Computer Life
ISSN: 1819-4818 | Vol. 13, No. 1, 2025

Open-World Class Incremental Learning with Adaptive Threshold

Han Luo *

School of Artificial Intelligence, Sun Yat-sen University, Zhuhai 519080, China
* **Corresponding author Email:** luoh226@mail2.sysu.edu.cn

Abstract: Existing class incremental learning (CIL) settings are mainly based on the closed-world assumption. Its training and testing sets only contain in-distribution (ID) samples. However, CIL methods are typically applied in open-world scenarios where out-of-distribution (OOD) samples are widely present, leading to unpredictable behavior of intelligent agents. The OOD categories may also change as the incremental tasks progress. In this paper, we focus on handling a realistic and challenging new setting called Open-World Class Incremental Learning (OWCIL). Specifically, OWCIL includes the accumulated OOD categories during testing to simulate the open-world continual learning scenario. Moreover, it does not provide any OOD samples for model training. Existing methods integrate techniques for CIL and OOD detection to address scenarios similar to the OWCIL. However, they either fail to handle evolving OOD categories or require OOD samples during training, limiting their performance under OWCIL. We propose an adaptive threshold (AT) method to handle the accumulated OOD categories. Additionally, a discriminative optimization method is introduced to determine the threshold without OOD samples. The effectiveness of our proposed method has been validated through extensive experiments on multiple benchmark datasets under OWCIL. Detailed analysis shows that both AT and discriminative optimization can clearly boost performance.

Keywords: Class incremental learning; Open world; Out-of-distribution detection.

1. Introduction

In recent years, continual learning (CL) [1-3] has attracted more and more attention from the research community. Class incremental learning (CIL) [4-9] is one of the important CL topics. CIL enables intelligent agents, e.g., deep neural networks (DNN), to incrementally learn new classes from streaming data while maintaining the knowledge of old classes. To further simulate the real-world constraints, e.g., privacy regulations and memory limitations, no old class samples can be preserved and replayed at the new task learning. It is known as exemplar-free class incremental learning (EFCIL) [10-13]. Existing CIL settings are based on the closed-world assumption, where the training and testing sets only contain in-distribution (ID) samples. In real-world scenarios, the agent usually encounters unexpected (out-of-distribution, OOD) categories along with the provided ones at new task learning, leading to cumulative increases in both ID and OOD categories. In this work, we introduce a new CIL setting called open-world class incremental learning (OWCIL). OWCIL requires that the training set does not contain any OOD samples, while the OOD categories are added cumulatively during testing to simulate a more realistic incremental learning scenario in the open world. The OWCIL setting is depicted in Fig. 1, where the training set of each task does not contain OOD samples, and the test set of Task 2 includes not only the new OOD class ”birds” but also the old OOD class ”sharks” from Task 1.

Existing closed-world CIL approaches fail to handle the OOD samples from OWCIL. For example, the trained CIL model 1 in Fig. 1 incorrectly predicts a shark (the OOD sample) as a dog. Therefore, the incremental learning methods should be capable of handling the OOD samples and taking responsible actions when deployed in open-world scenarios, especially the safety-prioritized applications such as autonomous driving. Traditional OOD detection methods [14-16] assume the range of ID and OOD classes remain static. They fail to handle the dynamic scenarios in OWCIL. Other settings such as continual out-of-distribution detection (COOD) [17-19] and open world recognition (OWR) [20-22] combine OOD and CIL. OWCIL differs significantly from them in the following aspects. On the one hand, both out-of-distribution (OOD) samples and old class exemplars are present during the learning process in the COOD setting. This allows for direct use of the OOD samples and old class exemplars from the training set to optimize an OOD threshold. However, in OWCIL, there are no OOD samples in the training set and old class exemplars are not available, so the existing COOD methods cannot be applied. On the other hand, the OWR setting excludes OOD samples during training but uses fixed OOD samples across different tasks during testing. Therefore, the OWR methods use static thresholds for OOD detection. On the contrary, OOD categories in OWCIL are accumulated along incremental tasks, and a fixed threshold would lead to sub-optimal results. Moreover, the exemplars of old ID classes are available in both the COOD and OWR settings, while OWCIL is under more challenging exemplar-free restrictions. Comparisons between OWCIL and the relevant settings are summarized in Table 1.

In the OWCIL setting, two main obstacles are recognized. In the context of determining the out-of-distribution (OOD) threshold without having OOD samples during training, current methods either rely on the empirical value of TPR95 as the threshold [23], or they randomly select in-distribution (ID) categories as OOD for optimization. However, these methods fail to identify the distinct thresholds between ID and OOD. Inspired by the concept of hard sample mining, we propose a method that does not rely on out-of-distribution (OOD) samples. Instead, we use misclassified in-distribution (MID) samples as pseudo OOD samples to optimize the OOD threshold. This approach selects samples that are generally located at the classifier edge, resulting in a more discriminative threshold compared to existing methods. Furthermore, as tasks progress and the OOD categories increase cumulatively, the corresponding OOD threshold also changes. Using a fixed threshold approach, e.g., TPR95, will lead to sub-optimal results. Therefore, we propose an adaptive threshold (AT) algorithm to optimize the OOD threshold at each incremental learning stage to accommodate the cumulative classes. The contributions of this paper are summarized as follows:

- ❖ We propose the open-world class incremental learning (OWCIL), a realistic continual learning setting. Existing methods fail to handle this challenging problem.
- ❖ A threshold optimization method is proposed to tackle the challenge of the lack of OOD samples, delivering superior performance compared to existing threshold optimization methods.
- ❖ An adaptive threshold (AT) is proposed to handle the dynamic OOD thresholds during testing. It is compatible with existing OOD threshold optimization methods and clearly boosts their performance.

Extensive experiments and analyses are conducted to demonstrate the effectiveness of the proposed method under the challenging OWCIL setting.

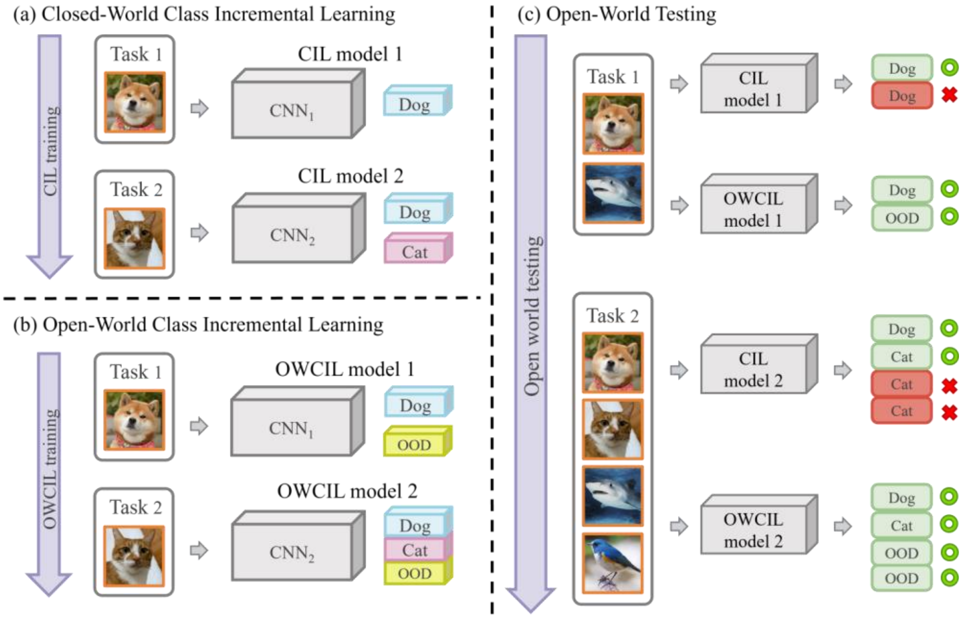

Fig. 1 Comparison between the closed-world CIL method and the proposed open-world CIL method.

Table 1. Comparisons between OWCIL and the relevant settings.

	Closed train set	Open test set	Accumulated OOD classes	Exemplar-free
CIL	√	×	×	×
EFCIL	√	×	×	√
OOD	×	√	×	√
COOD	×	√	×	×
OWR	√	√	×	×
OWCIL	√	√	√	√

2. Related Work

2.1. Class Incremental Learning

CIL methods [24, 25] aims to equip the deep neural networks (DNNs) with the continual learning ability to recognize the emerging classes in the dynamic environment. However, DNNs suffer from severe performance degradation of old classes after learning from the new task data. This is the key challenge of CIL and is known as catastrophic forgetting [26, 27]. To handle the forgetting issue, the proposed CIL methods can be categorized into three types: (1) Memory-based methods [28, 29] preserve the images and feature representations of old categories and replay them at the training of new tasks to defy forgetting; (2) Regularization-based methods [30, 31] constraint the updates of model features or parameters to retain the model knowledge of old classes. Techniques, such as distillation [32, 33], are used; (3) Isolation-based methods [34] decouple the parameters in the model and then assign different weights for learning different tasks during the incremental learning process. Among these methods, memory-based ones are widely adopted as the backbone and combined with other techniques to alleviate catastrophic forgetting. However, exemplars are usually not available in restricted real-world scenarios.

EFCIL prohibits the direct storage of exemplars from old classes. Therefore, EFCIL is significantly more challenging than its counterpart settings. Existing EFCIL settings usually assume a large initial task, i.e., half of the total classes, or models pre-trained on large datasets are available. The knowledge from model weights and feature statistics of previous tasks are still available. Knowledge distillation between old and new models is widely adopted by existing EFCIL methods. Pseudo-features of the old classes can also be sampled based on the old class statistics, e.g., class means and covariances, to compensate for the missing exemplars. Both EFCIL and the proposed Open-World CIL (OWCIL) settings are exemplar-free at incremental learning, but neither the EFCIL nor CIL considers the OOD cases during testing. Therefore, our OWCIL better simulates real-world scenarios than its counterpart CIL settings and becomes more challenging.

2.2. Out-of-Distribution Detection

The OOD samples can be detected based on a simple post-hoc approach, which utilizes the maximum softmax probability as the indicator score for in-distribution (ID) and OOD samples. The energy scores, as another post-hoc method, are built upon the theoretical explanations from the likelihood perspective for OOD detection. Such post-hoc methods require neither OOD samples for model training nor modification of the vanilla training procedure and objective. Therefore, they have been widely adopted in real world applications where the computational overhead of retraining is prohibitive [35]. Under some circumstances, the OOD samples are available for model training. It is also known as outlier exposure [36]. Generally speaking, OOD detection with outlier exposure can achieve superior performance. However, no OOD sample is available for model training under the proposed OWCIL setting. Therefore, the outlier exposure methods do not conform to our setting. Furthermore, all existing OOD detection methods focus on static scenarios and cannot dynamically update the range of ID categories, making them unsuitable for the OWCIL task.

Continual out-of-distribution detection (COOD) is proposed to address OOD detection tasks in the context of continual learning. In [37], the normalized classifier weights and information entropy are used to adjust predicted probabilities for detecting OOD samples. Exemplar sets are leveraged to train an additional and independent classifier for the model, aiming to detect OOD samples in each task, as done by [38]. Evidence deep learning is adopted to measure the uncertainty of samples, enabling OOD detection in incremental learning scenarios. A different paradigm is proposed in [39]. During training, the model learns the PCA transformation matrix specifically for ID samples. This leads to larger reconstruction errors for OOD samples during testing, facilitating OOD detection. However, it requires memory and pre-trained models. However, COOD and our OWCIL are two settings and their main differences are as follows. (1) COOD setting focuses on the OOD detection performance under continual learning. Its CIL performance is separately measured under the closed-world scenario with ID test data only. On the contrary, OWCIL focuses on the CIL performance under the open-world scenario, where both ID and OOD samples appear during testing. (2) Existing COOD methods focus on the overall performance of OOD detectors across different thresholds. However, under the OWCIL setting, a threshold is required to distinguish between ID and OOD samples in order to get final classification results. Moreover, empirical OOD thresholds are often suboptimal. It is essential to find an optimal OOD threshold in OWCIL. (3) OOD samples are required by most COOD methods for model training. However, no OOD sample is available in the OWCIL training set. Therefore, OWCIL is considered as a more challenging and realistic scenario than COOD. Most COOD methods are not applicable to OWCIL problems.

2.3. Open World Recognition

The OWR setting focuses on adding object categories incrementally while detecting outliers with the Nearest Non-Outlier (NNO) method proposed. However, the OOD samples in the test set remain the same for every task. Moreover, the OOD samples are not included in the training set. It randomly selects some classes as OOD and optimizes the OOD threshold by maximizing the F1 score under the condition that ID recall is greater than 90%. EVM is an exemplar-based method. The OOD samples are detected based on the distance between the test sample and its nearest exemplar neighbour from the memory, which is not feasible under the OWCIL setting. Some scholars simulates streaming data without task segmentation. When OOD samples are detected, they are incrementally learned once a certain number is reached. It also uses TPR95 as the OOD threshold. A recent work, further generalizes OWR to the variable-few-shot setting and focuses on CIL rather than OOD detection. OWCIL differs from OWR in two main aspects. First, in OWR, the OOD classes and samples in the test set are fixed, whereas in OWCIL, the classes are cumulatively increased. Second, OWR can use memory for auxiliary learning, while OWCIL prohibits the use of memory.

3. Open-World Class Incremental Learning

3.1. OWCIL Setting

Existing CIL research mainly focuses on enabling the incremental learning capabilities of intelligent agents within a closed world. However, incremental learning usually occurs in dynamic environments of realistic scenarios, where both ID classes and OOD classes are incrementally added. Therefore, the proposed Open-World Class Incremental Learning (OWCIL) setting has the following characteristics:

• The training and validation sets of OWCIL do not include out-of-distribution (OOD) samples. It is hard to anticipate and incorporate all potential OOD classes in advance. To simulate such uncertainty in real-world scenarios, we restrict the training and validation sets of OWCIL from containing OOD samples.

• The OWCIL test data includes out-of-distribution (OOD) samples. These OOD classes, similar to the ID ones, are accumulated over tasks. Therefore, the agent needs to handle both old and new OOD classes and distinguish them from the ID classes.

• Storing old class exemplars is prohibited under the OWCIL setting due to privacy concerns and storage limitations in the realistic scenario. Therefore, alleviating the catastrophic forgetting issues under the exemplar-free setting becomes one of the main concerns of OWCIL methods.

3.2. OWCIL Pipeline

A corresponding pipeline is proposed to handle the OWCIL setting. It consists of three stages: incremental learning, adaptive threshold optimization, and inference, as illustrated in Fig. 2 and detailed below.

3.2.1. Incremental learning

The incremental learning stage can be expressed as a series of N training tasks and corresponding training sets $\{D_1, D_2, \ldots, D_N\}$ without overlapping classes, where $D_n = \{(x_i, y_i)\}_{i=1}^{M_n}$ is the n-th training set with M_n training instances. $x_i \in \mathcal{X}$ is a picture and $y_i \in \mathcal{Y}_n$ is the corresponding label. Following the CIL setting, the label spaces of different tasks are disjoint $\mathcal{Y}_n \cap \mathcal{Y}_{n'} = \emptyset$ for $n \neq n'$ and we can only access training data from D_n in n-th training task. The ultimate goal of the n-th training task is to fit a classification model for all seen classes $f_n(x): \mathcal{X} \rightarrow Y_n$, where $Y_n = \cup_{i=1}^{n} \mathcal{Y}_i$, which minimizes the expected risk:

$$f_n^* = \arg\min_{f_n \in \mathcal{H}} E_{(x,y) \in D_n} I(y \neq f_n(x)), \quad (1)$$

where $\mathcal{H}$ is the hypothesis space. $I(\cdot)$ is the indicator function which outputs 1 if the expression holds and 0 otherwise. D_n denotes the training set of the n-th task. An ideal OWCIL model satisfying Eq. (1) and works well in both old and new classes.

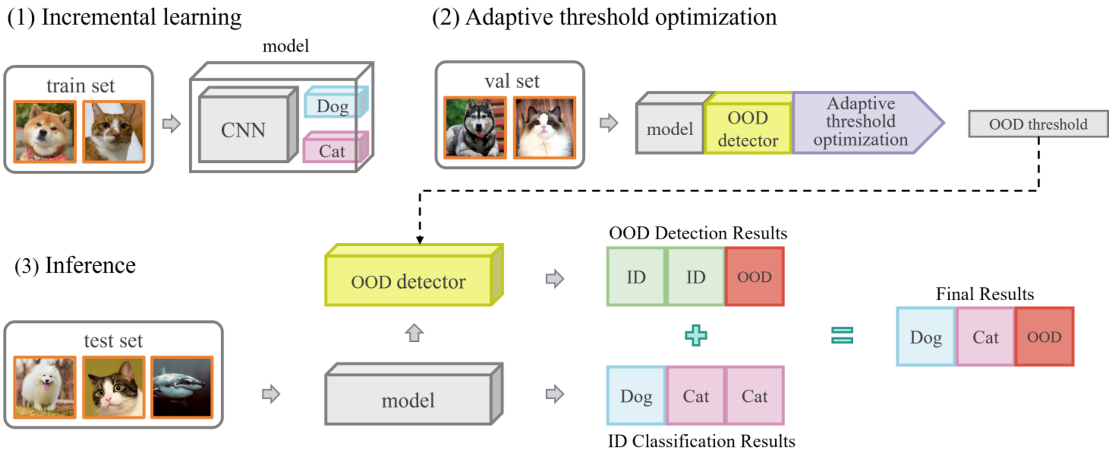

Fig. 2 Illustration of the proposed open-world class incremental learning (OWCIL) pipeline and its three stages.

3.2.2. Adaptive threshold optimization

Existing methods fix the network parameters and OOD thresholds, leading to sub-optimal results under continual learning settings. We propose an Adaptive Threshold (AT) method to optimize the OOD threshold after learning each task incrementally, enabling it to adapt to the dynamic environment. Specifically, a post-hoc OOD detection method is incorporated into the trained CIL model. The OOD score for each sample on n-th task can be obtained as the following:

$$o = \psi(f_n(x)), \quad (2)$$

where ψ can be any post-hoc OOD detection methods. We chose maximum softmax probability (MSP) as the default OOD method and the OOD score is defined as:

$$o_M = 1 - \frac{e^{\max f_n(x)}}{\sum_{i=1}^{C} e^{f_{n,i}(x)}}, \quad (3)$$

where $f_{n,i}(x)$ indicates the i-th index of outputs $f_n(x)$, i.e., the logit corresponding to the i-th class label. C represents the number of classes seen so far. Alternative to MSP, the energy function can be used as the OOD score:

$$o_E = -\log \sum_{i=1}^{C} e^{f_{n,i}(x)}. \quad (4)$$

The input x is an OOD sample or not can then be determined by the OOD detector:

$$G_n(x; \tau_n, f_n) = \begin{cases} 1 & \text{, if } \psi(x; f_n) \leq \tau_n, \\ 0 & \text{, if } \psi(x; f_n) > \tau_n, \end{cases} \quad (5)$$

where τ_n is the OOD threshold, 1 denotes an ID sample, and 0 denotes an OOD sample.

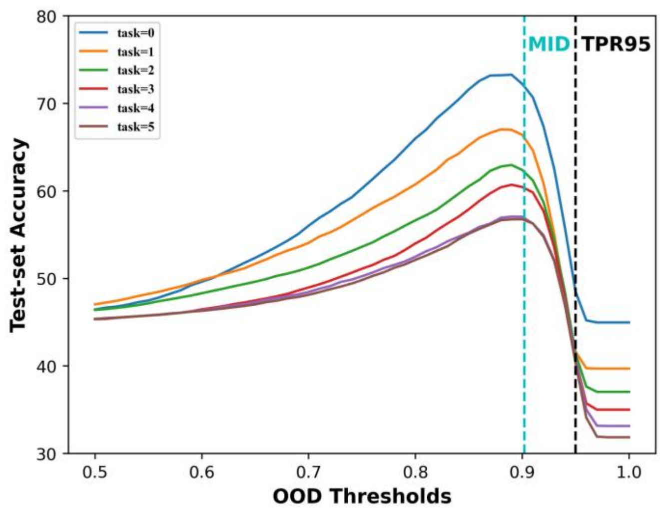

Fig. 3 The accuracy of PASS with MSP OOD detector at different thresholds. The solid lines represent the accuracy under different thresholds, and the dotted lines represent the accuracy under an optimized threshold.

As the OOD samples are excluded from the training set of OWCIL, the OOD thresholds cannot be optimized in a supervised manner. We propose a method for optimizing the threshold without using OOD samples. This approach utilizes hard sample mining to produce a more discriminative threshold, as depicted in Fig. 3. Specifically, the misclassified ID (MID) samples are treated as the pseudo OOD samples, and the goal is to minimize the Detection Error (DE) in order to obtain the optimal OOD threshold. The Detection Error measures the misclassification probability of OOD detector $G(x)$:

$$DE = \alpha P_{in}(G(x) = 0) + \beta P_{out}(G(x) = 1), \quad (6)$$

where P_{in} and P_{out} are assigned with equal weights.

3.2.3. Inference

The test set in OWCIL is open-set. It contains both ID and OOD samples. Therefore, for each test sample, we first perform OOD detection. For samples predicted as ID, we further classify them using $f(\mathrm{x})$. For samples predicted as OOD, we directly classify them as OOD class. The formulation of the final inference model is as follows:

$$H_n(x;\tau_n,f_n) = \begin{cases} arg\max_{\mathrm{i}} f_{n,i}(x) & ,if \quad G_n(x;\tau_n,f_n)=1, \\ OOD & ,if \quad G_n(x;\tau_n,f_n)=0. \end{cases} \tag{7}$$

where $arg\max_{\mathrm{i}} f_{n,i}(x)$ indicates the classification result, OOD denotes sample x is an out-of-distribution sample, $G_n(x;\tau_n,f_n)$ is the OOD detector.

After the preceding three stages, the final model $H_n(x;\tau_n,f_n)$ is capable of addressing the OWCIL problem. It can continuously learn new classes, retain knowledge of old classes, and handle out-of-distribution samples.

3.3. Evaluation Metrics for OWCIL

OWCIL highlights the effectiveness of CIL methods in real-world scenarios. Final classification results should be derived by combining OOD detection and classification results. Therefore, OOD samples are treated as the (K+1)-th class and are included in the evaluation alongside the K ID classes. For a fair comparison, we use the evaluation metrics of OWR:

$$\mathrm{ACC} = \frac{\mathrm{T_{OOD}}+\sum_{\mathrm{i=1}}^{\mathrm{C}}\mathrm{T_{ID,i}}}{\mathrm{T_{OOD}}+\mathrm{F_{OOD}}+\sum_{\mathrm{i=1}}^{\mathrm{C}}\mathrm{T_{ID,i}}+\mathrm{F_{ID,i}}}, \tag{8}$$

where T_{OOD} is the number of OOD samples that are correctly classified. F_{OOD} is the number of misclassified OOD samples. $T_{ID,i}$ is the number of i-th ID class samples that are correctly classified. $F_{ID,i}$ is the number of i-th ID samples that are misclassified. C represents the number of classes seen so far. In other words, the numerator of the ACC is all correctly classified samples (including OOD samples), and the denominator is the total number of samples. The higher ACC (Eq. (8)) is, the better overall performance the OWCIL method achieves in an open-world environment with OOD samples.

Considering the constantly updated CIL model, the accuracy usually decreases with the expansion of the semantic space. Hence, the accuracy of the last task is defined as:

$$\mathrm{A_L} = \mathrm{ACC_N}, \tag{9}$$

where N is the total number of tasks. The average incremental accuracy across all incremental stages is generally used to measure the overall effect of the model, which is expressed as:

$$\overline{\mathrm{A}} = \frac{1}{\mathrm{N}}\sum_{\mathrm{n=1}}^{\mathrm{N}}\mathrm{ACC_n}, \tag{10}$$

where $\mathrm{ACC_n}$ represents the accuracy in Eq. (8) at the n-th task.

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets

Three popular continual learning benchmark datasets are utilized. (1) CIFAR100 [40]. This dataset consists of 100 classes and 32×32 pixel images with 500 and 100 images per class for training and testing, respectively. (2) TinyImageNet [41]. This dataset is a subset of ImageNet with 200 classes, 64×64 pixel images, and 500 and 50 images per class for training and testing, respectively. (3) ImageNet100 [42]. This dataset is a subset of the ImageNet LSVRC dataset [43] consisting of 100 classes with 1300 and 50 images per class for training and testing, respectively.

4.1.2. Protocols

All datasets are partitioned into incremental setting. Three different incremental settings are adopted for CIFAR100: 1) 50 initial classes and 5 incremental learning (IL) tasks of 10 classes; 2) 50 initial classes and 10 IL tasks of 5 classes; 3) 40 initial classes and 20 IL tasks of 3 classes. For TinyImageNet, we use 100 initial classes and distribute the remaining classes into three incremental settings: 1) 5 IL tasks of 20 classes; 2) 10 IL tasks of 10 classes; 3) 20 IL tasks of 5 classes. For ImageNet100, we employ 50 initial classes and 5 IL tasks of 10 classes. During testing, the OOD detection dataset Places365 [44] is used as the source of OOD samples. The dataset contains 365 scene categories, all disjointing from the categories in our training and validation sets. We only use its validation set, which contains 36,500 images sized 64×64 pixels, with 100 images per category. Specifically, We take out the parts of the validation set with the same number of categories as the training set and divide them according to different increment settings.

4.1.3. Evaluation Metrics

The accuracy of the last task A_L (Eq. (9)) and the average incremental accuracy across all incremental stages $\overline{A}$ (Eq. (10)) are reported, as detailed in Section 3.3.

4.1.4. Backbones

The proposed OWCIL pipeline is compatible with a wide range of state-of-the-art EFCIL methods. We choose PASS, IL2A, SSRE, FeTrIL, as backbones.

4.1.5. Competitors

Two classic OOD threshold optimizations are selected as the competitors for our Adaptive threshold optimization. TPR95 determines thresholds by setting the true positive rate (TPR) to reach 95%, which is widely used in both OOD and OWR settings. Randomly selecting ID categories as OOD (RID) is also used for optimizing the OOD threshold. The proposed Adaptive Threshold (AT) is compatible with both TPR95 and RID, resulting in TPR95 w/ AT and RID w/ AT, respectively. Moreover, MSP is our default OOD score for different CIL methods. The impact of other OOD scores, e.g., ENERGY and NNO, are also evaluated.

4.1.6. Implementation Details

K-fold cross-validation is employed to create a validation set to determine the OOD threshold in OWCIL.We set K=3 in all experiments in this paper. Additionally, since K-fold cross-validation is utilized, K thresholds are obtained for each task, and we take the maximum value among them as the final threshold for inference: $\tau_{\mathrm{n}} = \max\tau_{\mathrm{n,i}}$, where $\tau_{\mathrm{n,i}}$ represents the threshold obtained by the i-th model for the n-th task. We use PyCIL [45] framework and the default parameters of each method in PyCIL for our experiments. The network architecture is ResNet-18 for all methods.

Table 2. The average incremental accuracy $\overline{A}$ of different methods on CIFAR100 and TinyImageNet. 'MID w/ AT' indicates the proposed full model for OWCIL. The incremental learning tasks (T) are equal to 5, 10, and 20.

CIL Backbone	OOD threshold	CIFAR100			TinyImageNet		
		T=5	T=10	T=20	T=5	T=10	T=20
PASS	TPR95	39.34	37.11	47.27	23.10	22.06	17.57
PASS	TPR95 w/ AT	53.40	52.48	55.99	38.18	38.62	33.13
PASS	MID w/AT	60.28	58.48	59.24	49.11	49.91	44.99
IL2A	TPR95	47.64	42.69	49.84	20.69	17.50	12.63
IL2A	TPR95 w/ AT	53.68	51.18	54.12	27.32	24.74	18.27
IL2A	MID w/AT	61.49	59.17	59.64	38.41	36.30	30.12
SSRE	TPR95	48.95	47.21	51.94	22.45	21.00	20.70
SSRE	TPR95 w/ AT	50.48	52.44	54.49	35.06	34.09	34.59
SSRE	MID w/AT	57.61	58.75	58.96	47.26	46.27	47.12
FeTrIL	TPR95	34.10	33.27	32.16	26.84	25.84	24.97
FeTrIL	TPR95 w/ AT	43.00	44.44	42.68	38.42	38.79	38.25
FeTrIL	MID w/AT	51.57	52.80	50.55	48.66	49.44	48.94

Table 3. OWCIL results of different methods on ImageNet100 5 IL tasks. 'MID w/ AT' indicates the proposed full model for OWCIL.

Method	A_L	$\overline{A}$
PASS+TPR95	35.26	37.50
PASS+MID w/AT	63.09	64.95
IL2A+TPR95	37.67	38.01
IL2A+MID w/AT	59.47	62.40
SSRE+TPR95	40.74	38.62
SSRE+MID w/AT	57.14	59.77
FeTrIL+TPR95	39.60	41.15
FeTrIL+MID w/AT	58.93	61.43

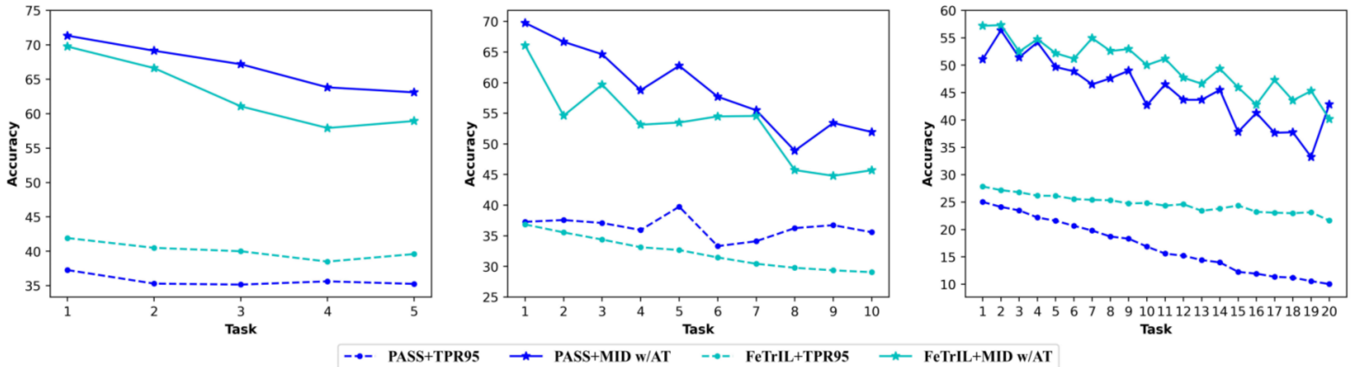

Fig. 4 Accuracy of each incremental task using different methods. '+TPR95' indicates the corresponding method under the baseline. '+MID w/ AT' represents the proposed full model. ImageNet100 5 IL tasks, CIFAR-100 10 IL tasks and TinyImagenet 20 IL tasks are reported.

4.2. Main Results

Table 2 presents the results of different threshold optimizations, i.e., TPR95, TPR95 w/ AT and the proposed MID w/ AT, based on different CIL backbones. Our MID w/ AT outperforms other threshold optimization methods significantly across all CIL methods. For instance, the proposed pipeline (MID w/ AT) boosts the PASS backbone (with TPR95) by 21.37% and 27.85% in the 10 IL tasks of CIFAR100 and TinyImageNet, respectively. The results of the large-scale dataset ImageNet100 are reposted in Table 3. Our MID w/ AT shows even more significant improvements over the baseline TPR95, increasing from 35.26% to 63.09% in A_L and from 37.50% to 64.95% in $\overline{A}$ with PASS as the CIL backbone. Detailed comparisons between different methods along the incremental learning procedure are illustrated in Fig. 4. Clear performance gaps between the curves of our MID w/ AT and the TPR95 can be observed.

4.3. Analysis of Different OOD Optimizations

As shown in Table 4, the proposed MID w/ AT consistently outperforms all competitors. On the one hand, our MID method finds more discriminative thresholds than TPR95 and RID without AT. For instance, MID is 8.30% better than the baseline TPR95 under 5 IL tasks setting with the PASS backbone. On the other hand, TPR95, RID, and MID have all been further improved after the addition of AT. Combining MID and AT further boosts the performance from 47.64% to 60.28%. Furthermore, equipping the CIL methods with OOD detectors may result in inferior performance under the OWCIL setting. As shown in Fig. 3, the performance of the OOD detector varies significantly at different thresholds, achieving optimal results only with the appropriate one. The improvement in our MID w/ AT mainly stems from the MID ability to identify distinctive pseudo OOD samples and the use of adaptive optimization. This allows the OOD detector to find optimal thresholds under the highly dynamic OWCIL setting.

Table 4. Comparisons between different OOD threshold optimization methods under the proposed OWCIL pipeline. The average incremental accuracy $\overline{A}$ of different methods on CIFAR100 and TinyImageNet are reported.

Method	CIFAR100			TinyImageNet		
	T=5	T=10	T=20	T=5	T=10	T=20
PASS+TPR95	39.34	37.11	47.27	23.10	22.06	17.57
PASS+RID	42.26	39.93	50.13	23.69	22.47	17.73
PASS+MID	47.64	44.87	54.03	24.61	23.11	17.98
PASS+TPR95 w/AT	53.40	52.48	55.99	38.18	38.62	33.13
PASS+RID w/AT	57.48	56.93	58.47	43.86	44.23	40.67
PASS+MID w/AT	60.28	58.48	59.24	49.11	49.91	44.99
IL2A+TPR95	47.64	42.69	49.84	20.69	17.50	12.63
IL2A+RID	51.30	46.25	51.26	21.13	17.71	12.74
IL2A+MID	55.33	50.32	55.38	21.67	17.99	12.89
IL2A+TPR95 w/AT	53.68	51.18	54.12	27.32	24.74	18.27
IL2A+RID w/AT	58.24	56.93	58.16	32.38	29.82	24.46
IL2A+MID w/AT	61.49	59.17	59.64	38.41	36.30	30.12
SSRE+TPR95	48.95	47.21	51.94	22.45	21.00	20.70
SSRE+RID	52.14	50.45	54.65	23.63	21.75	21.34
SSRE+MID	55.84	54.29	57.18	25.23	22.81	22.18
SSRE+TPR95 w/AT	50.48	52.44	54.49	35.06	34.09	34.59
SSRE+RID w/AT	54.82	57.20	57.79	41.18	40.61	42.67
SSRE+MID w/AT	57.61	58.75	58.96	47.26	46.27	47.12
FeTrIL+TPR95	34.10	33.27	32.16	26.84	25.84	24.97
FeTrIL+RID	35.38	34.24	33.43	28.58	27.30	26.13
FeTrIL+MID	38.88	37.35	36.18	31.54	29.91	28.28
FeTrIL+TPR95 w/AT	43.00	44.44	42.68	38.42	38.79	38.25
FeTrIL+RID w/AT	47.44	50.19	47.76	43.52	44.31	45.34
FeTrIL+MID w/AT	51.57	52.80	50.55	48.66	49.44	48.94

Table 5. Comparisons of different OOD scores based on our MID w/ AT. The average incremental accuracy $\overline{A}$ of different methods on CIFAR100 are reported.

Method	T=5	T=10	T=20
PASS+NNO	34.36	31.85	31.24
PASS+ENERGY	46.69	41.52	39.67
PASS+MSP	60.28	58.48	59.24
IL2A+NNO	35.61	34.35	34.16
IL2A+ENERGY	41.41	36.49	35.99
IL2A+MSP	61.49	59.17	59.64
SSRE+NNO	33.24	32.70	29.56
SSRE+ENERGY	36.63	33.39	32.08
SSRE+MSP	57.61	58.75	58.96
FeTrIL+NNO	48.69	48.95	46.67
FeTrIL+ENERGY	49.59	49.97	46.59
FeTrIL+MSP	51.57	52.80	50.55

4.4. Analysis of Different OOD Scores

The OOD score plays an important role in OOD detection. The default OOD score, MSP Eq. (3), is compared with ENERGY Eq. (4) and NNO, as shown in Table 5. These results are obtained based on the proposed MID w/ AT method.

MSP consistently outperforms ENERGY and NNO. For example, under the 20 IL task setting, the SSRE CIL backbone combined with MSP is 26.88% and 29.40% higher than that with ENERGY and NNO, respectively. Moreover, both MSP and ENERGY are exemplar-free methods, while NNO stores the feature centers of old classes. Therefore, MSP is more widely applicable to different scenarios than NNO. In Fig. 5, we visualize the accuracy heat map of SSRE plus different out-of-distribution scores for 10 incremental tasks on CIFAR100. From the overall color of the two matrices, MSP outperforms ENERGY on all tasks. From the diagonal elements, MSP performs better in learning new tasks. In addition, no matter which out-of-distribution score is used, the accuracy decreases with incremental learning, which reflects the catastrophic forgetting phenomenon in CIL.

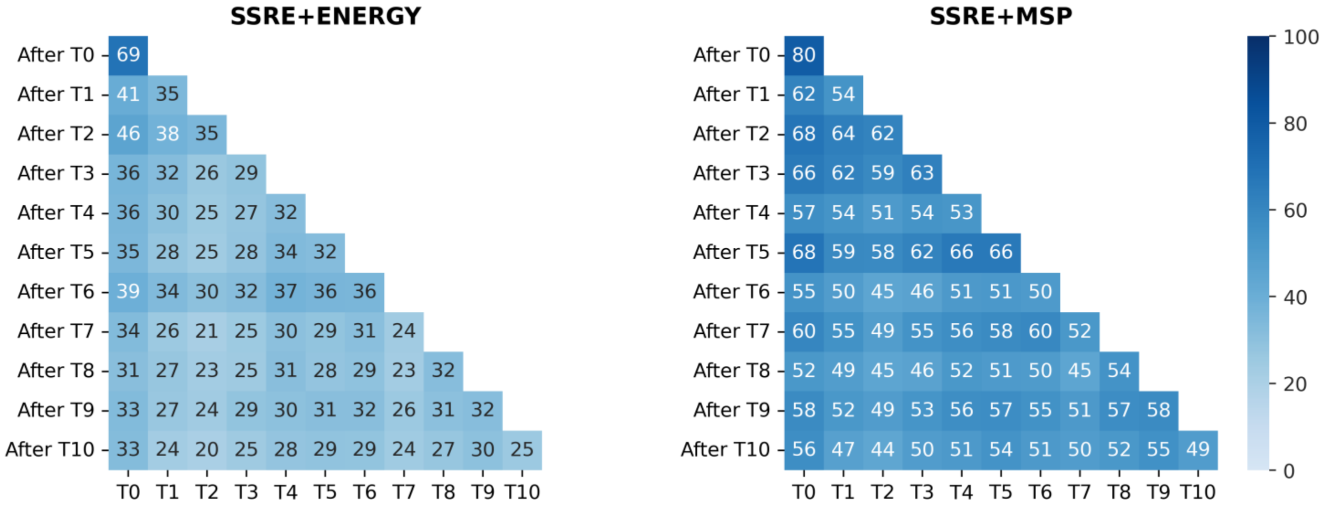

Fig. 5 Accuracy heatmap of SSRE with different out-of-distribution scores. The darker the color, the higher the accuracy.

5. Conclusion

In this paper, we propose a new setting, open-world CIL (OWCIL), to simulate the deployment of CIL models in real-world scenarios considering OOD samples. We introduce the discriminative optimization method, MID, to determine the threshold under the OWCIL setting, where no OOD sample is given. Additionally, we propose the adaptive threshold (AT) to dynamically identify the optimal OOD thresholds during testing. This method is compatible with existing OOD threshold optimizations and significantly improves their performance. Extensive experiments and analyses demonstrate that our MID with AT consistently enhances various incremental learning methods and achieves state-of-the-art performance.

References

[1] J. Read, A. Bifet, B. Pfahringer, and G. Holmes. Batchincremental versus instance-incremental learning in dynamic and evolving data. In Advances in Intelligent Data Analysis XI: 11th International Symposium, IDA 2012, Helsinki, Finland, October 25-27, 2012. Proceedings 11, pages 313–323. Springer, 2012. 1.

[2] G. M. Van de Ven, T. Tuytelaars, and A. S. Tolias. Three types of incremental learning. Nature Machine Intelligence, 4(12):1185–1197, 2022. 1.

[3] L. Wang, X. Zhang, H. Su, and J. Zhu. A comprehensive survey of continual learning: Theory, method and application. arXiv preprint arXiv:2302.00487, 2023. 1, 3.

[4] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13):3521–3526, 2017. 1, 3.

[5] Z. Li and D. Hoiem. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12):2935–2947, 2017. 1.

[6] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 2001–2010, 2017. 1, 3.

[7] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan. Foster: Feature boosting and compression for class-incremental learning. In European conference on computer vision, pages 398–414. Springer, 2022. 1, 3.

[8] S. Yan, J. Xie, and X. He. Der: Dynamically expandable representation for class incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3014–3023, 2021. 1, 3.

[9] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia. Maintaining discrimination and fairness in class incremental learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 13208– 13217, 2020. 1.

[10] G. Petit, A. Popescu, H. Schindler, D. Picard, and B. Delezoide. Fetril: Feature translation for exemplar-free classincremental learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 3911–3920, 2023. 1, 3, 4, 6.

[11] F. Zhu, Z. Cheng, X.-y. Zhang, and C.-l. Liu. Classincremental learning via dual augmentation. Advances in Neural Information Processing Systems, 34:14306–14318, 2021. 1, 3, 4, 6.

[12] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu. Prototype augmentation and self-supervision for incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5871– 5880, 2021. 1, 3, 4, 6.

[13] K. Zhu, W. Zhai, Y. Cao, J. Luo, and Z.-J. Zha. Selfsustaining representation expansion for non-exemplar classincremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9296–9305, 2022. 1, 3, 4, 6.

[14] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv e-prints, pages arXiv–1610, 2016. 1, 3, 4, 6.

[15] W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-ofdistribution detection. Advances in neural information processing systems, 33:21464–21475, 2020. 1, 3, 5, 6.

[16] M. Sensoy, L. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty. Advances in neural information processing systems, 31, 2018. 1, 3.

[17] E. Aguilar, B. Raducanu, P. Radeva, and J. Van de Weijer. Continual evidential deep learning for out-of-distribution detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3444–3454, 2023. 1, 3.

[18] R. Aljundi, D. O. Reino, N. Chumerin, and R. E. Turner. Continual novelty detection. In Conference on Lifelong Learning Agents, pages 1004–1025. PMLR, 2022. 1, 3.

[19] G. Kim, B. Liu, and Z. Ke. A multi-head model for continual learning via out-of-distribution replay. In Conference on Lifelong Learning Agents, pages 548–563. PMLR, 2022. 1, 3.

[20] T. Ahmad, A. R. Dhamija, M. Jafarzadeh, S. Cruz, R. Rabinowitz, C. Li, and T. E. Boult. Variable few shot class incremental and open world learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3688–3699, 2022. 1, 2, 4.

[21] A. Bendale and T. Boult. Towards open world recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1893–1902, 2015. 1, 2, 4, 6, 8.

[22] E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult. The extreme value machine. IEEE transactions on pattern analysis and machine intelligence, 40(3):762–768, 2017. 1, 2, 4, 6.

[23] D.-W. Zhou, Y. Yang, and D.-C. Zhan. Learning to classify with incremental new class. IEEE Transactions on Neural Networks and Learning Systems, 33(6):2429–2443, 2021. 2, 4.

[24] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(5):5513–5533, 2022. 3.

[25] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu. Deep class-incremental learning: A survey. arXiv preprint arXiv:2302.03648, 2023. 3.

[26] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. Psychological review, 102(3):419, 1995. 3.

[27] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Psychology of learning and motivation, volume 24, pages 109–165. Elsevier, 1989. 3.

[28] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi. Rainbow memory: Continual learning with a memory of diverse samples. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8218–8227, 2021. 3.

[29] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne. Experience replay for continual learning. Advances in Neural Information Processing Systems, 32, 2019. 3.

[30] Y. Yang, D.-W. Zhou, D.-C. Zhan, H. Xiong, and Y. Jiang. Adaptive deep models for incremental learning: Considering capacity scalability and sustainability. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 74–82, 2019. 3.

[31] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In International conference on machine learning, pages 3987–3995. PMLR, 2017. 3.

[32] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015. 3

[33] Z.-H. Zhou and Y. Jiang. Nec4. 5: Neural ensemble based c4. 5. IEEE Transactions on knowledge and data engineering, 16(6):770–773, 2004. 3.

[34] D.-W. Zhou, Q.-W. Wang, H.-J. Ye, and D.-C. Zhan. A model or 603 exemplars: Towards memory-efficient classincremental learning. arXiv preprint arXiv:2205.13218, 2022. 3.

[35] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection: A survey. arXiv preprint arXiv:2110.11334, 2021. 3.

[36] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606, 2018. 3.

[37] J. He and F. Zhu. Out-of-distribution detection in unsupervised continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3850–3855, 2022. 3.

[38] G. Kim, C. Xiao, T. Konishi, and B. Liu. Learnability and algorithm for continual learning. In International Conference on Machine Learning, pages 16877–16896. PMLR, 2023. 3.

[39] A. Rios, N. Ahuja, I. Ndiour, U. Genc, L. Itti, and O. Tickoo. incdfm: Incremental deep feature modeling for continual novelty detection. In European Conference on Computer Vision, pages 588–604. Springer, 2022. 3.

[40] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Tront, 2009. 6

[41] Y. Le and X. Yang. Tiny imagenet visual recognition challenge. CS 231N, 7(7):3, 2015. 6.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009. 6

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115:211–252, 2015. 6.

[44] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence, 40(6):1452–1464, 2017. 6.

[45] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, and D.-C. Zhan. Pycil: A python toolbox for class-incremental learning. arXiv eprints, pages arXiv–2112, 2021. 6.