ITERATIVE LABEL IMPROVEMENT:

ROBUST TRAINING BY CONFIDENCE BASED FILTERING & DATASET PARTITIONING

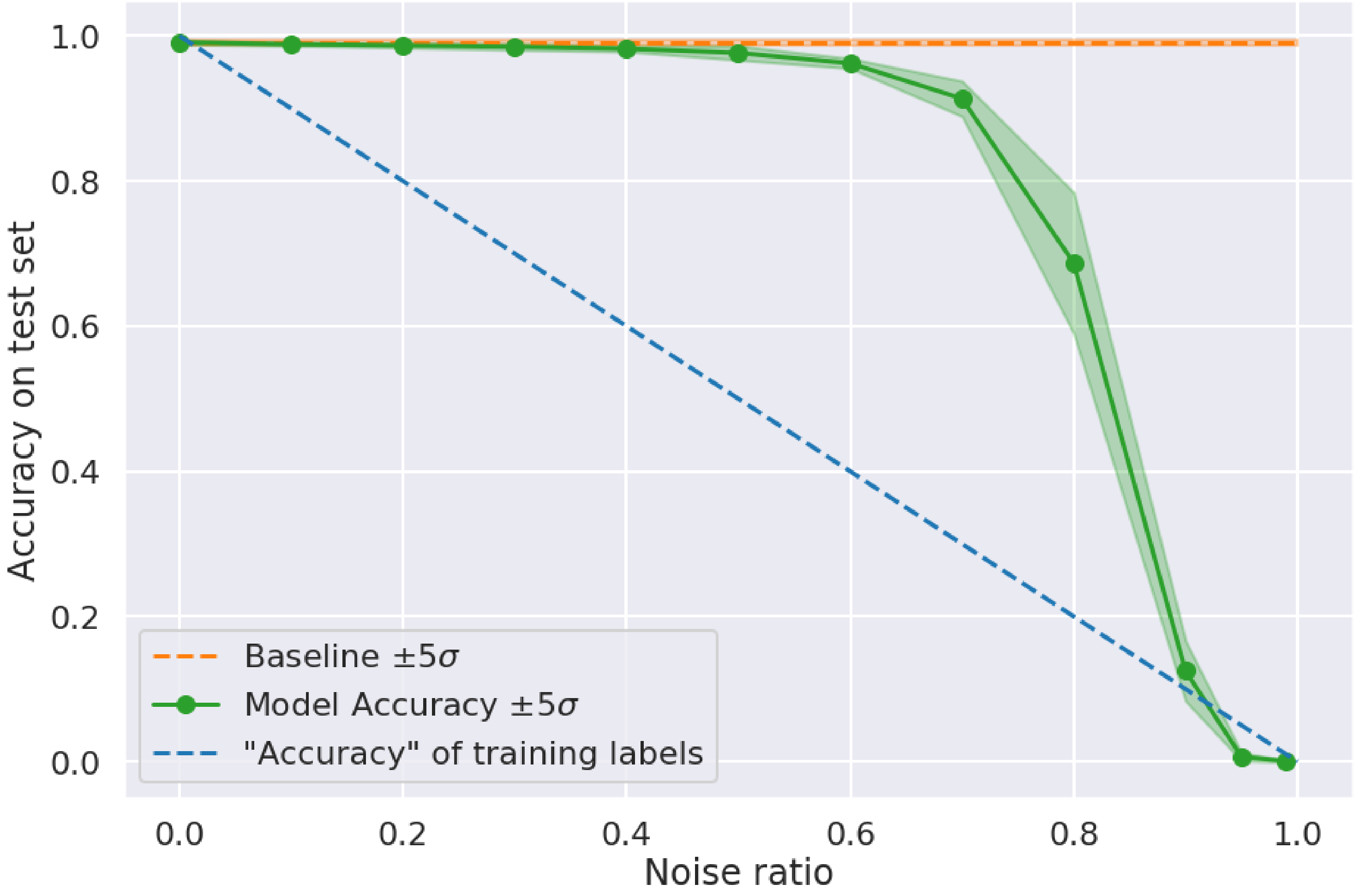
INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION

CHRISTIAN HAASE-SCHUETZ[1,2], RAINER STAL[1], HEINZ HERTLEIN[1], BERNHARD SICK[2]
[1]ENGINEERING COGNITIVE SYSTEMS, CHASSIS SYSTEMS CONTROL, ROBERT BOSCH GMBH, GERMANY
[2] INTELLIGENT EMBEDDED SYSTEMS GROUP, UNIVERSITY OF KASSEL, GERMANY

U N I KASSEL
V E R S I T Ä T

BOSCH

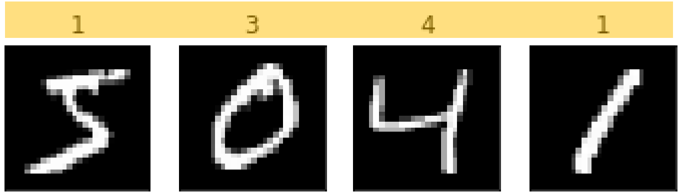
"Labelling quality influences detection performance"



UNIKASSEL VERSITÄT

BOSCH

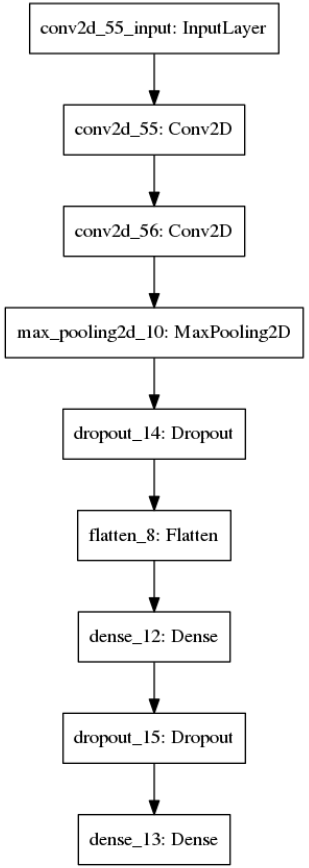
"By manually adding noise to well known datasets, we study the effect of erroneous labels and propose a technique to mitigate them"

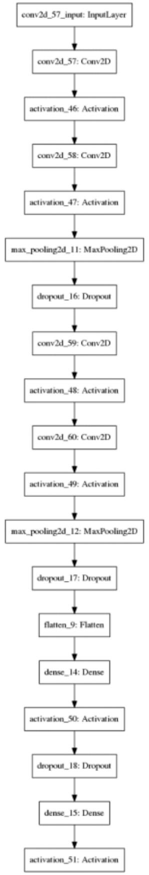


UNIKASSEL VERSITÄT

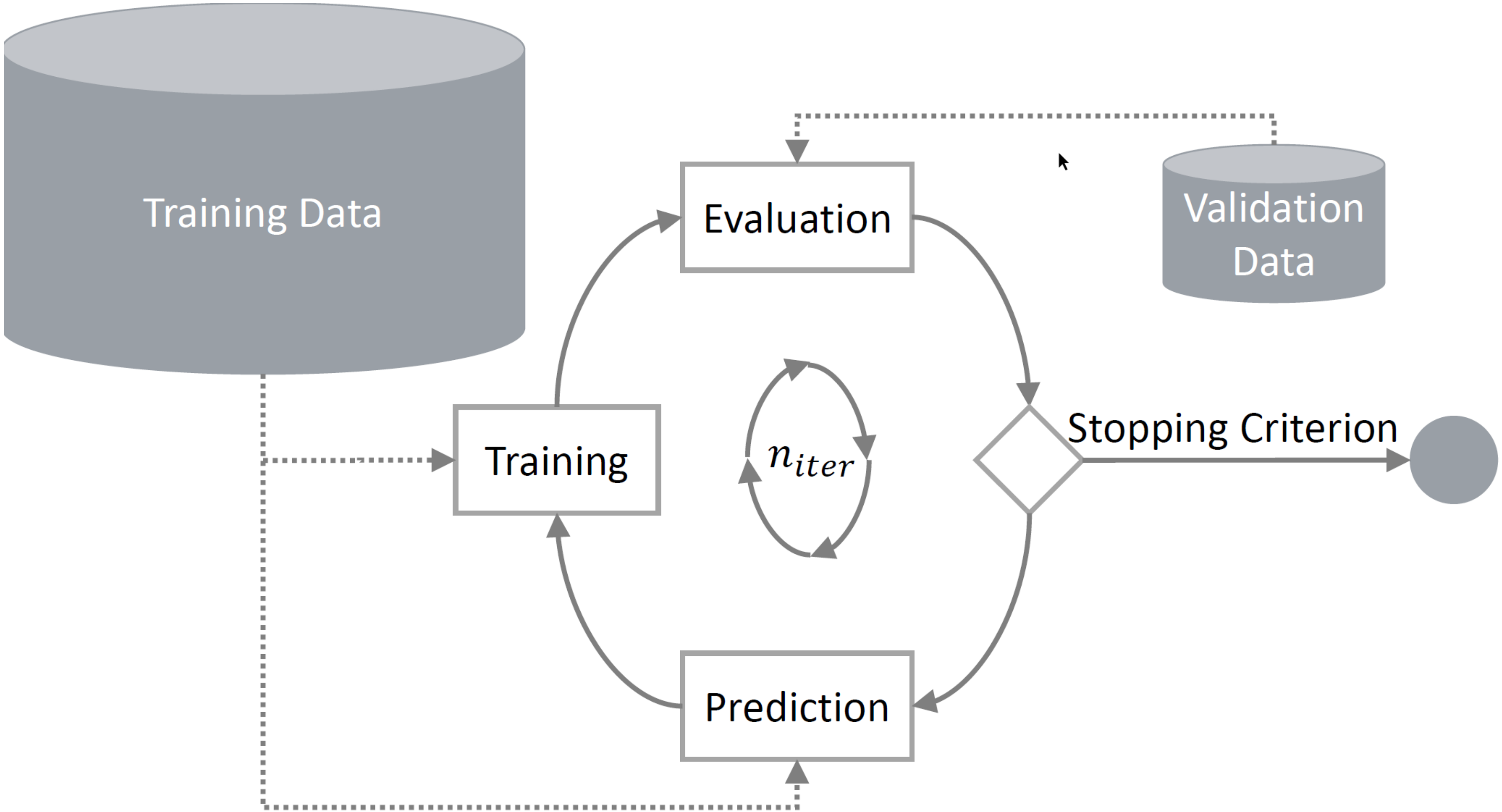
BOSCH

Three networks used
MNIST-CNN, CIFAR-CNN, ResNet32/50





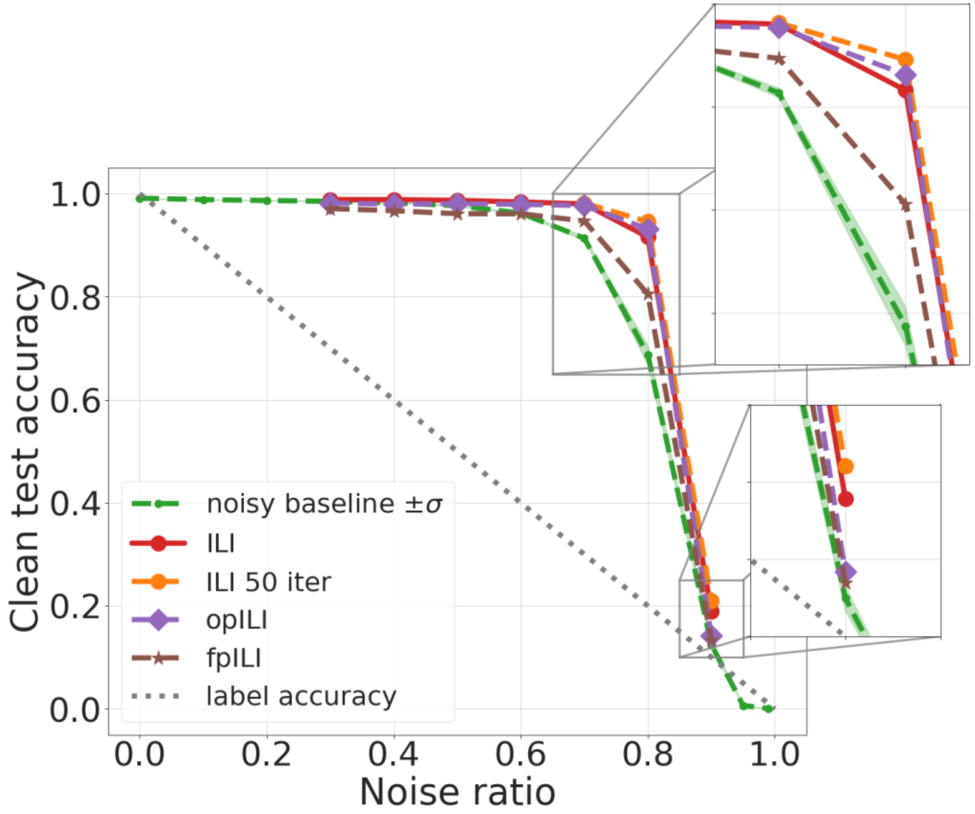
The Idea: Iterative Label Improvement – ILI



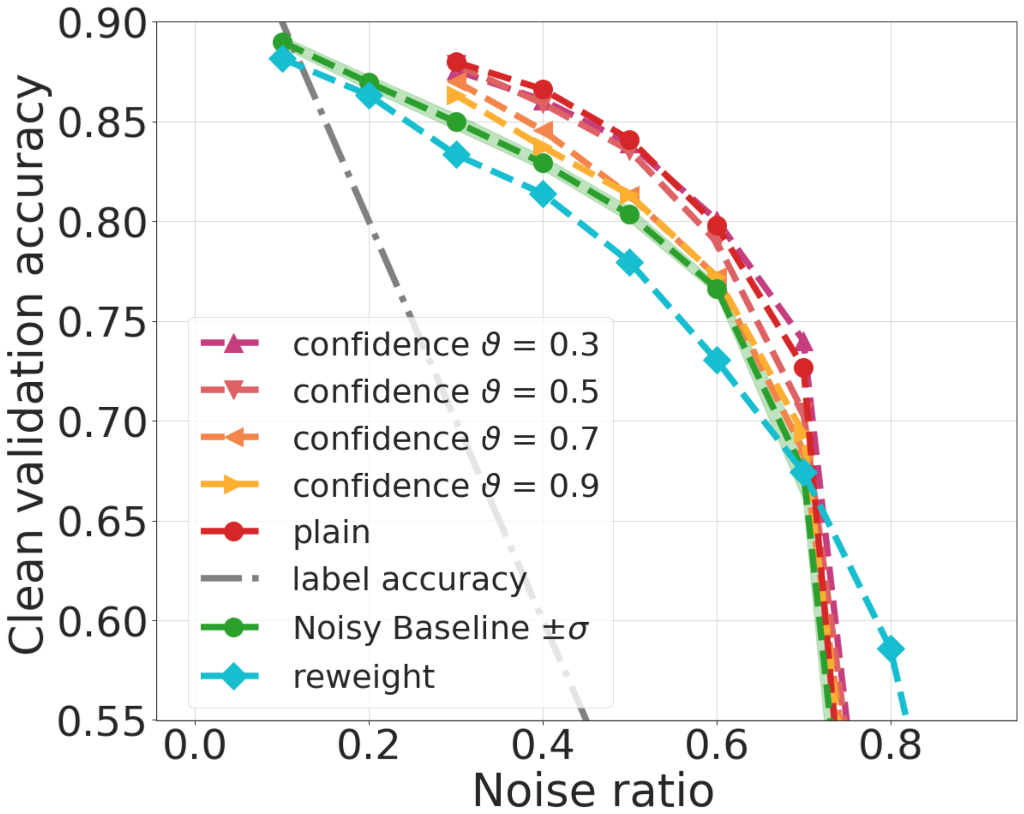
UNIKASSEL VERSITÄT

BOSCH

Some Results



ILI with MNIST-CNN on noisy MNIST data with random error



ILI with ResNet32 on noisy CIFAR10 data with random error, vs. "learning to reweight"

BACKUP

Different versions of ILI introduced

Algorithm 2 opILI with initILI

Input: $X_{\text{train, A}}, y_{\text{train}}, n_{\text{iter}}$

Output: Model $m^{n_{\text{iter}}}$, $y_{\text{train}}^{n_{\text{iter}}}$

$X_{\text{train, A}}, X_{\text{train, B}} = \text{Split}(X_{\text{train}})$

$m_{\text{A}}^{(0)}.initialize()$

$m_{\text{A}}^{(0)}.fit(X_{\text{train, A}}, y_{\text{train, A}}^{(0)})$

for $i = 1$ to n_{iter} **do**

    $y_{\text{train, B}}^{(i)} = m_{\text{A}}^{(i-1)}(X_{\text{train, B}})$

    $m_{\text{B}}^{(i)}.initialize()$

    $m_{\text{B}}^{(i)}.fit(X_{\text{train, B}}, y_{\text{train, B}}^{(i)})$

    $y_{\text{train, A}}^{(i)} = m_{\text{B}}^{(i)}(X_{\text{train, A}})$

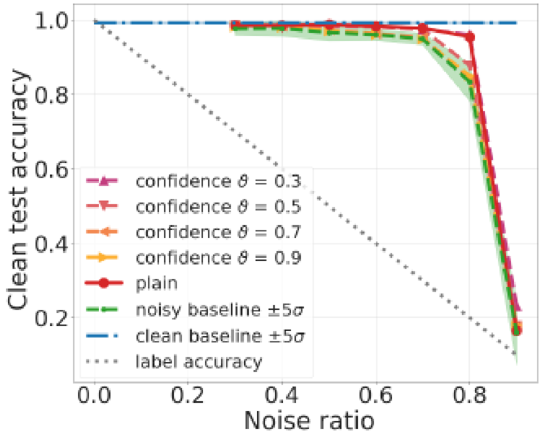
    $m_{\text{A}}^{(i)}.initialize()$

    $m_{\text{A}}^{(i)}.fit(X_{\text{train, A}}, y_{\text{train, A}}^{(i)})$

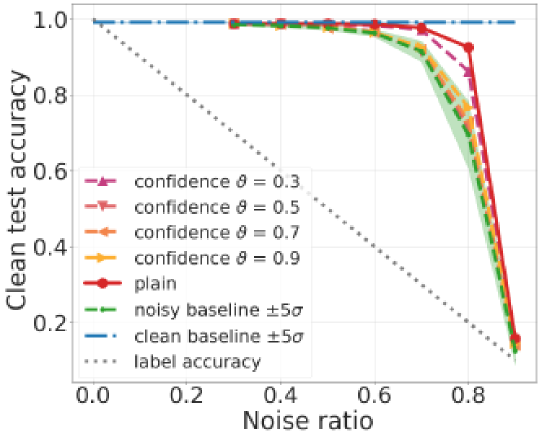
end for

$y_{\text{train}}^{n_{\text{iter}}} = \left[y_{\text{train, A}}^{(n_{\text{iter}})}, y_{\text{train, B}}^{(n_{\text{iter}})} \right]$

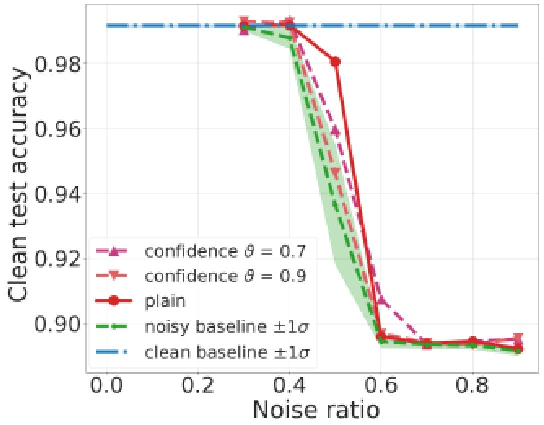
Results – different models & datasets



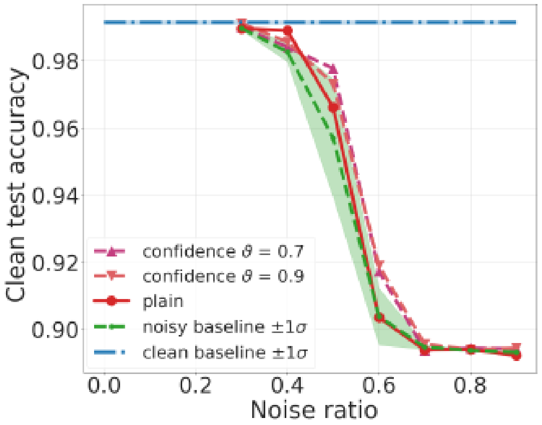
(a) CIFAR-CNN on noisy MNIST data with random error.



(b) MNIST-CNN on noisy MNIST data with random error.



(c) CIFAR-CNN on noisy MNIST data with bias error.



(d) MNIST-CNN on noisy MNIST data with bias error.

UNIKASSEL VERSITÄT

BOSCH

The effect of data augmentation

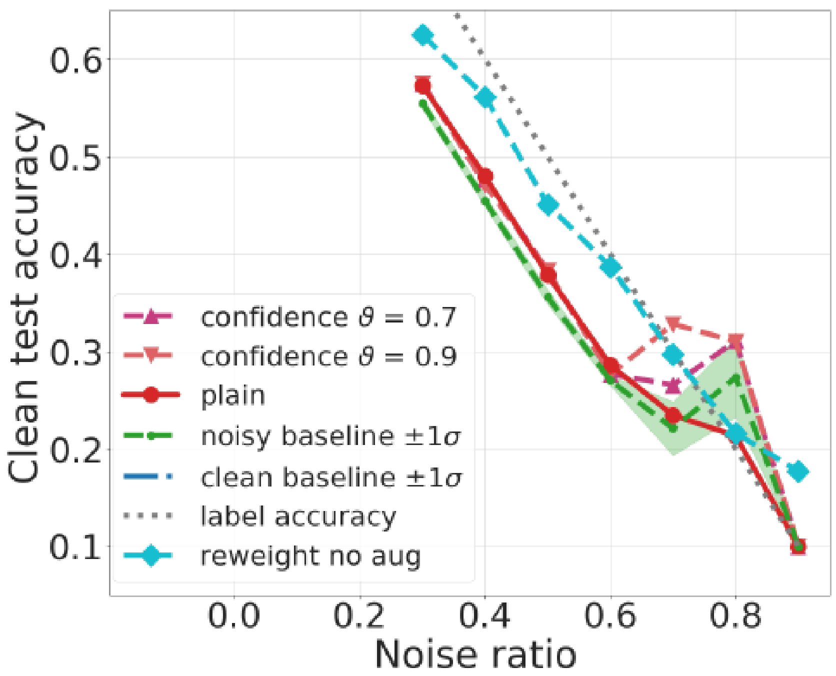


Figure 5. Comparison of different versions of our ILI algorithm on CIFAR10 data, using a ResNet32, with erroneous labels (randomly distributed). As a reference we compare to "learning to reweight" (Ren et al., 2018). Without data augmentation both methods fail to improve the accuracy significantly.

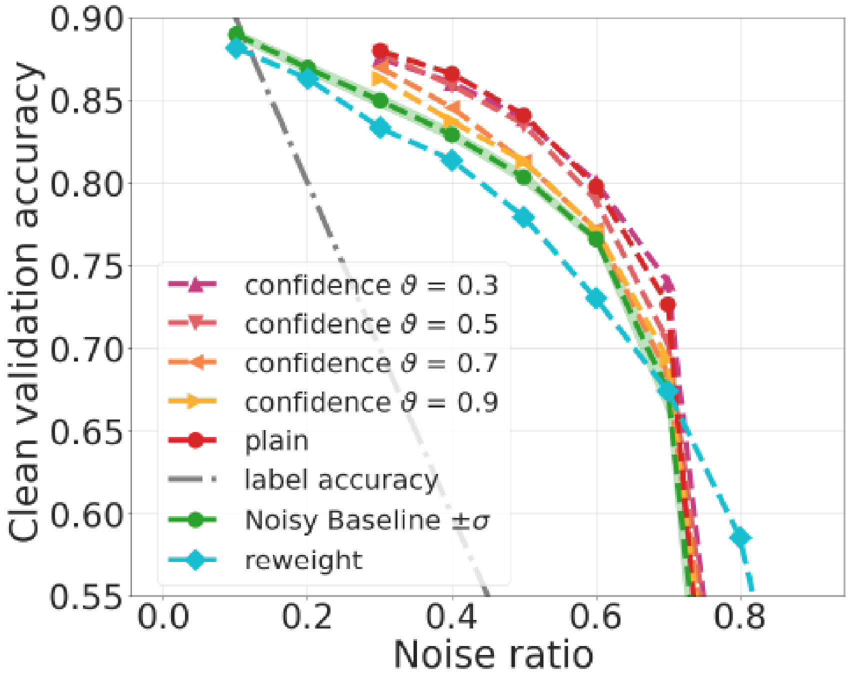


Figure 6. Comparison of different versions of our ILI algorithm on CIFAR10 data, using a ResNet32, with erroneous labels (randomly distributed) vs. "learning to reweight". With data augmentation our method outperforms "learning to reweight" for most noise fractions.

ResNet32, Noise fraction 0.1