CUAL: Continual Uncertainty-aware Active Learner

Amanda Rios
Intel
amanda.rios@intel.com

Ibrahima Ndiour
Intel
ibrahima.j.ndiour@intel.com

Parual Datta
Intel
parual.datta@intel.com

Jerry Sydir
Intel
jerry.sydir@intel.com

Omesh Tickoo
Intel
omesh.tickoo@intel.com

Nilesh Ahuja
Intel
nilesh.ahuja@intel.com

Abstract

AI deployed in many real-world use cases should be capable of adapting to novelties encountered after deployment. Here, we consider a challenging, under-explored and realistic continual adaptation problem: a deployed AI agent is continuously provided with unlabeled data that may contain not only unseen samples of known classes but also samples from novel (unknown) classes. In such a challenging setting, it has only a tiny labeling budget to query the most informative samples to help it continuously learn. We present a comprehensive solution to this complex problem with our model "CUAL" (Continual Uncertainty-aware Active Learner). CUAL leverages an uncertainty estimation algorithm to prioritize active labeling of ambiguous (uncertain) predicted novel class samples while also simultaneously pseudo-labeling the most certain predictions of each class. Evaluations across multiple datasets, ablations, settings and backbones (e.g. ViT foundation model) demonstrate our method's effectiveness. We will release our code upon acceptance.

1 Introduction and Related Work

Real-world artificial intelligence (AI) systems often contend with evolving data distributions, due to changes in operating conditions and the introduction of new classes after deployment. To enforce a robust response, AI systems should ideally be able to continuously learn and adapt to detected novelties with minimal computing and labeling costs. Nonetheless, most current adaptive continual learning (CL) solutions [1, 2, 3] still necessitate fully labeled data, a requirement that overlooks the feasibility and high cost of data labeling in practical AI applications. Unsupervised and semi-supervised CL solutions are scarce [4, 5, 6] and furthermore, operate with a highly simplified oracle assumption: that in unlabeled post-deployment data, old (known) classes cannot appear in conjunction with newly introduced (novel) classes. Yet, in many realistic use cases this assumption will not hold, i.e. new and old classes may co-occur. One reason current models continue to adopt this oversimplification is to limit error propagation: when unseen novel-class samples co-occur with old-classes, misclassifications can occur not only among new classes but also between novel and old classes. In this scenario, the CL model may also face prediction overconfidence [7], equivocally mapping novel-class samples to old-class labels. The question then becomes how to design a robust semi-supervised CL model that can properly function in a non-overly constrained setting? In what is still a parallel research area, cost-effective active labeling (AL) [8, 9, 10, 11] has been applied extensively on closed-world learning to minimize labeling costs to a human-in-the-loop. However, only a few works have incorporated Active Labeling concepts within continual learning [12, 13, 14] as a strategy to minimize labeling costs in challenging scenarios where data distributions are evolving.
Our contribution is as follows: We eliminate the commonly used oracle assumption employed by most semi-supervised CL techniques. This creates a more realistic continual learning setting where old and new classes can co-occur, while maintaining a minimal labeling budget. In this case, AL querying must be judiciously designed not only to prioritize novel-class sampling but also to select the most informative (ambiguous) samples among them [15]. We introduce our model, CUAL (Continual Uncertainty-aware Active Learner), which learns to classify new classes as they are introduced while

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

efficiently utilizing the limited labeling budget (e.g., 2.5%, an application-defined hyperparameter) by selecting only the most ambiguous samples among novel class predictions. CUAL uses its uncertainty scoring function to guide active querying. To our knowledge, ambiguity-based querying has not been previously covered in continual AL. Furthermore, unlike most existing continual AL methods that learn only from active queries [12, 13, 14], CUAL leverages its uncertainty formulation to include pseudo-labeling in its AL inner loop, significantly reducing labeling requirements.

2 Our Method

2.1. Problem Setting: Consider an agent $A(x,t)$ that is expected to learn to classify from a sequence of continual tasks. At each continual task t, the model is presented with an initially unlabeled set of samples $U(t)$ which consists of a mixture of unseen samples of old classes $U_{old}(t)$ and unseen samples of new (novel) classes $U_{new}(t)$:

$$U(t) = U_{old}(t) \cup U_{new}(t), \text{where } U_{old}(t) = \{x | x \sim \bigcup_{k=1}^{t-1} D_k\}, U_{new}(t) = \{x | x \sim D_t\}, \quad (1)$$

Here D_t comprises samples from the set of new classes C^t_{new} introduced at task t, while $\bigcup_{k=1}^{t-1} D_k$ are samples belonging to all the old classes C^t_{old} that have been learned up to and including task $t-1$. Samples in $U_{old}(t)$ are "unseen", meaning they were never used, neither in the initial training nor during prior tasks' learning. Note that addressing data drifts in U_{old} is beyond the scope of this work and will be addressed in future research.

2.2. Our solution: We introduce a 'short-term' active learner $A_s(x,i;t)$ whose goal is to acquire a sufficient number of samples from $U_{new}(t)$ via a combination of uncertainty-guided active-labeling and pseudo-labeling (i being the index for inner-loop AL iterations). These will be used to update the parameters of $A(x,t)$ via continuous learning procedures such as experience replay. Both modules, A and A_s, contain trainable classification heads, A^{cl} and A^{cl}_s respectively, that operate on features extracted from a frozen large/foundation vision model pre-trained on a separate dataset with no class overlap with the test dataset (implementation details in Appendix 4.2.2). While A^{cl} predicts labels for the main continual classification task, A^{cl}_s will perform pseudo-labeling by predicting among only new-class labels and will use an uncertainty metric 2 to guide AL and confident pseudo-labeling, as detailed in sec. 2.2.2.

2.2.1. Building block of CUAL's uncertainty formulation: CUAL's uncertainty estimation 2 uses the *feature reconstruction error* (FRE) metric introduced in [16]. FRE has been shown to effectively estimate novelty in the closed-world and continuous setting [17]. In multi-class settings, FRE learns a dimensionality-reducing PCA (principal component analysis) transform $\mathcal{T}_m$ and its inverse $\mathcal{T}^\dagger_m$ for each class m. For scoring, a test-feature $u = g(x)$ is first transformed into a lower-dimensional subspace by applying $\mathcal{T}_m$ and then re-projected back into the original higher dimensional space via $\mathcal{T}^\dagger_m$. FRE for the mth class is calculated as the ℓ_2 norm of the difference between the original and reconstructed vectors. Intuitively, if a sample does not belong to the same distribution as that m^{th} class, it should result in a large FRE_m score.

2.2.2. CUAL Step by Step: At the outset (task $t = 0$), we assume that $A(x, t = 0)$ has been previously trained to classify among a fixed set of pre-deployment classes C^0_{new}. Simultaneously, a set of FRE transforms, $\mathcal{T}_m$ have been learnt $\forall m \in C^0_{new}$. For each subsequent task $t > 0$, CUAL invokes its active learning inner-loop, indexed by i. At the first iteration $i = 0$ of a given task t, a first cycle of AL is performed by querying samples with the highest uncertainty scores defined as $S^0(u) \triangleq \min_{j \in C^t_{old}} FRE^0_j(u)$ as these are most likely to be novel (see Appendix 4.1.2 for details). At this point, novel classes are identified (denoted by $|C^t_{new}|$ in section 2.1, assuming $|C^t_{new}| > 0$) and the actively labeled samples are used to: (1) Initialize and train $A^{cl}_s(x, i = 0, t)$ to learn an imperfect initial mapping to the $|C^t_{new}|$ novel classes (A^{cl} contains output nodes only w.r.t novel classes); (2) compute rough estimates of per-novel-class PCA transforms $\{\mathcal{T}^{t,0}_m\}, m \in C^t_{new}$. Note that it's possible that not all true novel classes are found in this initial iteration and may be found in subsequent ones. For subsequent iterations $i > 0$, given an unlabeled sample $x \in U(t)$, $A^{cl}_s(x,i,t)$ predicts a pseudo-label $m, m \in C^t_{new}$ which then routes the selection of the corresponding PCA transform $\mathcal{T}^{t,i-1}_m$ resulting in that iteration's uncertainty score $S^i(u)$ 2:

$$S^i(u) = \min_{j \in C^t_{old}} \frac{FRE^0_j(u)}{FRE^{i-1}_m(u)}; i > 0, m = A^{cl}_s(u, i-1, t) \in C^{new}_t \quad (2)$$

$S^i(u)$ can be used to robustly categorize samples in $U(t)$ as: **(1) Novel with high-confidence:** These are samples with the highest score values, which will occur for a high numerator relative to the

denominator. A high value of numerator implies large distance from previously seen classes C^t_{old}, while a low value of the denominator implies low distance from novel class m. Such a sample likely belongs to $U_{new}(t)$ and is a strong candidate to be pseudo-labeled. From these, we select the topmost most confident α percent of samples to pseudo-label **(2) Old-class with high-confidence:** lowest score values corresponding to low numerator (low distance w.r.t C^{t-1}_{old}) and high denominator value (high-distance from the predicted novel class m). Such a sample likely belongs to $U_{old}(t)$, i.e. to an old class that has already been learned; **(3) Ambiguous:** Samples for which the score is neither definitively high nor definitively low. These could be old-class samples having relatively high scores, or new-class samples having relatively low scores. Owing to this ambiguity, a clear determination cannot be made. Hence, these samples are excellent candidates for active labeling (see Appendix 4.1.2 for details). At each inner-loop iteration, accumulated active and pseudo-labeled samples are used to re-update $A^{cl}_s(x, i, t)$ parameters. Finally, at the end of the inner-loop, the full set of actively labeled and pseudo-labeled samples is used to update the continual agent's classification head $A^{cl}(x, t)$ via Experience Replay [18], a common CL technique. The accumulated samples are also used to compute final estimates of the PCA transforms $\{\mathcal{T}^{t,i=I}_m\}$ for $m \in C^{new}_t$. Methodology details, including those for experience replay, stopping criteria and ambiguity formulation, can be found in the appendix sections.

CUAL Key Takeaways: (1) **Ambiguity**: In continual AL, the informativeness of a sample does not necessarily equate to its distance from old-classes or to its confusion among other new classes. Rather, CUAL is, to the best of our knowledge, the first continual active learning method to characterize samples as informative if they can be classified (pseudo-labeled) as new but with *ambiguous* certainty; (2) **Pseudo-labeling**: A design choice in current closed-world and also continual AL methods [13, 7] is to train only with the selected labeled samples for purpose of data efficiency. However, this can come at a high cost to performance when unlabeled samples are the vast majority (tiny AL budget). This effect is amplified in continual settings as other pressures inherent to CL, e.g. catastrophic forgetting, come into the mix. There are many techniques in semi-supervised learning to leverage information of unlabeled samples. CUAL uses pseudo-labeling, a generic semi-supervised technique, to help learn within the inner loop. Yet, if wrong pseudo-labels are used to train, it can lead to severe degradation of performance. This is why CUAL's uncertainty formulation is key to ensuring that only the most confident pseudo-labels are used to enhance AL and not undermine it, as shown in Fig 2. Note that other semi-supervised techniques, such as contrastive or prediction-consistency, etc, are orthogonal to CUAL and can be integrated in future research.

3 Experiments

3.1. Setup: Implementations for CUAL and baselines: All use a large/foundation frozen feature extractor, e.g. ResNet50 [19] pre-trained on ImageNet1K via SwAV [20] or ViTs16 [21] pre-trained on Imagenet1K via DINO [22]. A^{cl}_s (the pseudo-labeling head) is a fully connected layer and A^{cl}'s is a perceptron of size 4096 (the latter A_{cl} is same for all baselines). We test on 4 diverse datasets: Imagenet21K-OOD (Im21K-OOD) [23], Places365-OOD (Places) [24], Eurosat [25], iNaturalist-Plants-20 (Plants) [26] and Cifar100-superclasses [27]. All of the aforementioned datasets were constructed to have class orthogonality (be out-of-distribution) with respect to Imagenet1K with the exception of Cifar100. Our replay buffer stores a fixed amount of pre-logit deep-embeddings and labels/pseudo-labels. We set the maximum coreset size to 5000 for Im21K-OOD, Places and 2500 for Cifar100, Eurosat. While Cifar100 is not orthogonal to Imagenet1K, we include it for direct comparison with our main benchmarking approach, incDFM, which uses Cifar100 Superclasses as its hardest experimental dataset. Details on datasets are in the supplementary. At each incoming unlabeled pool, we fix a mixing ratio of 2:1 of old to new classes per task, with old classes drawn from a holdout set (0.35% of each dataset). We set pseudo-labeling selection to $\alpha = 20\%$ of samples predicted as novel (appendix 4.1.1). For experiments not purposely varying the tiny supervision budget, we fix a labeling budget of 1.25% for Im21K-OOD, Places, Plants and 0.5% for Eurosat, Cifar100 as guided by Fig 1 *right* which varies the AL budget from 0.5% to 5%. We compare CUAL to: (1) CCIC [5] a semi-supervised CL model that adapts MixMatch [28] to the CL setting; (2) Experience-Replay "ER" [18, 3]. ER is adapted to learn with active labeling via Entropy or Margin similarly to other continual AL works [13]; (3) PseudoER [29], same as ER, but uses the AL heuristic (Entropy or margin) to iteratively pseudo-label the most confident samples during the inner loop. Other baselines are constructed (Fig 2 right table) from removing elements of CUAL such as the iterativeness (i.e. doing AL/Pseudo-labeling in one shot during ER/PseudoER), etc. More details in appendix sections.

3.2. Results: Fig. 1 (left) shows the cumulative accuracy of the continual-learning classifier module A^{cl} over tasks, i.e. the average over all classes learned up to and including the task t. The "Oracle" method constitutes an upper-bound. It has perfect knowledge of old and new class labels (100% supervision) and is trained using the same architecture and experience replay hyper-parameters. The baseline ER only uses actively labeled data (chosen to have ambiguous classification entropy values akin to CUAL, for fair comparison). While this is superior to labeling randomly (see Fig 2 left table), ER performance is still capped by the tiny labeling budget available. On the other hand, CUAL as well as semi-supervised baselines PseudoER and CCIC all employ some technique (refer to section 3.1) to learn from unlabeled samples beyond those which are actively labeled. Nevertheless, CUAL outperforms all baselines by a large margin, even with stringent labeling budgets of $1.25-2.5\%$. Also, 1 (right) analyzes the effect of varying the tiny active labeling budget. We show CUAL over-performs the other methods over a large interval of supervision budgets (all tested). Note that CCIC (blue line or blue star symbol) drastically under-performs all other approaches even with a large supervision budget of 10%. As most semi-supervised CL methods, it originally assumed old and new classes would not co-occur and cannot properly quantify uncertainty when this assumption is lifted. Similar trends are observed in Table 2 (left), which lists the continual classification performance of CUAL and baselines averaged over all tasks for all 4 datasets. We showcase results for two architecture variations: Resnet50 trained via SWAV (R50) and ViTs16 trained via DINO (ViT). Equivalent results for Cifar100 and other class increment variations in Appendix.

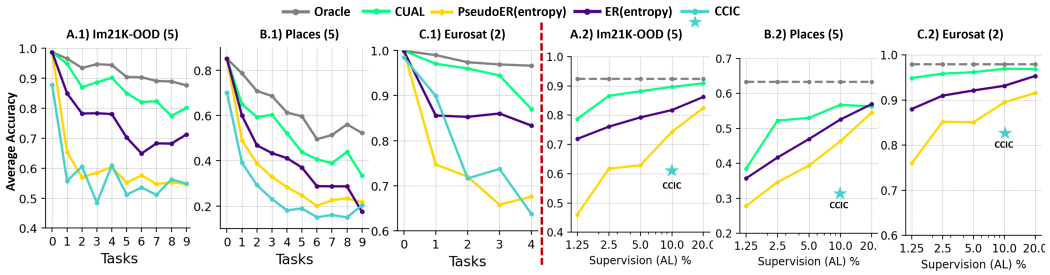

Figure 1: (*Left* A.1-C.1) Continual classification accuracy over continual tasks. The number of novel classes introduced per task is in parenthesis. CUAL over-performs other methods in this challenging setting. The Oracle (gray) is fully supervised ER.(*Right* A.2-C.2) Results varying the AL budget.

Table 2 (right) includes results for ablations of CUAL: *AL-Top* selects samples with highest uncertainty scores (i.e. most-confidently novel samples) for AL rather than ambiguous samples; *AL-Rand* refers to random sampling with the same supervision budget. These AL heuristic variations (Top and Rand) highlight the importance of querying Ambiguous samples instead of most novel samples as is customarily done. Benefit of pseudo-labeling is clearly underscored by the sharp drop in performance in the *only-AL* row in which pseudo-labeling is removed. In fact, AL only with an 8x higher labeling budget (*only-AL(8x-budget)*) still does not achieve the same performance as that obtained by pseudo-labeling. Note that in Fig 2 Table (left) we include additional results of the best-performing baseline ER, e.g. using Random or oneshot active labeling, akin to ablations of CUAL just described.

	Im21K		Places		Eurosat		Cifar100	
	R50	ViT	R50	ViT	R50	ViT	R50	ViT
Oracle	92.4	80.4	63.3	45.6	98.0	94.8	76.5	70.2
CUAL(ours)	**86.6**	**73.5**	**53.0**	**39.6**	**95.8**	**94.1**	**65.0**	**65.7**
ER-Ent	76.1	62.8	41.7	31.3	91.0	87.3	56.2	57.4
ER-Rand	71.9	-	35.6	-	84.4	-	48.2	-
ER-Ent-oneshot	75.3	-	39.4	-	85.6	-	53.8	-
PseudoER-Ent	61.7	49.0	34.7	28.5	83.6	77.6	46.9	48.1
CCIC	58.0	-	26.5	-	79.5	-	34.8	-

Variations (R50)	Im21K	Places	Eurosat	Cifar100
CUAL(default)	**86.6**	**53.0**	**95.8**	**65.0**
AL-Top	75.0	48.4	88.8	62.9
AL-Rand	81.4	42.9	85.5	60.3
AL-oneshot	81.8	46.7	94.2	63.2
only-AL	79.1	37.9	74.8	48.4
only-AL-Rand*	71.9	35.6	84.4	48.2
only-AL(8x-budget)	83.6	50.3	88.3	61.3

Figure 2: Continual Classification Results, averaged over all tasks. **Left** - Default CUAL and baselines ("ent" equates to Entropy). **Right** - CUAL ablations. Refer to main text for discussion. *see apx 4.3.

4. Conclusion: In this work, we presented CUAL, a solution to the still under-explored field of continual active learning. To the best of our knowledge, CUAL is the first continual active learner to propose and demonstrate that querying based on the ambiguity of being novel or non-novel may be more informative than just selecting the samples that are most likely to be novel. Further, this work shows that assigning pseudo-labels with trustworthy confidence can significantly alleviate labeling costs without degrading continual AL performance. Overall, the proposed approach outperforms

the baselines over multiple large-scale datasets and experimental variations. Yet, several challenges remain for continual active learning, which we hope to address in future work. One nontrivial example is how to continually query and update if distribution shifts of past classes (e.g. noise, illumination, etc) co-occur with novel classes.

References

[1] Amanda Rios and Laurent Itti. "Lifelong Learning Without a Task Oracle". In: *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2020, pp. 255–263.

[2] German I Parisi et al. "Continual lifelong learning with neural networks: A review". In: *Neural Networks* 113 (2019), pp. 54–71.

[3] Pietro Buzzega et al. "Rethinking experience replay: a bag of tricks for continual learning". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 2180–2187.

[4] Zhiqi Kang et al. "A soft nearest-neighbor framework for continual semi-supervised learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 11868–11877.

[5] Matteo Boschini et al. "Continual semi-supervised learning through contrastive interpolation consistency". In: *Pattern Recognition Letters* 162 (Oct. 2022), pp. 9–14. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2022.08.006. URL: http://dx.doi.org/10.1016/j.patrec.2022.08.006.

[6] Benedikt Bagus, Alexander Gepperth, and Timothée Lesort. *Beyond Supervised Continual Learning: a Review*. 2022. arXiv: 2208.14307 [cs.LG].

[7] Pengzhen Ren et al. "A survey of deep active learning". In: *ACM computing surveys (CSUR)* 54.9 (2021), pp. 1–40.

[8] Donggeun Yoo and In So Kweon. "Learning loss for active learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 93–102.

[9] Ozan Sener and Silvio Savarese. "Active Learning for Convolutional Neural Networks: A Core-Set Approach". In: *International Conference on Learning Representations*. 2018.

[10] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. "Deep bayesian active learning with image data". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1183–1192.

[11] Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. "How to measure uncertainty in uncertainty sampling for active learning". In: *Machine Learning* 111.1 (2022), pp. 89–122.

[12] Ali Ayub and Carter Fendley. "Few-shot continual active learning by a robot". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 30612–30624.

[13] Thuy-Trang Vu et al. "Active Continual Learning: On Balancing Knowledge Retention and Learnability". In: *arXiv preprint arXiv:2305.03923* (2023).

[14] Xiangli Nie et al. "Online active continual learning for robotic lifelong object recognition". In: *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[15] Manali Sharma and Mustafa Bilgic. "Most-surely vs. least-surely uncertain". In: *2013 IEEE 13th International Conference on Data Mining*. IEEE. 2013, pp. 667–676.

[16] Ibrahima Ndiour, Nilesh A Ahuja, and Omesh Tickoo. "Out-Of-Distribution Detection With Subspace Techniques And Probabilistic Modeling Of Features". In: *arXiv preprint arXiv:2012.04250* (2020).

[17] Amanda Rios et al. "incdfm: Incremental deep feature modeling for continual novelty detection". In: *European Conference on Computer Vision*. Springer. 2022, pp. 588–604.

[18] David Rolnick et al. "Experience replay for continual learning". In: *Advances in Neural Information Processing Systems* 32 (2019).

[19] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[20] Mathilde Caron et al. "Unsupervised learning of visual features by contrasting cluster assignments". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9912–9924.

[21] Dosovitskiy Alexey. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv: 2010.11929* (2020).

[22] Mathilde Caron et al. "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660.

[23] Tal Ridnik et al. *ImageNet-21K Pretraining for the Masses*. 2021. arXiv: 2104.10972 [cs.CV].

[24] Bolei Zhou et al. "Places: A 10 million Image Database for Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[25] Patrick Helber et al. *EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification*. 2019. arXiv: 1709.00029 [cs.CV].

[26] Grant Van Horn et al. *The iNaturalist Species Classification and Detection Dataset*. 2018. arXiv: 1707.06642 [cs.CV].

[27] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: *University of Toronto* (May 2012).

[28] David Berthelot et al. "Mixmatch: A holistic approach to semi-supervised learning". In: *Advances in neural information processing systems* 32 (2019).

[29] Dong-Hyun Lee. "Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks". In: *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)* (July 2013).

[30] Yen-Chang Hsu et al. "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*. 2020, pp. 10951–10960.

[31] Amanda Rios and Laurent Itti. "Closed-loop memory GAN for continual learning". In: *arXiv preprint arXiv:1811.01146* (2018).

[32] Utku Evci et al. "Head2toe: Utilizing intermediate representations for better transfer learning". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 6009–6033.

[33] Alexander A Petrov, Barbara Anne Dosher, and Zhong-Lin Lu. "The dynamics of perceptual learning: an incremental reweighting model." In: *Psychological review* 112.4 (2005), p. 715.

[34] Guneet S Dhillon et al. "A baseline for few-shot image classification". In: *arXiv preprint arXiv:1909.02729* (2019).

[35] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[36] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". English (US). In: *International Journal of Computer Vision* 115.3 (Dec. 2015). Publisher Copyright: © 2015, Springer Science+Business Media New York., pp. 211–252. ISSN: 0920-5691. DOI: 10.1007/s11263-015-0816-y.

[37] Rui Huang and Yixuan Li. "Mos: Towards scaling out-of-distribution detection for large semantic space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8710–8719.

[38] Yifei Ming et al. "Delving into out-of-distribution detection with vision-language representations". In: *Advances in neural information processing systems* 35 (2022), pp. 35087–35102.

[39] Yiyou Sun, Chuan Guo, and Yixuan Li. "React: Out-of-distribution detection with rectified activations". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 144–157.

4 Appendix

4.1 CUAL Methodology Details

4.1.1 Thresholds for Stopping the AL inner-loop and Pseudo-labeling

The inner-loop in CUAL is guided by two simple thresholds: (1) Threshold T_{inner} "roughly" estimates if there are any possible novel-class samples in the unlabeled task input data pool and is controlled by a single hyper-parameter, the number of standard deviations above the mean of an in-distribution validation set (2 STDs in our experiments). If no samples are found to be above T_{inner}, we reach the stopping criterion for our iterations. Our in-distribution validation set is conventionally defined to

include a portion (0.1%) of the previous tasks' $k = 1 : t-1$ novelty predictions that were held-out at previous tasks, i.e. not used to update $A(x,t)$ parameters. Importantly, the same in-distribution validation set is used for all compared baselines in our results section, as is common practice in the OOD/novelty-detection literature [17, 30]; (2) Finally, Threshold α tunes pseudo-labeling selection and is set to $\alpha = 20\%$ highest $S^i(u)$ scores (most confident) from the test samples found above T_{inner}. These two thresholds are not highly sensitive.

4.1.2 How to define Ambiguity

CUAL selects the most ambiguous samples at each AL inner-loop iteration, i.e. scores $S^i(u)$ which are neither too high or too low. Our mathematical formulation uses the threshold T_{inner} defined in the previous section: we formulate ambiguousness as the inverse squared distance $\frac{1}{\|S^i(u)-T_{inner}\|^2}$ of scores to T_{inner}. Intuitively, this formula favors selecting samples that cannot be unambiguously predicted as either old or new since T_{inner} represents this rough decision boundary. Active selection is stopped when the tiny labelling budget is exhausted. The only exception to this Ambiguity formulation is at the first iteration $i = 0$ where we select homogeneously from samples above T_{inner}. This is the case because at $i = 0$ only old classes are used to compute the score function, $S^0(u) = \min_{j \in C^t_{old}} FRE^0_j(u)$ and so ambiguity cannot be defined in the same way as for the remainder of iterations.

4.1.3 Measuring per-class uncertainty in CUAL $S^i(u)$ formulation

CUAL is agnostic to the elemental uncertainty metric used in its uncertainty scoring function ($S^i(u)$ Eq. 2 in section 2) as long as it can reliably estimate uncertainty w.r.t each novel class or old class. However, this is not an easy feat since many existing static uncertainty quantification approaches are not fully reliable [16, 17]. As discussed in the main text, CUAL currently leverages the *feature reconstruction error* (FRE) metric introduced in [16] to build Eq 2. For each in-distribution class, FRE learns a PCA (principal component analysis) transform $\{\mathcal{T}_m\}$ that maps high-dimensional features u from a pre-trained deep-neural-network backbone $g(x)$ onto lower-dimensional subspaces. During inference, a test-feature $u = g(x)$ is first transformed into a lower-dimensional subspace by applying $\mathcal{T}_m$ and then re-projected back into the original higher dimensional space via the inverse $\mathcal{T}_m^\dagger$. The FRE measure is calculated as the ℓ_2 norm of the difference between the original and reconstructed vectors:

$$FRE_m(u) = \|f(\mathbf{x}) - (\mathcal{T}_m^\dagger \circ \mathcal{T}_m)u\|_2. \tag{3}$$

Intuitively, FRE_m measures the distance of a test-feature to the distribution of features from class m. If a sample does not belong to the same distribution as that mth class, it will usually result in a large reconstruction score FRE_m. FRE is particularly well suited for the continual setting since for each new class discovered at test-time, an additional principle component analysis (PCA) transform can be trained without disturbing the ones learnt for previous classes.

4.1.4 Continual Model Update

After the end of CUAL's inner loop, $i = I$ with I corresponding to the last iteration, CUAL's short-term A_s has accumulated a final set of actively labeled and pseudo-labeled novel samples. This final set is then used to update the continual agent $A(x,t)$: (1) obtain the final estimates of the PCA transformations corresponding to the novel classes, $\mathcal{T}^t_m, m \in C^{new}_t$, for task t and to update long-term A^{cl} (long term classification head of $A(x,t)$) via experience replay. From then on, in subsequent tasks, these novel classes $m, m \in C^{new}_t$ will no longer be flagged as "novel" and will be treated as "old".

Here we expand upon the technique of "Experience Replay" (ER) [18, 3] used to update long-term A^{cl} at the end of each task. Note that the same ER algorithm and hyper-parameters are used for baselines ER and PseudoER. In our implementation of ER a limited number of deep-embeddings belonging to old classes (from the frozen foundation model) must be stored in a Buffer of fixed size ($B_t = U^*_{old,k}, k : 1 : t-1$). For a given task, samples B_t are interleaved with those in $U^*_{new,t}$ to train $A(x,t)$ without catastrophic forgetting. After updating A^{cl} the replay buffer itself must also be updated. We use a fixed-size memory buffer B_t with the same building strategy as in [31]: a buffer of fixed size and prioritizing homogeneous distribution among classes. That is, an equivalent number of samples of each class are removed if room is required for new classes and the buffer is full. Formally,

the ER training loss is defined in equation (4), which includes cross-entropy loss, L, calculated over the set of actively-labeled new class samples (new_{AL}), pseudo-labeled (PL) samples (new_{PL}) and the buffer samples (B_t), respectively. We set the weights β, γ and θ to 0.25, 0.25 and 0.5 respectively for a roughly even importance of old and new class samples.

$$L_t = \beta L(new_{AL}) + \gamma L\left(new_{PL}\right) + \theta L(B_t) \tag{4}$$

4.2 Experimental Methodology

4.2.1 Implementation Details for CUAL and Baselines

We use two large-scale/foundation models as feature extraction backbones, kept frozen throughout CUAL and baselines' training: (1) Most results use ResNet50 [19] unsupervisedly pre-trained on ImageNet1K via SwAV [20]. We extract features from the pre-logit AvgPool layer of size 2048 as deep-embeddings. We also experimented with other feature extraction points [16] but those under-performed w.r.t the pre-logit layer. (2) We also show results (Fig 2 Left Table) using ViTs16 [21] pretrained on Imagenet1K via DINO [22]. For ViTs16 we tried several extraction points, e.g. head, last norm later, different transformer block outputs with different pool factors (e.g. 2,4). Best results were obtained with Block 9 features with pooling range of 2, yielding deep features of size 18816. Note that learning on frozen deep features is commonplace in vision CL and domain-adaptation fields [1, 17, 32]. It is theoretically based on the principle that low-level visual features from a large-scale/foundation frozen model are task nonspecific and do not need to be constantly re-learned. Rather, learning may happen upstream by utilizing the extracted deep features (at the last or inner-layers, or a combination thereof - an active research area) [33, 34, 32].

CUAL and baseline parameters are trained using the extracted deep features as outlined above. The classification heads are trained with ADAM [35] and learning rate of 0.001. For long-term A^{cl} we use a batchsize of 50 and an average of 20 epochs for each task during ER. CUAL A_s^{cl} is trained at each inner-loop for an average of 5 epochs and mini-batch ranging from 10-20. As mentioned in the main text, we implement A_s^{cl} as a fully-connected layer (we also tried 1-layer perceptrons ut with marginal performance gain). Baselines' long term classifier and CUAL's A^{cl} are implemented as a one layer perceptron of size 4096 (also tested variations with marginal variations in results). The ER replay buffer is set to a size 5000 deep-embeddings for Places365 [24], Imagenet21K-OOD [23] and 2500 for eurosat and cifar100. Lastly, baseline CCIC [5] was trained using same hyperparameters proposed by the authors and their open-source code.

4.2.2 Datasets:

Since the employed large/foundation feature extractor were pretrained on Imagenet1K, we evaluate CUAL on datasets that either do not contain class overlap with Imagenet1K (out-of-distribution w.r.t Imagenet1K [36]), or curated them by excluding any overlapping classes. The exception is cifar100, which was included due to it being a very popular and widespread dataset.

1. *Imagenet21K-OOD (Im21K-OOD) [23]*: We curated a subset of Imagenet21K containing the top-most populous 50 classes and that do not overlap with the classes present in Imagenet1K. We use a random set of 500 samples from each of the 50 classes. Because Imagenet21K is a superset of Imagenet1K, by excluding any overlapping class we guarantee orthogonality in our curated subset. We will release the full list of images chosen in this curation for reproducibility.
2. *Places365-OOD (Places) [24, 37]*: is a subset of Places365 also originally curated by [37] to contain 51 "environment" categories orthogonal to Imagenet1K, containing a total of 9822 images. It has also been used as a test OOD dataset in [37, 38, 39] with respect to Imagenet1K.
3. *Eurosat [25]*: An RGB dataset of 10 classes and 27K images of Sentinel-2 satellite images, which is also orthogonal to Imagenet1K.
4. *Cifar100-Superclasses (Cifar100) [27]*: We use the super-label granularity of Cifar100 dataset. This totals 20 labels (super) and 50K images. While Cifar100 is not orthogonal to Imagenet1K, we decided to showcase its results since it is a widespread dataset in CL.

4.2.3 Baselines

As discussed, we use a range of methods for comparison: (1) CCIC [5] leverages experience replay, a MixMatch [28] derived label sharpening technique and nearest neighbor classification to solve semi-supervised CL but does not differentiate between old and new classes which may be unlabeled. we used the author's publicly released code to generate results. CCIC, similar to other semi-supervised CL models, assumes that new and old classes will not co-occur (discussed in Section 1) which ultimately explains its performance degradation when the oracle assumption (of only novel classes present in an unlabeled pool) is no longer valid, as in our experimental scenario; (2) ER [18, 3], originally proposed for supervised CL is adapted in a similar fashion to continual active learning work [13] to only use actively labeled samples (as embeddings) for replay; (3) We also adapt PseudoER [29] to continual AL by further incorporating pseudo-labeling for high confidence unlabeled samples in addition to actively labeled samples to be used for training. In both ER and PseudoER, we utilize the cumulative classification entropy as an uncertainty score to actively-label and Pseudo-Label (PseudoER). Similar to CUAL, we actively label "ambiguous" samples according to the same formula as outlined in appendix 4.1.2 for superior results, then sampling according to the TOP heuristic (see section 3 discussion). We also tested with other common uncertainty metrics such as margin [7] but with inferior results.

4.3 Additional Results

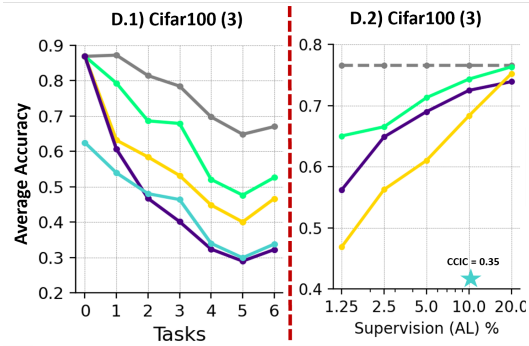

Figure 3: Results for Cifar100; (Left D.1) Continual classification accuracy over continual tasks. The number of novel classes introduced per task is in parenthesis. Al budget is 1.25% as outlined in section 3.1. CUAL over-performs other methods in this challenging setting. The Oracle (gray) equates to classic fully supervised experience-replay.(Right D.2) Varying the Active supervision budget and it's effect on CUAL and AL baselines.

Class-inc	1	3	5	7
Oracle	92.4	92.2	92.4	91.4
CUAL(ours)	**90.7**	**88.3**	**86.6**	**83.6**
ER-ent	77.9	74.2	76.1	78.2
PseudoER-ent	65.5	55.9	61.7	59.6
CCIC	63.1	-	58.0	-

Table 1: Averaged continual classification performance when varying class increments per task. Results for Im21K-OOD with AL budget of 2.5%.

Result clarification for Fig 2 right table (ablations): The ablation only-AL-rand is equivalent to ER-rand since sampling is done randomly during the inner loop. In other words, it does not use either entropy or Eq 2 to guide AL. Furthermore, all of the only-AL ablations do not use pseudo-labeling.