Active Label Correction Using Robust Parameter Update and Entropy Propagation

Kwang In Kim(✉)

UNIST, Ulsan, Korea
kimki@unist.ac.kr

Abstract. Label noise is prevalent in real-world visual learning applications and correcting all label mistakes can be prohibitively costly. Training neural network classifiers on such noisy datasets may lead to significant performance degeneration. Active label correction (ALC) attempts to minimize the re-labeling costs by identifying examples for which providing correct labels will yield maximal performance improvements. Existing ALC approaches typically select the examples that the classifier is least confident about (e.g. with the largest entropies). However, such confidence estimates can be unreliable as the classifier itself is initially trained on noisy data. Also, naïvely selecting a batch of low confidence examples can result in redundant labeling of spatially adjacent examples. We present a new ALC algorithm that addresses these challenges. Our algorithm robustly estimates label confidence values by regulating the contributions of individual examples in the parameter update of the network. Further, our algorithm avoids redundant labeling by promoting diversity in batch selection through propagating the confidence of each newly labeled example to the entire dataset. Experiments involving four benchmark datasets and two types of label noise demonstrate that our algorithm offers a significant improvement in re-labeling efficiency over state-of-the-art ALC approaches.

Keywords: Active label correction · Uncertainty sampling · Diffusion

1 Introduction

Deep neural networks can provide state-of-the-art performance on a variety of inference problems. This success often relies on the availability of large amounts of annotated data, but building large-scale annotations is a costly and erroneous process. For example, reliable annotations for medical imaging or astronomical imaging require expensive domain experts, and hence less costly (but less reliable) crowdsourcing might be employed [22,36,44]. Noisy labels can also be

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19803-8_1.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13681, pp. 1–16, 2022.
https://doi.org/10.1007/978-3-031-19803-8_1

found in automatically annotated data [26], data collected by noisy sensors [18], and label corruptions caused by adversarial attacks [1,42].

As such, *noisy annotations* are invariably used in training. Naïvely training deep neural networks on noisy data can severely limit their generalization capability as they tend to memorize data [41]. One approach to reduce labeling costs (or label correction or proofreading costs) is *active label correction* (ALC). ALC approaches attempt to identify examples for which correct labeling will provide the most significant performance improvement.

A common approach to ALC is to incrementally determine important examples. Initially, a deep network h is trained on the noisy training set D where only a small portion is provided with *clean* labels. By analyzing the predictions $h|_D$ of h on D, an ALC algorithm suggests examples to query. Once an oracle provided correct labels to these examples, h is retrained yielding improved predictions $h|_D$. This process is repeated until the labeling budget is exhausted.

Inspired by the success of active learning (AL), most existing ALC algorithms select examples with the largest entropy (or loss) values of the class-conditional probability distributions of $h|_D$ [2,19,25]. Using entropy is an intuitive and reliable approach to AL as the highest entropy examples are the most ambiguous to classify by h, and hence labeling them could significantly reduce the uncertainties that h has on D. However, its application to ALC is limited in that the estimated entropy values can be unreliable as h itself is trained on the initially noisy dataset: It is possible that a small portion of *clean* examples exhibit relatively high entropy values, even though re-labeling them would not help improve network performance. Also, selecting a large *batch* of high entropy points is redundant: If an example x_* has a high entropy value then its spatial neighbors tend to show similar high entropies (Fig. 1), but labeling them along with x_* would be unnecessary as re-training h on x_* can resolve the uncertainties of its neighbors. On the other hand, employing small batches of high entropy points would require frequent retraining of h, leading to increased computational costs.

Our algorithm addresses these difficulties by robustly estimating the entropy values of $h|_D$. During the update of h-parameters, the loss gradients of individual examples are weighted by the respective contribution parameters. These parameters are continuously adjusted based on the progressions of loss estimates, gradually suppressing the influence of outlier examples. Further, we explicitly diversify batch selection by iteratively regulating the entropy estimates of h: Once a point x_* is labeled, its updated entropy value (of zero) is propagated along D such that the entropies of neighboring points are suppressed. Instantiating these contributions into a learning framework lets our algorithm robustly train deep networks with efficient acquisition of labels.

We conducted experiments on four benchmark datasets with two different types of label noise. The results demonstrate that our robust parameter update strategy and entropy propagation approach contribute individually and collectively to improving ACL performance, outperforming existing approaches by a large margin.

2 Related Work

Active Label Correction. Typically, active label correction algorithms select examples to query by assessing the *confidences* of predictions made by the classifier on given training sets. For example Nallapati et al.'s *CorrActive learning* prioritizes misclassified examples (those exhibiting high training losses) [21] while Rebbapragada et al.'s *Active Label Correction* algorithm queries examples with the highest entropy values [25]. For support vector machine classifiers, predictive confidences can be evaluated based on their margins, leading to margin-based sample strategies [31]. Similarly, Henter et al.proposed to label examples showing the smallest difference between the class probabilities of the best and second-best hypotheses [13] and Bernhardt et al.'s *Active Label Cleaning* algorithm selects examples with the highest predictive losses and predictive entropy [2]. These algorithms demonstrated significant performance gains over random selection, but they do not explicitly model the underlying noise generation processes.

Kremer et al.'s robust ACL algorithm employs explicit noise modeling to measure the expected change of the classifier when examples are newly labeled: Such changes are measured based on the difference between the total losses obtained with and without labeling candidate examples. For logistic regression, with the aid of its noise model, this quantity can be evaluated without having to actually label the candidates. However, its extension to deep neural networks is not straightforward. Similarly to [2,25], Li et al.'s *Dual Active Label Correction* algorithm queries high-entropy examples [19]. Further, this algorithm achieved noise robustness by incorporating a noise model into classifier training: It estimates the probabilities of class transitions caused by noise and uses them to rectify the classifier outputs in the loss evaluation achieving significant improvements over existing ALC algorithms. In the experiments, we show that our method outperforms these existing methods [2,19,21,25].

Inspired by the intuition that examples that lie in label-homogeneous regions are likely to be clean, Urner et al.presented a theoretical sample complexity analysis [34]. Instantiating this theoretical analysis into practical algorithms remains an open problem.

Related Problems. Active learning (AL) with noisy annotators is a closely related problem. Zhang and Chaudhuri presented a theoretical AL framework where weak and strong oracles respectively provide clean and noisy labels [40]. This problem differs from ALC in that the identities of clean labels (provided by strong oracles) are known a priori [40]. Similarly, Yan et al.presented an AL scenario where oracles occasionally provide noisy labels [38]. Younesian et al.considered a learning problem where multiple annotators have varying levels of experience, presenting labels of diverse quality [39]. Their algorithm actively selects not only the examples to query but also the oracles who will annotate the queried examples. Under the presence of multiple noisy annotators (e.g.from crowdsourcing) Parde and Nielsen proposed to assign varying weights to the labels of each example via estimating the reliability of individual annotators [22].

Sheng et al.'s algorithm queries an example for additional labels when existing labels are discrepant [30].

Explicit detection of noisy examples is another closely related domain. Shen and Sanghavi used the loss values to determine noisy examples. Their algorithm alternates between filtering out examples with the highest losses and retraining the classifier on the remaining training set [29]. Zhu et al. proposed to construct *soft labels* based on local feature aggregations and used them to define a score function [43]. This enables to detect noisy examples without having to train a task-specific model. Huang et al.proposed to avoid memorization of noisy data by cyclically transferring the status of the learner from overfitting to underfitting via controlling its learning rate [14]. This strategy achieved improved performance over traditional noise-robust learning approaches. However, it relies on known numbers of noisy examples. We show that our algorithm offers improved labeling efficiency even without requiring the number of noisy examples (Sect. 3.1). Park et al.'s algorithm assesses how labeling individual examples influences the change of the classifier parameters and their evaluations on validation data [23]. Its application to ALC is not straightforward as clean validation sets are seldom available in ALC problems.

3 Robust Active Label Acquisition

Problem Setting and Algorithm Overview. We consider classification problems where one learns a neural network as a function h from the input space $\mathcal{X}$ to the output class-encoding space $\mathcal{Y}$. We employ one-hot class encoding such that $\mathcal{Y}$ forms a probability simplex of dimension M where M is the number of classes, and h generates class-conditional probabilities as outputs. When h does not generate probabilistic outputs, one could apply softmax activations to construct pseudo probabilities.

For a given *clean* training set $\widehat{D} = \{(x_j, y_j)\}_{j=1}^{N}$ sampled from an underlying distribution p of $\mathcal{X} \times \mathcal{Y}$, h can be constructed by minimizing the sum of losses:

$$L(h) = \sum_{j=1}^{N} l(h(x_j), y_j) \tag{1}$$

for a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$. We use the cross-entropy loss $l(z, y) = -\sum_{k=1}^{M} z^k \log(y^k)$ where z^k is the k-th component of z, while our method is applicable to other losses as well. In active label correction (ALC) problems, a subset $D^N \subset \widehat{D}$ of data is contaminated with label noise forming a (partially) noisy dataset D, e.g.the labels in $D^N \subset D$ are randomly flipped or altered with underlying class transition probabilities [16,20,35], and the number and identities of such noisy examples are not known. ALC algorithms are then provided with a labeling budget G such that they can select and query the true labels of G examples to an oracle. Typically, these algorithms identify iteratively such examples: Initially, a network h^0 is trained on $D^0 = D$. At iteration t, the outputs

$h^{t-1}|_{D^{t-1}}$ of h^{t-1} trained on D^{t-1} are analyzed to determine a batch $B^t \subset D$ of examples to query. Once $B^t \subset D$ is labeled, D^t is accordingly updated.

In active learning (AL), B^t is often selected as the most *ambiguous* examples, i.ethose with the highest entropy values of the corresponding class-conditional probabilities $h(x)$. However, naïvely applying this strategy to ALC can be sub-optimal as it can select already clean examples (in $D^C = D \setminus D^N$): As D^N is not known a priori, h^t trains on the entire dataset D^t, possibly memorizing noisy data $D^N \subset D^t$, and hence it can generate relatively high entropy values even on the examples in D^C.

Similarly to existing AL and ALC approaches, we employ entropy as the main batch selection criterion but we enhance its labeling efficiency by 1) robustifying the training of h^t via suppressing the contributions of label noise in its parameter update (Sect. 3.1). This helps improve the estimation of entropy values as well as classification accuracy. Also, 2) we iteratively regulate the estimated entropy values during batch selection (Sect. 3.2). Each time a single example is labeled, its updated entropy value is instantly propagated to D^t suppressing the entropies of the other points. This helps avoid selecting accumulations of adjacent examples and diversify label selection without having to retrain h per stage.

3.1 Robust Update of Classifier Parameters

In the standard stochastic gradient descent-based learning, the parameter vector W of h is iteratively updated using a mini-batch subset D_i of D, minimizing L:

$$W(i+1) = W(i) - \eta(i) \sum_{(x_k, y_k) \in D_i} \overline{\alpha}_k(i) \nabla_W l(h(x_k), y_k), \tag{2}$$

where i is the pass index of an iteration, $\eta(i)$ is the learning rate, and the contribution parameters $\{\overline{\alpha}_k\}$ are kept at a constant value of $\frac{1}{N}$. In this case, clean and noisy examples contribute equally to the update of W potentially distracting the training process.

Our algorithm dynamically adjusts $\{\overline{\alpha}_k(i)\}$ according to the learning progress of h. At each epoch, the global weights $\{\alpha_j\}_{j=1}^N$ are determined as convex combination coefficients of the entire dataset D ($\alpha_j \geq 0$, $\sum_{j=1}^N \alpha_j = 1$) and $\{\overline{\alpha}_k(i)\}$ is selected from $\{\alpha_j\}_{j=1}^N$ according to its mini-batch index[1]. In the first epoch, $\boldsymbol{\alpha}(1) = [\alpha_1(1), \ldots, \alpha_N(1)]^\top$ is uniformly initialized and W is updated according to Eq. 2. Thereafter, at epoch q, $\boldsymbol{\alpha}(q)$ is updated based on the following rule:

$$\boldsymbol{\alpha}(q+1) = (1 - \delta^\alpha)\boldsymbol{\alpha}(q) + \delta^\alpha \frac{\mathbf{g}(q)}{\|\mathbf{g}(q)\|_1}, \tag{3}$$

where $\mathbf{g}(q) = [g(l(h^q(x_1), y_1)), \ldots, g(l(h^q(x_N), y_N))]^\top$ and

$$g(z) = \exp\left(-\frac{z^2}{\sigma^\alpha}\right) \tag{4}$$

[1] We denote a single update step of W for a given mini-batch (Eq. 2) by 'pass' while an 'epoch' involves multiple mini-batch passes including all the training examples.

for the step hyperparameter $0 \leq \delta^\alpha \leq 1$ and scale hyperparameter $\sigma^\alpha \geq 0$. Once $\boldsymbol{\alpha}(q+1)$ is obtained, it is normalized such that $\|\boldsymbol{\alpha}(q+1)\|_1 = 1$. As $g(l(h(x_j), y_j))$ is inversely proportional to the loss l incurred at (x_j, y_j), the iterative update process of Eq. 2 tends to ignore examples that *consistently* (during the iteration) exhibit large errors. The parameter δ^α controls the speed of α evolution: At large (close to one) δ^α, $\{\alpha_j\}$ evolves rapidly, emphasizing the latest observed loss values while small δ^α places more emphasis on the previous loss trajectory of each example. σ^α determines how aggressively losses are penalized: For small δ^α values, even small losses are heavily penalized and only a small number of examples contribute to the parameter update, while as $\delta^\alpha \to \infty$, $g \to 1$ independently of individual loss values, producing an even $\{\alpha_j\}$ distribution.

Discussion. Using the training loss as a noise indicator is a common practice in ACL. For example, Bernhardt et al.'s active cleaning strategy queries points with large loss values [2]. Empirically evaluating this approach in preliminary experiments, we observed that determining the optimal timing to measure losses is challenging: At early training epochs, the classifier h might not have gained sufficient information on the problem to faithfully estimate the target outputs of clean data. On the other hand, at later epochs, h can overfit to D, generating low loss values even on D^N. Our approach bypasses this step by accumulating the contribution parameters over time and *gradually* suppressing the outliers. When our algorithm suggested candidates for labeling, on average (across a varying number of labels) 97.9% of these examples were noisy while using [2] achieved only around 91.1% accuracy: As (re-)labeling already clean examples is redundant, ALC algorithms need to select noisy examples for querying (Table 1).

Our framework (Eq. 3) can be considered as an instance of example reweighting for robust learning: As the gradient is a linear operator, weighting the loss gradient per example is equivalent to weighting individual examples. However, adapting these approaches to ALC is non-trivial. For example, Ren et al.'s meta-learning approach requires clean validation labels [27]. While this might be reasonable for general robust learning, in ALC, such labels are seldom available.

An alternative to our strategy is to explicitly pre-select clean examples $D^C \subset D$: One could first apply data cleaning algorithms e.g. [14,23,43] to identify clean examples and subsequently apply active learning. Once D^C can be successfully estimated, in principle, this choice would lead to improved performance. However, Fig. 4 demonstrates that precisely identifying D^C is challenging even when the number of noisy examples is assumed known as in [14].

3.2 Entropy Propagation for Iterative Label Selection

Querying the most *uncertain* examples (e.g.with the highest entropy values) for labeling has been commonly exercised in AL and ALC. An ideal setting in this case would be *fine-grained* incremental learning: At stage t, the single most uncertain example $x_* \in D^{t-1}$ is queried for labeling and D^t is accordingly updated. Then, a new classifier h^t is trained on D^t yielding the uncertainty estimates for the next stage. However, in deep learning, this strategy is not

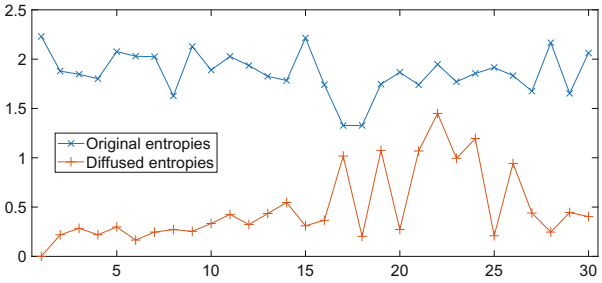

Fig. 1. An example of entropy diffusion on *CIFAR-10* dataset. A point x_* is newly labeled and the corresponding entropy is updated to zero. The entropies of the remaining examples in D are accordingly adjusted. The x-axis shows the indices of data points ordered inversely according to the distance to x_*. The first entry is x_*. When x_* has originally a high entropy value, its spatial neighbors also exhibit high entropy values (the average entropy on D was less than 0.7). Applying diffusion on D suppressed the entropies of points near x_*. Note that the degrees of suppression are proportional to the similarity to x_*.

directly applicable as it requires frequently retraining the classifier, incurring prohibitively high computational costs. Instead, a batch B^t of examples are selected at once as examples with the highest entropies in D^{t-1}.

This naïve batch selection strategy often generates redundant labelings: When h^t assigns a high entropy value $e(x_*)$ to an example x_*, it is likely that other examples in its neighborhood $\mathcal{N}(x_*)$ also have high entropies (Fig. 1) and therefore, included in B^t along with x_* (Fig. 1). However, when h^t is trained with the ground-truth label of x_*, it may acquire sufficient information to resolve uncertainty in $\mathcal{N}(x_*)$. In this case, it would be more efficient to select examples outside $\mathcal{N}(x_*)$. Our approach is to *diversify* batch selection by simulating a diffusion process on the manifold of data points $\mathcal{X}$.

Entropy Diffusion on Data Manifolds. On a Riemannian manifold $\mathcal{X}$ equipped with a data-generating distribution $p(x)$, a diffusion of a smooth function $g \in C^\infty(\mathcal{X})$ is described as a time evolution equation:

$$\frac{\partial g}{\partial t} = \Delta_p g, \tag{5}$$

where Δ_p is the (p-normalized) Laplacian on $\mathcal{X}$. This process gradually *propagates* the *mass* $g(x)$ at location x to the entire manifold weighted by p [28] and it has been used in denoising smooth functions and data points [12], semi-supervised learning [3], and simulating information spread on social networks [24]. We consider the entropy values e as a smooth function to be diffused along $\mathcal{X}$. Suppose that at stage t, h^t is trained and the corresponding entropy estimation on $\mathcal{X}$ is made. Then, the example x_* with the maximum entropy is queried for labeling, and the corresponding entropy value $e(x_*)$ is set to zero. This *new information* is spread over $\mathcal{X}$ suppressing the entropies of the related points.

Entropy Propagation Algorithm. In practice, the manifold $\mathcal{X}$ is not directly observed and instead a point cloud $X = \{x_j\}_{j=1}^N$ sampled from p is presented as an embedding of $\mathcal{X}$ onto a Euclidean space (i.e. $X \subset \mathbb{R}^d$ with d being the data dimensionality). In this case, the analytic diffusion process can be spatially discretized as

$$\frac{\partial \mathbf{e}}{\partial t} = -L\mathbf{e}, \tag{6}$$

where $\mathbf{e} = [e(x_1), \ldots, e(x_N)]^\top$, L is the probability-normalized graph Laplacian constructed based on X:

$$\begin{aligned} L &= I - D^{-1}A, \\ A^{jk} &= \begin{cases} \exp\left(-\frac{\|x_j - x_k\|^2}{\sigma^L}\right), & \text{if } x_j = \mathcal{N}(x_k) \text{ or } x_k = \mathcal{N}(x_j) \\ 0, & \text{otherwise,} \end{cases} \\ D^{jk} &= \begin{cases} \sum_m^N A^{jm}, & \text{if } j = k \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{7}$$

$\mathcal{N}(x)$ is the nearest neighbors of x, and $\sigma^L > 0$ is a scaling hyperparameter. When the sample size N grows to infinity, X becomes a precise representation of $\mathcal{M}$ and in this case, $-L$ converges to the true Laplacian Δ_p [11].

Our entropy propagation algorithm is obtained by time-discretizing the continuous evolution of Eq. 6 using the explicit Euler scheme [4]:

$$\mathbf{e}(i+1) = \mathbf{e}(i) - \delta^L L\mathbf{e}(i) \tag{8}$$

with a time-discretization step size $\delta^L > 0$. This helps simulate fine-grained incremental learning without the need to actually retrain h for each newly added label. Figure 1 shows that high entropy values are indeed spatially correlated, and our diffusion process can effectively suppress the entropies of nearby points.

Discussion. The diffusion process in Eq. 6 jointly updates the entropy values $\mathbf{e}$ of the entire training set D. This offers the capability of not only suppressing the entropies of neighbors $\mathcal{N}(x_*)$ of a newly labeled example x_* but also regularizing potentially noisy entropies (as h trains on noisy data D). The latter can be seen by noting that for a manifold $\mathcal{X}$ with a compactly supported data distribution p, the time-discretization of the diffusion process in Eq. 5 corresponds to a single gradient-descent step for minimizing the following energy:

$$E(g; g^{t-1}) = \|g - g^{t-1}\|^2 + \delta\langle \nabla g, \nabla g \rangle \tag{9}$$

where ∇g is the gradient of g. This is a direct consequence of Stokes' theorem [10]: Our diffusion promotes *first-order smoothness* of the solution $\mathbf{e}$ along $\mathcal{X}$.

3.3 Active Label Correction Algorithm

Our final algorithm is obtained by incorporating the robust classifier training steps (Sect. 3.1) into the incremental label selection process (Sect. 3.2): At iteration t, the classifier parameter W and contribution parameters $\{\alpha_j\}_{j=1}^N$ are

Algorithm 1. Robust active label correction algorithm

Input: Noisy data $D = \{(x_i, y_i)\}_{i=1}^N$, and labeling batch size Q and budget G.
Initialization: $D^0 = \emptyset$, $B^0 = \emptyset$, $S^0 = \emptyset$, and $t = 1$.
repeat
  $\alpha_j = \frac{1}{N}$ for $j = 1 \ldots, N$.
  if $\mod(t, Q) = 0$ **then**
    $S^t = S^{t-1} \cup B^{t-1}$.
    repeat
      Update the classifier h parameter W according to Eq 2.
      Update the contribution coefficients $\{\alpha_j\}_{j=1}^N$ using Eq 3.
    until maximum epoch reached.
    Train h with $\{\alpha_j\}_{j=1}^N$.
    Evaluate the entropy values $\mathbf{e}$ using h on D.
    $B^t = \emptyset$.
  end if
  Sample the candidate set C from $D \setminus S^t$ using p^S.
  Select an example $x_* \in C$ with the largest entropy e.
  repeat
    Assign zero to $e(x_*)$.
    Update $\mathbf{e}$ using Eq. 8.
  until maximum diffusion steps reached.
  $B^t = B^t \cup \{x_*\}$.
  $t = t + 1$.
until labeling budget reached.
Output: Trained classifier h^t and (partially) cleaned label set S^t.

estimated using Eqs. 2 and 3, and the entropy estimates $\{e(x_j)\}_{j=1}^N$ are obtained by evaluating h^t on D^t. Then, a batch B^t of data are constructed by iterating through 1) sampling a set C of candidate examples from the probability distribution p^S on $\{1, \ldots, N\}$ formed by $p_j^S = \frac{1-\alpha_j}{\sum_{k=1}^N 1-\alpha_k}$: For experiments, we sampled candidates from p^S by ranking p^S); 2) adding the example x_* with the highest entropy value in C to B^t; and 3) iteratively updating the entropy estimates $\{e(x_j)\}_{j=1}^N$ using Eq 8. Algorithm 1 summarizes the training process.

Hyperparameters and Complexity. Our algorithm requires determining several hyperparameters. This is a difficult problem in the ALC setting: Often, the hyperparameters of learning algorithms are tuned based on separate validation sets. However, in ALC, labels are inherently limited and such validation sets might not be available. For our experiments (Sect. 4), we determined these parameters based on heuristics commonly employed in related problem domains and fixed them across the entire datasets. The Laplacian scaling parameter σ^L (Eq. 7) was determined as the squared mean of pairwise data distances in D following [32]. The scale parameter $\sigma^\alpha > 0$ (Eq. 4) was determined similarly. The number of entropy diffusion steps (Eq. 8) was fixed at 10. The size of the neighborhood $\mathcal{N}(x)$ in building L was fixed at 10 as commonly exercised in semi-supervised learning [12]. The explicit Euler discretization (Eq. 8) of the

continuous diffusion process (Eq. 6) is numerically stable only for small step sizes $\delta^L > 0$ [15] and we fixed it at 0.1. The update parameter $\delta^\alpha > 0$ is fixed at the same value.

The time complexity of our algorithm is linear in the number N of total examples and the number M of classes: The main computational bottleneck is in the assessment of the loss and entropy values on D, and the diffusion step of the entropies (Eq. 8). As the graph Laplacian L is sparse, the multiplication $L\mathbf{e}$ takes linear time $\mathcal{O}(|\mathcal{N}| \times N)$. On *CIFAR-100* dataset, selecting a single label took 0.003 s on average.

4 Experiments

Datasets. We evaluated our method on four benchmark datasets: *CIFAR-10* [17], *CIFAR-100* [17], Fashion MNIST (*F-MNIST*) [37], and *Caltech-256* [7]. These datasets are widely used in active label correction (ALC) and learning with noisy labels [2,5,8,14,19]. For all datasets, initially 80% of the ground-truth labels were corrupted using noise models and the remaining clean labels were augmented by performing ALC with a labeling budget G and a batch size Q of 15,000 and 1,000, respectively: The classifier h was trained at every 1,000-th stages, and during the intermediate stages, ALC algorithms queried 1,000 examples to label. The numbers and identities of the original clean labels were not known to ALC algorithms. We considered two label noise models. The *uniform noise* model replaces the ground-truth labels with labels randomly selected from uniform class distributions [2,25]. In the *class-symmetry flipping* model, a class transition probability matrix $T \in \mathbb{R}^{M \times M}$ is first constructed such that T^{ij} is the probability of transition from class i to class j. For each point with class i, a noise label is sampled from the distribution corresponding to the i-th row of T. The entries of T are randomly sampled from the uniform distribution in $[0, 1]$ and each row was probabilistically normalized.

Baselines. We compared with random sampling of labels (*Random*), Nallapati et al.'s CorrActive learning (*CorrActive*) [21], Rebbapragada et al.'s ACL approach [25] which selects a batch of examples with the highest entropies (*Entropy*; the ALC disagreement criterion in [25]), a method that combines the loss and entropy values, inspired by Bernhardt et al.'s active label cleaning approach [2] (*LossEnt*)[2], and Li et al.'s dual active label correction (*DALC*) [19]. Similarly to *Entropy*, *DALC* selects high-entropy examples, and additionally, it estimates the class transition matrix T (as a noise model) to adjust the estimated outputs of h during training: For *uncertain* examples identified during training, a modified loss l' was applied:

$$l'(h(x), y) = l(T^\top h(x), y). \tag{10}$$

[2] This algorithm cannot be directly applied to our setting as it requires multiple annotations for each newly labeled example. Our approach selects the points with the largest sums of the loss and entropy values.

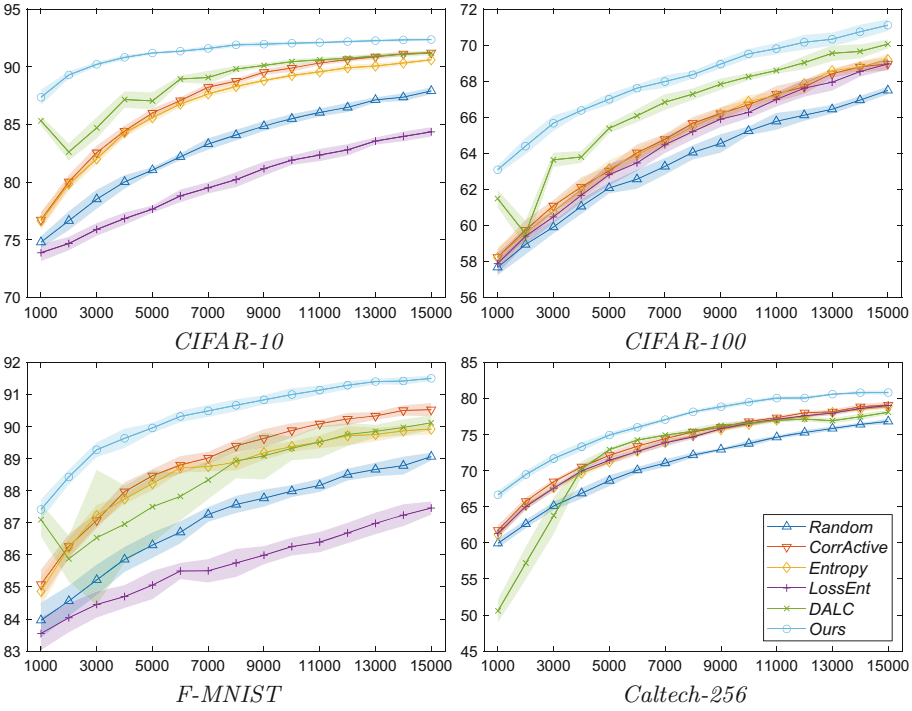

Fig. 2. Mean accuracy (%) with standard deviation (shaded) of different active label correction algorithms under uniform noise. The x-axis corresponds to the number of queried labels. All ALC algorithms outperformed *Random* except for *LossEnt* on *CIFAR-10* and *F-MNIST*. *DALC* demonstrated competitive performance in *CIFAR-10*, *CIFAR-100*, and later learning stages of *Caltech-256*. Our algorithm achieved further significant and consistent improvements.

For all datasets and ALC methods, we conducted experiments 10 times and averaged the results. All experiments were performed on a machine with NVIDIA RTX 3090 GPU, Intel Core i7-11700KF CPU, and 32 GB of RAM. We used the classifier that consists of fixed ResNet-101 [9] pretrained on ImageNet and four fully-connected layers. This configuration constantly outperformed fully trained ResNet-50 and VGG-16 on *CIFAR-10*. We used stochastic gradient descent with an initial learning rate of 0.01, a momentum value of 0.9, and a weight decay factor of 10^{-4}. The learning rate was scaled by 0.1 every 10 epochs.

Results. Figures 2 and 3 summarize the results. All algorithms showed increasing accuracy as more labels were corrected, and they showed similar accuracy progressions for both uniform noise and class-symmetry flipping noise. *Entropy* and *CorrActive* achieved noticeable improvements from *Random* while *LossEnt* was worse than *Random* when the number of classes are limited (*CIFAR-10* and *F-MNIST*). *DALC* demonstrated further significant performance gains by

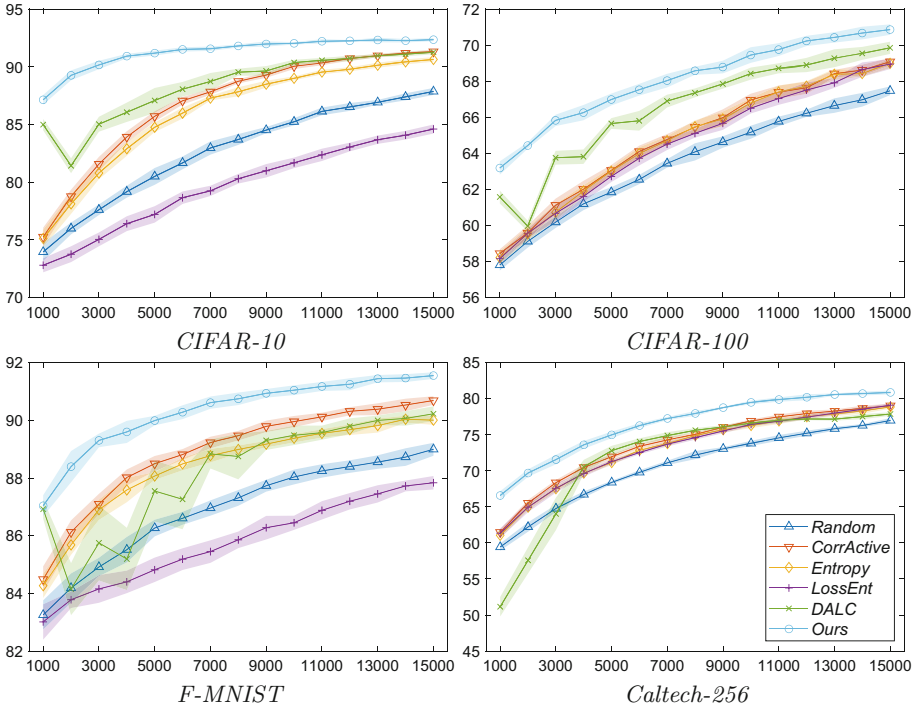

Fig. 3. Mean accuracy (%) with standard deviation (shaded) of different active label correction algorithms under class-symmetry flipping noise.

learning the class transition matrices[3]. Interestingly, it achieved high accuracy even when uniform noise was used: *DALC*'s noise model does not directly match this type of noise. However, simultaneously selecting a batch of examples with the highest entropy values can produce redundant labeling (Fig. 1 shows that high entropy values are indeed spatially correlated). By robustifying classifier training and entropy estimation through the contribution-weighted parameter update (Eq. 2), and promoting diversity in batch selection using entropy diffusion (Eq. 8), our algorithm achieved even more pronounced performance improvements. On *CIFAR-10*, our algorithm reached 90% accuracy at only 3,000 labels, while *DALC* and *CorrActive* required, 9,000 and 11,000 labels, respectively, offering 3- and 3.6-times higher labeling efficiency. Importantly, our algorithm was never significantly worse than other algorithms. Our results on *Caltech-256* (with 256 classes) indicate that it gracefully scales with the number of classes.

[3] DACL's accuracy often decreased in the second iteration as it switches from the entire dataset D to the labeled dataset S^t in estimating the class transition matrix T. At early ALC stages, these data points are limited and the corresponding T estimation is unreliable, leading to degraded performances.

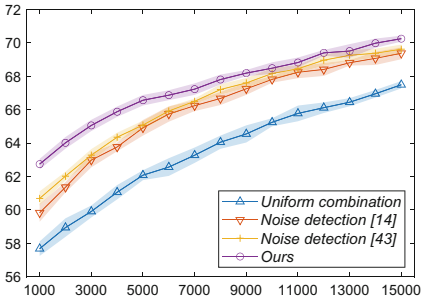

Fig. 4. Performance of our robust parameter update approach (Eq. 2), the standard uniform gradient combination, and the explicit noise detection methods of [14,43]. x- and y-axes show the number of acquired labels and the corresponding classification accuracy (%), respectively. Our *soft* gradient combination approach provides considerably higher labeling efficiency than *hard* noise detection and uniform gradient averaging.

Selection Accuracy of Noisy Examples, and Contributions of Robust Parameter Update and Entropy Propagation. As the number and identities of clean examples D^C are not known and (re-)labeling such examples is redundant, ALC algorithms must select examples for querying from $D \setminus D^C$. Table 1 shows that the superior overall performance of our algorithm can be (partially) explained by its ability to select examples from $D \setminus D^C$.

Figures 4 and 5 demonstrate the effectiveness of our robust parameter update and entropy propagation components.

Table 1. Average noisy example selection accuracy (%) of different ALC algorithms defined as the ratio between the number of queried points in $D \setminus D^C$ and the total number of queries; *CIFAR-100*. Our algorithm consistently achieved the highest selection accuracy.

# labels	1,000	3,000	5,000	7,000	9,000	11,000	13,000	15,000
Random	81.20	80.50	81.50	82.00	80.50	80.00	81.00	82.10
Entropy	90.30	91.30	90.00	91.50	92.60	90.70	90.80	91.40
LossEnt	93.70	93.98	93.02	93.41	93.70	93.79	94.27	93.22
CorrActive	87.60	89.70	87.50	87.60	89.20	89.10	89.70	90.20
DALC	89.10	92.10	91.70	91.10	91.10	90.20	88.30	88.80
Ours	97.90	98.00	97.30	97.80	97.80	97.80	98.10	97.60

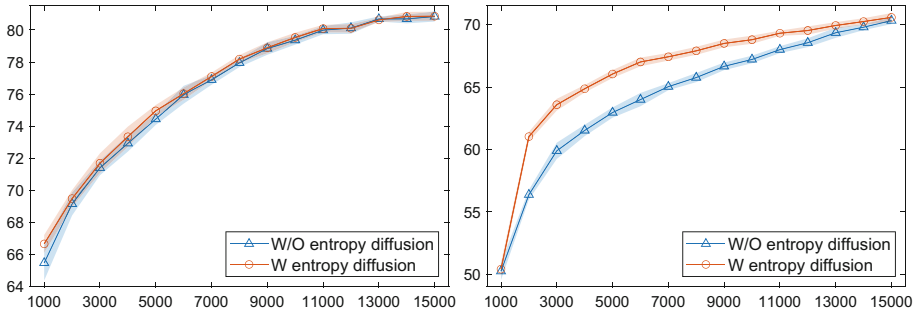

Fig. 5. (Left) Performance of our algorithm with and without entropy diffusion (*Caltech-256*). (Right) Pure label acquisition efficiency of the conventional batch entropy-based selection method and our entropy diffusion method: 15,000 labels were acquired from initial 1,000 known labels (*CIFAR-100*). Entropy diffusion contributes to consistent and statistically significant performance improvements.

5 Conclusions

Existing active label correction (ALC) approaches rely on uncertainty predictions made by unreliable classifiers (trained on noisy samples). Further, simultaneously selecting a batch of ambiguous examples can lead to redundant labeling. Our method addresses these limitations by regulating the contributions of individual parameter gradients via monitoring the progression of losses, and diffusing the entropy value of each newly labeled point avoiding the selection of spatial accumulations. Combining these contributions into a learning framework, our algorithm offers robustness in training under label noise and efficiency in label acquisition without having to know the identity or number of noisy examples. Evaluated on four benchmark datasets, our algorithm demonstrated a significant and consistent performance gain over state-of-the-art methods.

Limitations and Future Work. Our method assumes a uniform cost per label while in practice, the labeling cost can vary across examples: Labeling the most ambiguous examples can rapidly improve classifier performance, but it could also involve considerable annotation time and effort. In such cases, one should carefully trade between the gain of the information and the associated annotation cost. Adjusting our original entropy-based label selection criterion by incorporating a label cost estimation module (e.g., [6,33]) might be possible, but this would involve modifying the entire label acquisition process. Our method is agnostic to the noise generation process and therefore, complementary to noise model-based approaches including DALC. Future work should investigate the possibility of combing the strengths of model-based approaches and ours.

Acknowledgments. We thank James Tompkin for fruitful discussions and the anonymous reviewers for their insightful comments. This work was supported by the National Research Foundation of Korea (NRF) grant (No. 2021R1A2C2012195, Data efficient machine learning for estimating skeletal pose across multiple domains, 1/2) and Institute of Information & Communications Technology Planning & Evaluation (IITP)

grant (No. 2021–0–00537, Visual common sense through self-supervised learning for restoration of invisible parts in images, 1/2) both funded by the Korea government (MSIT).

References

1. Arachie, C., Huang, B.: A general framework for adversarial label learning. JMLR **22**, 1–33 (2021)
2. Bernhardt, M., et al.: Active label cleaning: improving dataset quality under resource constraints. In: arXiv:2109.00574 (2021)
3. Budninskiy, M., Abdelaziz, A., Tong, Y., Desbrun, M.: Laplacian-optimized diffusion for semi-supervised learning. Comput. Aided Geom. Des. **79** (2020)
4. Shampine, L.F.: Tolerance proportionality in ODE codes. In: Bellen, A., Gear, C.W., Russo, E. (eds.) Numerical Methods for Ordinary Differential Equations. LNM, vol. 1386, pp. 118–136. Springer, Heidelberg (1989). https://doi.org/10.1007/BFb0089235
5. Fang, T., Lu, N., Niu, G., Sugiyama, M.: Rethinking importance weighting for deep learning under distribution shift. In: NeurIPS (2020)
6. Gao, R., Saar-Tsechansky, M.: Cost-accuracy aware adaptive labeling for active learning. In: AAAI, pp. 2569–2576 (2020)
7. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical report. California Institute of Technology (2007)
8. Han, B., et al.: Co-teaching: robust training of deep neural networks with extremely noisy labels. In: NIPS (2018)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
10. Hein, M.: Geometrical Aspects of Statistical Learning Theory. Ph.D. thesis. Technical University of Darmstadt, Germany (2005)
11. Hein, M., Audibert, J.-Y., von Luxburg, U.: From graphs to manifolds – weak and strong pointwise consistency of graph laplacians. In: Auer, P., Meir, R. (eds.) COLT 2005. LNCS (LNAI), vol. 3559, pp. 470–485. Springer, Heidelberg (2005). https://doi.org/10.1007/11503415_32
12. Hein, M., Maier, M.: Manifold denoising. In: NIPS, pp. 561–568 (2007)
13. Henter, D., Stahlt, A., Ebbecke, M., Gillmann, M.: Classifier self-assessment: active learning and active noise correction for document classification. In: ICDAR, pp. 276–280 (2015)
14. Huang, J., Qu, L., Jia, R., Zhao, B.: O2U-Net: a simple noisy label detection approach for deep neural networks. In: ICCV, pp. 3326–3334 (2019)
15. Iserles, A.: A First Course in the Numerical Analysis of Differential Equations. Cambridge University Press, 2nd edn. (2012)
16. Kremer, J., Sha, F., Igel, C.: Robust active label correction. In: AISTATS, pp. 308–316 (2018)
17. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Technical report. University of Toronto (2009)
18. Krüger, M., Novo, A.S., Nattermann, T., Mohamed, M., Bertram, T.: Reducing noise in label annotation: a lane change prediction case study. In: IFAC Symposium on Intelligent Autonomous Vehicles, pp. 221–226 (2019)
19. Li, S.-Y., Shi, Y., Huang, S.-J., Chen, S.: Improving deep label noise learning with dual active label correction. Mach. Learn. **111**, 1–22 (2021). https://doi.org/10.1007/s10994-021-06081-9

20. Liu, T., Tao, D.: Classification with noisy labels by importance reweighting. IEEE TPAMI **38**(3), 447–461 (2016)
21. Nallapati, R., Surdeanu, M., Manning, C.: CorrActive learning: learning from noisy data through human interaction. In: IJCAI Workshop on Intelligence and Interaction (2009)
22. Parde, N., Nielsen, R.D.: Finding patterns in noisy crowds: regression-based annotation aggregation for crowdsourced data. In: EMNLP, pp. 1907–1912 (2017)
23. Park, S., Jo, D.U., Choi, J.Y.: Over-fit: noisy-label detection based on the overfitted model property. In: arXiv:2106.07217 (2021)
24. Pierri, F., Piccardi, C., Ceri, S.: Topology comparison of twitter diffusion networks effectively reveals misleading information. Sci. Rep. **10**(1372), 1–19 (2020)
25. Rebbapragada, U., Brodley, C.E., Sulla-Menashe, D., Friedl, M.A.: Active label correction. In: ICDM, pp. 1080–1085 (2012)
26. Rehbein, I., Ruppenhofer, J.: Detecting annotation noise in automatically labelled data. In: ACL, pp. 1160–1170 (2018)
27. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: ICML (2018)
28. Rosenberg, S.: The Laplacian on a Riemannian Manifold. Cambridge University Press (2009)
29. Shen, Y., Sanghavi, S.: Learning with bad training data via iterative trimmed loss minimization. In: ICML (2019)
30. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get another label? improving data quality and data mining using multiple, noisy labelers. In: KDD, pp. 614–622 (2009)
31. Stokes, J.W., Kapoor, A., Ray, D.: Asking for a second opinion: re-querying of noisy multi-class labels. In: ICASSP, pp. 2329–2333 (2016)
32. Szlam, A.D., Maggioni, M., Coifman, R.R.: Regularization on graphs with function-adapted diffusion processes. JMLR **9**, 1711–1739 (2008)
33. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., Ding, X.: Embracing imperfect datasets: a review of deep learning solutions for medical image segmentation. Med. Image Anal. **63** (2020)
34. Urner, R., David, S.B., Shamir, O.: Learning from weak teachers. In: AISTATS, pp. 1252–1260 (2012)
35. van Rooyen, B., Menon, A.K., Williamson, R.C.: Learning with symmetric label noise: the importance of being unhinged. In: NIPS (2015)
36. Wang, S., et al.: Annotation-efficient deep learning for automatic medical image segmentation. Nat. Commun. **12**(1), 1–13 (2021)
37. Xiao, H., Rasul, K., Vollgraf, R.: FashionMNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017)
38. Yan, S., Chaudhuri, K., Javidi, T.: Active learning from imperfect labelers. In: NIPS (2016)
39. Younesian, T., Epema, D., Chen, L.Y.: Active learning for noisy data streams using weak and strong labelers. arXiv:2010.14149v1 (2020)
40. Zhang, C., Chaudhuri, K.: Active learning from weak and strong labelers. In: NIPS (2015)
41. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: ICLR (2017)
42. Zhang, M., Hu, L., Shi, C., Wang, X.: Adversarial label-flipping attack and defense for graph neural networks. In: ICDM (2020)
43. Zhu, Z., Dong, Z., Liu, Y.: Detecting corrupted labels without training a model to predict. In: ICML (2022)
44. Ørting, S.N., et al.: A survey of crowdsourcing in medical image analysis. Hum. Comput. **7**, 1–26 (2020)