# QActor: Active Learning on Noisy Labels

**Taraneh Younesian** T.YOUNESIAN@TUDELFT.NL
*Delft University of Technology, Delft, Netherlands*

**Zilong Zhao** Z.ZHAO-8@TUDELFT.NL
*Delft University of Technology, Delft, Netherlands*

**Amirmasoud Ghiassi** S.GHIASSI@TUDELFT.NL
*Delft University of Technology, Delft, Netherlands*

**Robert Birke** ROBERT.BIRKE@CH.ABB.COM
*ABB Corporate Research, Baden-Dättwil, Switzerland*

**Lydia Y.Chen** LYDIAYCHEN@IEEE.ORG
*Delft University of Technology, Delft, Netherlands*

**Editors:** Vineeth N Balasubramanian and Ivor Tsang

## Abstract

Noisy labeled data is more a norm than a rarity for self-generated content that is continuously published on the web and social media from non-experts. Active querying experts are conventionally adopted to provide labels for the informative samples which don't have labels, instead of possibly incorrect labels. The new challenge that arises here is how to discern the informative and noisy labels which benefit from expert cleaning. In this paper, we aim to leverage the stringent oracle budget to robustly maximize learning accuracy. We propose a noise-aware active learning framework, QActor, and a novel measure *CENT*, which considers both cross-entropy and entropy to select informative and noisy labels for an expert cleansing. QActor iteratively cleans samples via quality models and actively querying an expert on those noisy yet informative samples. To adapt to learning capacity per iteration, QActor dynamically adjusts the query limit according to the learning loss for each learning iteration. We extensively evaluate different image datasets with noise label ratios ranging between 30% and 60%. Our results show that QActor can nearly match the optimal accuracy achieved using only clean data at the cost of only an additional 10% of ground truth data from the oracle.

**Keywords:** Active Learning, Noisy Labels, Robustness, Filtering

## 1. Introduction

We are in the era of big data, which are continuously generated on different web platforms, e.g., social media, and disseminated via search engines often in a casual and unstructured way. Consequently, such a big data analysis suffers from diversified quality issues, e.g., images tagged with incorrect labels, so-called noisy labels. According to (Redman, 2016), noisy data costs the US industry more than $3 trillion per year to cleanse or to mitigate the impact of derived incorrect analyses. While the learning models conveniently leverage such a free source of data, its quality greatly undermines the learning efficiencies and their associate utilities (Jiang et al., 2018). For instance in (Zhang et al., 2017), using an image

classifier trained from data with highly noisy labels can significantly degrade classification accuracy and hinder its applicability on different domains.

Noisy label issue has been a long-standing challenge (Ghiassi et al., 2019), from standard machine learning (ML) models to deep neural networks (DNN), whose large learning capacities can have detrimental memorization effects on dirty labels (Zhang et al., 2017). The central theme here is to filter out the suspicious data which might have corrupted labels via quality estimates. The drawback of filtering approaches is the risk of dropping informative data points which can be influential for the underlying learning models. It might be worthwhile to actively cleanse such data due to its high potential in improving the learning tasks, even at a certain expense.

Active learning (AL) techniques (Settles, 2009) are designed to query extra information from an oracle for the data whose (true) labels are not readily available. Such an oracle is assumed to know the ground truth, but at high costs, e.g. a human expert. Hence, only the informative/uncertain data is queried within a certain query budget. The efficacy of active learning relies on uncertainty measurements of learning tasks, e.g., class probability (Schohn and Cohn, 2000), entropy value (Holub et al., 2008) or posterior predictive densities (Haußmann et al., 2019).

As the majority of active learning approaches focus on the unsupervised scenarios and constant budget, it is not clear how the active query approach can be adopted when encountering noisy data - a kind of noisy supervision.In the noisy labeled data scenario different from traditional active learning, the sample selection process ought to identify both informative and noisy samples. Using a limited query budget to cleanse clean samples leads to the waste of queries. Such a noisy supervision calls for a new measure that asks for the expert query on highly informative and noisy samples and retain the clean samples.

In this paper, we focus on a challenging multi-class learning problem whose labels are extremely noisy. Our objective is to enhance the noise-resiliency of the underlying classifier by selectively learning from good data as well as noisy labels that are critical for training the classifier. In order to turn the noisy labels into a learning advantage, we resort to the oracle for recovering their label ground truth under a given query budget. Ultimately, we aim to optimize classification accuracy with a minimum number of oracle queries.

To such an end, we design an active learning framework termed Quality-driven Active Learning (QActor), which marries quality models with active learning. Upon receiving new data instances, QActor first filters it via the *quality model* into "clean" and "noisy" categories. Second, we propose a novel measure noise-aware informative measure, *CENT*, combining cross-entropy and regular entropy, to identify erroneous and informative samples and send them for oracle cleansing. Another unique feature of QActor is that the overall query budget is fixed but the number of queries per batch is dynamically adjusted based on the current training loss value. Our results show that in the presence of very large label noise, i.e., up to 60% corrupted labels, QActor can achieve remarkable accuracy, i.e., almost match the optimal accuracy obtained excluding all noisy labels, at the cost of just a small fraction of oracle information, i.e., up to 10% oracle queried labels. Moreover, compared to state of the art on noise resilient DNN, QActor achieves higher accuracy by 15% and 8% for CIFAR-10 and CIFAR-100, respectively.

Our contributions are threefold. We design a novel and efficient learning framework, termed QActor, whose core combines a quality model with active learning. Secondly, we

propose a novel noise-aware informative measure, *CENT*. Thirdly we propose a dynamic learning strategy that can adapt to the dynamic nature of iterative active learning and achieve better results than the static one. To the best of our knowledge, this is the first study on the dynamic allocation of an active learning budget.

## 2. Related Work

Human error and careless annotators result in unreliable datasets with mistakes in labels available in public domains (Yan et al., 2014; Blum et al., 2003). Adversaries are another source of label noise attacking the performance of (deep) learning systems (Goodfellow et al., 2015). Corresponding to the contribution of QActor, we categorize the related work of learning from noisy supervision into two categories: (i) noise resilient models that filter the noise or alter the loss function without ground truth, (ii) active learning from the oracle supervision.

**Noise resilient model.** Learning with noise in the labels with no quality filtering shows the effect of noise in the degradation of the classification accuracy of deep neural networks (Ghiassi et al., 2021). As mentioned in (Zhang et al., 2017), the accuracy of using trained AlexNet to classify CIFAR-10 images with random label assignment drops from 77% to 10% due to network memorization of the noisy samples. Co-teaching (Han et al., 2018) trains two neural networks simultaneously on two different data and exchanges the model information trained by the data causing the lowest loss. RAD and its extensions (Zhao et al., 2019, 2021) cascadically train two models to find out "clean" data. They also use the help of external experts to verify the labels and optimize the cost with a limited budget. On the other hand, the study in (Patrini et al., 2017), Forward, assumes there is a noise transition matrix to cleanse the noisy labels for a deep neural network. Furthermore, D2L (Wang et al., 2018b) uses the Local Intrinsic Dimension (LID) as a measure to filter the noisy labeled instances during training. Re-weighting samples based on their similarity to a clean set to increase the robustness against label noise has been studied in (Ren et al., 2018). Noise confusion matrix estimation is another method to improve learning on corrupted data (Ghiassi et al., 2020).

**Active learning.** Active learning has been employed at a growing rate in recent studies with deep networks due to the expenses of large dataset collection. Various studies focus on the identification of informative data instances. The studies in (Sener and Savarese, 2018, 2017) consider geometrical approaches to select the data instances, i.e. the core-set, that is the representative of the data space. Meanwhile, (Gal et al., 2017) uses deep Bayesian neural networks with monte-carlo dropout to identify the most uncertain samples for labeling. Following the same framework, (Kirsch et al., 2019) argues the effectiveness of batch labeling via a expert in a deep neural network. Furthermore, (Stanitsas et al., 2017) uses the probability output of the convolutional neural network to label the instances based on discrete entropy and best-vs-second-best. A relevant line of research in active learning is to deal with noisy oracles which can not accurately provide the labels. For instance, (Bouguelia et al., 2018) benefits from the disagreement of a committee of models with the given label. In (Lin et al., 2016) the goal of the active learner is to identify the informative noisy instances and ask the oracle their true label. (Zhang and Chaudhuri, 2015) assumes that a strong labeler is sided by a weak labeler, termed quality model which is cheaper

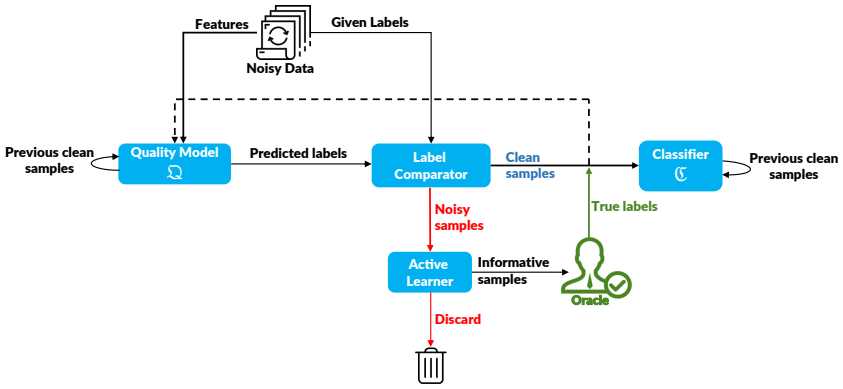



Figure 1: Overview of QActor: workflow of quality model, active learner, and classifier.

than an expert, and only queries the oracle when the two labelers disagree. However, these studies fail to consider the noisy data characteristic in their query selection strategies.

## 3. Quality-aware Active Learning QActor

### 3.1. Problem Statement

We consider multi-class classification problems that map data inputs $\mathbf{x}$ of $K$ features into labels $y$ of $C$ classes, $\mathbf{x} \in \mathbf{X}^{N \times K}$ into $y \in \mathbf{C} = \{1, \ldots, C\}$. We assume that given dataset $\mathbf{D} = \{(\mathbf{x}_j, \hat{y}_j), j = 1, ..., N\}$ has noisy labels, i.e. the label of a fraction of the data is altered from its true label. A small set of initial data instances with clean labels used as the initial seed is given, together with a testing set for evaluation. A clean data instance refers to a sample whose given label is properly annotated, without any alteration. In this paper, $\eta$ shows the noise rate which indicates the ratio of the noisy label data to the entire dataset size.

The goal is to identify the noisy labeled data samples and clean their labels by an expert labeler, i.e the oracle[1], within a limited budget. Since expert labeling is expensive, we aim to identify informative noisy labeled data and it to send the oracle for relabeling. In the end, the evaluation is done by training a classifier on the filtered and cleansed data. In the following section, we demonstrate the procedure of the introduced method QActor.

### 3.2. Architecture and Methodology of QActor

Figure 1 depicts the architecture. The main components are the following: 1)quality model $\mathfrak{Q} : \mathbf{x} \rightarrow \tilde{y} \in \mathbf{C}$ where $\tilde{y}$ is the predicted label by the quality model, 2) the label comparator which discerns noisy from clean labels, 3) the active leaner which determines which and how many data instances to send to the oracle, and 4) the classifier $\mathfrak{C} : \tilde{\mathbf{x}} \rightarrow \tilde{y} \in \mathbf{C}$. $\tilde{X} = \{\tilde{\mathbf{x}}\}$ is a subset of $X$ defined below.

1. In this paper, we interchangeably use terms of expert and oracle.

We use Deep Neural Networks (DNN) as the classifier since they have shown extremely promising results in classifying complex image datasets (Goodfellow et al., 2015). Due to the high training costs of deep neural networks, to reduce the computational burden, instead of having two different models for $\mathfrak{Q}$ and $\mathfrak{C}$, we leverage the time difference between data arrivals to use the previously trained $\mathfrak{C}$ as $\mathfrak{Q}$ for the following time period, i.e., $\mathfrak{Q}(t) = \mathfrak{C}(t-1)$. This optimization allows us to train only one model per time period.

At each iteration $t$ the data $x$ is first sent to the quality model from the previous time instance $\mathfrak{Q}(t-1)$ which predicts their labels $\tilde{y}$. If the predicted labels are the same as the given ones, i.e. $\tilde{y} = \hat{y}$, the label comparator marks them as clean, denoted as $\mathbf{x}^c(t) \in X^c(t)$; otherwise as suspicious, $\mathbf{x}^s(t) \in X^s(t)$. After this filtering step, the goal is to efficiently clean the suspicious data. Since single sample relabeling is very inefficient for the training process of deep neural networks, we relabel a batch of informative samples at each iteration as (Kirsch et al., 2019). To this end, the suspicious data samples are sent to the active learner to rank them based on their informativeness and select a batch of $o(t)$ data instances $X^o(t)$ to send to the oracle to query their true label. Note that $o(t)$ is limited by the given available budget $B$, i.e., $\sum_t o(t) \leq B$. Since $o(t)$ is typically much smaller than the number of noisy data instances, i.e. $o(t) \ll |X^s(t)|$, instances are ranked and selected based on an uncertainty metric. Highly uncertain samples indicate high informativeness and thus we argue that re-labeling these samples would increase the performance of the classifier.

The clean and relabeled instances are denoted by $\tilde{X}(t) = (\cup_{\tau=1}^{t} X^o(\tau)) \cup X^c(t)$ and used to re-train $\mathfrak{C}(t)$ and $\mathfrak{Q}(t)$. We repeat this procedure again for a few iterations until the budget is exceeded or the performance reaches to a desired level. To avoid pitfalls in learning we monitor the accuracy on a small hold-out of the initial set. If performance drops by more than $a$ we roll back the model before processing the next data batch.

In the following section, we explain our proposed noise-aware informativeness measure that identifies the useful samples to be relabeled by the oracle.

### 3.3. Noise-aware Informative Measure: *CENT*

Since relabeling all the data in the suspicious set is an expensive and time-consuming task, we aim to identify the most informative and useful samples to relabel by the oracle. We argue that in noisy labeled data settings different from traditional active learning where no label information is available, relabeling is more effective if the budget is spent on informative and noisy data. To overcome this issue, we introduce a novel noise-aware active learning measure, *CENT*, which consists of two parts, informative data identification, and clean/noisy separation. *CENT* queries the oracle to relabel an informative noisy set, and furthermore, identifies the informative clean set within the suspicious set and keeps their own label and then adds to clean set $X^c(t)$. The motivation behind this strategy is to leverage the clean data that is mistakenly categorized as noisy and is then discarded by the traditional AL methods for their lack of informativeness.

To measure the informativeness of the data, first, we calculate the entropy of the suspicious set:

$$\mathcal{L}_E(\mathbf{x}_i^s) = -\sum_{c=1}^{C} p(y=c|\mathbf{x}_i^s) \log p(y=c|\mathbf{x}_i^s) \tag{1}$$

where $p$ is the neural network's softmax prediction output for each class label $c$: $p(c|\mathbf{x}) = \frac{e^{z_c}}{\sum_{j=1}^{C} e^{z_j}}$ where $z_j$ are the logits.

Entropy is an information-theoretic measure and the higher this value, the higher in information in the data sample is. We pick $2 \cdot o(t)$ samples with the highest $\mathcal{L}_E$ value and put them in the informative set $X^I(t)$. Although entropy is a popular informativeness measure in active learning literature where the data is unlabeled, in noisy labeled data problems it doesn't necessarily identify the noisy informative data. Therefore, we employ another metric, cross-entropy, which can distinguish between noisy and clean data:

$$\mathcal{L}_{CE}(\mathbf{x}_i^I, y_i) = -\sum_{c=1}^{C} q(c|\mathbf{x}_i^I) \log p(y = c|\mathbf{x}_i^I) \tag{2}$$

where $q(c|\mathbf{x}_i)$ denotes the given label probability distribution over the $C$ class labels where $q(c|\mathbf{x}_i) = 1$ for $c$ equal to the given class $\hat{y}_i$ and $q(c|\mathbf{x}_i) = 0$ for all $c \neq \hat{y}_i$. We show in Theorem 1 that the higher values of CE are the indication of noisy labeled data. Therefore, we pick the $o(t)$ data with the highest CE values among $X^I(t)$ as $X^o(t)$ and send them to the oracle to relabel. Moreover, since smaller values of CE represent the clean data among the informative set, we pick the other half of $X^I(t)$ that have the lowest CE values and call them the *semi-clean* data $X^{semi-c}(t)$ to add to the clean set, keeping their own labels. Algorithm 1 shows the overview of our proposed method.

**Theorem 1** *With uniform label noise with the rate $\eta$ and the accuracy $A$ of the classifier in a $C$ class classification task, the average of $\mathcal{L}_{CE}(\mathbf{x}, \hat{y})$ for noisy samples is higher than $\mathcal{L}_{CE}(\mathbf{x}, \hat{y})$ for clean samples with the given label $\hat{y}$, if $\frac{(1-A)}{A+(1-A)\gamma}\frac{\eta}{(1-\eta)} < C-1$ where $\gamma << 1$ is a positive number.*

**Proof** Let $\hat{y}$ and $\tilde{y}$ be the given (noisy) and the predicted label for the true label $y$. We show the clean and noisy set as $\Omega = \{(\mathbf{x}_j, \hat{y}_j)|\hat{y}_j = y_j\}$ with $|\Omega|$ samples and $\Phi = \{(\mathbf{x}_i, \hat{y}_i)|\hat{y}_i \neq y_i\}$ with $|\Phi|$ samples respectively. We argue that the average $\mathcal{L}_{CE}$ value for the noisy samples is higher than the clean samples:

$$\mathbb{E}_{\mathbf{x}\sim\Omega(\mathbf{x},\hat{y})}\mathcal{L}_{CE}(\mathbf{x}, \hat{y}) < \mathbb{E}_{\mathbf{x}\sim\Phi(\mathbf{x},\hat{y})}\mathcal{L}_{CE}(\mathbf{x}, \hat{y}) \tag{3}$$

According to equation 2, since $q$ is a one-hot vector of the labels, $\mathcal{L}_{CE}(\mathbf{x}_i, \hat{y}_i) = -\log p(\hat{y}_i|\mathbf{x}_i)$. By removing logarithm from both sides, we have:

$$\mathbb{E}_{\mathbf{x}\sim\Phi(\mathbf{x},\hat{y})}\{p(\hat{y}|\mathbf{x})\} < \mathbb{E}_{\mathbf{x}\sim\Omega(\mathbf{x},\hat{y})}\{p(\hat{y}|\mathbf{x})\} \tag{4}$$

where $p$ is the neural network's softmax prediction output. Consider Figure 2 which categorizes the clean and noisy data based on the noise ratio and the classifier's accuracy. With the uniform noise patter, the probability of the noisy label being equal to the predicted label is $\frac{1}{C-1}$, and $\frac{C-2}{C-1}$ otherwise. Therefore, the inequality above is equivalent to the following:

$$\begin{aligned} &A\eta\mathbb{E}_{\mathbf{x}\sim\Phi(\mathbf{x},\hat{y})}\{p(\hat{y}|\mathbf{x}, \tilde{y} = y)\} + \frac{(1-A)\eta}{C-1}\mathbb{E}_{\mathbf{x}\sim\Phi(\mathbf{x},\hat{y})}\{p_{max}(\mathbf{x})\} + \\ &\frac{(1-A)(C-2)\eta}{C-1}\mathbb{E}_{\mathbf{x}\sim\Phi(\mathbf{x},\hat{y})}\{p(\hat{y}|\mathbf{x}, y \neq \tilde{y} \neq \hat{y})\} < \\ &A(1-\eta)\mathbb{E}_{\mathbf{x}\sim\Omega(\mathbf{x},\hat{y})}\{p_{max}(\mathbf{x})\} + (1-A)(1-\eta)\mathbb{E}_{\mathbf{x}\sim\Omega(\mathbf{x},\hat{y})}\{p(\hat{y}|\mathbf{x}, \tilde{y} \neq \hat{y} = y)\} \end{aligned} \tag{5}$$

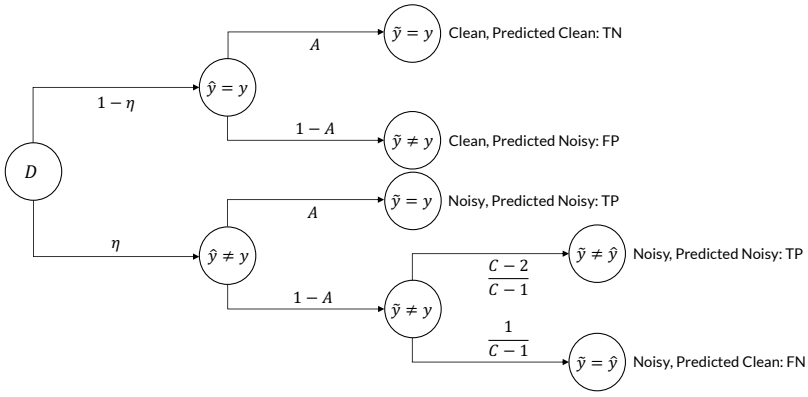



Figure 2: Categorization of the data $D$ based on the true, given and predicted labels. TP and FP indicate true and false positives and, TN and FN indicate true and false negatives respectively, where noisy and clean data are considered positive and negative respectively.

where $p_{max}$ for the data $\mathbf{x}_j$ is the maximum value of prediction vector for $\mathbf{x}_j$, associated with $\tilde{y}$. The inequality will hold by omitting the rest of the $p$ values in the left hand side which are significantly smaller than $p_{max}$. Moreover, on the right had side, we can use $\mathbb{E}_{\mathbf{x}\sim\Omega(\mathbf{x},\hat{y})}\{p(\hat{y}|\mathbf{x}, \tilde{y} \neq \hat{y} = y)\} = \gamma\mathbb{E}_{\mathbf{x}\sim\Omega(\mathbf{x},\hat{y})}\{p_{max}(\mathbf{x})\}$ where $\gamma << 1$. Therefore we will have:

$$\frac{(1-A)\eta}{C-1}\mathbb{E}_{\mathbf{x}\sim\Phi(\mathbf{x},\hat{y})}\{p_{max}(\mathbf{x})\} < (1-\eta)(A+(1-A)\gamma)\mathbb{E}_{\mathbf{x}\sim\Omega(\mathbf{x},\hat{y})}\{p_{max}(\mathbf{x})\} \tag{6}$$

Since $\mathbb{E}_{\mathbf{x}\sim\Phi(\mathbf{x},\hat{y})}\{p_{max}(\mathbf{x})\} \leq \mathbb{E}_{\mathbf{x}\sim\Omega(\mathbf{x},\hat{y})}\{p_{max}(\mathbf{x})\}$, therefore, the inequality holds when:

$$\frac{(1-A)}{A+(1-A)\gamma}\frac{\eta}{(1-\eta)} < C-1 \tag{7}$$

■

### 3.4. Active Learner Query Policies

The aforementioned uncertainty measures are used by the active learner in combination with two different policies on how to deplete the query budget over time:

**Static policy**. The active learner asks a constant number $o(t) = M, \forall t$ of queries at every iteration of learning. Essentially, for each iteration, the active learner queries the most uncertain $M$ data instances that are considered noisy by the quality model.

**Dynamic policy**. The active learner dynamically adjusts $o(t)$ based on the value of the loss function of the quality model. The rationale behind this is to increase the number of queries when the quality model has a low learning capacity, reflected by high loss function values, and to decrease the number of queries when the loss function converges to lower values. Specifically, we propose to adjust $o(t)$ as following:

$$o(t) = o(t-1)(1 - \frac{L^{\Omega}(t-2) - L^{\Omega}(t-1)}{L^{\Omega}(t-1)}) \tag{8}$$

**Algorithm 1:** Quality Driven Active Learning.

**Input :** Initial Dataset $D^I$, Noisy Labeled Dataset $D = \{(\mathbf{x}_j, \hat{y}_j)\}$, Budget $B$, Total Number of Iterations $T$

**Output:** Quality model $\mathfrak{Q}$, Classifier $\mathfrak{C}$

Train $\mathfrak{Q}$ and $\mathfrak{C}$ with $D^I$

**while** *iteration* $t < T$ **do**

- $\tilde{y}_j :=$ Predicted label by Quality model $\mathfrak{Q}$ for $\mathbf{x}_j$
- $X^c(t) = \{\forall \mathbf{x}_j \in D, \tilde{y}_j = \hat{y}_j\}$
- $X^s(t) = \{\forall \mathbf{x}_j \in D, \tilde{y}_j \neq \hat{y}_j\}$
- $o(t) =$ Query size according to Section 3.4
- **if** *Informativeness Measure is CENT* **then**
  - $X^I(t) =$ The first $2o(t)$ high entropy samples selected from $X^s(t)$
  - Sort $X^I(t)$ based on their CE value
  - $X^o(t) =$ The $o(t)$ samples from $X^I(t)$ with the highest CE
  - $X^{semi-c}(t) =$ The $o(t)$ samples from $X^I(t)$ with the lowest CE
- **else**
  - $X^o(t) =$ The $o(t)$ samples with the highest informativeness from $X^s(t)$
- **end**
- Send $X^o(t)$ to the oracle to relabel
- Train $\mathfrak{Q}$ and $\mathfrak{C}$ with $\tilde{X}(t) = (\cup_{\tau=1}^{t} X^o(\tau)) \cup X^c(t) \cup X^{semi-c}(t)$

**end**

where

$$L^{\mathfrak{Q}}(t) = \frac{-1}{|\tilde{\mathbf{x}}(t)|} \sum_{x_i \in \tilde{\mathbf{x}}} \sum_{c=1}^{C} p(y = c|\mathbf{x}_i) \log p(y = c|\mathbf{x}_i) \tag{9}$$

is the average entropy loss across all training samples used at period $t$. Since the re-training of the model(s) happens after the oracle querying, we use the loss from the periods $t-1$ and $t-2$. Finally, we note that the number of active queries is capped by the given budget $B$, i.e. the number of active queries used is $\min(B - \sum_{\tau=1}^{t-1} o(\tau), o(t))$.

Standard active learning studies query one instance at a time and train the model by adding that instance to the training set. Then the learner queries the next instance based on the retrained model and repeats the procedures recursively until all the budget is spent or the desired performance is achieved. However, it is computationally too expensive to retrain the model after each oracle query and repeat for the next one. Therefore, we decide to query $o(t)$ instances per round. This applies to both policies: static and dynamic.

## 4. Experimental Setup

Here we describe the datasets, the model parameters and the baselines used for comparison.

**Datasets.** We consider image datasets using the pixel values as inputs. In particular we use the well-known CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009). These datasets try to classify colored 32×32-pixel images into ten and hundred classes, respectively. CIFAR-100 is more complex due to both the higher number of classes and lower number

of data per class. For both dataset we use 50000 samples for training and the rest 10000 samples for testing.

**Label Noise.** We inject label noise into the training set by corrupting the label of randomly sampled data instances. We term the sampling probability as noise rate. Corrupted samples are subject to symmetric noise, e.g., the true label is exchanged with a random different label with uniform probability. We consider noise rates of 30% and 60%. Test data is not subject to label noise.

**Baselines.** To better show the overall effectiveness of our proposed QActor method we compare it two sets of baselines. First we compare against different active query selection baselines:

- **No-Sel**: uses all samples that arrive in the batch to train the classifier without filtering.
- **Q-only**: in this case the quality model filters the suspicious samples but there is no active learner to relabel the informative noisy instances. Therefore the classifier will train only on the clean data instances identified by the quality model.
- **Opt-Sel**: which assumes a perfect quality model able to identify all the true clean and noisy samples and uses all the clean samples for training the classifier without active learning.
- **Entropy (ENT)**: ranks the data based on their entropy value, i.e. $\sum_{c=1}^{C} p(y = c|\mathbf{x}_i^s) \log p(y = c|\mathbf{x}_i^s)$. Entropy is an information theoretic measure and the higher this value, the higher in information in the data sample is.
- **Re-Active**: based on the idea discussed by (Lin et al., 2016), where a weighted average of entropy and cross-entropy values are use for informative sample selection, i.e. $(1 - \alpha)\mathcal{L}_E + \alpha\mathcal{L}_{CE}$.
- **Random**: among the suspicious set, randomly selects $o(t)$ samples in each iteration to be relabeled by the oracle.
- **Semi-Random**: among the suspicious set, randomly selects $2o(t)$ samples in each iteration, one half to be relabeled by the oracle and another half to be added directly to the clean set keeping their own label.

Second we put QActor in the context of other noise-resistant techniques drawn from the related work on learning with noise. For a fair comparison we have used extra clean data also when training the noise resilient baselines, so that they have the same number of initial clean data during their training processes.

- **D2L** (Wang et al., 2018a): estimates the dimensionality of sub-spaces during training to adapt the loss function.
- **Forward** (Patrini et al., 2017): corrects the loss function based on the noise transition matrix.
- **Bootstrap** (Reed et al., 2015): using convex combination of the given and predicted labels for training.
- **Co-teaching** (Han et al., 2018): exchanges mini-batches between two networks trained in parallel.
- **Re-weighting** (Ren et al., 2018): Trains the neural network with a weighted loss function per samples, where the weights are learned based on the similarity of the data to a clean set.

### 4.1. Model Parameters

As QActor classifier for CIFAR-10 and CIFAR-100 we use the Convolutional Neural Network (CNN) architectures defined in (Wang et al., 2018a) with ReLU and softmax activation functions as image classifier and cross-entropy as loss function. We train the models by using stochastic gradient descent with momentum 0.9, learning rate 0.01, and weight decay $10^{-4}$. QActor and all baselines are implemented using Keras v2.2.4 and Tensorflow v1.12, except Co-teaching and Reweight. They are based on PyTorch v1.1.0, and we reproduce the same CNN structure in PyTorch as we use for other baselines implemented with Keras.

With CIFAR-10 QActor is trained initially with clean 1000 instances and 40 epochs. We inject noise for the remaining 49000 instances with 30% (60%) rate. Under static policy, in each iteration we query $o(t) = \{100, 300\}$ samples actively from the oracle and retrain the model for 10 epochs for 50 iterations which adds up to 540 epochs overall. Similarly, the dynamic policy uses $B = 5000$ which is equal to the budget used in the static policy with $o(t) = 100$. At the end of each batch, we test the model with the test set of 10000 instances. Rollback uses $a = 20\%$. For CIFAR-100 we increase the initial set size to size to 5000 and 100 epochs per batch to cope with the higher complexity. We also lower the total training iterations for CIFAR-100 to 30, but the epochs per iteration and $o(t)$ stay the same, which would be 400 epochs overall. For fair comparison, baselines are also trained under the same initial set and the same CNN structure. Therefore, we change *Re-weighting* neural network from Wide-ResNet to our own network structure. All baselines use the same parameters as from their papers except for *D2L*. Here we reduce the dimensionality estimation interval to 40 and 10 for the initial and subsequent batches, respectively. This keeps roughly the original ratio against the overall training period. We repeat each experiment for 3 times and for each experiment we report the average accuracy of the last 5 iterations.

## 5. Results

Here we present the accuracy achieved by QActor on the CIFAR-10 and CIFAR-100 datasets. We first compare QActor against six noise-resistant models (without using active learning strategies) from four state-of-the-art related papers, followed by the analysis over the uncertainty metrics. Finally, we unravel our model sensitivity analysis and eventually show the effect of the dynamic policy.

### 5.1. Noise-resistant Models

We compared our proposed QActor with the measure *CENT*, with the noise-resistant described in Section 4. Table 1 summarizes the results for different noise-resistant prior arts and QActor. For a fair comparison, we use the same initial clean set and neural network architecture to train all models. As the results illustrate, QActor is outperforming all the prior arts significantly by relabeling only 10% of the data. Although these state-of-the-art models are successful in classification tasks of samples affected by label noise, they don't benefit from an expert labeler during their training procedure. The best performance is achieved by *D2L* and *Bootstrap hard* which however are still 15 percent points lower than our QActor with 100 active queries per iteration for CIFAR-10. Increasing the active queries to 300 per iteration increases the gap by another 6 percent point. The other models, i.e.

Table 1: Accuracy (%) of different noise resilient networks under 60% label noise.

| Methods | Baselines | | | | | | Our | | |
|---|---|---|---|---|---|---|---|---|---|
| | D2L | Forward | Co-teaching | Bootstrap soft | Bootstrap hard | Re-weighting | QActor(100) | QActor(300) | QActor $^{D}$(100) |
| CIFAR-10 | 69.33 | 62.89 | 35.45 | 64.46 | 69.20 | 46.36 | 76.94 | 81.26 | 77.83 |
| CIFAR-100 | 37.35 | 39.52 | 6.92 | 38.51 | 29.99 | 8.54 | 47.63 | 50.57 | 48.40 |

*Co-teaching*, *Forward*, *Bootstrap soft*, and *Re-weighting* only achieve at best about 46% accuracy. This underlines how our method copes very well with noisy labeled data. For CIFAR-100, *Forward* and *Bootstrap soft* are best performing models while being almost 9% and 12% below QActor with only 100 and 300 queries per iteration respectively. As the last row of the table demonstrates, dynamic allocation of the budget boosts the accuracy of QActor with 100 up to one more percent. We will analyze this dynamic policy with more details in section 5.4.

### 5.2. Information Metrics

Here we compare the effect on the accuracy obtained on the test set when changing the underlying uncertainty measure.

In particular, we consider the uncertainty measures *CENT*, highest Entropy (*ENT*), and random baselines *Random* and *Semi-Random* to select samples from the noisy set to be queried for their labels. Moreover, we also analyze the methodology discussed by (Lin et al., 2016) and we set $\alpha = 0.8$. Figure 3 summarises the results on CIFAR-10 for 60% noise and $o(t) = 100$ over 50 iterations. As the figure shows, *CENT* is outperforming all the baselines over the iteration particularly in the earlier stages. As the number of queried samples grows, the performance of all the methods converges close to each other except for *Semi-Random*. The reason is that in the final iterations a large number of cleaned samples weigh more than the informativeness of them which is observed in all AL studies. Comparing *ENT* and *CENT* overtime shows the that leveraging the ability of CE function to detect the clean samples in the suspicious set and training on them is beneficial especially with small query number in the early stages. Moreover, results show the weighted average of entropy and cross-entropy values in *Re-Active* is not a suitable measure for informativeness. Analyzing the performance of Random, however, shows that random selection of samples to be cleaned by the oracle is less effective than smartly querying the informative samples.

The decline in the performance of *Semi-Random* over time is due to the selection of the noisy samples as the *semi-clean* data and adding them to the training pool. The reason is that *Semi-Random* randomly chooses samples from the informative suspicious set and directly sends them to the training set without cleaning, while *CENT* chooses these samples based on their low CE value. This comparison shows the effectiveness of using CE values to distinguish between clean and noisy samples.

Table 2 shows the detailed statistics of clean samples in the active for *CENT* and the sample selection baselines for both CIFAR-10 and CIFAR-100. Observing the last column that indicates the ratio of the FP (samples that are clean but have been considered noisy in the suspicious set by the quality model) to the suspicious set size, shows the total ratio of the clean samples in the suspicious set. This number is close to the *Semi-Random* clean ratio which shows the *Semi-Random* fails smartly select clean samples for the *semi-clean* set. Moreover, to see the effect of the *semi-clean* set in *CENT*, we compare the percentage

Table 2: Comparison of *CENT* and the query selection baselines for CIFAR-10 and CIFAR-100 with 60% noise and $o(t) = 100$ over 50 and 30 iterations respectively. FP and TP are false and true positives in the suspicious set where positive is considered the noisy data.

| Dataset | Method | Accuracy(%) | Clean(%) in Active Set | FP/(FP+TP)(%) |
|---|---|---|---|---|
| CIFAR-10 | CENT | **76.94** | 9.58 | 11.62 |
| | ENT | 75.77 | 39.92 | 11.27 |
| | Re-Active | 73.62 | 1.16 | 12.67 |
| | Random | 75.37 | 12.56 | 13.01 |
| | Semi-Random | 71.52 | 12.04 | 11.55 |
| CIFAR-100 | CENT | **47.63** | 9.06 | 27.34 |
| | ENT | 45.80 | 38.07 | 26.98 |
| | Re-Active | 47.00 | 1.97 | 26.75 |
| | Random | 47.59 | 25.24 | 26.50 |
| | Semi-Random | 46.76 | 26.99 | 26.41 |

of the clean samples in that set with the same value with *Semi-Random* strategy, where the data in the *semi-clean* set is selected randomly. For CIFAR-10 this number is 63.24% while it is only 11.98% for *Semi-Random*. The numbers are 73.86% and 26.24% for CIFAR-100 respectively. This illustrates how the cross-entropy value helps *CENT* to select mostly clean samples as the *semi-clean* set.

Furthermore, comparing the number of clean samples in the active set for *ENT* and *CENT* validates our argument over the ability of CE value to recognize noisy samples. It should be noted that although *Re-Active* is successful in including mostly noisy samples in the active set, the selected samples are the least helpful for the performance. Our detailed analysis indicates that this measure results in selecting the data mainly from a few classes instead of a more homogeneous selection. The high value of clean samples in the active set for *ENT* indicates that over half of the budget is being wasted on the samples that, although informative, were already clean. As mentioned earlier, our motivation to introduce *CENT*, was to overcome this waste and have a measure that focuses on both informativeness and noisiness of the samples that are being selected to be relabeled. As the experimental results confirm, we believe having such a measure is essential in active learning applications on noisy labeled data.

### 5.3. QActor Model Sensitivity Analysis

Here we present the results using the static policy termed QActor(100) and QActor(300) with constant 100 and 300 queries per iteration, respectively. We compare our QActor using *CENT* with the selection baselines from Section 4 with the different numbers of active queries and noise rates. Table 3 summarises the results. As expected, *No-Sel* which directly learns from the data has the worst results in the presence of noise, achieving at most 68.11% accuracy. Note that the accuracy of *No-Sel* declines over time and we reported the highest accuracy that the model achieved over epochs. However, our static QActor with *CENT* with only 100 active querying per iteration from the oracle comes remarkably close to *Opt-Sel*, where the model is trained with only the clean samples.

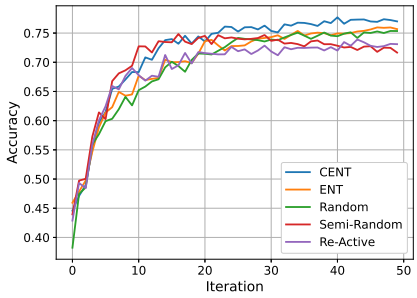
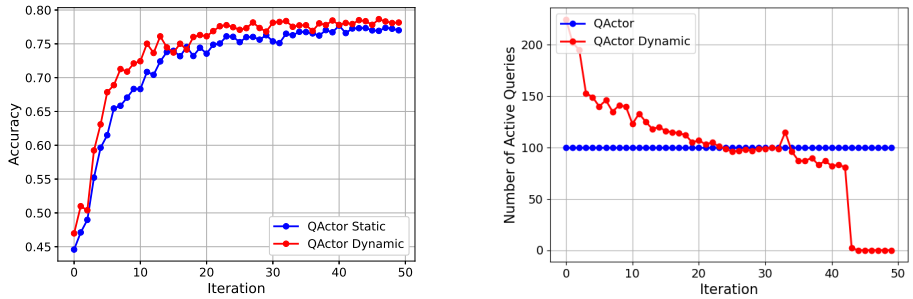



Figure 3: The impact of noise-aware informative measures: accuracy and loss for CIFAR-10 with 60% noise where at each iteration 100 queries are labeled by the oracle.




a) Accuracy b) Number of Queries

Figure 4: The impact of dynamic queries: comparison between QActor(100) and QActor$^D$ on CIFAR-10 with 60% noise.

Applying either sample selection or active querying alone achieves intermediate results compared to *No-Sel*. *Q-only* alone is more efficient driving up the accuracy to approximately 78% for CIFAR-10. However, *Q-only* has very poor performance for CIFAR-100. This comparison shows the effectiveness of both the quality model and active learning since neglecting any of both would result in a decrease in the accuracy. Moreover, we can observe with the increase in the number of the active query per iteration the performance increases over time.

### 5.4. Dynamic QActor

We compare our dynamic policy QActor$^D$ using a given budget of $B = 5000$ with our static policy QActor (100) that queries over the whole time horizon the same number of instances. Figure 4(a) and Table 3 summarize the accuracy results for CIFAR-10. Using dynamic query allocation policy (red line) of the budget across the batches leads to a better performance of 77.83% than the static policy (blue line) 76.94%. This increase in performance is more visible in higher noise rate as it is more crucial to clean more samples by the oracle. Looking at the evolution, we observe a higher performance compared to the static policy particularly in the first iterations.

Table 3: Accuracy on noisy CIFAR. Three alternative approaches v.s. static and dynamic versions of QActor, namely QActor(100) and QActor$^D$(100) which uses the same budget as QActor(100).

| Dataset | **Noise** | **Opt-sel** | **No-sel** | **Q-only** | **QActor(100)** | **QActor$^D$(100)** |
|---|---|---|---|---|---|---|
| CIFAR-10 | 30% | 88.52 | 68.11 | 78.74 | 86.24 | 86.31 |
| | 60% | 85.19 | 64.86 | 75.04 | 76.94 | 77.83 |
| CIFAR-100 | 30% | 59.29 | 50.50 | 36.52 | 54.28 | 54.74 |
| | 60% | 52.67 | 33.71 | 35.75 | 47.63 | 48.40 |

This result stems from the fact that the dynamic model queries more in the earlier batches when the model is less accurate and confident. Figure 4(b) shows the evolution of the number of active queries used in QActor$^D$ across the time periods. We see that indeed in the beginning the number of queries increases goes above the static assignment (100). In later batches, the number of queries goes then near and lower than the static case. The fluctuation of the query number per iteration shows how this number imitates the model's overall performance. Eventually all the budget is used over the whole time.

## 6. Conclusion

In this paper, we consider a challenging problem of image classification with corrupted labels. We propose QActor, a learning algorithm for very noisy label datasets, introducing an active learning methodology suited for noisy labeled data called *CENT*. The core of QActor is composed of a quality model that filters out the noisy labels and an active learner that smartly selects informative noisy instances to be relabeled by an oracle. The unique feature of QActor is its noise-awareness while selecting the informative data and its dynamic query allocation over training iterations based on the training loss. The flexible design enables QActor to be generalized on both standard and deep learning models that have limited clean data labels. Our extensive evaluation on CIFAR-10 and CIFAR-100 show that QActor can effectively combine the merits of the quality model and active learning when encountering extremely noisy labels, i.e., up to 60%. Compared to the state of the art addressing noisy labels, QActor can achieve higher accuracy by at least 8% at the cost of querying oracle to cleansing 10% suspicious images. For future work, we plan to explore different types of DNN models which can either provide more accurate predictions on probability, e.g., Bayesian neural networks. Moreover, to further improve the performance of the noise resilient prior arts we plan to combine QActor with them benefiting from expert relabeling in those models.

## Acknowledgments

This work has been partly funded by the Swiss National Science Foundation NRP75 project 407540_167266.

## References

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *JACM*, 50(4):506–519, 2003.

Mohamed-Rafik Bouguelia, Slawomir Nowaczyk, K. C. Santosh, and Antanas Verikas. Agreeing to disagree: active learning with noisy labels without crowdsourcing. *Int. J. Mach. Learn. Cybern.*, 9(8):1307–1319, 2018.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *ICML*, volume 70, pages 1183–1192, 2017.

Amirmasoud Ghiassi, Taraneh Younesian, Zhilong Zhao, Robert Birke, Valerio Schiavoni, and Lydia Y Chen. Robust (deep) learning framework against dirty labels and beyond. In *IEEE TPS-ISA*, pages 236–244, 2019.

Amirmasoud Ghiassi, Taraneh Younesian, Robert Birke, and Lydia Y Chen. Trustnet: Learning from trusted data against (a) symmetric label noise. *arXiv:2007.06324*, 2020.

Amirmasoud Ghiassi, Robert Birke, Rui Han, and Lydia Y Chen. Labelnet: Recovering noisy labels. In *IJCNN*, pages 1–8. IEEE, 2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8536–8546, 2018.

Manuel Haußmann, Fred A. Hamprecht, and Melih Kandemir. Deep active learning with adaptive acquisition. In *IJCAI*, pages 2470–2476, 2019.

Alex Holub, Pietro Perona, and Michael C. Burl. Entropy-based active learning for object recognition. In *CVPR Workshops*, pages 1–8, 2008.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2309–2318, 2018.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *NeurIPS*, pages 7024–7035, 2019.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10/100 (canadian institute for advanced research). 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

Christopher H Lin, M Mausam, and Daniel S Weld. Re-active learning: Active learning with relabeling. In *AAAI*, 2016.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952, 2017.

Thomas C. Redman. Bad data costs the U.S. $3 trillion per year. https://hbr.org/2016/09/bad-data-costs-the-u-s-3-trillion-per-year, 2016. Accessed: 2020-02-20.

Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML 2018*, volume 80, pages 4331–4340, 2018.

Greg Schohn and David Cohn. Less is more: Active learning with support vector machines. In *ICML*, pages 839–846, 2000.

Ozan Sener and Silvio Savarese. A geometric approach to active learning for convolutional neural networks. *CoRR*, abs/1708.00489, 2017.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Panagiotis Stanitsas, Anoop Cherian, Alexander Truskinovsky, Vassilios Morellas, and Nikolaos Papanikolopoulos. Active convolutional neural networks for cancerous tissue recognition. In *ICIP*, pages 1367–1371, 2017.

Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *CVPR*, pages 8688–8696, 2018a.

Yisen Wang, Xingjun Ma, Michael E Houle, Shu-Tao Xia, and James Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, 2018b.

Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. Learning from multiple annotators with varying expertise. *Machine learning*, 95(3):291–327, 2014.

Chicheng Zhang and Kamalika Chaudhuri. Active learning from weak and strong labelers. In *NeurIPS*, pages 703–711, 2015.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Zilong Zhao, Sophie Cerf, Robert Birke, Bogdan Robu, Sara Bouchenak, Sonia Ben Mokhtar, and Lydia Y. Chen. Robust anomaly detection on unreliable data. In *IEEE/IFIP DSN*, pages 630–637, 2019.

Zilong Zhao, Robert Birke, Rui Han, Bogdan Robu, Sara Bouchenak, Sonia Ben Mokhtar, and Lydia Y. Chen. Enhancing robustness of on-line learning models on highly noisy data. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2177–2192, 2021.