# OpenAL: Evaluation and Interpretation of Active Learning Strategies

**William Jonas** Dataiku williamjonas@hotmail.fr

Alexandre Abraham Dataiku abraham.alexandre@gmail.com

### Léo Dreyfus-Schmidt

Dataiku leo.dreyfus-schmidt@dataiku.com

#### **Abstract**

Despite the vast body of literature on Active Learning (AL), there is no comprehensive and open benchmark allowing for efficient and simple comparison of proposed samplers. Additionally, the variability in experimental settings across the literature makes it difficult to choose a sampling strategy, which is critical due to the one-off nature of AL experiments. To address those limitations, we introduce OpenAL, a flexible and open-source framework to easily run and compare sampling AL strategies on a collection of realistic tasks. The proposed benchmark is augmented with interpretability metrics and statistical analysis methods to understand when and why some samplers outperform others. Last but not least, practitioners can easily extend the benchmark by submitting their own AL samplers.

## 1 Introduction

Active Learning (AL) has proved its worth in practice to optimize labeling tasks [15]. However, it remains challenging to apply in practice as its benefit can vary significantly depending on the task [12]. The optimal sampler may depend on several experimental hyperparameters, such as the initial labeled set size, the batch size, the ML model used, or the number of iterations, among others. Those hyperparameters values vary substantially between studies, even for similar tasks, as shown in Table 1.

This diversity in experimental settings impairs reproducibility and makes methods comparisons arduous. Existing AL benchmarks have tackled this variability by fixing some parameters arbitrarily or targeting specific AL problems, such as using only Logistic Regression as a base learner [22], outlier detection [19], or structural reliability [13]. But comparing sampling strategies reliably requires to repeat the experiments several times [9] using various tasks and models [14]. OpenAL follows those best practices and encompasses various realistic tasks, models, and use cases. We designed them as close as possible to real tasks. We address the following caveats:

**Initialization induced variability.** It has been proven that the variance in performance induced by the initial set of selected samples can be greater than the difference between sampling strategies [9]. We propose to use a 10-fold stratified shuffle split to get enough significance when comparing methods [5].

**Plausibility of the experimental setting.** Research task settings must be well representative of real-life ones to be helpful. Experiments on CIFAR-10 in the literature often vary from 6 batches of 5% of the whole dataset to 3 batches of 10% [16]. According to earlier work on realistic

Table 1: AL experiment parameters

Paper	Dataset	Init size	Batch size	nb iterations
Active Learning for convolutional	CIFAR 10	10%	10%	3
neural networks: a core set	CIFAR 100	10%	10%	3
approach [16]	SVHN	1%	8% then 43%	3
Deep batch active learning by diverse, uncertain gradient lower bounds [3]	SVHN	100	100	350
	OpenML #156	100	1000	4
	CIFAR 10	100	10000	4
Variational Adversarial Active Learning [18]	CIFAR 10	10%	5%	6
	CIFAR 100	10%	5%	6
	Caltech-256	10%	5%	6
	ImageNet	10%	5%	6
BatchBALD: Efficient and Diverse	MNIST	10	10	25
Batch Acquisition for Deep Bayesian	<b>EMNIST</b>	10	10	25
Active Learning [8]	CINIC-10	200	10	120

applications of AL [21], it is usually used to reduce data labeling between 1% and 10%. OpenAL's default is to label 1% of the data in 10 iterations on tabular and image classification tasks. We kickstarted the image classification models using transfer learning or self-supervision, following the industry best practices.

**Reproducibility.** Our framework is open source, and all experiments results are made available and can be easily run again. We provide the accuracies and other AL metrics for the most common AL samplers, along with all train, test, and initial batch indices used for those experiments.

Online evaluation of sampling strategies. Research works rely on the area under the accuracy curve of a left-out test set to evaluate the performance of AL strategies. This testing set is not available in real experiments making it hard to trust their behavior online [10]. OpenAL logs unsupervised metrics to improve the offline strategies' interpretability and be able to interpret their behavior online [1].

We first start by describing the setup of our tasks, the model selection methodology, and the evaluation criteria for sampling strategies. Then we present the results of our experiments per strategy across all tasks. We finish by focusing on the metrics observed and how they explain the performance of some strategies. Finally we open new perspectives on AL experiments and how this benchmark could be useful and extended in the future.

#### 2 Evaluation framework

OpenAL features eleven classification tasks on tabular datasets and four on image datasets. Tabular datasets come from OpenML [20, 7] and must be plausible enough *i.e.* having at least 10000 samples to justify the cost of setting up an AL pipeline and being non-trivial, or not solvable easily with 1% randomly selected samples. We were left with 11 tasks, which we deemed sufficient to obtain reliable results to compare AL strategies.

**Cross-validation.** Each task is repeated ten times with different test sets and batch initialization. We use a stratified shuffle split with 20% of the data dedicated to the test. This amount of repetition is said to provide enough significance for method comparison [9]. Our accuracy plots display confidence intervals of  $10^{th}$  to  $90^{th}$  quantiles over the ten folds.

Active learning experimental setting. We chose experimental parameters to be as close as possible to industrial use cases. Each experiment starts with 0.1% randomly selected labeled data with at least one sample from each class. Nine iterations follow it with batch size 0.1% to end up with a total of 1% of the data labeled. We do not use a specific stopping criterion and stop the experiment when this labeling budget is exhausted. In most experiments, this budget allows the best AL method to reach a performance plateau, as shown in experiments in Section 4. OpenAL includes seminal uncertainty-based strategies [17] (Margin, Confidence, and Entropy), weighted

KMeans (WKMeans) [23], incremental weighted KMeans (IWKMeans) [2], and k-center greedy (Kcenter) [16]. Note that what most literature works call core-sets use k-center greedy because of the latter's high computational cost. We call it by its original name to avoid any confusion. Since KCenter relies on the weights of the penultimate layer of a neural network for its computation, we used the embedding method proposed in scikit-learn for embedding tree models. It vectorizes the data using a PCA computed on the activation of the tree leaves.

Selection of the best model. Models are usually selected using cross-validation, which is tricky to perform in Active Learning where labeled data is scarce [11]. We expect the practitioners to have prior knowledge of which models could perform well for the task at hand. For tabular datasets and MNIST, we simulate this prior knowledge by doing model selection over the whole dataset using a 5-fold cross-validation. We consider a multi-layer perceptron and two tree-based models, Random Forest and Gradient Boosting Tree, as they are known to excel on tabular data. For CIFAR-10 and CIFAR-100, we use embeddings precomputed on ImageNet and finetune the last layer. For CIFAR-10 only, we also consider embeddings precomputed on unlabeled data using contrastive learning [4].

**Experiment caching for easy comparison.** All the benchmark elements are seeded, which guarantees reproducibility at the machine level. Because seeded number generation may change from one machine to another, we also provide the indices of all train and test indices used in our benchmark. Once a strategy has run, all its corresponding metrics results are cached and can be used for plotting or method comparison. Running a new strategy is as simple as taking the dedicated notebook, wrapping the strategy in our sampler formalism, and running it. Submitting the results can then be done through a GitHub pull request.

#### 3 Software

OpenAL is coded in Python and available through the GitHub platform<sup>1</sup>. We also provide documentation explaining how to install, use, and publish results using our framework<sup>2</sup>. The repository contains all results of previous experiences. Running the benchmark on all reference samplers or on a new one is as simple as 3 lines of code that are contained in the main\_run.py file:

```
initial_conditions = load_initial_conditions(dataset_id)
experimental_parameters = load_experiment(dataset_id, initial_conditions)
run(experimental_parameters, methods)
```

All experiments are modular and split in blocks for easy running and customization. *Initial conditions* contain the samples initially labeled and the number of folds. *Experimental parameters* include the batch size and the number of iterations. The *run* function runs the experiment and generates accuracy and metrics results in a dedicated folder that the user can submit through a pull request for validation. After replicating the results on our side, we will integrate this new sampler into OpenAL and share the results with the community.

### 4 Experiments and results

The tasks included in OpenAL are listed in Table 2. We report accuracy and the following set of metrics measured during our experiments:

**Agreement.** Agreement ratio between the inductive model and a 1-nearest-neighbor (1-NN) classifier trained on labeled data. We expect a high agreement to be correlated with good exploration.

**Contradictions.** Ratio of test samples where the models at the previous iteration and current iteration disagree. It is an upper bound on accuracy change from one iteration to the other.

**Hard exploration.** Ratio of test samples where the 1-NN of the previous iteration and current iteration disagree.

**Top exploration.** Mean difference of the distance between test samples and their nearest neighbour in the labeled pool from one iteration to the next.

<sup>1</sup>https://github.com/dataiku-research/OpenAL

<sup>&</sup>lt;sup>2</sup>https://dataiku-research.github.io/OpenAL/

**Violations.** This unsupervised metric measures how many data compliance rules computed on the test set are violated in the labeled dataset [6]. Conformance rules are computed by extracting eigenvectors on the reference dataset and setting conformance boundaries based on the standard deviation of the projected reference data. Sample conformance is given by the number of times its projections fall outside the conformance boundaries. Overall, the highest the violation, the more the labeled samples deviate from the test set.

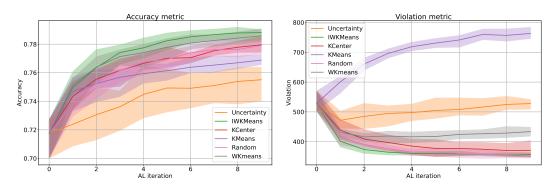


Figure 1: Results for dataset #42803: Accuracy (left) and Violation (right). Note that the violation metrics is at 0 on the first iteration because the dataset is too small to compute them.

Best active learning strategy. WKMeans and IWKMeans have similar performances and dominate the benchmark in terms of accuracy on all tasks, as observed in Table 3 and in previous work [1, 2]. One notable difference is that IWKMeans has fewer violations than WKMeans which means that its training set is more representative of the test set, as observed in Figure 1. We expected this result as IWKMeans is designed to sample data more uniformly than WKMeans. If this does not impact accuracy, we could expect a different generalization power between the two models which advocates for adding a domain adaptation task in the future.

Uncertainty-based AL strategies. As all uncertainty metrics have the same rank in binary classification, we resort to multi-class tasks to compare them. The only non-binary tabular classification task of our benchmark is #42803. We observe that Confidence and Entropy strategies perform poorly, even worse than random. Looking at the metrics, we notice that most of the two strategies' values do not stand out except for higher violations for Confidence and Entropy as seen in Figure 2. This means that the training set is very different from the test set, which may be due to those samplers focusing on noisy samples [2]. Margin reaffirms its dominance which explains why it is preferably used in many studies [23].

Name	#samples	#classes	#features	class balance
#1461 Bank-marketing	45221	2	7/9	0.88 / 0.12
#1471 Eeg-eye-state	14980	2	14/0	0.55 / 0.45
#1502 Skin-segmentation	245057	2	3/0	0.21 / 0.79
#1590 Adult	48842	2	6/8	0.76 / 0.24
#40922 Run or walk information	88588	2	6/0	0.5 / 0.5
#41138 APSFailure	76000	2	170/0	0.98 / 0.02
#41162 Kick	72983	2	14/18	0.88 / 0.12
#42395 Santander Customer Satisfaction	200000	2	200/0	0.9 / 0.1
#42803 Road-safety	363243	3	61/5	0.66 / 0.29 / 0.05
#43439 Medical-Appointment-No-Shows	110527	2	8/4	0.8 / 0.2
#43551 Employee-Turnover-at-TECHCO	34452	2	9/1	0.02 / 0.98
MNIST	70000	10	28x28	0.1 each
CIFAR10	60000	10	32x32x3	0.1 each
CIFAR100	60000	100	32x32x3	0.01 each

Table 2: OpenAL datasets' characteristics. For tabular data, features correspond ton continuous/categorical features. For images, the shape of one image is given.

Dataset	Random	KMeans	Confidence	Margin	KCenter	WKmeans
1471	$68.6 \pm 0.8$	$68.7 \pm 1.2$	$69.7 \pm 1.0$	$69.7 \pm 1.0$	$67.4 \pm 0.6$	<b>71.2</b> ±1.1
41138	$98.4 \pm 0.1$	$98.4 \pm 0.1$	$98.9 \pm 0.1$	$98.9 \pm 0.1$	$98.7 \pm 0.1$	<b>99.0</b> $\pm 0.1$
1502	$98.7 \pm 0.4$	$99.3 \pm 0.0$	$99.2 \pm 0.2$	$99.2 \pm 0.2$	<b>99.5</b> $\pm 0.1$	<b>99.5</b> $\pm 0.1$
1590	$81.8 \pm 1.0$	$80.7 \pm 0.8$	$81.5 \pm 1.0$	$81.5 \pm 1.0$	<b>82.0</b> $\pm 0.6$	<b>82.9</b> $\pm 0.6$
41162	$85.0 \pm 0.7$	$84.3 \pm 0.9$	$85.7 \pm 0.9$	$85.7 \pm 0.9$	<b>86.7</b> $\pm 0.5$	<b>86.3</b> $\pm 0.8$
43439	$76.6 \pm 0.3$	$76.1 \pm 0.6$	$76.5 \pm 0.4$	$76.5 \pm 0.4$	<b>77.3</b> $\pm 0.9$	<b>77.0</b> $\pm 0.5$
40922	$96.6 \pm 0.4$	$96.0 \pm 0.6$	<b>97.7</b> $\pm 0.5$	<b>97.7</b> $\pm 0.5$	<b>97.3</b> $\pm 0.1$	<b>97.7</b> $\pm 0.5$
42395	$89.7 \pm 0.1$	$89.6 \pm 0.1$	<b>89.8</b> $\pm 0.1$	<b>89.8</b> $\pm 0.1$	<b>89.8</b> $\pm 0.1$	<b>89.8</b> $\pm 0.1$
43551	$97.2 \pm 0.5$	$95.2 \pm 1.8$	<b>97.0</b> $\pm 0.8$	<b>97.0</b> $\pm 0.8$	<b>97.5</b> $\pm 0.7$	<b>97.5</b> $\pm 0.9$
40922	$96.6 \pm 0.4$	$96.0 \pm 0.6$	<b>97.7</b> $\pm 0.5$	<b>97.7</b> $\pm 0.5$	<b>97.3</b> $\pm 0.1$	<b>97.7</b> $\pm 0.5$
1461	$88.8 \pm 0.3$	$88.8 \pm 0.2$	<b>89.4</b> $\pm 0.1$	<b>89.4</b> $\pm 0.1$	$88.8 \pm 0.3$	<b>89.4</b> $\pm 0.1$
41138	$98.4 \pm 0.1$	$98.4 \pm 0.1$	<b>98.9</b> $\pm 0.1$	<b>98.9</b> $\pm 0.1$	$98.7 \pm 0.1$	<b>99.0</b> $\pm 0.1$
42803	$76.1 \pm 0.4$	$75.6 \pm 0.5$	$74.4 \pm 1.2$	<b>76.9</b> $\pm 0.4$	$76.2 \pm 0.3$	<b>76.9</b> $\pm 0.4$
CIFAR-10	$70.2 \pm 0.7$	$70.4 \pm 0.4$	$68.9 \pm 1.0$	<b>71.2</b> $\pm 0.3$	$66.9 \pm 1.2$	<b>71.6</b> $\pm 0.4$
CIFAR-10S	$85.0 \pm 0.8$	$84.3 \pm 0.5$	$84.8 \pm 0.8$	<b>86.3</b> $\pm 0.5$	$85.9 \pm 0.5$	<b>86.4</b> $\pm 0.6$
MNIST	$82.6 \pm 0.8$	$82.5 \pm 0.5$	$82.0 \pm 0.9$	$85.4 \pm 0.5$	$80.5 \pm 1.2$	<b>87.1</b> $\pm 0.4$

Table 3: Benchmark results per dataset and sampling strategy. We show the average accuracy over 10 folds. Entropy (*resp.* IWKMeans) has been omitted because their results were close to Confidence (*resp.* WKMeans). Datasets are ordered to display patterns of dominance for samplers.

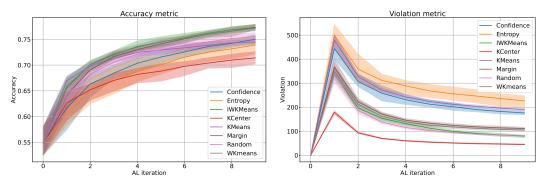


Figure 2: Results for dataset CIFAR-10: Accuracy (left) and Violation (right).

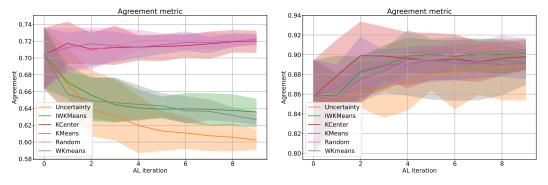


Figure 3: Agreements for datasets: #42803 (left) and #43439 (right).

The importance of data representation. IWKMeans and WKMeans are overall the best methods, but we observe a subset of methods on which uncertainty-based Margin is on par with them and another one where exploration-based KCenter reaches the same accuracy as displayed in Table 3. Surprisingly, the agreement metric seems to be a good indicator of which method is on par with the best. When all samplers have a similar agreement score, KCenter manages to reach the best accuracy. Conversely, when the agreement score of Margin is significantly below Random or Kcenter, Margin reaches the best performance. Figure 3 illustrates the case of dominance of Margin on #42803 and dominance of KCenter on #43439. We hypothesize that the quality of the representation learned is responsible for this effect. When the distance in the representation space is inconsistent with the labels, diversity becomes ineffective or even counter-productive.

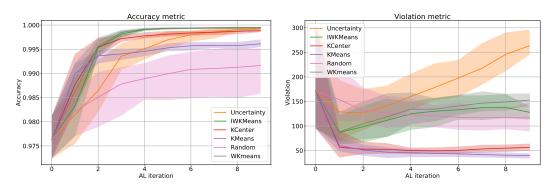


Figure 4: Results for dataset #1502: Accuracy (left) and Violation (right).

The peculiar case of #1502. Task #1502 is peculiar since we considered it tabular, but its three features are an image's red, green, and blue channels. It is the only task where KMeans has better accuracy than random. More than that, we observe two regimes in this experiment. KMeans and KCenter-Greedy, two purely exploratory techniques, dominate the two first iterations of the experiment. After that, they plateau at a suboptimal accuracy, while uncertainty based-methods take the lead. We hypothesize that at iteration 1, the training set is too small for the model to produce meaningful uncertainty scores. WKMeans, which combines uncertainty and exploration, manages to take the best of both worlds. This unusual behavior could be correlated to the unique pattern shown in the violations metric where KMeans minimize this score while WKMeans keeps it increasing, as displayed in Figure 4. Unfortunately, we cannot draw a conclusion from one task, and we hope that adding further tasks could help us reproduce this behavior and understand it better.

### 5 Limitations

We have limited ourselves to OpenML datasets and the most common image ones for this proof of concept. In the future, we plan to explore other data sources, such as Kaggle, and other modalities or tasks, such as text and object detection. We could also explore other models and settings, such as different batch sizes, to observe their effect on overall performance. Given the high computational cost of the benchmark, we have set aside very costly methods such as BADGE [3] or BatchBALD, but we plan to add them in the near future.

### 6 Conclusion

This first version of OpenAL proves the value of comparing methods on fixed predefined tasks. Although the global outcome that the more sophisticated methods dominate the others was expected, the systematic monitoring and analysis of the metrics helped dig into the results. We believe that our benchmark is a first step towards helping the practitioners to be more confident in their choice of samplers.

By ensuring a complete reproducibility of the results, we also allow strategy developer to test their method against our references quickly. Thanks to the metrics, they can also understand faster why their method may fail on a peculiar dataset and why other methods perform better. For example, we

have highlighted that diversity in active learning strategies is as good as the data representation on which it relies.

In the end, we hope this benchmark will accelerate research in active learning and facilitate its adoption in industrial contexts.

#### References

- [1] Alexandre Abraham and Léo Dreyfus-Schmidt. Rebuilding trust in active learning with actionable metrics. 2020 IEEE International Conference on Data Mining Workshops (ICDMW), 2020.
- [2] Alexandre Abraham and Léo Dreyfus-Schmidt. Sample noise impact on active learning. *IAL* 2021 workshop, ECML PKDD, 2021.
- [3] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [5] Thomas G Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.
- [6] Anna Fariha, Ashish Tiwari, Arjun Radhakrishna, Sumit Gulwani, and Alexandra Meliou. Conformance constraint discovery: Measuring trust in data-driven systems. In *Proceedings of the 2021 International Conference on Management of Data*, pages 499–512, 2021.
- [7] Matthias Feurer, Jan N Van Rijn, Arlind Kadra, Pieter Gijsbers, Neeratyoy Mallik, Sahithya Ravi, Andreas Müller, Joaquin Vanschoren, and Frank Hutter. Openml-python: an extensible python api for openml. *The Journal of Machine Learning Research*, 22(1):4573–4577, 2021.
- [8] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *arXiv preprint arXiv:1906.08158*, 2019.
- [9] Daniel Kottke, Adrian Calma, Denis Huseljic, GM Krempl, and Bernhard Sick. Challenges of reliable, realistic and comparable active learning evaluation. In *Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning*, pages 2–14, 2017.
- [10] Daniel Kottke, Jim Schellinger, Denis Huseljic, and Bernhard Sick. Limitations of assessing active learning performance at runtime. *arXiv preprint arXiv:1901.10338*, 2019.
- [11] Christian Limberg, Heiko Wersing, and Helge Ritter. Beyond cross-validation—accuracy estimation for incremental and active learning models. *Machine Learning and Knowledge Extraction*, 2(3):327–346, 2020.
- [12] David Lowell, Zachary C Lipton, and Byron C Wallace. Practical obstacles to deploying active learning. *arXiv preprint arXiv:1807.04801*, 2018.
- [13] Maliki Moustapha, Stefano Marelli, and Bruno Sudret. Active learning for structural reliability: Survey, general framework and benchmark. *Structural Safety*, 96:102174, 2022.
- [14] Prateek Munjal, Nasir Hayat, Munawar Hayat, Jamshid Sourati, and Shadab Khan. Towards robust and reproducible active learning using neural networks. *arXiv*, pages arXiv–2002, 2020.
- [15] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.
- [16] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

- [17] Burr Settles. Active learning literature survey. Technical report, Department of Computer Sciences, University of Wisconsin-Madison, 2009.
- [18] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.
- [19] Holger Trittenbach, Adrian Englhardt, and Klemens Böhm. An overview and a benchmark of active learning for outlier detection with one-class classifiers. *Expert Systems with Applications*, 168:114372, 2021.
- [20] Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.
- [21] Jenna Wiens and John Guttag. Active learning applied to patient-adaptive heartbeat classification. *Advances in neural information processing systems*, 23, 2010.
- [22] Yazhou Yang and Marco Loog. A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83:401–415, 2018.
- [23] Fedor Zhdanov. Diverse mini-batch active learning. arXiv preprint arXiv:1901.05954, 2019.

# Appendix

A Tabular dataset metrics

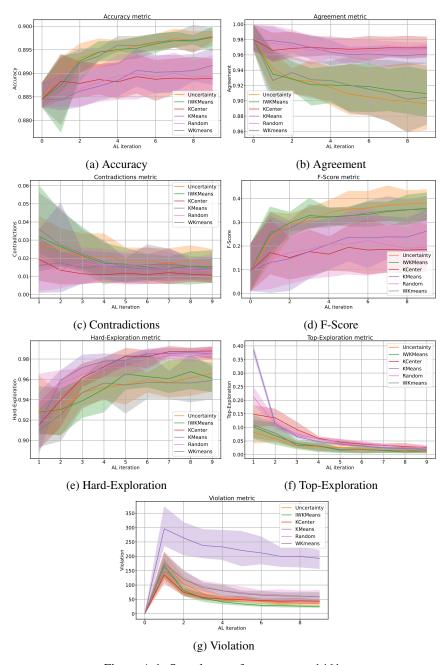


Figure A.1: Samplers performances on 1461.

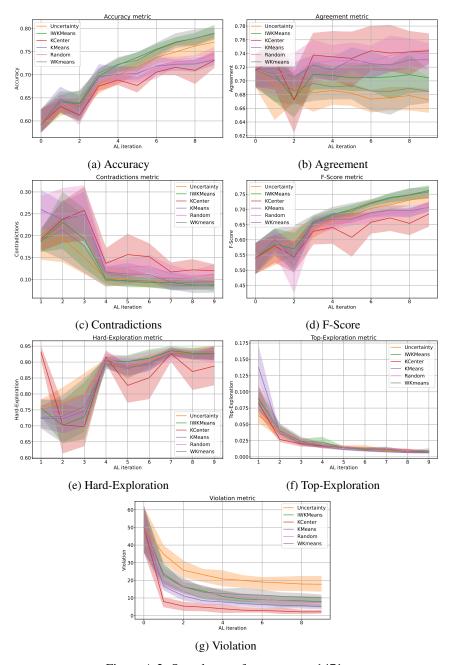


Figure A.2: Samplers performances on 1471.

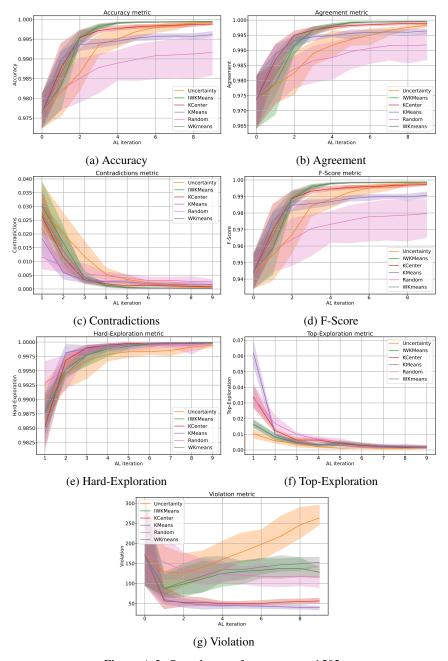


Figure A.3: Samplers performances on 1502.

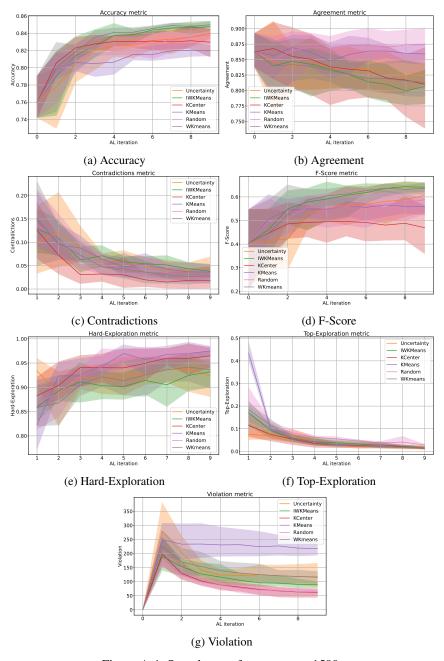


Figure A.4: Samplers performances on 1590.

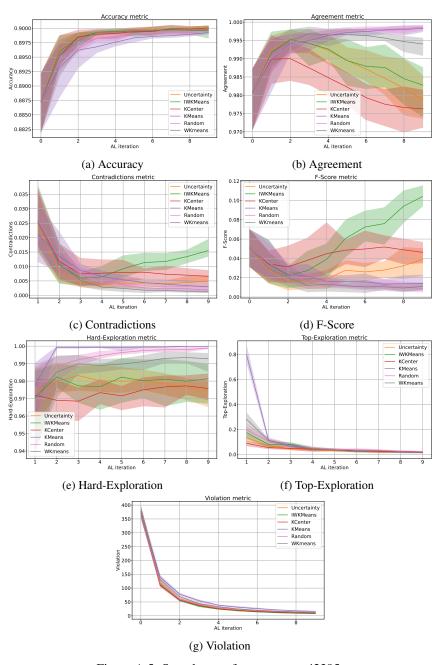


Figure A.5: Samplers performances on 42395.

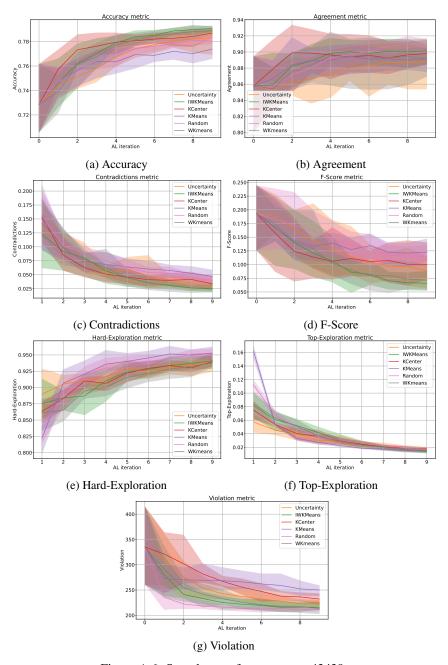


Figure A.6: Samplers performances on 43439.

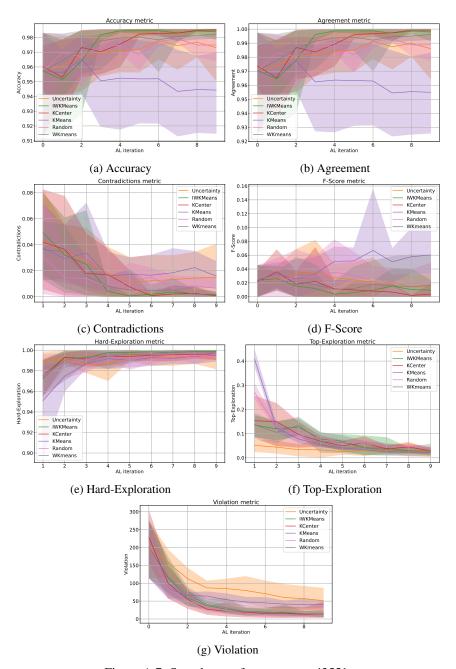


Figure A.7: Samplers performances on 43551.

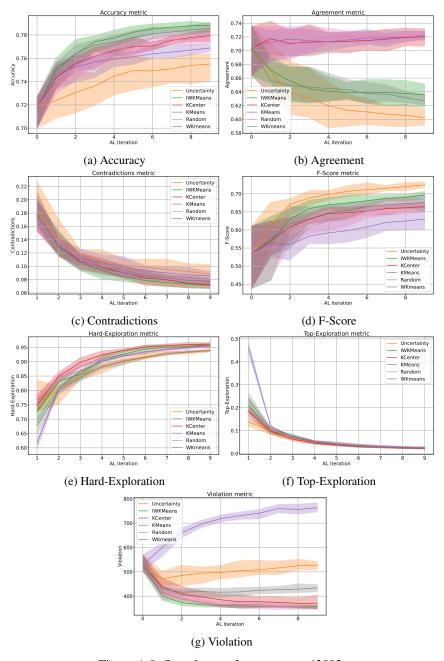


Figure A.8: Samplers performances on 42803.

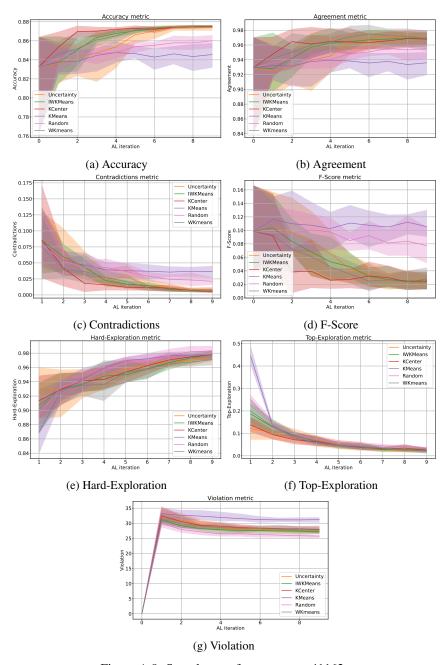


Figure A.9: Samplers performances on 41162.

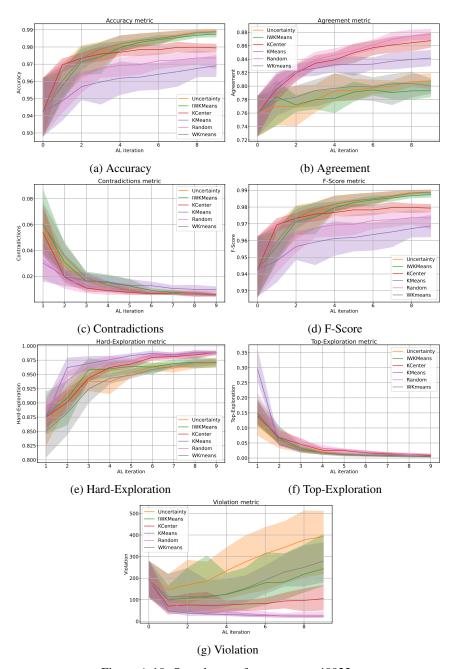


Figure A.10: Samplers performances on 40922.

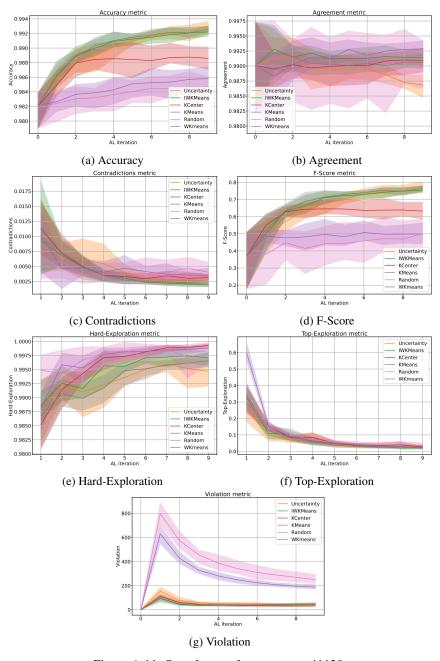


Figure A.11: Samplers performances on 41138.

B Image dataset metrics

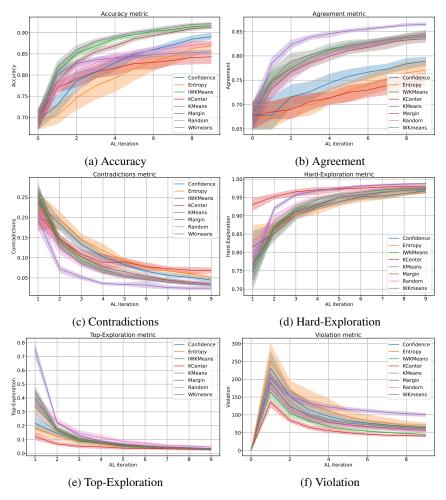


Figure B.1: Samplers performances on mnist.

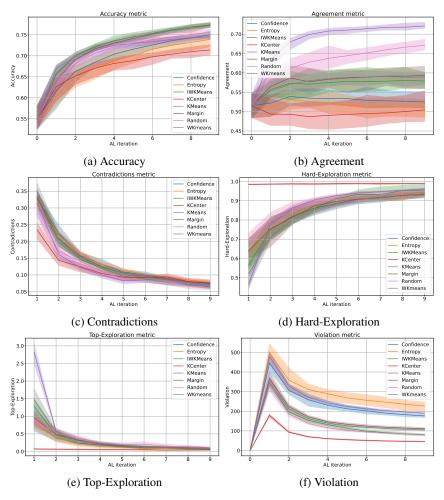


Figure B.2: Samplers performances on cifar10.

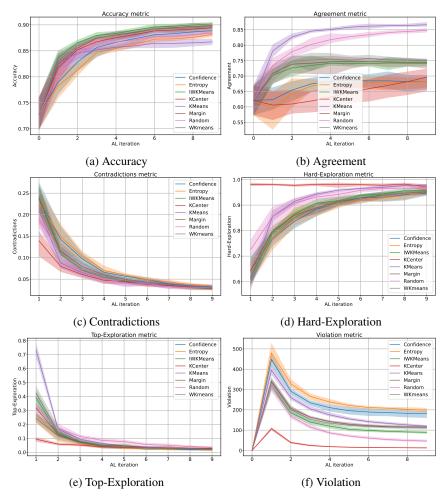


Figure B.3: Samplers performances on cifar10-simclr.