. BN layers normalize the pre-activations of a layer using the statistics from mini-batches of data to yield zero mean, unit variance and diagonal covariance. We emphasize that the diagonal covariance assumption is a simplification (pertaining to de-correlations), yet BOWLL delivers a strong simple monolithic baseline using just the diagonal values. We list other limitations as follows:

Quality and computation of Deep Inversion: The Deep Inversion module relies on the BN outputs to synthesize reliable image representations for the model to further train on. Hence one aspect of knowledge transfer from previous timesteps to the current one is conditioned on competitive model performance. The pseudo-data from Deep Inversion help with mitigating forgetting and since BOWLL trains on both memory buffer and pseudo-images higher quality synthesized images are beneficial. One particular aspect to ensure the latter is the choice of a data modality prior, in the case of images total variation. Although Deep Inversion can technically be used for any data type, as it simply aims to match BN statistics, finding such priors can significantly aid in performance, for instance by assuring that neighboring pixels are correlated in images. As such, if BOWLL is applied to other data modalities, respective similar priors are likely necessary for the Deep Inversion module (see also appendix A.4 for a discussion on computational complexity and removing Deep Inversion altogether) More details on the limitations of Deep Inversion pertaining to data synthesis time and quality can be found in (Yin et al., 2020).

Memory buffer size: As in all memory-based continual learning methods, BOWLL requires a good estimate on the size of the memory buffer to store data. At present, all approaches, including BOWLL, thus motivate buffer size from a storage and compute constraint and make a choice a priori. The memory requirement can however increase or vary with time and one can investigate methods to dynamically allocate memory, depending on both how informative novel data is and the expected performance of the model. The a priori choice is thus a current limitation, requiring some intuition of task complexity, and the dynamic extension of memory size an enticing future prospect.

Generalization and application beyond supervised open world learning: Already in the current form, BOWLL's OoD module detects the outlier on small batch size in an unsupervised fashion and without any prior training on out-of-distribution data, using only population statistics maintained in the batch-norm layers. Similarly, the active query module acquires novel samples using entropy obtained from the BN layer activations weighted with data similarity. This again is done without any information about the label or any form of supervision. Finally, the dual memory for continual training formulated using pseudo-images and a memory buffer for storing informative samples is also driven purely batch-norm statistics. In essence, one can thus trivially extend BOWLL to unsupervised learning, reinforcement learning, or other prediction tasks, such as e.g. semantic segmentation. We thus expect BOWLL to fuel further development of open world lifelong learning beyond the supervised examples in this paper.