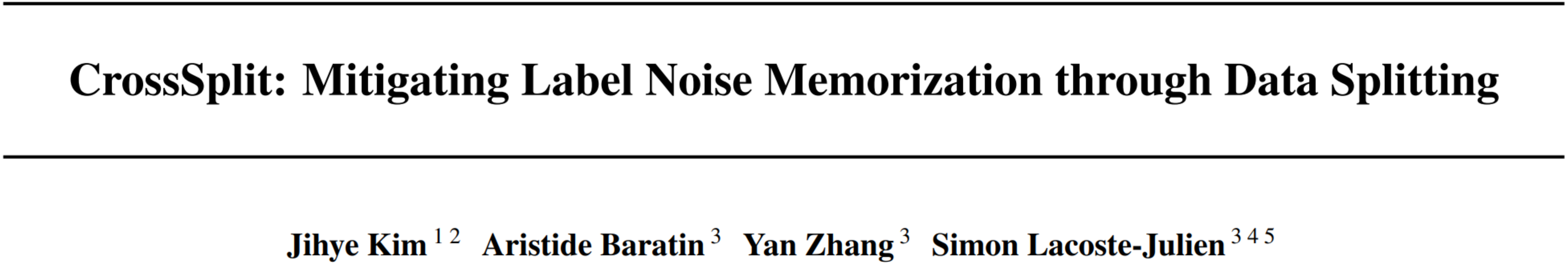


CrossSplit: Mitigating Label Noise Memorization through Data Splitting

Jihye Kim [1] [2] Aristide Baratin [3] Yan Zhang [3] Simon Lacoste-Julien [3] [4] [5]

ICML 2023

2023.7.21

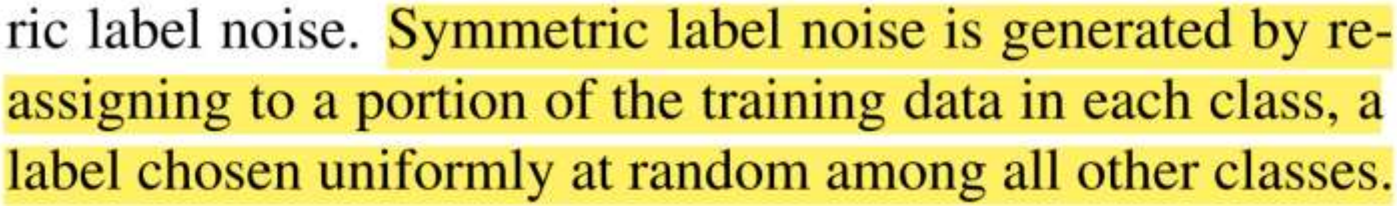
Motivation: Improving robustness of model in the presence of **label noise.**
(数据集中有时会存在一些错误的标注，而这些错误标注会对训练带来影响)

Label noise:

1. **Simulated label noise** controls the noise level, analyze the memorization behavior of our algorithm and test a variety of scenarios.
2. **Natural label noise** enables practical evaluation in situations where the type and level of noise are unknown.

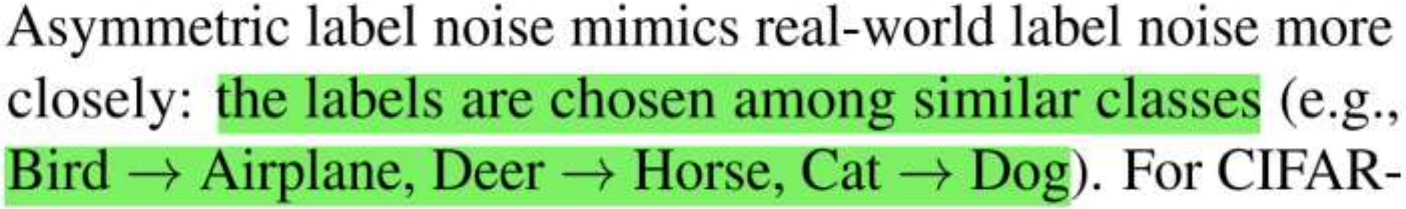
Simulated label noise :

1. **Symmetric:**



ric label noise. Symmetric label noise is generated by re-assigning to a portion of the training data in each class, a label chosen uniformly at random among all other classes.

2. **Asymmetric:**



Asymmetric label noise mimics real-world label noise more closely: the labels are chosen among similar classes (e.g., Bird → Airplane, Deer → Horse, Cat → Dog). For CIFAR-

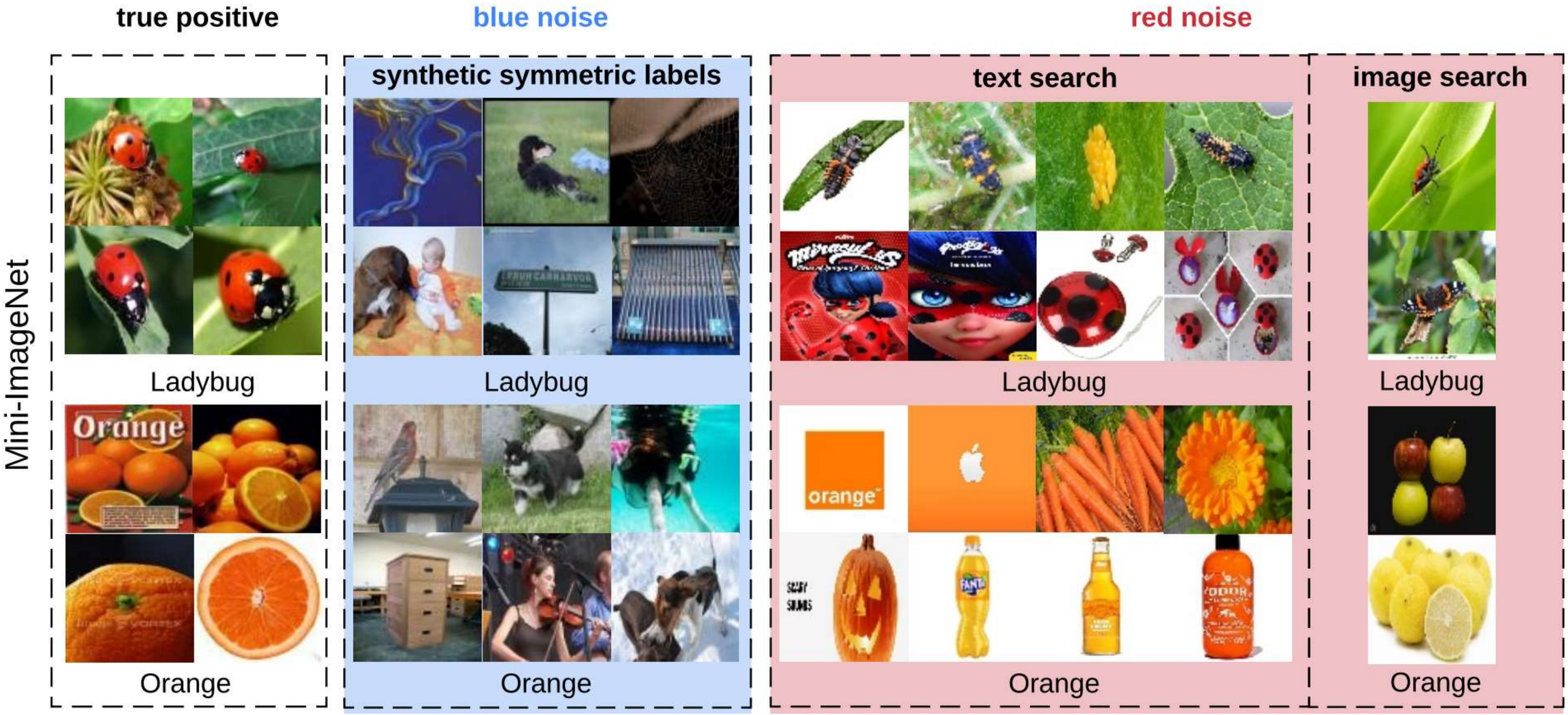
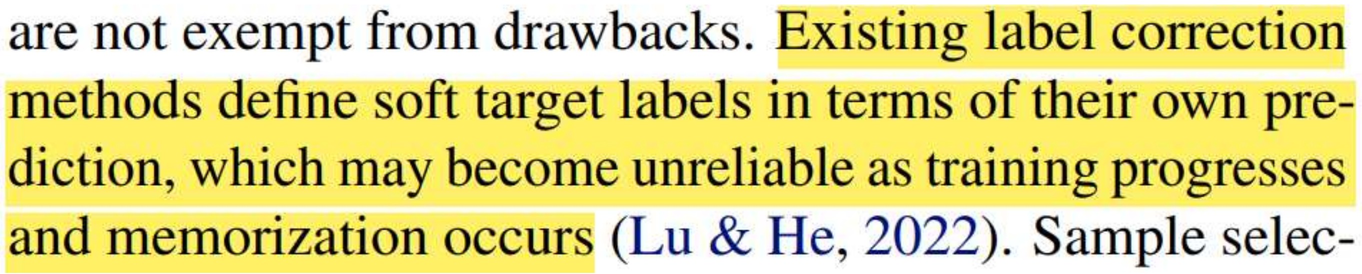


Figure 1. Comparison of symmetric label noise (Blue noise) and web label noise (Red noise). From left to right, columns are true positive images, images with incorrect symmetric labels, and images with incorrect web labels from text-to-image search and image-to-image search, respectively. The image-to-image search results (the last column) only account for 18% in our dataset and fewer images are shown as a result.

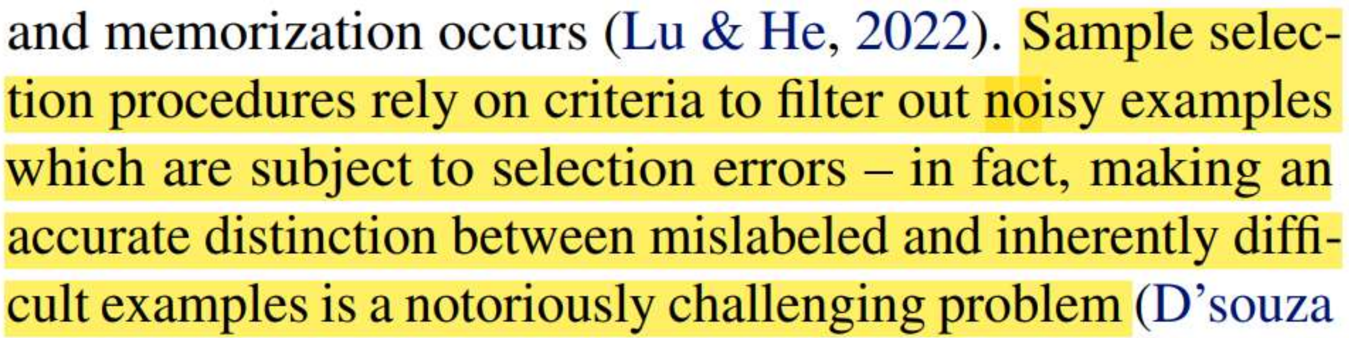
Two approaches: 1. label correction (noisy->clean)

2. sample selection (select clean)

Drawbacks:



are not exempt from drawbacks. Existing label correction methods define soft target labels in terms of their own prediction, which may become unreliable as training progresses and memorization occurs (Lu & He, 2022). Sample selec-



and memorization occurs (Lu & He, 2022). Sample selection procedures rely on criteria to filter out noisy examples which are subject to selection errors – in fact, making an accurate distinction between mislabeled and inherently difficult examples is a notoriously challenging problem (D'souza

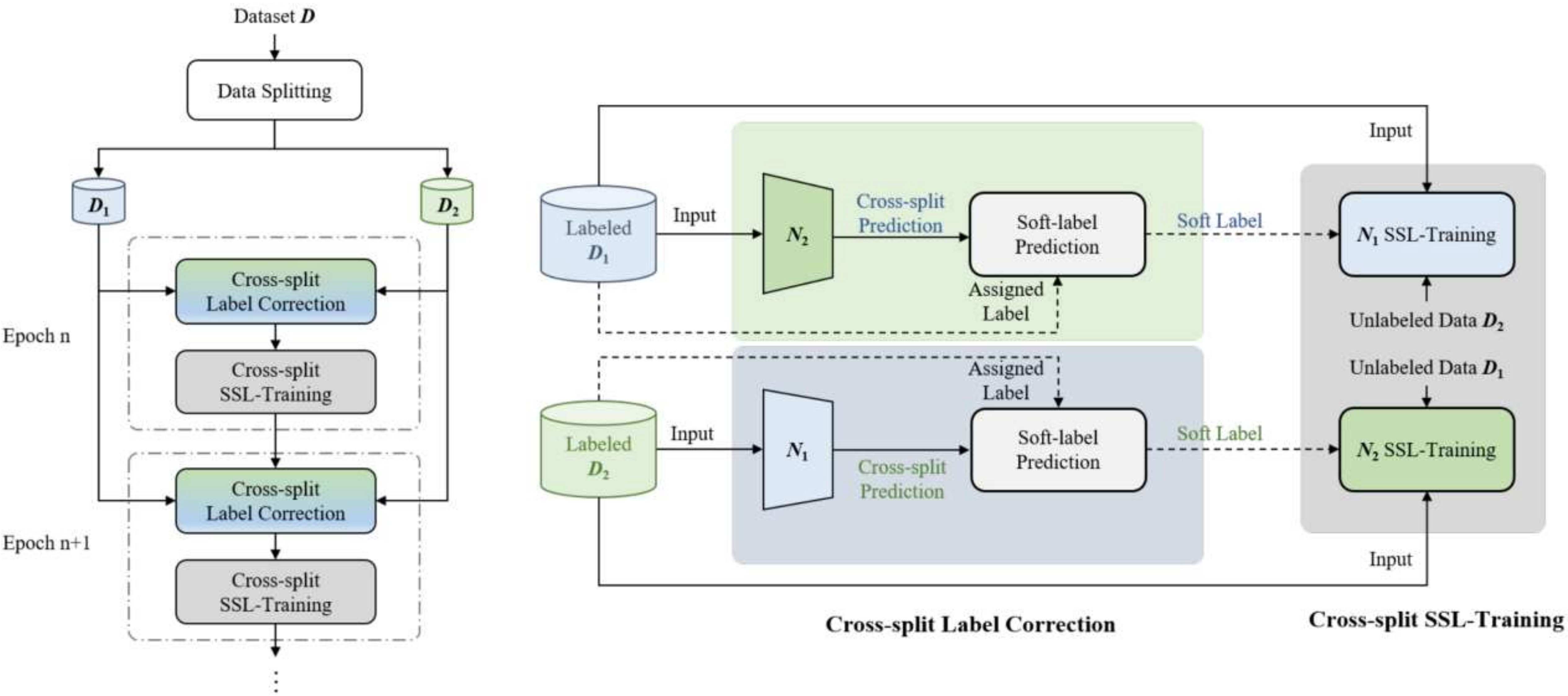
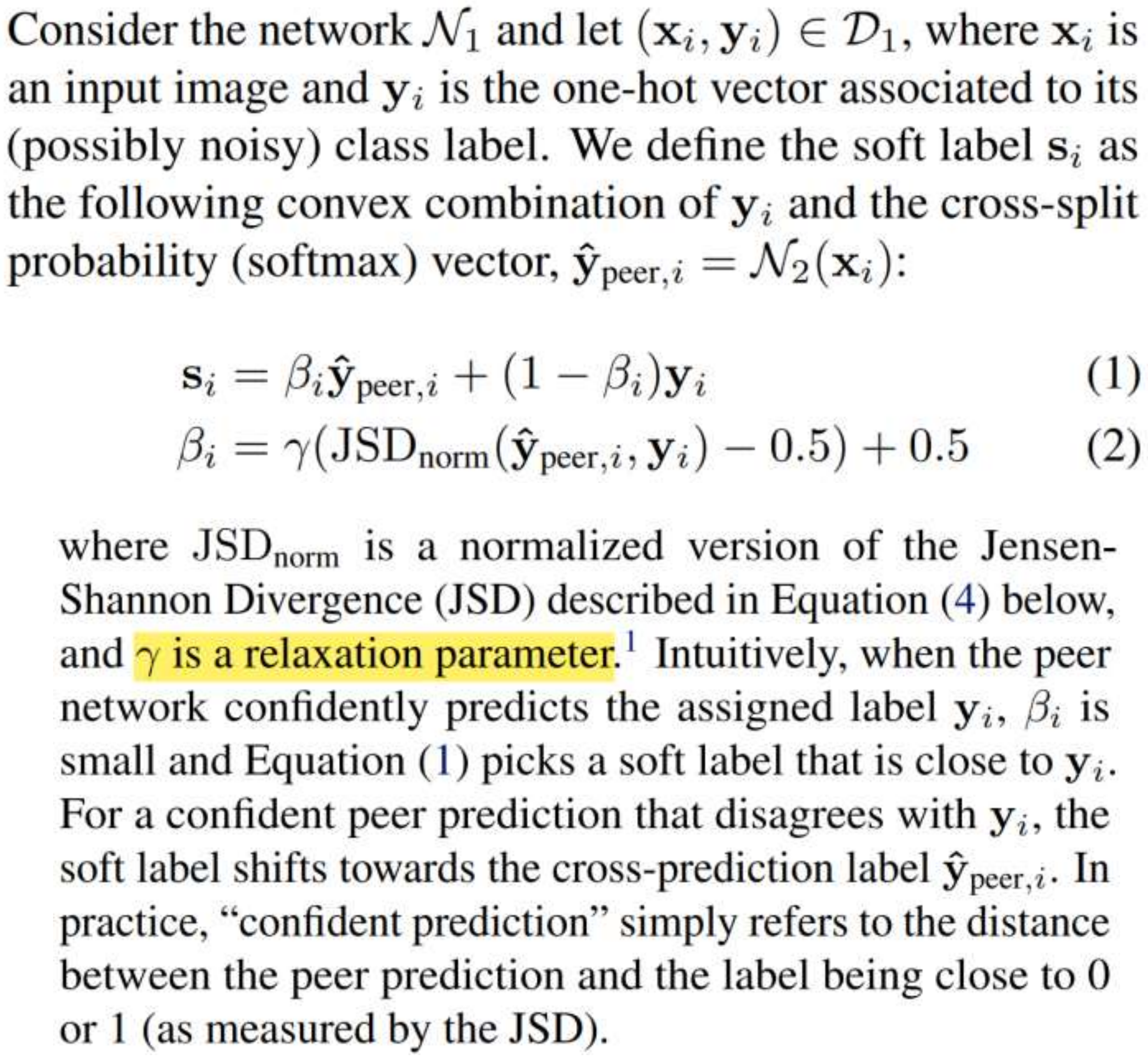


Figure 1. CrossSplit splits the original training labelled dataset into two disjoint parts and trains a separate network on each of these splits The dataset each network is trained on is also used by the peer network as unlabeled data for semi-supervised learning (SSL). At each training epoch, *CrossSplit* uses a cross-split label correction scheme that defines soft labels in terms of the peer prediction.

1. Cross-split Label Correction

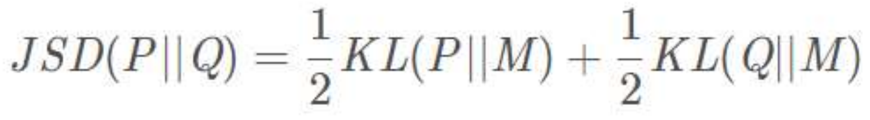


Consider the network $\mathcal{N}_1$ and let $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_1$, where $\mathbf{x}_i$ is an input image and $\mathbf{y}_i$ is the one-hot vector associated to its (possibly noisy) class label. We define the soft label $\mathbf{s}_i$ as the following convex combination of $\mathbf{y}_i$ and the cross-split probability (softmax) vector, $\hat{\mathbf{y}}_{\text{peer},i} = \mathcal{N}_2(\mathbf{x}_i)$:

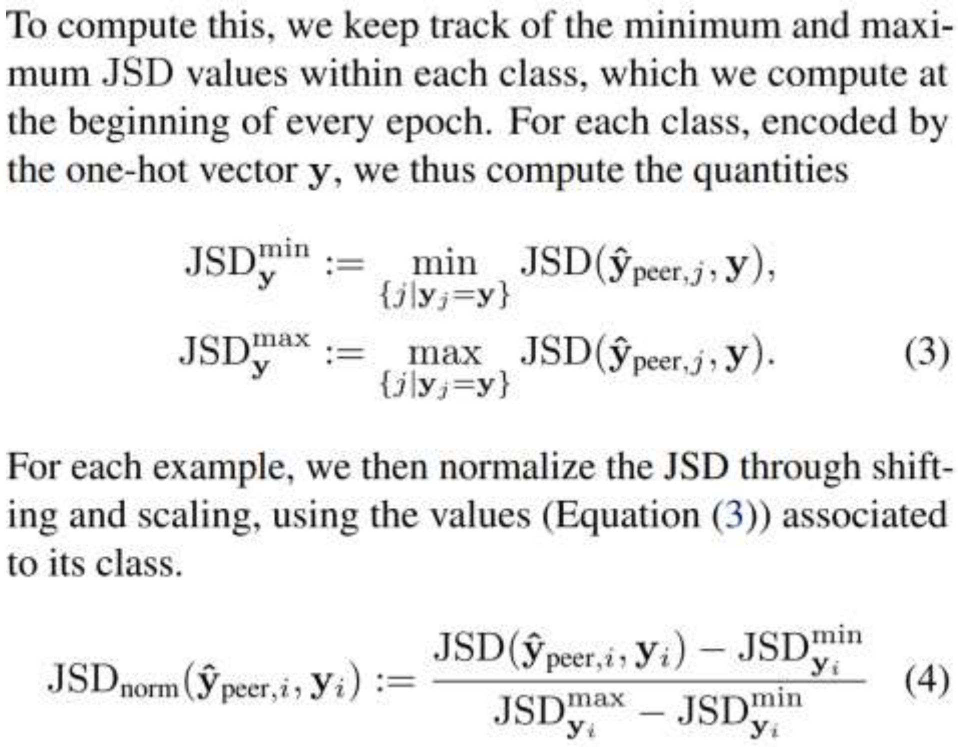
$$\mathbf{s}_i = \beta_i \hat{\mathbf{y}}_{\text{peer},i} + (1 - \beta_i)\mathbf{y}_i \tag{1}$$

$$\beta_i = \gamma(\text{JSD}_{\text{norm}}(\hat{\mathbf{y}}_{\text{peer},i}, \mathbf{y}_i) - 0.5) + 0.5 \tag{2}$$

where JSD_{norm} is a normalized version of the Jensen-Shannon Divergence (JSD) described in Equation (4) below, and γ is a relaxation parameter.[1] Intuitively, when the peer network confidently predicts the assigned label $\mathbf{y}_i$, β_i is small and Equation (1) picks a soft label that is close to $\mathbf{y}_i$. For a confident peer prediction that disagrees with $\mathbf{y}_i$, the soft label shifts towards the cross-prediction label $\hat{\mathbf{y}}_{\text{peer},i}$. In practice, "confident prediction" simply refers to the distance between the peer prediction and the label being close to 0 or 1 (as measured by the JSD).



$$JSD(P||Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M)$$



To compute this, we keep track of the minimum and maximum JSD values within each class, which we compute at the beginning of every epoch. For each class, encoded by the one-hot vector $\mathbf{y}$, we thus compute the quantities

$$\begin{aligned} \text{JSD}_{\mathbf{y}}^{\min} &:= \min_{\{j|\mathbf{y}_j=\mathbf{y}\}} \text{JSD}(\hat{\mathbf{y}}_{\text{peer},j}, \mathbf{y}), \\ \text{JSD}_{\mathbf{y}}^{\max} &:= \max_{\{j|\mathbf{y}_j=\mathbf{y}\}} \text{JSD}(\hat{\mathbf{y}}_{\text{peer},j}, \mathbf{y}). \end{aligned} \tag{3}$$

For each example, we then normalize the JSD through shifting and scaling, using the values (Equation (3)) associated to its class.

$$\text{JSD}_{\text{norm}}(\hat{\mathbf{y}}_{\text{peer},i}, \mathbf{y}_i) := \frac{\text{JSD}(\hat{\mathbf{y}}_{\text{peer},i}, \mathbf{y}_i) - \text{JSD}_{\mathbf{y}_i}^{\min}}{\text{JSD}_{\mathbf{y}_i}^{\max} - \text{JSD}_{\mathbf{y}_i}^{\min}} \tag{4}$$

A. Implementation Details

A.1. Detail on Relaxation Parameter

As mentioned in Equation (2) of the main paper, we use a relaxation parameter γ as a way to control the range of the combination coefficients in our definition of the soft labels. In our experiments, γ gradually increases from 0.6 to 1.0 during training according to the following schedule:

$$\gamma = \begin{cases} 0.6, & \text{if } epoch \in [E_{\text{warm}}, E_{\text{warm}} + 2\delta] \\ 0.8, & \text{else if } epoch \in [E_{\text{warm}} + 2\delta, E_{\text{warm}} + 3\delta] \\ 1.0, & \text{otherwise} \end{cases} \tag{A.1}$$

where the parameter δ determines the relaxation period. We empirically set δ to 10.

It is interesting to show how the value of γ affects the results. We investigate two additional settings: (i) constant small (= 0.6) γ, (ii) constant large (= 1.0) γ. The results are shown in Table A.1.

Table A.1. Effect of γ setting. The best scores are **boldfaced**, and the second best ones are underlined.

Noise type	Symmetric			
Dataset	CIFAR-10		CIFAR-100	
Noise ratio	50%	90%	50%	90%
0.6 (constant)	**96.38**	89.65	**79.16**	45.77
1.0 (constant)	96.29	**93.56**	75.24	50.71
$0.6 \rightarrow 0.8 \rightarrow 1.0$ (ours)	96.34	91.25	75.72	**52.40**

We find different outcomes depending on the noise ratio. For a 50% noise ratio, setting (i) performs better than (ii); for a 90% noise ratio this is the other way around. The setting of our paper yields a good performance across various noise ratios.

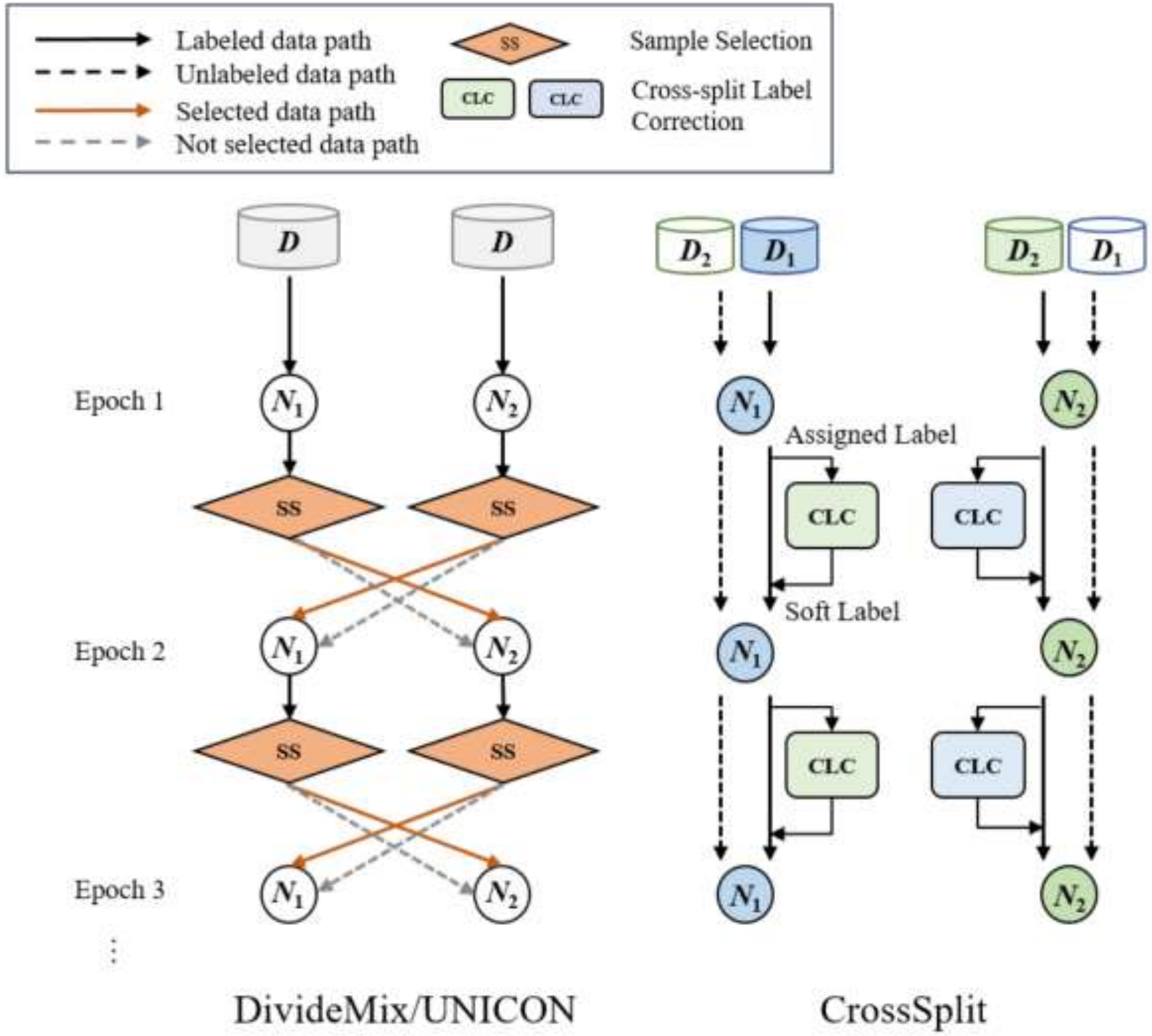
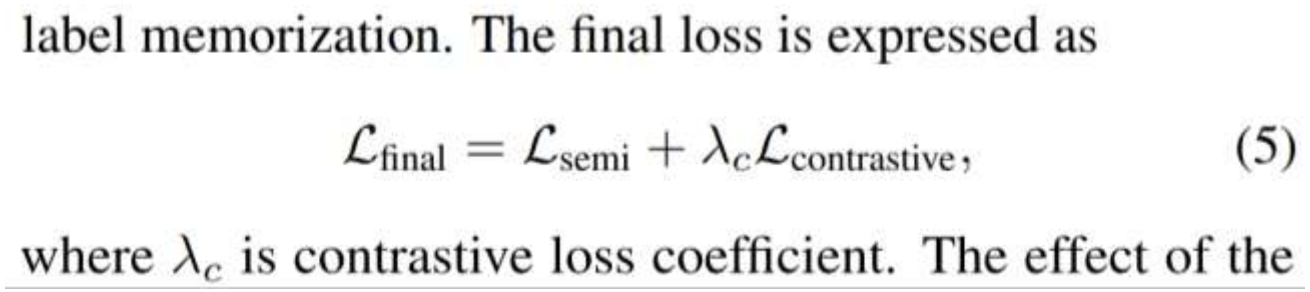


Figure 3. Comparison of the DivideMix (Li et al., 2020), UNICON (Karim et al., 2022), and *CrossSplit* co-teaching pipelines. The data flow is represented with solid lines for labeled data and dotted lines for unlabeled data. All three methods train two networks ($\mathcal{N}_1$ & $\mathcal{N}_2$) simultaneously. In DivideMix and UNICON, at every epoch, each network separates clean samples (orange solid line) and noisy samples (gray dotted line) using a small loss criterion, and transfers the two subsets to its peer network for subsequent semi-supervised learning. By contrast, *CrossSplit* splits the original training dataset into two halves and trains each network on one of these splits. For each of the two networks, we use soft labels defined as convex combinations of the assigned label and the peer network prediction via cross-split label correction (CLC) process. The data each network is trained on is also used by the peer network as unlabeled data for semi-supervised learning.

2. Cross-split SSL-Training



label memorization. The final loss is expressed as

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{semi}} + \lambda_c \mathcal{L}_{\text{contrastive}}, \tag{5}$$

where λ_c is contrastive loss coefficient. The effect of the

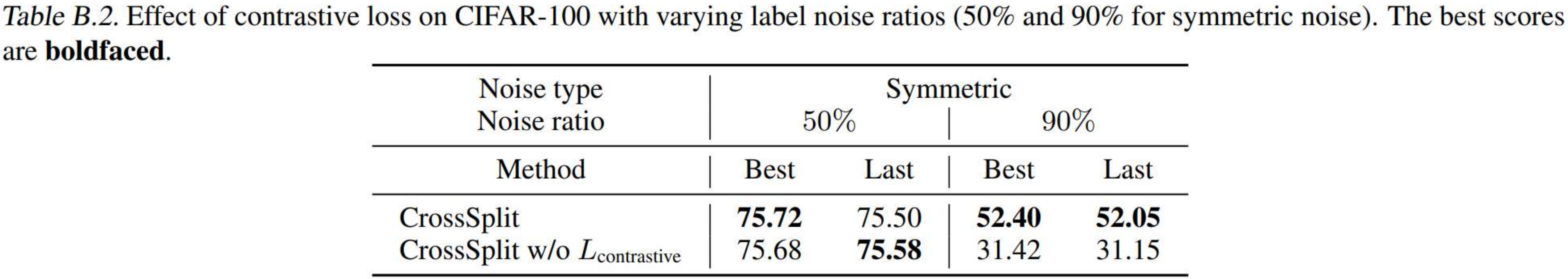
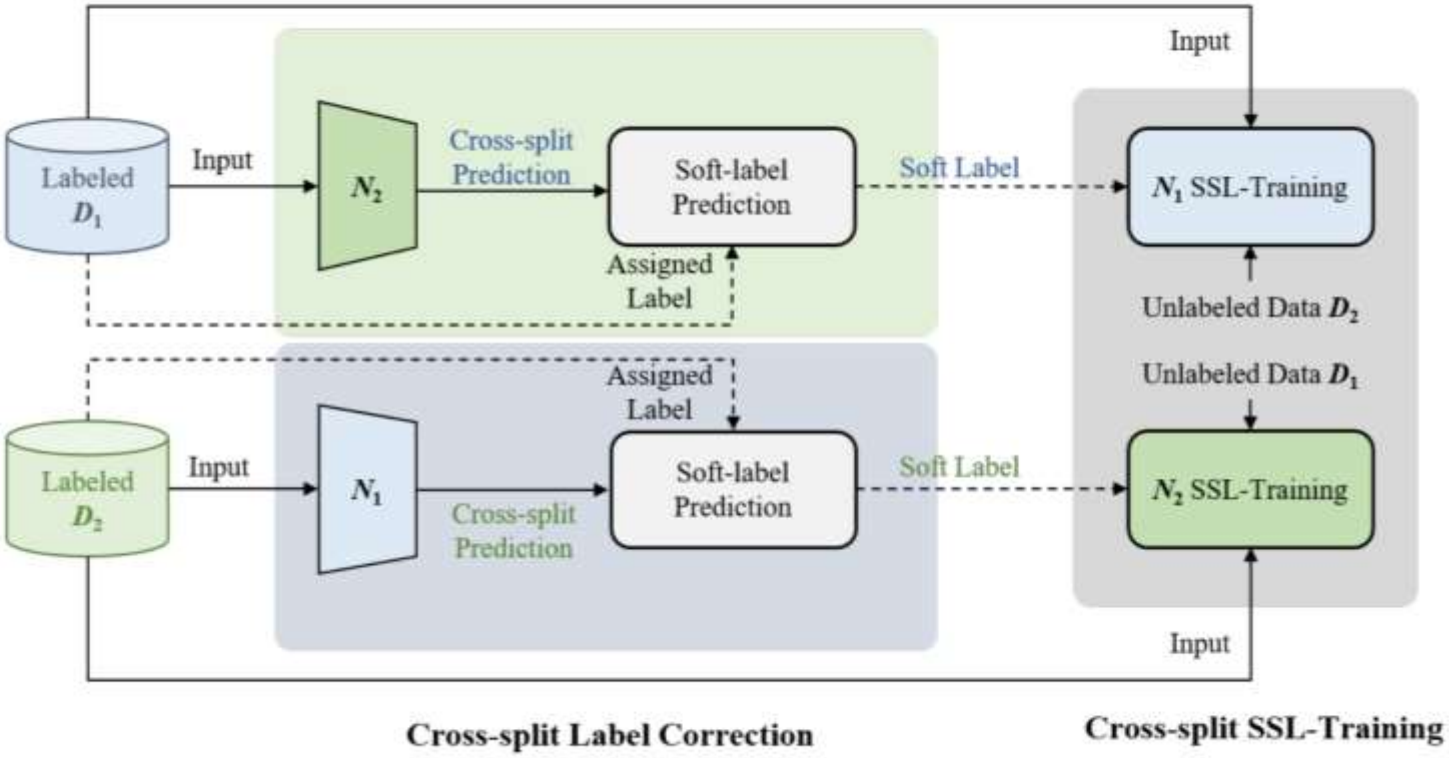


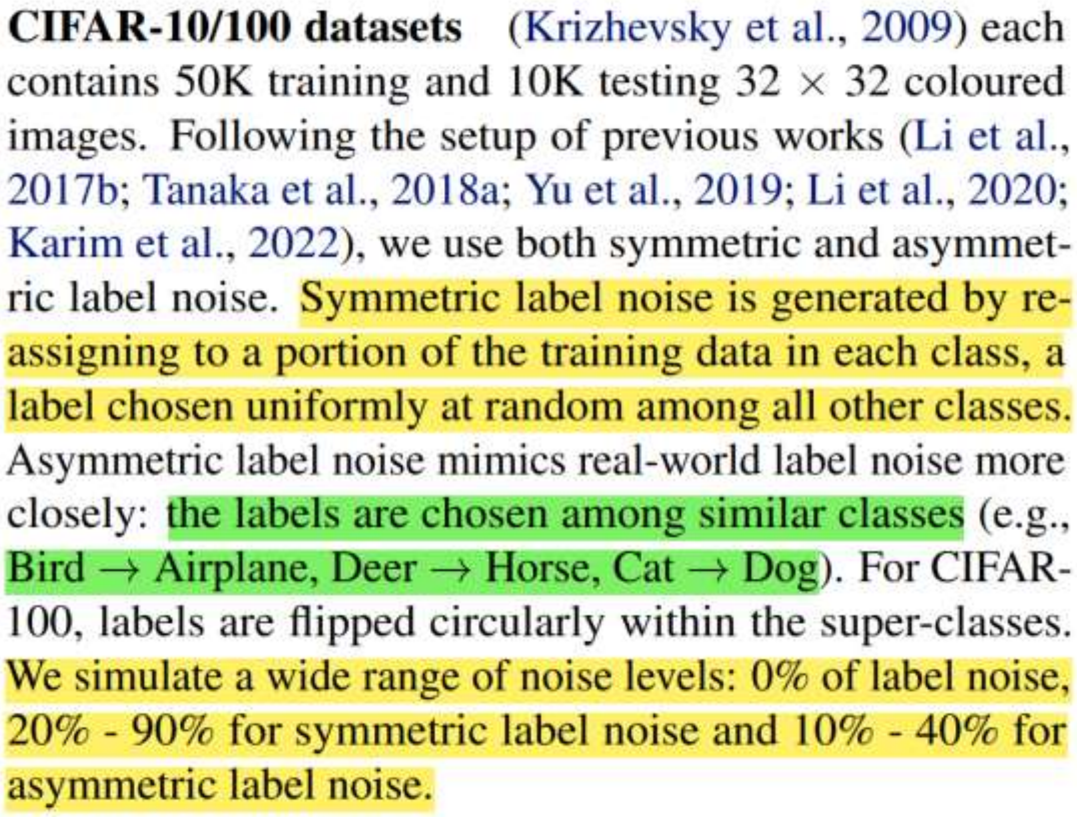
Table B.2. Effect of contrastive loss on CIFAR-100 with varying label noise ratios (50% and 90% for symmetric noise). The best scores are **boldfaced**.

Noise type	Symmetric			
Noise ratio	50%		90%	
Method	Best	Last	Best	Last
CrossSplit	**75.72**	75.50	**52.40**	**52.05**
CrossSplit w/o $L_{\text{contrastive}}$	75.68	**75.58**	31.42	31.15

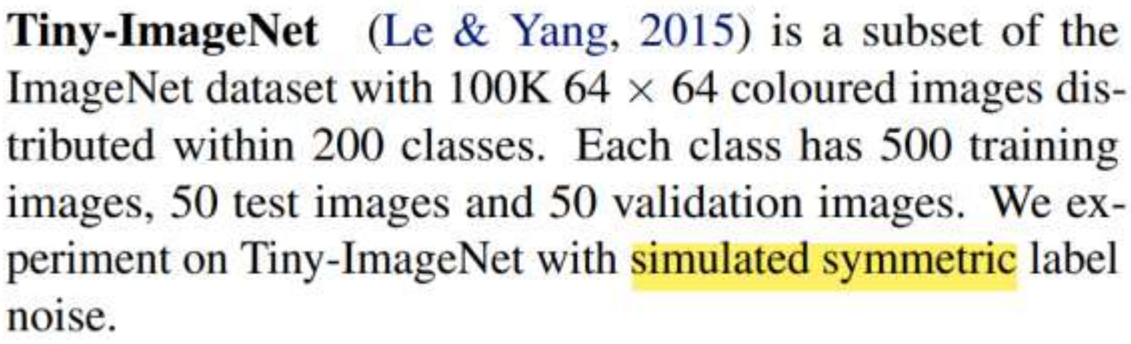
Algorithm 1 CrossSplit: Cross-split SSL training based on cross-split label correction

```
Input: Split training set $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2\}$, pair of networks $\mathcal{N}_1, \mathcal{N}_2$, warmup epoch $E_{\text{warm}}$, total number of epochs $E_{\max}$.
$\theta_1, \theta_2 \leftarrow$ Initialize network parameters
$\theta_1, \theta_2 \leftarrow$ Warmup supervised training on whole dataset for $E_{\text{warm}}$ epochs
for $epoch \in [E_{warm} + 1, \ldots, E_{max}]$ do
    1. Training $\mathcal{N}_1$:
    1.1: Perform cross-split label correction Equation (1) for labeled $\mathcal{D}_1$ using the predictions of $\mathcal{N}_2$ (see Section 2.1).
    1.2: Perform SSL training (Sohn et al., 2020; Zhang et al., 2018) using (soft)-labeled $\mathcal{D}_1$ as labeled data and $\mathcal{D}_2$ as unlabeled data (see Section 2.2).
    2. Analogous training for $\mathcal{N}_2$.
end
Return: $\theta_1, \theta_2$.
```

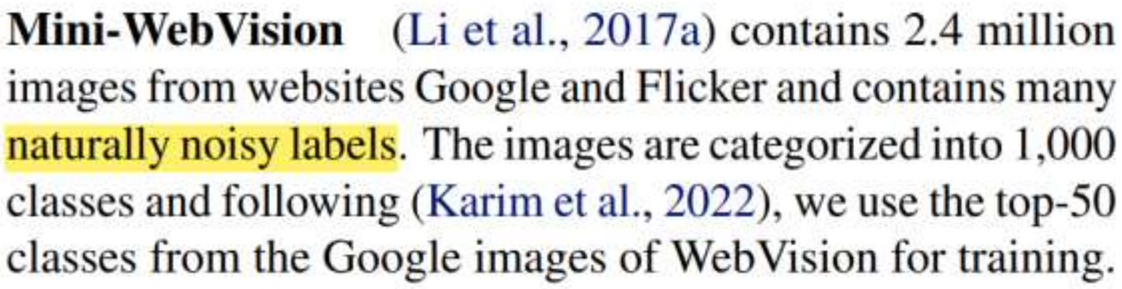




CIFAR-10/100 datasets (Krizhevsky et al., 2009) each contains 50K training and 10K testing 32 × 32 coloured images. Following the setup of previous works (Li et al., 2017b; Tanaka et al., 2018a; Yu et al., 2019; Li et al., 2020; Karim et al., 2022), we use both symmetric and asymmetric label noise. Symmetric label noise is generated by reassigning to a portion of the training data in each class, a label chosen uniformly at random among all other classes. Asymmetric label noise mimics real-world label noise more closely: the labels are chosen among similar classes (e.g., Bird → Airplane, Deer → Horse, Cat → Dog). For CIFAR-100, labels are flipped circularly within the super-classes. We simulate a wide range of noise levels: 0% of label noise, 20% - 90% for symmetric label noise and 10% - 40% for asymmetric label noise.



Tiny-ImageNet (Le & Yang, 2015) is a subset of the ImageNet dataset with 100K 64 × 64 coloured images distributed within 200 classes. Each class has 500 training images, 50 test images and 50 validation images. We experiment on Tiny-ImageNet with simulated symmetric label noise.



Mini-WebVision (Li et al., 2017a) contains 2.4 million images from websites Google and Flicker and contains many naturally noisy labels. The images are categorized into 1,000 classes and following (Karim et al., 2022), we use the top-50 classes from the Google images of WebVision for training.

A.2. Training Details

Table A.2. Training details on CIFAR-10, CIFAR-100, Tiny-ImageNet and mini-WebVision datasets.

Dataset	CIFAR-10	CIFAR-100	Tiny-ImageNet	mini-WebVision
Batch size	256	256	40	128
Network	PRN-18	PRN-18	PRN-18	ResNet-18
Epochs	300	300	360	140
Optimizer	SGD	SGD	SGD	SGD
Momentum	0.9	0.9	0.9	0.9
Weight decay	5e-4	5e-4	5e-4	5e-4
Initial LR	0.1	0.1	0.005	0.02
LR scheduler	Cosine Annealing LR			Multi-Step LR
$T_{\max}$/LR decay factor	300	300	360	0.1 (80, 105)
Warm-up period	10	30	10	1

The training details are summarized in Table A.2. For CIFAR-10 and CIFAR-100, we train each network using stochastic gradient descent (SGD) optimizer with momentum 0.9 and a weight decay of 0.0005. Training is done for 300 epochs with a batch size of 256. We set the initial learning rate as 0.1 and use a a cosine annealing decay (Loshchilov & Hutter, 2017). Just like in (Li et al., 2020; Karim et al., 2022), a warm-up training on the entire dataset is performed for 10 and 30 epochs for CIFAR-10 and CIFAR-100, respectively.

Table 1. Test accuracy (%) comparison on CIFAR-10 (left) and CIFAR-100 (right) without label noise and with symmetric and asymmetric label noise. Our model achieves state-of-the-art performance on almost every dataset-noise combination. The best scores are **boldfaced**, and the second best ones are underlined. The baseline results are imported from (Karim et al., 2022; Li et al., 2020; 2022). For CrossSplit, mean and standard deviation of best accuracy are calculated over 3 repetitions of the experiments. The results are sorted according to their performance in the case of a 20% symmetric noise ratio.

	CIFAR-10								CIFAR-100							
Noise type	-	Symmetric				Asymmetric			-	Symmetric				Asymmetric		
Method/Noise ratio	0%	20%	50%	80%	90%	10%	30%	40%	0%	20%	50%	80%	90%	10%	30%	40%
CE	95.4	86.8	79.4	62.9	42.7	88.8	81.7	76.1	77.3	62.0	46.7	19.9	10.1	68.1	53.3	44.5
Bootstrapping (Reed et al., 2015)	-	86.8	79.8	63.3	42.9	-	-	-	-	62.1	46.6	19.9	10.2	-	-	-
JPL (Kim et al., 2021)	-	93.5	90.2	35.7	23.4	94.2	92.5	90.7	-	70.9	67.7	17.8	12.8	72.0	68.1	59.5
M-Correction (Arazo et al., 2019)	-	94.0	92.0	86.8	69.1	89.6	92.2	91.2	-	73.9	66.1	48.2	24.3	67.1	58.6	47.4
MOIT (Ortego et al., 2021)	-	94.1	91.1	75.8	70.1	94.2	94.1	93.2	-	75.9	70.1	51.4	24.5	77.4	75.1	74.0
SELC (Lu & He, 2022)	-	95.0	-	78.6	-	-	-	92.9	-	76.4	-	37.2	-	-	-	73.6
Sel-CL (Li et al., 2022)	-	95.5	93.9	89.2	81.9	95.6	95.2	93.4	-	76.5	72.4	59.6	48.8	78.7	76.4	74.2
MixUp (Zhang et al., 2018)	95.8	95.6	87.1	71.6	52.2	93.3	83.3	77.7	78.9	67.8	57.3	30.8	14.6	72.4	57.6	48.1
ELR (Liu et al., 2020)	-	95.8	94.8	93.3	78.7	95.4	94.7	93.0	-	77.6	73.6	60.8	33.4	77.3	74.6	73.2
UNICON (Karim et al., 2022)	-	96.0	95.6	93.9	90.8	95.3	94.8	94.1	-	78.9	**77.6**	63.9	44.8	78.2	75.6	74.8
DivideMix (Li et al., 2020)	-	96.1	94.6	93.2	76.0	93.8	92.5	91.7	-	77.3	74.6	60.2	31.5	71.6	69.5	55.1
CrossSplit (PRN-18)	**97.0** ±0.16	**96.9** ±0.05	**96.3** ±0.05	**95.4** ±0.64	**91.3** ±0.79	**96.9** ±0.04	**96.4** ±0.16	**96.0** ±0.12	**81.7** ±0.25	**79.9** ±0.19	75.7 ±0.18	**64.6** ±1.43	**52.4** ±1.78	**80.7** ±0.05	**78.5** ±0.19	**76.8** ±0.66
CrossSplit (PRN-34)	**97.3** ±0.16	**97.1** ±0.16	**96.5** ±0.24	**95.2** ±0.59	85.3 ±3.61	**97.2** ±0.09	**96.6** ±0.11	**96.1** ±0.08	**83.0** ±0.15	**81.4** ±0.38	77.2 ±0.25	**67.0** ±0.49	**52.6** ±3.43	**82.6** ±0.15	**80.5** ±0.27	**79.1** ±0.40

Tables 2 & 3. Test accuracy (%) comparison on Tiny-ImageNet (left) and mini-WebVision (right). Our model is competitive with the state-of-the-art (only small differences in performance) on Tiny-ImageNet with artificial noise, and surpasses the state-of-the-art on mini-Webvision with real-world noise. The best scores are **boldfaced**, and the second best ones are underlined. In Table 2, Best and Avg. mean highest and average accuracy over the last 10 epochs. The baseline results are imported from (Karim et al., 2022) and sorted according to their best performance in the case of a 20% noise ratio. In Table 3, the baseline results are sorted by best performance.

Table 2. Tiny-ImageNet

Noise type	Symmetric			
Noise ratio	20%		50%	
Method	Best	Avg.	Best	Avg.
CE	35.8	35.6	19.8	19.6
Decoupling (Malach & Shalev-Shwartz, 2017)	37.0	36.3	22.8	22.6
MentorNet (Jiang et al., 2018)	45.7	45.5	35.8	35.5
Co-teaching+ (Yu et al., 2019)	48.2	47.7	41.8	41.2
M-Correction (Arazo et al., 2019)	57.2	56.6	51.6	51.3
NCT (Sarfraz et al., 2021)	58.0	57.2	47.8	47.4
UNICON (Karim et al., 2022)	**59.2**	58.4	**52.7**	**52.4**
CrossSplit (ours)	59.1	**58.8**	52.4	52.0

Table 3. Mini-WebVision

Method	Best	Last
Decoupling (Malach & Shalev-Shwartz, 2017)	62.54	-
MentorNet (Jiang et al., 2018)	63.00	-
Co-teaching (Han et al., 2018)	63.58	-
Iterative-CV (Chen et al., 2019)	65.24	-
ELR (Liu et al., 2020)	73.00	71.88
SELC (Lu & He, 2022)	74.38	-
MixUp (Zhang et al., 2018)	74.96	73.76
DivideMix (Li et al., 2020)	76.08	74.64
UNICON(Karim et al., 2022)	77.60	-
CrossSplit (ours)	**78.48**	**78.07**

Table 4. Ablation study on CIFAR-10: Test accuracy (%) of different setting on CIFAR-10 with varying noise rates (50% - 90% for Symmetric and 10% - 40% for Asymmetric noise). We see that there is a minor difference when removing class-balancing normalization with lower noise ratios, but a large degradation in performance if it is removed for high noise ratios. Mean and standard deviation of best and average of last 10 epochs are calculated over 3 repetitions of the experiments. The best results are highlighted in **boldfaced** and scores that differ from them by more than 5% are marked in red.

Noise type	Symmetric				Asymmetric			
Noise ratio	50%		90%		10%		40%	
Method	Best	Last	Best	Last	Best	Last	Best	Last
CrossSplit	96.34±0.05	96.23±0.07	**91.25**±0.79	**91.02**±0.77	96.85±0.04	96.74±0.07	96.01±0.12	95.88±0.13
w/o data splitting	96.10±0.04	95.96±0.00	90.30±0.13	89.93±0.24	96.76±0.05	96.63±0.06	92.16±0.09	86.24±0.37
w/o class-balancing normalization	**96.73**±0.13	**96.61**±0.07	75.54±2.82	74.88±2.50	**97.33**±0.02	**97.20**±0.02	**96.22**±0.07	**96.04**±0.12
w/o cross-split label correction	96.12±0.05	95.99±0.03	90.83±0.25	90.08±0.40	**97.33**±0.08	97.15±0.09	96.12±0.14	95.95±0.10

Table 5. Ablation study on CIFAR-100: Test accuracy (%) of different settings on CIFAR-100 with varying noise rates (50% - 90% for Symmetric and 10% - 40% for Asymmetric noise). With its higher difficulty than CIFAR-10, each component of *CrossSplit* is crucial when the noise ratios are high. Mean and standard deviation of best and average of last 10 epochs are calculated over 3 repetitions of the experiments. The best results are highlighted in **boldfaced** and scores that differ from them by more than 5% are marked in red.

Noise type	Symmetric				Asymmetric			
Noise ratio	50%		90%		10%		40%	
Method	Best	Last	Best	Last	Best	Last	Best	Last
CrossSplit	75.72±0.18	75.50±0.18	**52.40**±1.78	**52.05**±1.94	80.71±0.05	80.50±0.06	**76.78**±0.66	**76.56**±0.55
w/o data splitting	73.63±0.18	73.36±0.14	14.19±1.30	13.28±2.21	78.97±0.07	78.77±0.43	72.12±0.43	71.83±0.42
w/o class-balancing normalization	**77.67**±0.03	**77.17**±0.17	33.37±0.52	18.53±0.19	**82.86**±0.14	**82.57**±0.18	71.59±0.28	60.35±0.37
w/o cross-split label correction	70.20±0.16	65.74±0.10	31.77±0.32	15.93±0.21	82.38±0.16	82.10±0.23	69.61±0.65	59.67±0.11

B.1. Effects of Training Set Size on Performance

There might be a tradeoff between a potential performance loss due to halving of the dataset for the two individual networks (although it is mitigated by our use of the semi-supervised setup), and the performance loss due to noise memorization. Our empirical results show that this tradeoff is worthwhile for learning with noisy labels (see Table 4 and Table 5). In Section 4.6, our ablation study shows that when we do not perform data splitting (i.e. use the whole dataset), the results are consistently worse.

In addition, we investigate the effect of training set size on performance further. One way to do this is to extend the method to multiple networks trained on multiple splits. Depending on the number of data splits, the ratios of labeled and unlabeled datasets (S_{labeled}, $S_{\text{unlabeled}}$) can vary. We set the number of data splits from two (default) to four. Figure B.1 shows five possible scenarios, (a–e). When we train two (a), three (b–c) or four (d–e) models, each model is trained via the semi-supervised training process with the data configuration depicted on the left side of the model. Then, each model performs cross-split label correction using the peer network prediction (the remaining models' predictions) for the labeled data.

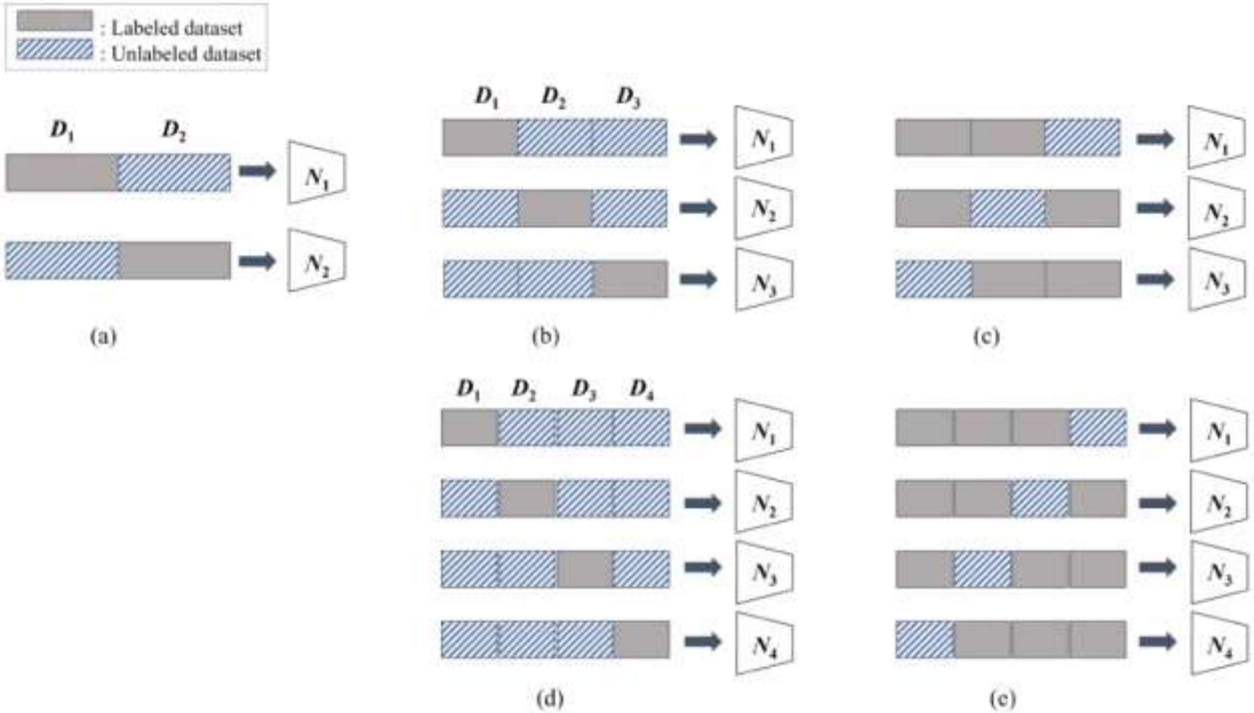


Figure B.1. Visualizations of five possible data configurations: (a) (N_{splits}, S_{labeled}, $S_{\text{unlabeled}}$) = (2, 0.50, 0.50), (b) (3, 0.33, 0.67), (c) (3, 0.67, 0.33), (d) (4, 0.25, 0.75), and (e) (4, 0.75, 0.25)

Table B.1. Effect of the number of splits and data configurations: Test accuracy (%) of different setting on CIFAR-10 with varying label noise ratios (50% and 90% for symmetric noise). We see that there is a minor difference when changing the number of splits and data configurations with lower noise ratios, but a large degradation in performance if it is changed for high noise ratios. Among five scenarios, the setting of our paper (a) yields a good performance across various noise ratios. Mean and standard deviation of best and average of last 10 epochs are calculated over 3 repetitions of the experiments. The best results are highlighted in **boldfaced**.

Noise type	Symmetric							
Dataset	CIFAR-10				CIFAR-100			
Noise ratio	50%		90%		50%		90%	
($\mathrm{N}_{\mathrm{splits}}$, S_{labeled}, $S_{\mathrm{unlabeled}}$)	Best	Last	Best	Last	Best	Last	Best	Last
(a) (2, 0.50, 0.50)	**96.34**±0.05	**96.23**±0.07	**91.25**±0.79	**91.02**±0.77	**75.72**±0.18	**75.50**±0.18	52.40±1.78	52.05±1.94
(b) (3, 0.33, 0.67)	96.10±0.15	95.95±0.09	88.73±0.72	88.53±0.75	75.46±0.06	75.27±0.02	50.73±0.65	50.15±0.84
(c) (3, 0.67, 0.33)	96.02±0.06	95.95±0.06	88.74±0.05	88.57±0.07	74.97±0.10	74.79±0.10	52.46±1.98	52.27±1.92
(d) (4, 0.25, 0.75)	**95.59**±0.19	**95.50**±0.16	88.13±0.25	87.89±0.36	73.83±0.30	73.66±0.28	46.04±0.57	44.76±0.77
(e) (4, 0.75, 0.25)	96.08±0.05	95.97±0.02	88.74±1.00	88.57±0.96	75.71±0.14	75.45±0.25	**53.79**±0.94	**53.61**±0.97

Table B.1 demonstrates that there is a considerable difference in performance due to the ratio of labeled information especially in the case of 90% noise on CIFAR-100. That is, the larger the ratio of labeled information, the better the performance ((d) $S_{\mathrm{labeled}} = 0.25 <$ (b) $0.33 <$ (a) $0.50 <$ (c) $0.67 <$ (e) 0.75). However, when the amount of noise is small, the performance difference due to the proportion of labeled datasets is not large and (a) ($S_{\mathrm{labeled}} = 0.50$) performs better than other cases in most cases. This results in an optimal number of splits that is 2 in favour of our original setting.

B.5. T-SNE Visualization

In this section, we provide a visual comparison of the features (penultimate layer) learned by UNICON (Karim et al., 2022) and *CrossSplit*. Both are trained with PreAct ResNet-18. Figure B.2 and Figure B.3 show the class distribution of the features corresponding to test images on CIFAR-10 and CIFAR-100 with asymmetric noise (40%) and symmetric noise (50%, 90%), respectively. This suggests that the representations learned by *CrossSplit* do a better job at separating the classes than UNICON.

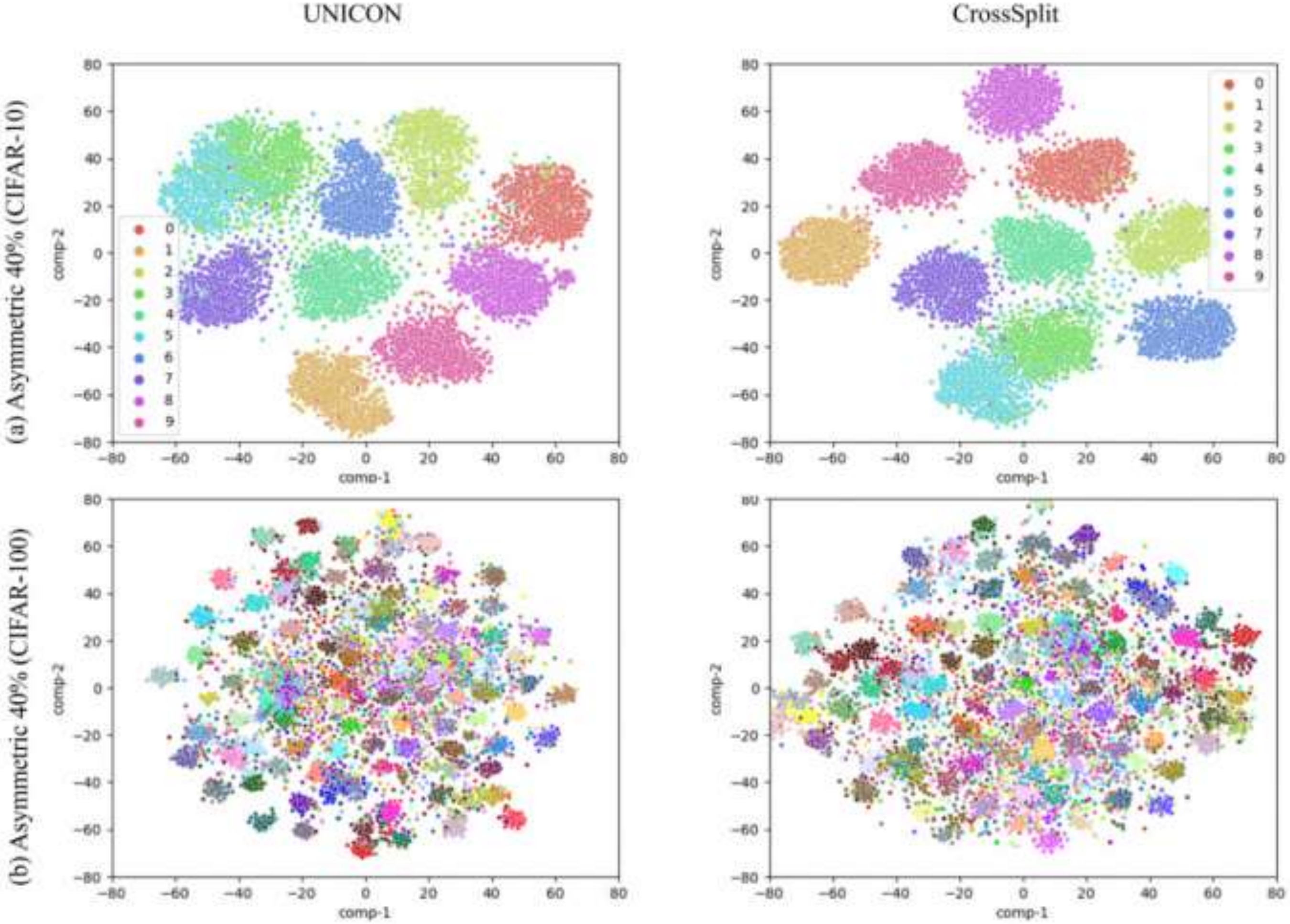


Figure B.2. T-SNE visualizations of learned features of test images by UNICON (Karim et al., 2022) and CrossSplit with asymmetric noise of 40%. In general, the clusters for CrossSplit are significantly better separated than for UNICON.

THANKS