南京航空航天大學
Nanjing University of Aeronautics and Astronautics

Boosting Active Learning via Improving Test Performance

Tianyang Wang,[1*] Xingjian Li,[2 6*] Pengkun Yang,[3] Guosheng Hu,[4]
Xiangrui Zeng,[7] Siyu Huang,[5] Cheng-Zhong Xu,[6] Min Xu[7†]

[1]Austin Peay State University, [2]Baidu Inc, [3]Tsinghua University,
[4]Oosto, [5]Harvard University, [6]University of Macau, [7]Carnegie Mellon University
toseattle@siu.edu, lixingjian@baidu.com, yangpengkun@tsinghua.edu.cn, huguosheng100@gmail.com,
xiangruz@andrew.cmu.edu, huang@seas.harvard.edu, czxu@um.edu.mo, mxu1@cs.cmu.edu

AAAI 2022

Active Learning

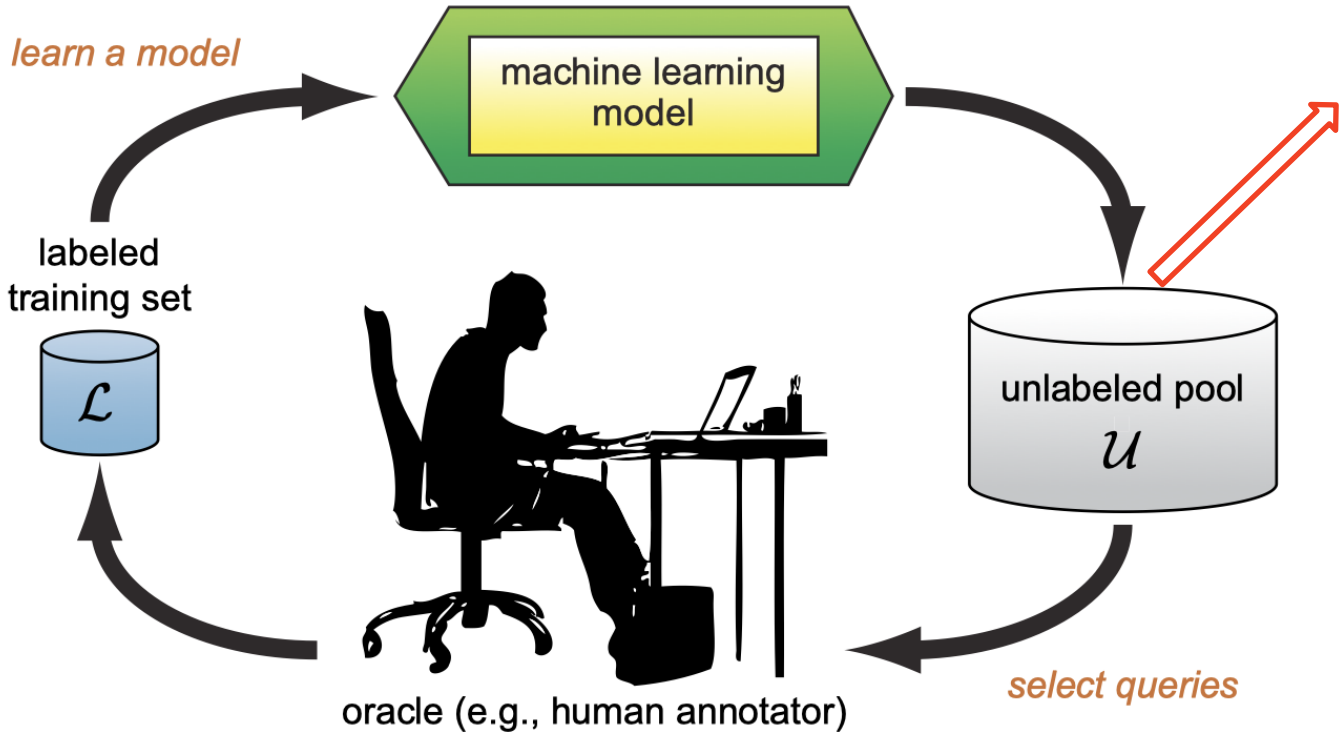


Figure 1: The pool-based active learning cycle.

Question:

查询一个无标注样本进行训练，是如何影响模型在测试集上的性能？

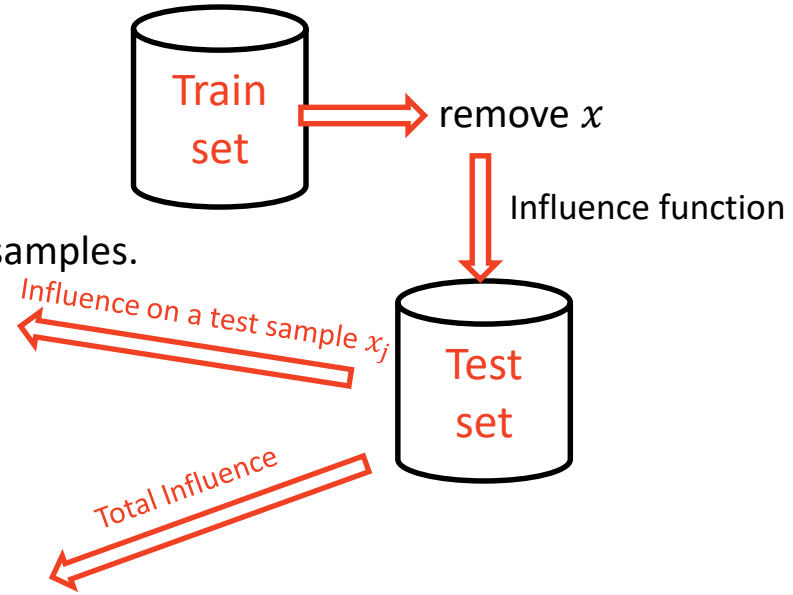
Influence Functions

Given a model f_θ, removing a sample x from its training set will approximately influence the loss at a test sample x_j by

$$I_{loss}(x, x_j) = \frac{1}{n} \nabla_\theta L\left(f_\theta(x_j)\right)^\top H_\theta^{-1} \nabla_\theta L(f_\theta(x))$$

$H_\theta = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 L(f_\theta(x_i))$ is the average Hessian over all training samples.

f_θ : the logits output of the model



Since we want to compute its influence on all the samples in a test dataset, we compute the total influence as follows

$$\sum_j I_{loss}(x, x_j) = \frac{1}{n} \sum_j \nabla_\theta L\left(T^{c+1}(x_j)\right)^\top H_\theta^{-1} \nabla_\theta L(T^{c+1}(x))$$

f_θ : the task model in AL cycle $c+1$ (i.e. T^{c+1}) ,

j : the index over the test test

$\sum_j I_{loss}(x, x_j)$ is generally Positive,

$I_{loss}(x, x_j)$ could be Negative.

-Understanding Black-box Predictions via Influence Functions. ICML 2017

AL Settings

data selection occurs in cycle c , and the selected data is used in cycle $c+1$,The test loss of T^{c+1} is L_{test}^{c+1} x is removed from the labeled pool and not involved in training T^{c+1} , then the influenced test loss L'^{c+1}_{test}

$$\begin{aligned} L'^{c+1}_{test} &= L^{c+1}_{test} + \sum_j I_{loss}\left(x, x_j\right) \\ &= L^{c+1}_{test} + \frac{1}{n}\sum_j \nabla_\theta L\left(T^{c+1}\left(x_j\right)\right)^{\top} H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x)\right) \end{aligned} \tag{2}$$

Since test data is unknown, computing $\frac{1}{n}\sum_j \nabla_\theta L\left(T^{c+1}\left(x_j\right)\right)^{\top} H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x)\right)$ is intractable:

$$\begin{aligned} L'^{c+1}_{test} &= \| L^{c+1}_{test} + \frac{1}{n}\sum_j \nabla_\theta L\left(T^{c+1}\left(x_j\right)\right)^{\top} H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x)\right) \| \\ &\leq L^{c+1}_{test} + \frac{1}{n}\| \sum_j \nabla_\theta L\left(T^{c+1}\left(x_j\right)\right)^{\top} H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x)\right) \| \\ &= L^{c+1}_{test} + \frac{1}{n}\| \nabla_\theta L\left(T^{c+1}(x)\right)^{\top} \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}\left(x_j\right)\right) \| \\ &\leq L^{c+1}_{test} + \frac{1}{n}\| \nabla_\theta L\left(T^{c+1}(x)\right) \| \cdot \| \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}\left(x_j\right)\right) \| \end{aligned} \tag{3}$$

Upper bound

$$L'^{c+1}_{test} \leq L^{c+1}_{test} + \frac{1}{n} \| \nabla_\theta L\left(T^{c+1}(x)\right) \| \cdot \| \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x_j)\right) \| \quad (4)$$

can be regarded as a fixed term

The upper bound of L'^{c+1}_{test} is mainly determined by $\| \nabla_\theta L\left(T^{c+1}(x)\right) \|$

However, computing $\| \nabla_\theta L\left(T^{c+1}(x)\right) \|$ is infeasible:

- The model T^{c+1} is not available
  use $\| \nabla_\theta L\left(T(x)\right) \|$ to bound $\| \nabla_\theta L\left(T^{c+1}(x)\right) \|$
- x does not have ground truth

$$\begin{aligned} L'^{c+1}_{test} &\leq L^{c+1}_{test} + \frac{1}{n} \| \nabla_\theta L\left(T^{c+1}(x)\right) \| \cdot \| \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x_j)\right) \| \\ &\lesssim L^{c+1}_{test} + \frac{1}{n} \| \nabla_\theta L\left(T^{c}(x)\right) \| \cdot \| \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x_j)\right) \| \end{aligned} \quad (5)$$

Eq (5) : removing a training sample x of higher $\| \nabla_\theta L\left(T^{c}(x)\right) \|$ result in a higher upper-bound of L'^{c+1}_{test}

So higher $\| \nabla_\theta L\left(T^{c}(x)\right) \|$ should be selected.

Unlabeled data of higher gradient norm should be selected for annotation in AL

Computing Gradient Norm

$$
\begin{aligned}
L'^{c+1}_{test} &\leq L^{c+1}_{test} + \frac{1}{n} \| \nabla_\theta L\left(T^{c+1}(x)\right) \| \cdot \| \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x_j)\right) \| \\
&\lesssim L^{c+1}_{test} + \frac{1}{n} \| \nabla_\theta L\left(T^{c}(x)\right) \| \cdot \| \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x_j)\right) \|
\end{aligned} \tag{5}
$$

Computing $\| \nabla_\theta L\left(T^c(x)\right) \|$ remains challenging due to the lack of the label information

Expected-Gradnorm Scheme(N分类任务): expected empirical loss to approximate the real empirical loss:

$$L_{exp}\left(T^c(x)\right) = \sum_{i=1}^{N} P\left(y_i \mid x\right) L_i\left(T^c(x), y_i\right)$$

Entropy-Gradnorm Scheme (分割任务): the differentiable entropy of the softmax output of the network:

$$L_{ent}\left(T^c(x)\right) = -\sum_{i=1}^{N} P\left(y_i \mid x\right) logP\left(y_i \mid x\right)$$

Quantitative Evaluations

$$L'^{c+1}_{test} \leq L^{c+1}_{test} + \frac{1}{n} \| \nabla_\theta L\left(T^{c+1}(x)\right) \| \cdot \| \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x_j)\right) \|$$
$$\lesssim L^{c+1}_{test} + \frac{1}{n} \| \nabla_\theta L\left(T^{c}(x)\right) \| \cdot \| \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x_j)\right) \| \quad (5)$$

Derived bound is reasonable.

$$\| \nabla_\theta L\left(T^{c+1}(x)\right) \| \leq \| \nabla_\theta L\left(T^{c}(x)\right) \|$$

Cycle	1	2	3	4	5	6
#	2488	2481	2466	2439	2386	2255

Table 4: Number of the selected samples that satisfy $||\nabla_\theta L(T^{c+1}(x))|| \leq ||\nabla_\theta L(T^c(x))||$. Cycle 0 is ignored since right after cycle 0 model T^1 is not available yet. All the values are out of 2500, which is the number of the samples selected in each cycle. In this evaluation (on Cifar10), the *Expected-Gradnorm* scheme is used.

Cycle	1	2	3	4	5	6
#	2487	2477	2467	2447	2388	2250

Table 5: Number of the selected samples that satisfy $||\nabla_\theta L(T^{c+1}(x))|| \leq ||\nabla_\theta L(T^c(x))||$. Cycle 0 is ignored since right after cycle 0 model T^1 is not available yet. All the values are out of 2500, which is the number of the samples selected in each cycle. In this evaluation (on Cifar10), the *Entropy-Gradnorm* scheme is used.

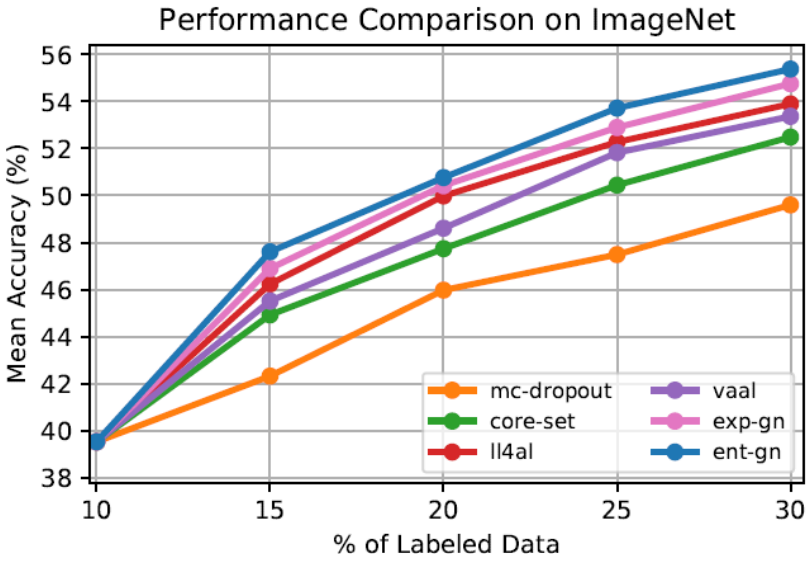
Active Learning Framework

Algorithm 1: Proposed Active Learning Framework.

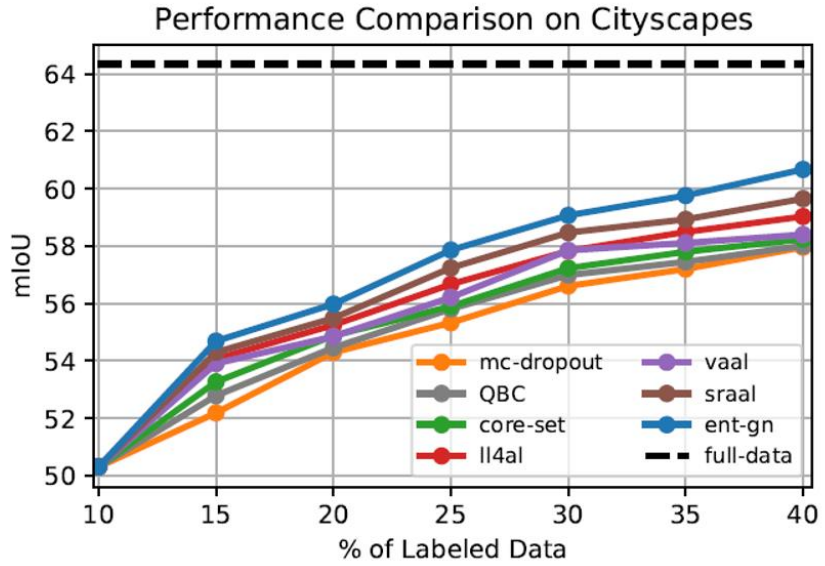
Input:
$\mathcal{T}$: task model; $\mathcal{U}$: unlabeled pool; $\mathcal{L}$: labeled pool; $\mathcal{Y}$: test dataset; $\mathcal{C}$: number of AL cycles $\mathcal{E}$: number of epochs within each cycle; L: empirical loss; K: annotation budget in each cycle; x: each unlabeled sample;

Output:
the task model $\mathcal{T}$

begin
- initialize $\mathcal{T}$;
- **for** $i \leftarrow 1$ **to** $\mathcal{C}$ **do**
  - **for** $j \leftarrow 1$ **to** $\mathcal{E}$ **do**
    - train $\mathcal{T}$ with L on $\mathcal{L}$;
  - **if** *expected-gradnorm* **then**
    - compute the expected loss L_{exp} for x according to Eq. 6;
    - select K samples of the highest $||\nabla_\theta L_{exp}(T^c(x))||$ for annotation;
  - **if** *entropy-gradnorm* **then**
    - compute the entropy loss L_{ent} for x according to Eq. 7;
    - select K samples of the highest $||\nabla_\theta L_{ent}(T^c(x))||$ for annotation;
  - update $\mathcal{L}$ and $\mathcal{U}$, respectively;
- return $\mathcal{T}$

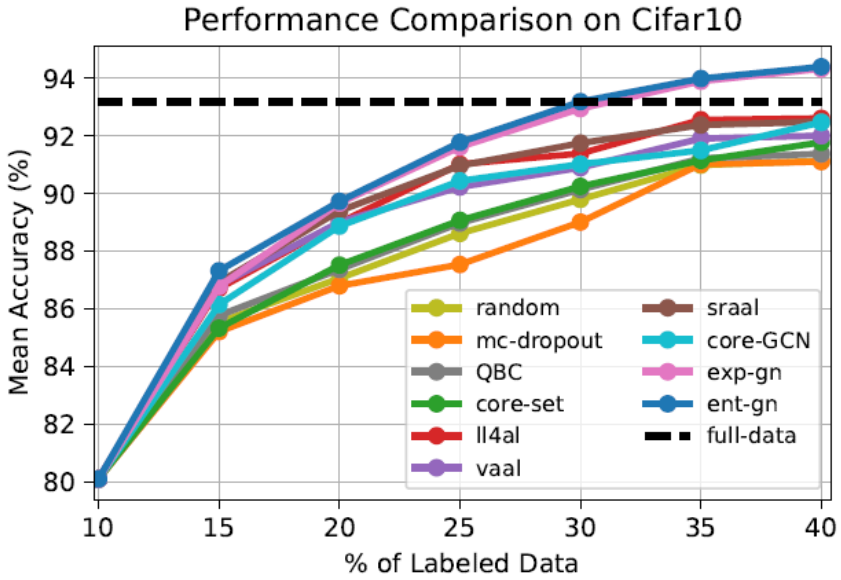


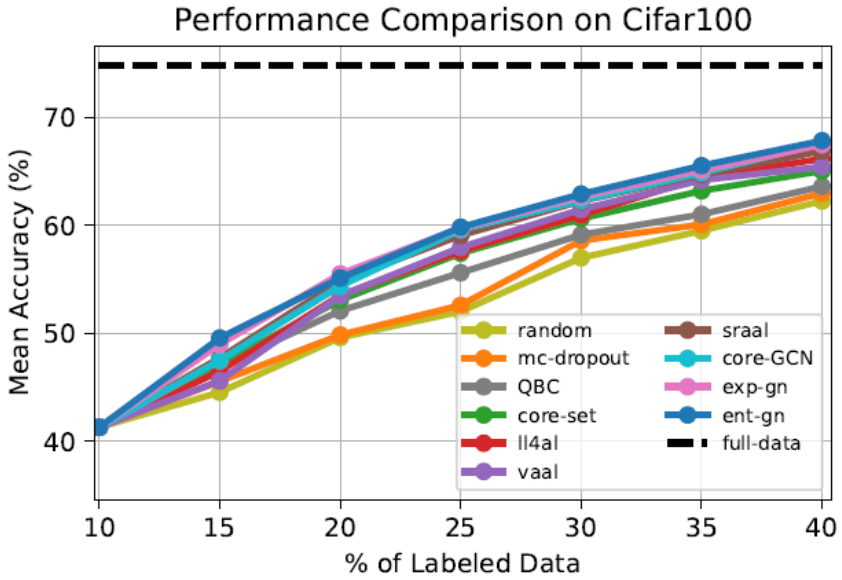
classification

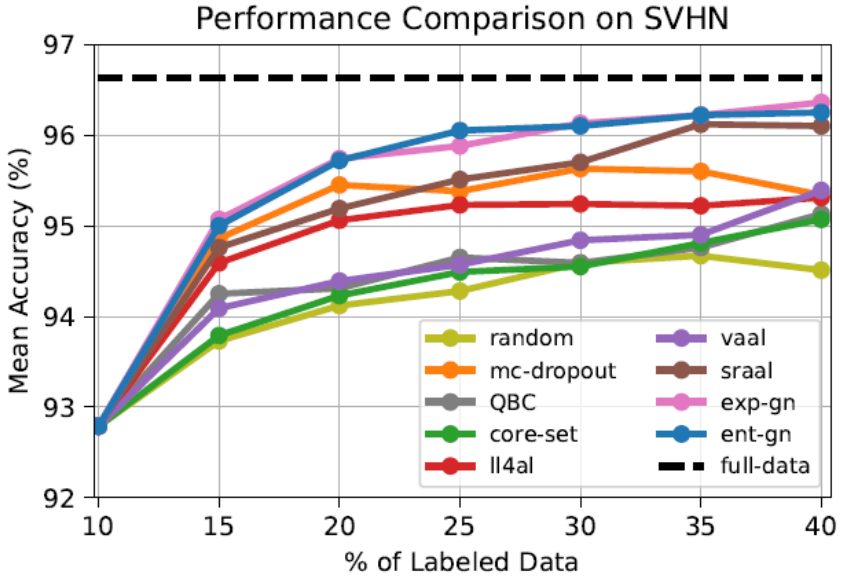


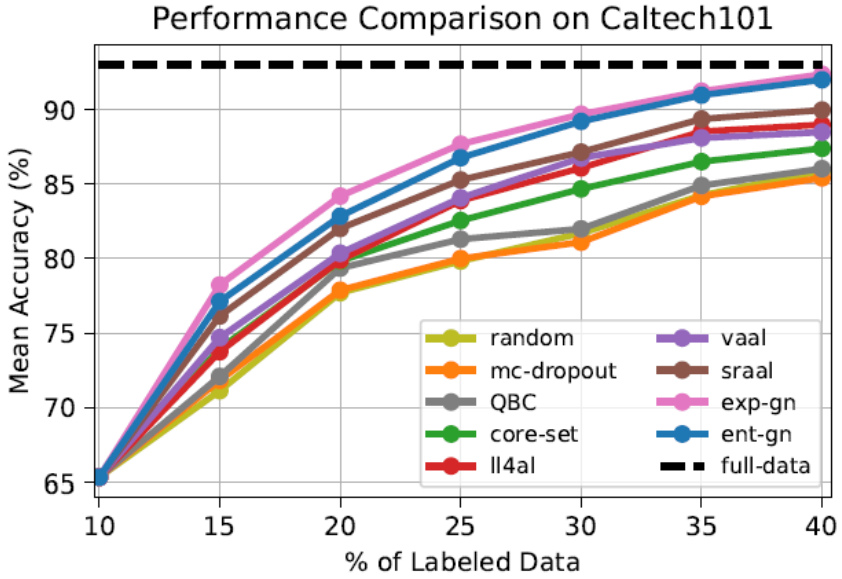
segmentation

Image Classification









Analysis

从cifar-10上看，AL-methods选择越多更高梯度范数的样本，模型性能越好：

budget (%)	10	15	20	25	30	35	40
core-set (Sener and Savarese 2018)	217	230	284	339	329	289	255
vaal (Sinha, Ebrahimi, and Darrell 2019)	268	259	215	231	256	240	256
ll4al (Yoo and Kweon 2019)	391	893	1084	1710	1698	2035	1817
sraal (Zhang et al. 2020)	506	819	1120	1765	1842	2133	2196
expected-gradnorm	**703**	**980**	**1216**	**1942**	**2318**	**2427**	**2448**
entropy-gradnorm	**677**	**932**	**1141**	**1993**	**2317**	**2429**	**2443**

Table 1: Comparison of the AL methods via the number of the selected samples (out of $K = 2500$) of higher gradient norm. The Cifar10 dataset is used.

Quantitative evaluation of the Bounds in Eq.4 and Eq.5

$$L'^{c+1}_{test} = \| L^{c+1}_{test} + \frac{1}{n}\sum_j \nabla_\theta L\left(T^{c+1}(x_j)\right)^\top H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x)\right) \| \quad (3)$$

$$\lesssim L^{c+1}_{test} + \frac{1}{n}\| \nabla_\theta L\left(T^{c}(x)\right) \| \cdot \| \sum_j H_\theta^{-1} \nabla_\theta L\left(T^{c+1}(x_j)\right) \| \quad (5)$$

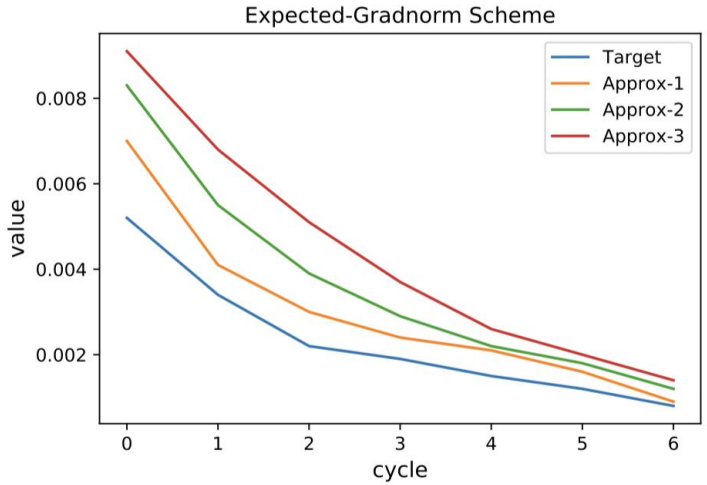
Eq.	Cycle						
	0	1	2	3	4	5	6
3	2500	2500	2500	2500	2500	2500	2500
5	2290	2377	2396	2432	2449	2438	2482

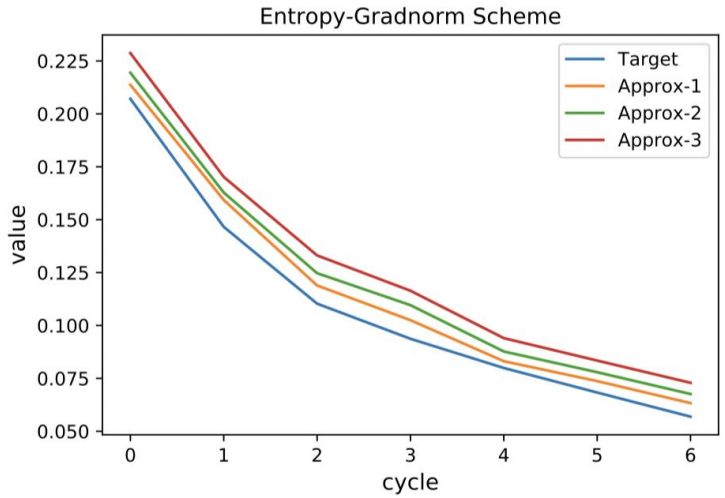
Table 2: How the samples selected by Eq. 5 are consistent with that selected by Eq. 3, evaluated on Cifar10.

Analysis

Compute four terms:

- Target: $\frac{1}{n}\sum_j \nabla_\theta L\left(T^{c+1}(x_j)\right)^\top H_\theta^{-1}\nabla_\theta L\left(T^{c+1}(x)\right)$
- Approx-1: $\| \sum_j \nabla_\theta L\left(T^{c+1}(x_j)\right)^\top H_\theta^{-1}\nabla_\theta L\left(T^{c+1}(x)\right) \|$
- Approx-2: $\| \sum_j \nabla_\theta L\left(T^{c+1}(x_j)\right)^\top H_\theta^{-1}\nabla_\theta L\left(T^{c+1}(x)\right) \|$
- Approx-3: $\| \sum_j \nabla_\theta L\left(T^{c+1}(x_j)\right)^\top H_\theta^{-1}\nabla_\theta L\left(T^{c+1}(x)\right) \|$





Better Generalization

budget (%)	15	20	25	30	35	40
mc-dropout	16.14	10.25	9.47	8.77	8.24	7.35
core-set	14.59	12.73	10.88	9.95	8.92	8.31
vaal	15.01	12.24	11.08	9.66	8.57	8.08
ll4al	13.83	10.7	8.89	8.78	7.64	7.23
sraal	14.09	11.55	10.24	8.62	8.01	6.94
exp-gn	**12.74**	10.04	8.06	7.05	**6.09**	5.66
ent-gn	13.05	**9.9**	**7.76**	**6.79**	6.29	**5.65**

Table 3: Gap (%) between training and test accuracy after each AL cycle, evaluated on Cifar10. Annotation budget of 10% is ignored since all the methods use the same randomly selected initial data for the first cycle training.

Active label cleaning for improved dataset quality under resource constraints

Mélanie Bernhardt[1,3], Daniel C. Castro[1,3], Ryutaro Tanno[1], Anton Schwaighofer[1], Kerem C. Tezcan[1], Miguel Monteiro[1], Shruthi Bannur[1], Matthew P. Lungren[2], Aditya Nori[1], Ben Glocker[1], Javier Alvarez-Valle[1] & Ozan Oktay[1✉]

Nature communications 2022

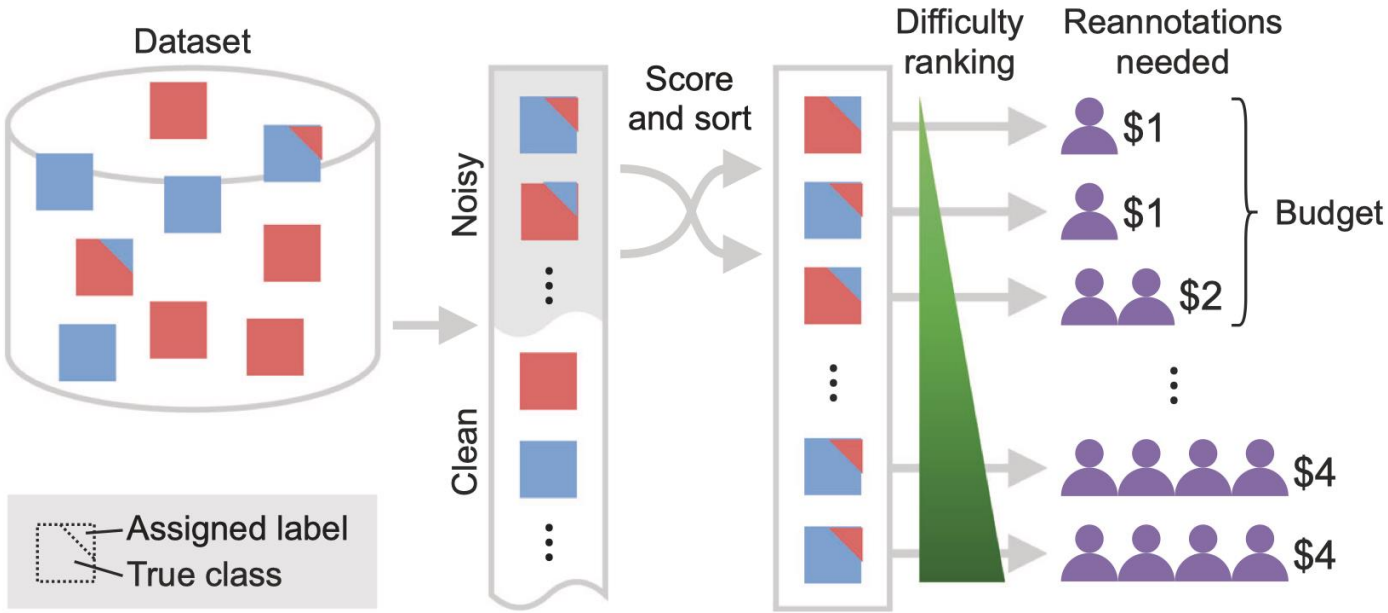
Active label cleaning

Overview of active label cleaning:

数据集 $\mathcal{D} = \{(\mathbf{x}_i, \hat{1}_i)\}_{i=1}^{N}$ 带噪，在资源有限*的情况下，减少最多的噪声样本。

资源有限*：假设在众包场景下，每个样本会分配给多个 worker 进行标记，此时资源有限是指限制worker标记次数。

CIFAR10H: 10K 个样本，10个类，平均每个样本有 51 个人工标记（类似软标记）。



Active label cleaning

Active label cleaning:

$$\max \underbrace{\frac{1}{N}\sum_{i=1}^{N} 1\left[\hat{y}_i = y_i\right]}_{\text{Correctness of majority labels}} \quad s.t. \quad \underbrace{\sum_{i=1}^{N} \| \hat{1}_i \|_1 \leq B}_{\text{Budge constraint}} \qquad (\| \hat{1}_i \|_1 = \sum_{c=1}^{C} \hat{l}_i^c \geq 1)$$

$$\hat{y}_i = argmax_{c\in\{1,\dots,C\}} \hat{l}_i^c \qquad \hat{l}_i^{\hat{y}_i} > \hat{l}_i^c, \forall c \neq \hat{y}_i, \text{ with } \sum_{c=1}^{C} \hat{l}_i^c > 1$$

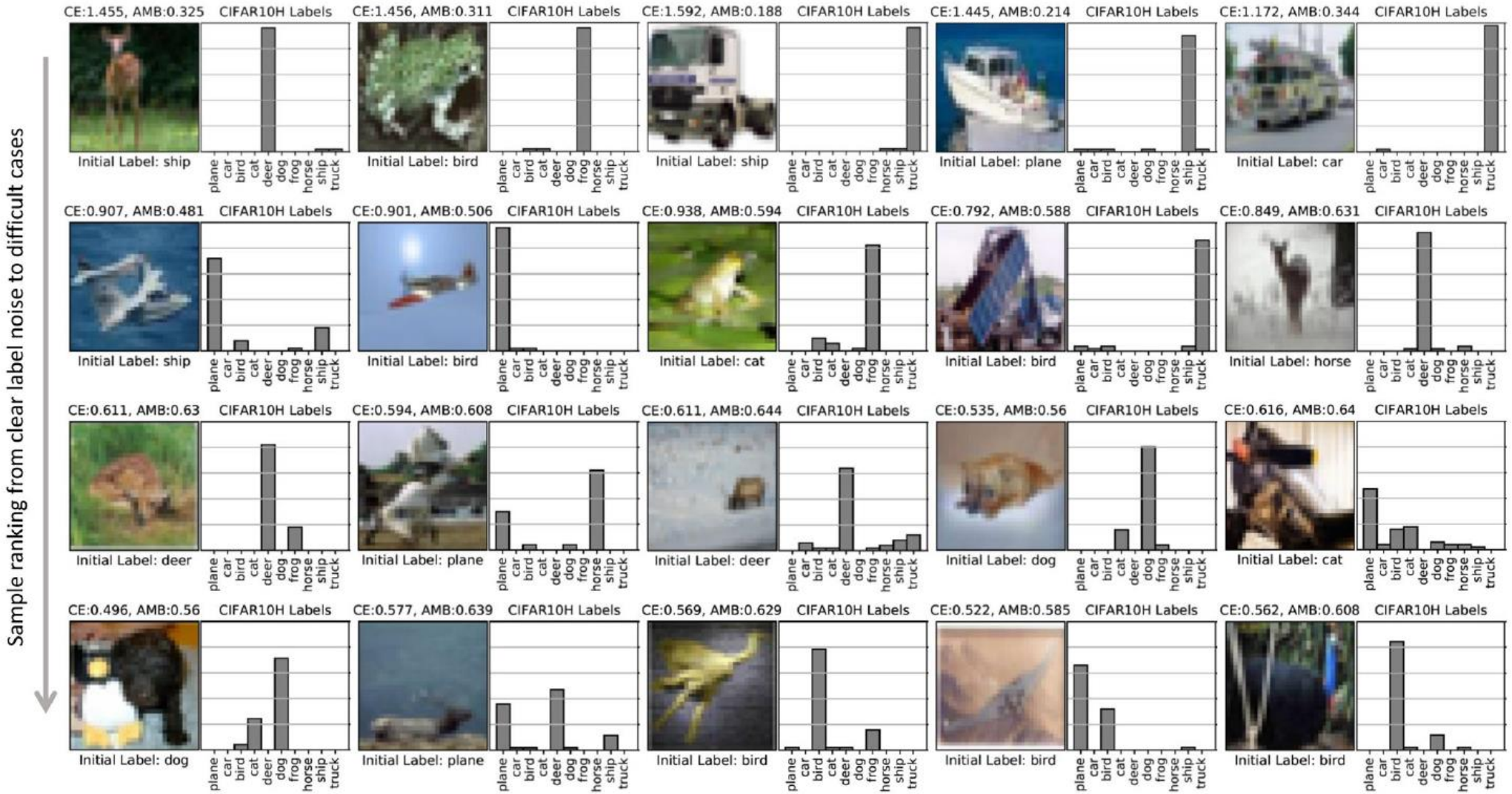
Scoring Function Φ:

$$\Phi(\mathbf{x}, \hat{1}; \boldsymbol{\theta}) = \underbrace{\mathrm{CE}(\hat{\mathbf{l}}, \mathrm{p}_{\boldsymbol{\theta}})}_{\text{noisiness}\uparrow} - \underbrace{\mathrm{H}(\mathrm{p}_{\boldsymbol{\theta}})}_{\text{ambiguity}\downarrow}$$

$$\mathrm{CE}(\hat{1}, \mathrm{p}_{\boldsymbol{\theta}}) = -\mathbb{E}_{\hat{1}/\|\hat{1}\|_1}[log \mathrm{p}_{\boldsymbol{\theta}}(\hat{y} \mid \mathbf{x})]$$

$$\mathrm{H}(\mathrm{p}_{\theta}) = -\mathbb{E}_{\mathrm{p}_{\theta}(\hat{y}|\mathbf{x})}[log \mathrm{p}_{\theta}(\hat{y} \mid \mathbf{x})]$$

Active label cleaning



Algorithm 1. Active label cleaning

Algorithm 1. Active label cleaning:

Given: $Y = \{\mathbf{l}_i\}_{i=1}^N$: True label distributions
Input: $\mathcal{D} = \{(\mathbf{x}_i, \hat{\mathbf{l}}_i)\}_{i=1}^N$: Dataset with noisy labels
$B \in \mathbb{N}$: Relabelling budget
$b \in \mathbb{N}$: Update frequency

$\theta \leftarrow$ TrainRobustModel($\mathcal{D}$)
$\mathcal{I}_{\text{avail}} \leftarrow \{1, \ldots, N\}, \quad \mathcal{I}_{\text{cleaned}} \leftarrow \emptyset$
count $\leftarrow 0$
while count $< B$ **do** ▷ If budget remains
  $j \leftarrow \arg\max_{i \in \mathcal{I}_{\text{avail}}} \Phi(\mathbf{x}_i, \hat{\mathbf{l}}_i; \theta)$ ▷ Rank (Eq. (2))
  repeat
    $\hat{\mathbf{l}}_j \leftarrow \hat{\mathbf{l}}_j +$ Sample($\mathbf{l}_j$) ▷ Acquire one-hot label
    count $\leftarrow$ count $+ 1$
  until majority formed in $\hat{\mathbf{l}}_j$
  $\mathcal{I}_{\text{avail}} \leftarrow \mathcal{I}_{\text{avail}} \setminus \{j\}, \quad \mathcal{I}_{\text{cleaned}} \leftarrow \mathcal{I}_{\text{cleaned}} \cup \{j\}$
  $\mathcal{D} \leftarrow \{(\mathbf{x}_i, \hat{\mathbf{l}}_i) : i \in \mathcal{I}_{\text{avail}} \cup \mathcal{I}_{\text{cleaned}}\}$
  if count divisible by b **then**
    $\theta \leftarrow$ Update($\theta, \mathcal{D}$) ▷ Fine-tune model
  end if
end while
return $\mathcal{D}$

Experiment

Selector 指主动标记清洗所用的模型
Classifier 指清洗后单独训练的模型
Vanilla 指普通的CNN

Table 1 Classification accuracy (%) before and after label cleaning.

	Selector	Scoring	Classifier	Before cleaning	After cleaning
(1)	Vanilla	Eq. (2)	Vanilla	64.1	68.4
(2)	Vanilla	BALD[34]	Vanilla	64.1	68.3
(3)	SSL	Eq. (2)	Vanilla	64.1	70.9
(4)	SSL	Eq. (2)	SSL	78.7	80.3
(5)	Vanilla	Eq. (2)	SSL	78.7	79.4
(6)	Co-teaching	Eq. (2)	Co-teaching	66.5	68.8
(7)	-	-	ELR[29]	67.0	-
(8)	(Clean training)		Vanilla	73.6	-
(9)	(Clean training)		SSL	80.7	-

Models are evaluated on a clean test set ($N = 50$ k) before and after relabelling 32.7% of samples in the training set (CIFAR10H, $N = 5$ k, $\eta = 30\%$).

Results

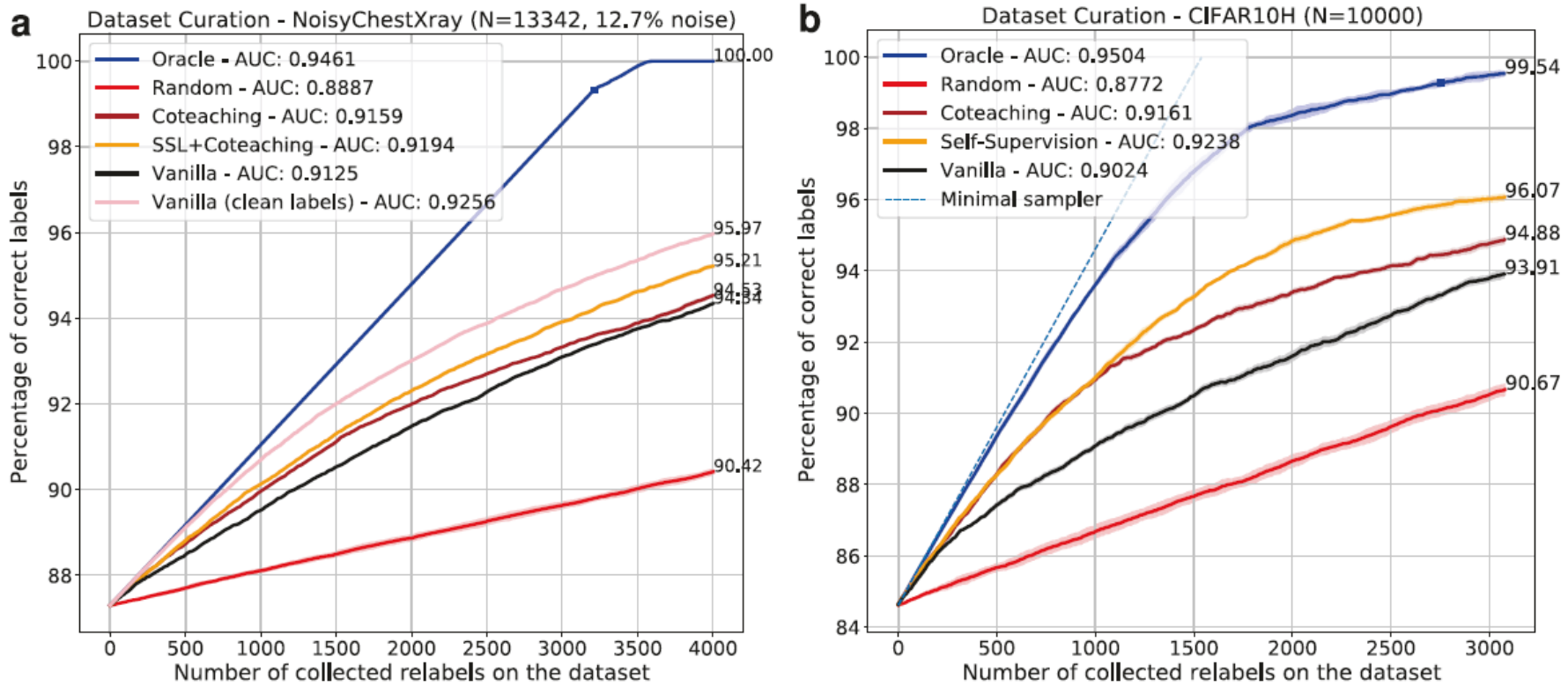


Fig. 3 Results of the label cleaning simulation on training datasets. a NoisyCXR ($\eta = 12.7\%$); **b** CIFAR10H ($\eta = 15\%$). For a given number of collected labels (x-axis), a cost-efficient algorithm should maximise the number of samples that are now correctly labelled (y-axis). The correctness of acquired labels is measured in terms of accuracy. The area-under-the-curve (AUC) is reported as a summary of cleaning efficiency of each selector across different relabelling budgets. The upper and lower bounds are set by oracle (blue) and random sampling (red) strategies. The pink curve (**a**) illustrates the practical "model upper bound" of cleaning performance when the selector model is trained solely on clean labels, its performance being bound to the capacity of the model to fit the data. Shaded areas represent ± standard deviation over 5 random seeds for relabelling.