南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

CLIPCleaner: Cleaning Noisy Labels with CLIP

Chen Feng
Queen Mary University of London
London, UK
chen.feng@qmul.ac.uk

Georgios Tzimiropoulos
Queen Mary University of London
London, UK
g.tzimiropoulos@qmul.ac.uk

Ioannis Patras
Queen Mary University of London
London, UK
i.patras@qmul.ac.uk

MM 2024

Background

Noisy labels learning:
1. develop robust loss functions
2. model the labeling error patterns with a label transition matrix

Problem: these methods are often sub-optimal in dealing with high noise ratios and complicated noise patterns.

Sample selection based the fact that the model tends to fit clean samples earlier than noisy samples in the training process.

Problem:
1. some of the label noise is between classes that are visually very similar (‘hard noise’).
2. ‘self-confirmation’ bias: the in-training model, is at least partially trained on the noisy labels.

Method

Problem:
1. some of the label noise is between classes that are visually very similar (‘hard noise’).
2. ‘self-confirmation’ bias: the in-training model, is at least partially trained on the noisy labels.

Introduce CLIP:
1. the sample selection is aware of visual and semantic similarities between the classes and therefore compensates for biases that may arise from relying solely on visual information for sample selection.
2. the sample selection is independent of the in-training model, and therefore immune to the influence of noisy labels and the ‘self-confirmation’ bias.

Note:
1. the final classifier, is different from the VL model that is used for sample selection.
2. we adhere to using CLIP solely for sample selection and refrain from training/fine-tuning it

Method

Preliminary on CLIP

$$L(x_i', z_i; g, h) = \frac{1}{2}\Big(- \log \frac{\exp(g(x_i')^T h(z_i))}{\sum_{j=1}^{M} \exp(g(x_i')^T h(z_j))} - \log \frac{\exp(g(x_i')^T h(z_i))}{\sum_{j=1}^{M} \exp(g(x_j')^T h(z_i))}\Big). \tag{2}$$

Estimate P (y|x = xi) with CLIP zero-shot classifier.

$$\begin{aligned} P_{zeroshot}(\mathrm{y} = y_i | \mathrm{x} = x_i) &= \int Q(\mathrm{y} = y_i | \mathrm{z} = z_i) Q(\mathrm{z} = z_i | \mathrm{x} = x_i) dz \\ &\propto \int Q(\mathrm{y} = y_i | \mathrm{z} = z_i) Q(\mathrm{z} = z_i, \mathrm{x} = x_i) dz. \end{aligned} \tag{3}$$

$$Q(\mathrm{z} = z_i, \mathrm{x} = x_i) \propto \exp(g(x_i)^T h(z_i))).$$

$$Q(\mathrm{y} = y_i | \mathrm{z} = \text{‘}\texttt{A photo of class name of } y_i\texttt{.}\text{’}) \approx 1.$$

$$P_{zeroshot}(\mathrm{y} = y_i | \mathrm{x} = x_i) \underset{\sim}{\propto} \sum_{j=1}^{J} \tilde{Q}(\mathrm{z} = \mathcal{P}_j, \mathrm{x} = x_i). \tag{4}$$

Method

Theoretical justification of CLIPCleaner

$$P_{induced}(y|\mathbf{x}=\boldsymbol{x}_i) = \mathrm{softmax}(f'(g(\boldsymbol{x}_i))). \qquad (8)$$

An immediate question is: how does the zero-shot classifier (eq. (4)) compare to the induced classifier here (eq. (8)) in estimating the clean conditional probability?

Estimation with zero-shot classifier $d(P_{zeroshot}, P) \le \varepsilon_{domain} + \Delta(\lambda_0 \varepsilon_{clip} + \lambda_1 \mathfrak{R}(\mathcal{G} \circ \mathcal{H}) + \lambda_2 l_{\infty}^{clip} \sqrt{\frac{\log 1/\delta}{M}} + \lambda_3 \varepsilon_n)$

Estimation with induced classifier $d(P_{induced}, P) \le \varepsilon_{noise} + \lambda_0 \varepsilon_{induced} + \lambda_1 \mathfrak{R}(\mathcal{F}) + \lambda_2 l_{\infty}^{noisy} \sqrt{\frac{\log 1/\delta}{N}}$

ignoring the uncontrollable and common bound error terms (marked in gray),

ε_domain denotes the bias term induced by the domain gap between Q and P_true
ε_noise denotes the difference term induced by the label noise in the training dataset

the zero-shot classifier is affected by domain gap and prompts quality
the induced classifier is affected by the label noise of the noisy dataset

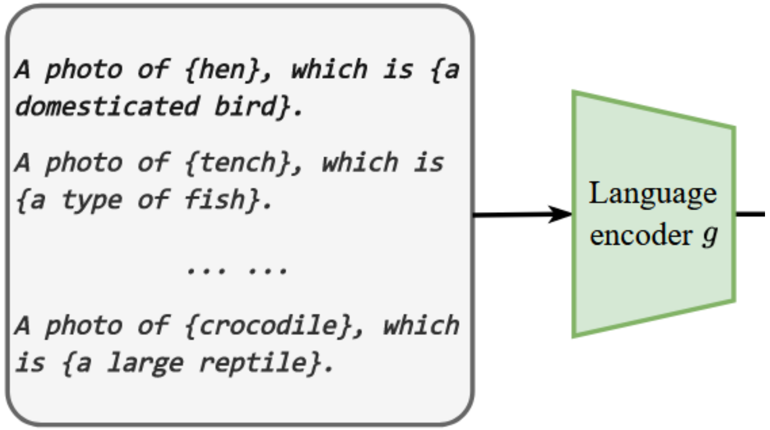
Method

better prompt engineering, and ε_domain can be minimized by training CLIP with a more diverse dataset

$$\mathcal{P}_j = \text{`A photo of \{class name of } y_i\text{\}, which is/has \{class-specific feature } j \text{ of class } y_i\text{\}.'}$$

using class-specific features such as the unique color or habitat of different animal species in an animal classification task.

Prompts with class-specific features

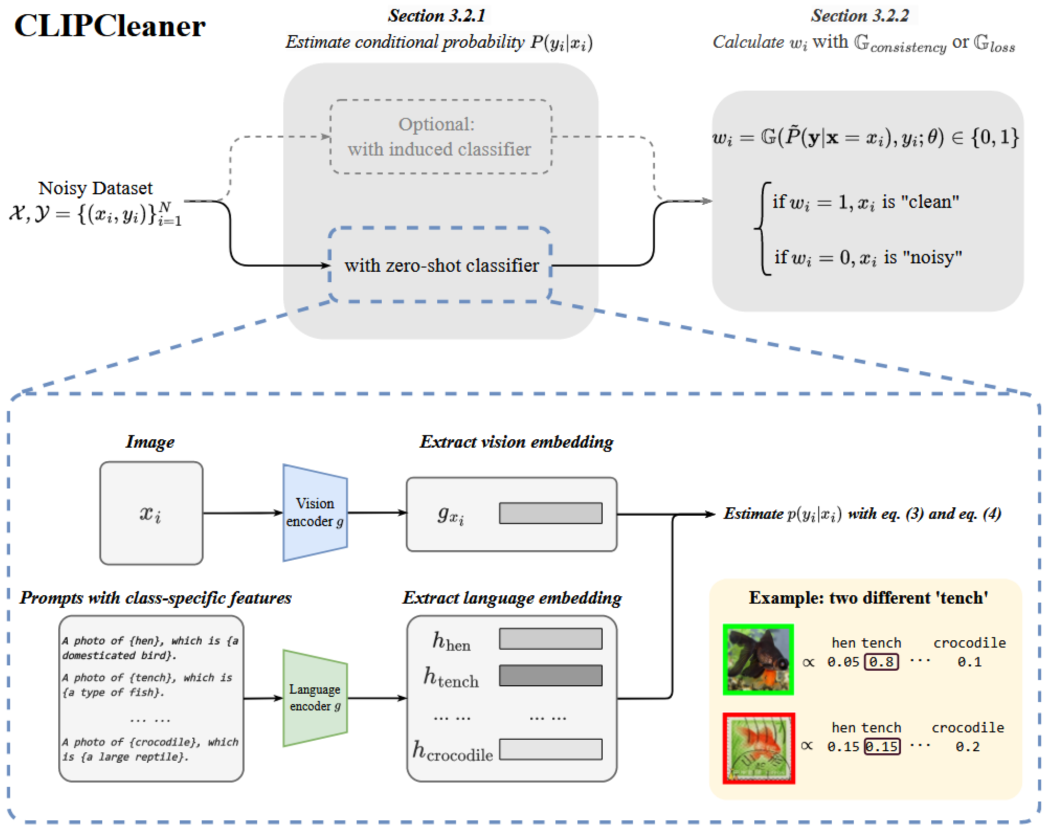


Method

$$\mathbb{G}_{consistency} = \mathbb{I}\left(\frac{\tilde{P}(\mathrm{y} = y_i|\mathbf{x} = x_i)}{\max_k \tilde{P}(\mathrm{y} = k|\mathbf{x} = x_i)} \geq \theta_{consistency}\right). \quad (5)$$

$$\mathbb{G}_{loss} = \mathbb{I}(\mathbb{P}(-\log \tilde{P}(\mathrm{y} = y_i|\mathbf{x} = x_i) \in \mathrm{GMM}_{small}) \geq \theta_{loss}). \quad (6)$$

In this work, we adopt a conservative strategy by taking the intersection of different sample selection results, prioritizing the precision of sample selection.



MixFix: Efficient semi-supervised training by absorbing and relabelling

Motivated by FixMatch, we also inspect in unlabeled subset each sample's current prediction p_i based on the in-training model *f*

$$(w_i, y_i) = \begin{cases} (0, y_i), \text{ if } p_m < \theta_r \text{ and } p_m < \theta'_r & \texttt{*Drop*} \\ (1, y_i), \text{ if } p_m > \theta_r \text{ and } y_i = y_m & \texttt{*Absorb*} \\ (1, y_m), \text{ if } p_m > \theta'_r \text{ and } y_i \neq y_m & \texttt{*Relabel*} \end{cases} \quad (7)$$

Different from FixMatch [43] using one threshold for all samples, we typically set $\theta r \leq \theta' r$. This allows us to fully leverage noisy labels to distinguish between the 'absorb' and 'relabel' processes.

Experiment

Table 1: Ablations on *MixFix* with synthetic CIFAR100 noisy dataset. The *top-3* results are bolded.

θ_r	θ'_r	Noise ratio 20%	50%	80%	90%
0.7	0.7	76.46	74.69	**69.50**	62.91
	0.8	**76.63**	**75.23**	**69.72**	**63.11**
	0.9	**77.06**	**75.17**	67.76	59.17
0.8	0.7	75.49	74.30	67.95	**63.29**
	0.8	76.36	**74.90**	68.86	**63.42**
	0.9	**76.66**	74.50	67.37	58.09
0.9	0.7	74.53	73.49	68.74	62.22
	0.8	75.98	74.25	**68.94**	62.81
	0.9	75.78	74.23	67.17	59.38

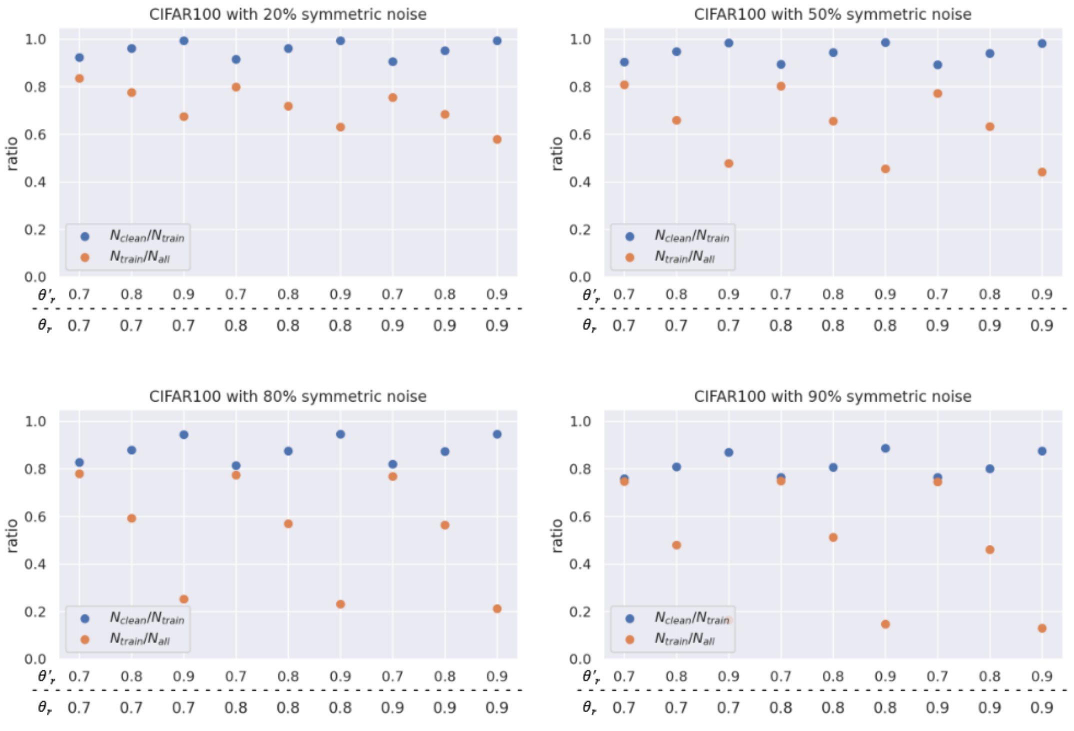


Figure 3: N_{train} denotes number of training samples, N_{clean} denotes number of clean training samples and N_{all} denotes number of clean training samples.

Especially, after reducing the 'absorb' threshold θ′r , the proportion of training samples increases and the accuracy of training samples decreases.

Experiment

Analyzing CLIP Zero-shot classification as a baseline

Table 2: Testing accuracy (%) with CLIP zero-shot classifier.

Model	CIFAR10	CIFAR100	Red Mini-ImageNet	WebVision	Clothing1M	ANIMAL-10N
CLIP	89.97	63.72	78.12	73.36	39.73	76.12
SOTA	92.68 [22]	67.7 [22]	49.55 [16]	80.9 SSR+ [10]	74.84 C2D [68]	88.5 SSR+ [10]
Ours	**95.15**	**71.17**	**54.21**	**81.56**	**74.87**	**88.85**

Experiment

Analyzing sample selection w.r.t different classifiers and different mechanisms

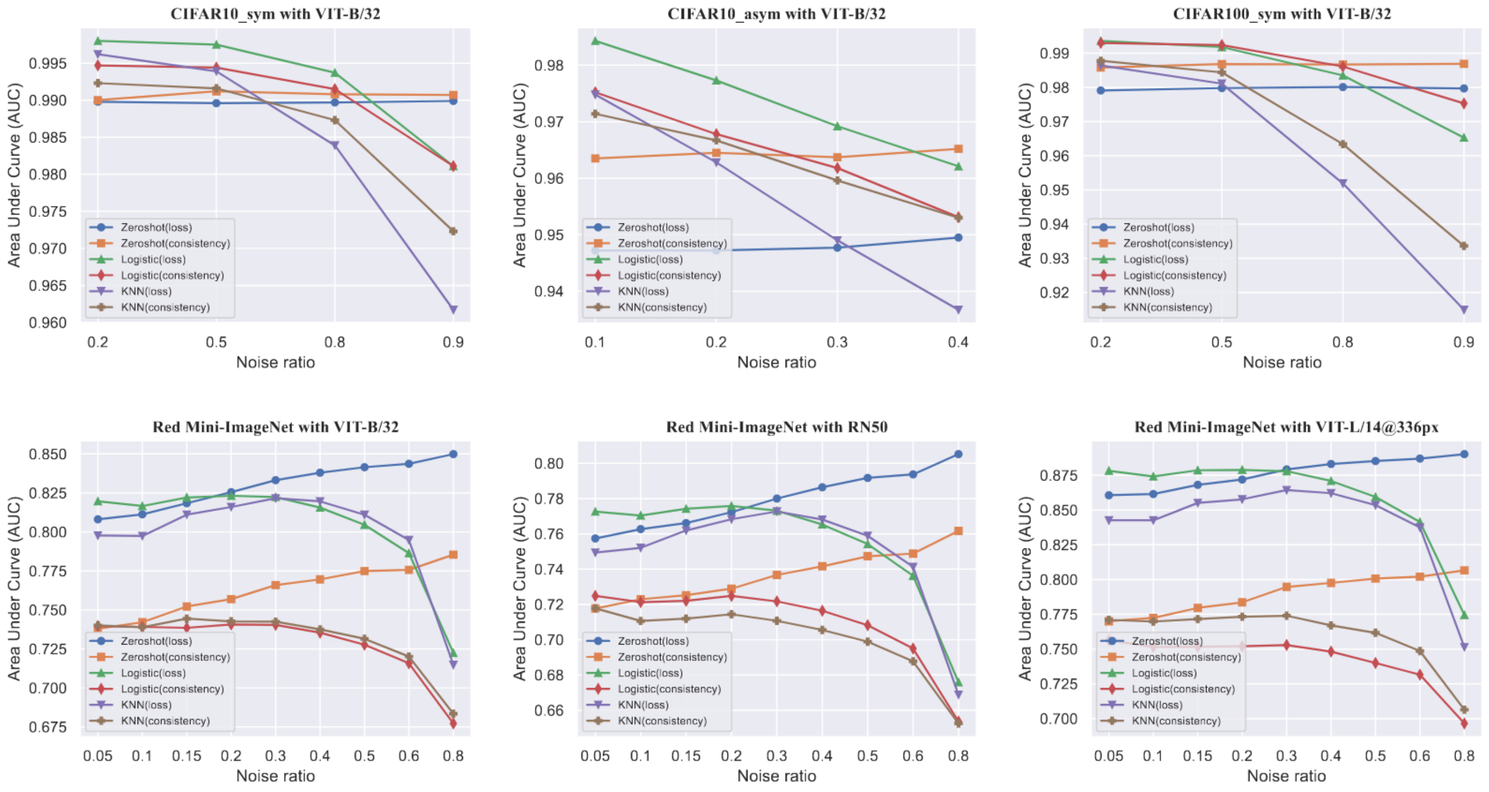


Figure 4: Comparisons of various sample selection methods *w.r.t* different dataset/noise type/noise ratio. Here, we show the ROC AUC score of binary identification of clean samples.

the zero-shot classifier gradually outperforms the induced classifier that the latter is affected by label noise while the former is not;
we find that different sample selection mechanisms show distinct advantages and disadvantages on different datasets.
the LogisticRegression classifier empirically exhibits superior performance to the kNN classifier.

Experiment

Table 3: Testing accuracy (%) on CIFAR10 with instance-dependent noise.

Method	Noise ratio			
	10%	20%	30%	40%
CE	91.25	86.34	80.87	75.68
F-correction [41]	91.06	86.35	78.87	71.12
Co-teaching [19]	91.22	87.28	84.33	78.72
GCE [67]	90.97	86.44	81.54	76.71
DAC [47]	90.94	86.16	80.88	74.80
DMI [58]	91.26	86.57	81.98	77.81
SEAL [4]	91.32	87.79	85.30	82.98
CE*	90.76	86.08	80.64	75.27
CLIPCleaner + CE	**92.33±0.37**	**91.06±0.37**	**89.71±0.37**	**88.26±0.37**

Table 4: Testing accuracy (%) on CIFAR-10 and CIFAR-100 with synthetic noise.

Dataset	CIFAR10					CIFAR100			
Noise type	Symmetric				Assymetric	Symmetric			
Noise ratio	20%	50%	80%	90%	40%	20%	50%	80%	90%
CE	86.8	79.4	62.9	42.7	85.0	62.0	46.7	19.9	10.1
Co-teaching+ [63]	89.5	85.7	67.4	47.9	-	65.6	51.8	27.9	13.7
F-correction [41]	86.8	79.8	63.3	42.9	87.2	61.5	46.6	19.9	10.2
PENCIL [62]	92.4	89.1	77.5	58.9	88.5	69.4	57.5	31.1	15.3
LossModelling [1]	94.0	92.0	86.8	69.1	87.4	73.9	66.1	48.2	24.3
DivideMix [29]	**96.1**	94.6	93.2	76.0	93.4	77.3	74.6	60.2	31.5
ELR+ [34]	95.8	94.8	93.3	78.7	93.0	77.6	73.6	60.8	33.4
MOIT [38]	93.1	90.0	79.0	69.6	92.0	73.0	64.6	46.5	36.0
SelCL+ [31]	95.5	93.9	89.2	81.9	93.4	76.5	72.4	59.6	48.8
TCL [22]	95.0	93.9	92.5	89.4	92.6	78.0	73.3	65.0	54.5
Ours	95.92±0.15	**95.67±0.28**	**95.04±0.37**	**94.23±0.54**	**94.89±0.16**	**78.20±0.45**	**75.23±0.29**	**69.72±0.61**	**63.11±0.89**

Table 5: Testing accuracy (%) on Clothing1M.

CE	F-correction [41]	RRL [30]	C2D [68]	DivideMix [29]	ELR+ [34]	SSR+ [10]	TCL [22]	Ours	Ours (Co-training)	CLIPCleaner + DivideMix
69.21	69.84	74.30	74.84	74.76	74.81	74.83	74.80	73.41±0.65	74.01±0.47	**74.87±0.44**

incorporated to two additional schemes

Table 5: Testing accuracy (%) on Clothing1M.

CE	F-correction [41]	RRL [30]	C2D [68]	DivideMix [29]	ELR+ [34]	SSR+ [10]	TCL [22]	Ours	Ours (Co-training)	CLIPCleaner + DivideMix
69.21	69.84	74.30	74.84	74.76	74.81	74.83	74.80	73.41±0.65	74.01±0.47	**74.87±0.44**

Clothing1M dataset is more fine-grained than other datasets. For such fine-grained noisy datasets, sample selection may not be the optimal strategy.

Experiment

Table 6: Testing accuracy (%) on WebVision.

Methods	WebVision		ILSVRC2012	
	Top1	Top5	Top1	Top5
Co-teaching [19]	63.5	85.20	61.48	84.70
DivideMix [29]	77.32	91.64	75.20	90.84
ELR+ [34]	77.78	91.68	70.29	89.76
NGC [54]	79.16	91.84	74.44	91.04
FaMUS [59]	79.4	92.8	77.0	92.8
RRL [30]	76.3	91.5	73.3	91.2
SelCL+ [31]	79.9	92.6	76.8	**93.0**
SSR+ [10]	80.9	92.8	75.8	91.8
TCL [22]	79.1	92.3	75.4	92.4
Ours	**81.56±0.29**	**93.26±0.65**	**77.80±0.25**	92.08±0.44

Table 7: Testing accuracy (%) on Red Mini-ImageNet.

Method	Noise ratio			
	20%	40%	60%	80%
CE	47.36	42.70	37.30	29.76
Mixup [64]	49.10	46.40	40.58	33.58
DivideMix [29]	50.96	46.72	43.14	34.50
MentorMix [23]	51.02	47.14	43.80	33.46
FaMUS [59]	51.42	48.06	45.10	35.50
InstanceGM [16]	58.38	52.24	47.96	39.62
Ours	**61.44±0.45**	**58.42±0.66**	**53.18±0.47**	**43.82±0.87**

Table 8: Testing accuracy (%) on ANIMAL-10N.

Method	Accuracy
CE	79.4
SELFIE [44]	81.8
PLC [66]	83.4
NCT [6]	84.1
InstanceGM [16]	84.6
SSR+ [10]	88.5
Ours	**88.85±0.61**

Thank you