# 702 A SUPPLEMENTAL MATERIAL

704

## A.1 DISCUSSION ABOUT DISAGREEMENT

705 706

We test a three-layer MLP on *CIFAR-10* with symmetric noise ratio  $\epsilon = 0.5$  and  $\epsilon = 0.8$ , and quantify the disagreement for three update strategies. *Intersection over Union* (IoU) between two selections are utilized for the quantitative evaluation, while the update strategies are the Cross-update, Self-update, and Jump-update. Specifically, Intersection computes the number of commonly selected samples in two selections, while Union calculates the total number of selected samples in two selections. In the cross-update strategy, the disagreement emerges in a different network, while in the other two strategies, it emerges during different epochs.

714 As shown in Fig. 2(a) and Fig. 2(b), the single network in Self-Update exhibits significantly greater 715 disagreement between epochs than the dual-network setup in Cross-update, which suggests a huge potential for mitigating bias. However, self-update does not leverage this advantage, resulting in the 716 worst performance among strategies. In contrast, jump-update strategy, which leverages previous 717 selections for model updates, displays slightly lower disagreement compared to self-update while 718 achieving remarkable performance. Furthermore, we observe that the error band of Jump-Update 719 is smaller than the other two methods, indicating more stable network updates. This reflects the 720 robustness of the network against noise to some extent. These results are shown in Fig. 2(c) and Fig. 721 2(d): Jump-Update, with the smallest error band, achieves the best performance, while self-update, 722 with the largest error band, performs the worst. 723

724 725

726

## A.2 DISCUSSIONS ABOUT "DISAGREEMENT"

727 In this section, we discuss the concept of "disagreement" and its relationship with prior related works 728 (Wei et al.) 2022 Yuan et al., 2023). The "disagreement" discussed in our paper differs fundamentally 729 from similar concepts in previous studies in terms of its definition, described objects, application, 730 calculation, and focus. Unlike the "fluctuation" defined in (Wei et al.) [2022], which refers to a sample 731 being correctly classified at one step but misclassified in the next, or the "First-time k-epoch Learning 732 (FkL)" introduced in (Yuan et al., 2023), which measures the first training epoch where an instance is 733 consistently predicted to its given label for k consecutive epochs, our "disagreement" is derived from 734 (Han et al., 2018) and refers to differences in selection behaviors within networks. These differences are not tied to individual samples or labels but rather describe variations in network behaviors over 735 iterations. 736

The objects described in these concepts are also distinct. While "fluctuation" and "FkL" capture variations in the predictions of individual samples, "disagreement" focuses on the characteristics of networks. For instance, "disagreement" in our work examines the behavior of a single network across different iterations, whereas in Han et al. (2018), "disagreement" describes an attribute of two networks.

742 The applications of these concepts further emphasize their differences. As a network-level character-743 istic, "disagreement" is employed to design strategies for updating models. In contrast, "fluctuation" 744 and "FkL," being sample-level characteristics, are primarily used to define selection criteria for 745 determining whether a sample should be included in the training process. Additionally, the methods 746 for calculating and interpreting these concepts vary significantly. Both "fluctuation" and "FkL" 747 emphasize consistency between model predictions and true labels, aiming to identify clean samples. On the other hand, "disagreement" is independent of labels and is quantified using the Intersection 748 Over Union (IoU) metric between selected sets, enabling it to address selection bias in model updates. 749

Our work contributes novel insights into the concept of "disagreement." We are the first to demonstrate that disagreement within a single network not only exists but also persists during training, as illustrated in Fig. 2(a) and Fig. 2(b). Additionally, we find that single-network disagreement is even more significant than that observed between two networks, a discovery that underscores its importance in understanding network behavior. To further solidify this understanding, we introduce the IoU metric to quantify and visualize disagreement, providing a new lens through which to analyze and mitigate selection bias in model updates.

## A.3 EXAMPLE OF JUMP-UPDATE STRATEGY

758 We would like to give a simple example to illustrate the Jump-update Strategy for quick understanding. 759 Suppose we have three iterations, iter0, iter1, and iter2. After the parameters of iter1 are updated 760 through backpropagation, the resulting updated model is referred to as iter2. When training in iter2, 761 the model only used the clean samples selected by iter0, which creates a 'disagreement' due to the 762 inconsistency between the iteration where sample selection occurs (iter0) and the iteration where 763 the model is updated (iter1). The behavior of the update exhibits a jump form. In this example, the 764 current iteration i equals 2, and jump-step S equals 2. Iter0 is the ancestor iteration of iter2.

765 766

## A.4 PROOF OF PROPERTY 1

The key of proving  $D_A \propto n$  is to prove  $N_A = N_{iteration}$ . Under the hypothesis that the error flow is an uninterrupted model, the situation can be categorized into two types: (1) errors accumulate in a single network; (2) errors accumulate across multiple networks. Through the analysis of the following two types, Property [] is proved.

**Errors Accumulate in A Single Network.** When errors accumulate in a single network, one accumulation is generated from one sample selection and one model update. As is obvious that the single network operates only one sample selection and a parameter updating in each iteration. Therefore,  $N_A = N_{iteration}$ .

**Errors Accumulate across Multiple Networks.** When errors accumulate across multiple networks, errors first pass through each network once before being transmitted back to the original network. We denote the number of networks by  $N_{net}$ . Error accumulation can only occur when the updates of the network are guided by its own selections. Thus, this process requires  $N_{net}$  sample selections and  $N_{net}$  parameter updates. Similarly, in each iteration, all networks collectively undergo  $N_{net}$  sample selections and  $N_{net}$  parameter updates.

781 782

## 783 A.5 DISCUSSION ON ERROR FLOW

784 First of all, let us review the formulation  $N_a = \frac{N_A}{N_f}$ , mentioned in Property 2. The following 785 discussion is helpful for the understanding of this formulation. We denote the number of samples as 786  $n_s$  and the number of mini-batches as  $n_b$ . If the network updates the identifier table after updating 787 the model when the data for selection is a single sample,  $N_f = n_s$ . When it comes to mini-batches, 788  $N_f = n_b$ . In all of these conditions,  $N_a = e$ . Thus the single network can be enhanced by integrating 789 the jump-update strategy. If the network updates the identifier table before updating parameters, 790  $N_f$  will be doubled so that  $N_a = 0.5e$ . The cross-Update strategy uses two networks to mitigate 791 bias, it changed  $N_f$  to 2. This shows that the cross-update strategy can be incorporated into our 792 framework and does not require the use of two networks as well as the maintenance of two training processes. Moreover, we provide two methods to reduce  $N_a$ : the most direct method is to control the 793 update frequency of the identifier table, or by adding new identifier tables to maintain the updates 794 per iteration, each doubling of the identifier table reduces  $N_a$  by half. This allows us to control  $N_a$ flexibly. Empirical evidence has shown that maintaining  $N_a$  in the hundreds is reasonable. Limiting 796  $N_a$  to a single-digit number can lead to the model updating too slowly and over-fitting noise. 797

798 799

## A.6 DISCCUSION ON INTIAL BIAS

800 Initial selection bias and number of accumulations, *i.e.*,  $N_a$ , will jointly affect the overall degree of 801 accumulated error  $D_A$ . Initial Bias is manifested in the form of initial identifier tables which can be 802 obtained in three ways: the first is randomly selection, the second involves updates during warm-up, 803 and the third approach involves labeling before training, which is the usual practice. For ease of 804 subsequent discussion, we call them Random initiation, Warm-up initiation, and Instant initiation. 805 To analyze the impact of initial bias on error accumulation, we observe the subsequent performance 806 of the network by controlling the initial bias, using a single network as the baseline. The update 807 frequency of identifier table *i.e.*, r is set at 10%, and three different initiations are employed. The first two tables are unbiased as they are not affected by selection, whereas Instant initiation is biased as it 808 acquires tables during the second epoch of training and is affected by an increasingly deepening bias. Experimental results are shown in Fig. 5. The identifier table from Random initiation is less accurate

than that obtained through Instant initiation, yet its performance is significantly better, demonstrating
 the effectiveness of avoiding bias. Warm-up initiation gets the best result, as it avoids performance
 degradation caused by the high error rate of the initial table.



Figure 5: Effect of initial bias on training.

#### A.7 LOSS CONVERGENCE

To show how the different values of scaler T affect the convergence rates, we present the results of four different temperature values in Fig. 6. When T=1, the Binary Cross-Entropy (BCE) loss fails to converge properly. The cases for T = 2, 3, and 4 illustrate the effects of temperature scaling. Specifically, temperature scaling slows down the convergence speed of the classification head, which can be obviously observed during the warm-up phase, while it accelerates the convergence of the detection head, which can be obviously observed during the training phase. Notably, as the temperature T increases, the convergence rate of the Cross-Entropy (CE) loss decreases.



Figure 6: Convergence of losses with different temperature scaling factors T under CIFAR-100 Sym.  $\epsilon = 0.8$ .

#### 

## A.8 ABLATION STUDY

We evaluate three components of Jump-teaching: data augmentation (AUG), jump-update strategy (JU), and sample selection (SS) with an ablation study. Results are shown in Table 6

**Effectiveness of Data Augmentation.** When only a limited number of samples are available, such as in the presence of symmetric noise ratio  $\epsilon = 0.8$ , data augmentation enables the network to gain knowledge of the intrinsic characteristics of the images themselves. Therefore, it can significantly enhance the learning capabilities of the networks for these samples.

**Effectiveness of Jump-update Strategy.** Jump-update strategy can effectively overcome error accumulation, allowing it to perform excellently with extreme noise. Without the jump-update strategy, the network is almost unable to learn on *CIFAR-10* with the symmetric noise ratio  $\epsilon = 0.8$ .

Effectiveness of Sample Selection. If noisy samples are not filtered out at all, the network can easily fit them, resulting in poor generalization performance. This has been verified among diverse noise settings.

#### Table 6: Ablation Study.

N	lodu	ules			CIE	AR-10		CIFAR-100						
AUC	3 J	U	SS	Sym. 20%	Sym. 50%	Sym. 80%	Asym. 40%	Sym. 20%	Sym. 50%	Sym. 80%	Asym. 40%			
×	v	$\checkmark$	$\checkmark$	94.12	89.59	74.94	85.34	74.69	65.87	32.66	67.30			
$\checkmark$	>	×	$\checkmark$	88.33	81.27	10.01	56.28	68.32	65.27	28.71	50.40			
$\checkmark$	v	$\checkmark$	Х	85.42	63.21	30.18	78.19	61.88	39.35	12.90	45.17			
×	>	×	$\checkmark$	92.21	87.24	9.99	57.56	74.19	65.98	29.23	57.05			
$\checkmark$	v	$\checkmark$	$\checkmark$	94.77	92.11	82.81	90.91	72.15	66.91	40.81	67.66			

## A.9 THE SIZE OF HADAMARD CODEBOOK

As demonstrated in Table 7, we find that the performance of the network does not exhibit a clear correlation with the code length. This indicates that our approach differs from traditional deep hashing methods (Yang et al.) 2015; Yuan et al.) 2020; Jose et al., 2022) which rely on learning the semantic features of samples. It is common sense that the performance of such methods is closely associated with the dimensionality of the high-dimensional space, as more dimensions typically provide richer information. However, our method only employs non-mutual exclusion of encoding, not high-dimensional properties, to reflect patterns of memorization effects. Therefore, our approach is robust to code length.

Table 7: Test accuracy(%) on CIFAR-10 and CIFAR-100 with symmetric and asymmetric noise.

Dataset	CIFAR-10				CIFAR-100				
Noise type		Sym.		Asym.		Sym.		Asym.	
Code length/Noise ratio	0.2	0.5	0.8	0.4	0.2	0.5	0.8	0.4	
8	94.66	91.85	82.75	90.56	-	-	-	-	
16	94.7	92.00	82.87	90.79	-	-	-	-	
32	94.74	92.3	83.28	90.99	-	-	-	-	
64	94.67	92.10	83.41	90.51	72.14	66.91	40.81	67.66	
128	94.62	91.95	83.1	90.55	72.04	66.39	38.25	68.86	

## A.10 DETAILS OF SAMPLE SELECION

We record the number and ratio of clean samples selected in each epoch. In moderate noise, this data is reported for noise condition Sym.  $\epsilon = 0.2$ , Sym.  $\epsilon = 0.5$ , Sym.  $\epsilon = 0.8$  in the CIFAR-10 and CIFAR-100 datasets and is presented in Fig. 7. In extreme noise, we report the results in Sym.  $\epsilon = 0.9$  in the CIFAR-10 and is presented in Fig. 8.

Moderate Noise. It can be observed that the number of clean samples continuously increases and stabilizes during the last 20 epochs, presented in Fig. 7(a) and Fig. 7(c). As for Fig. 7(b) and Fig. 7(d), the ratio of clean samples increases steadily during the early and middle stages of training, indicating that the selection ability of the network improves as its performance increases. However, in the later stages (epochs range from 150 to 200), the ratio of clean samples begins to decline, primarily due to the model starting to fit the noisy data, which results in more noisy samples being misclassified as clean.

914 Extreme Noise. As illustrated in Fig. 8(b), methods utilizing the jump-update strategy achieve a
915 higher ratio in clean samples, demonstrating the effectiveness of the jump-update strategy in mitigating
916 selection bias. Furthermore, Fig. 8(a) highlights that Jump-teaching significantly outperforms
917 others in the number of selected clean samples, underscoring the effectiveness of Semantic Loss
918 Decomposition in detecting noisy samples.



Figure 7: Clean number (num.) and ratio (rat.) of selected samples on CIFAR-10 (C-10) and CIFAR-100 (C-100) with different noise types and ratios.



Figure 8: Clean number and ratio of selected samples on CIFAR-10 and CIFAR-100 with extreme noise.

944

939

925 926

927

#### A.11 THE SUMMARY OF THE BENCHMARK DATASETS

The summary of three benchmark datasets, CIFAR-10, CIFAR-100 and Clothing1M is illustrated 945 in Table 8 Synthetic noisy benchmark in Sec. 4.1 is evaluated on CIFAR-10, and CIFAR-100, 946 respectively, while real-world noisy benchmark in Sec. 4.1 is evaluated on *Clothing1M*. Specifically, 947 the CIFAR-10 dataset is a standard dataset widely used for image classification tasks. It consists of 948 60,000 color images with a resolution of  $32 \times 32$  pixels, divided into 10 categories, each containing 949 6,000 images. The dataset is divided into 50,000 training images and 10,000 testing images. Similar 950 to the former, the CIFAR-100 dataset contains more categories. It consists of 60,000 color images of 951  $32 \times 32$  pixels, divided into 100 categories, with each category containing 600 images. *Clothing1M* 952 dataset is a large-scale image classification dataset specifically designed for clothing recognition. It 953 contains over 1,000,000 labeled images of clothing, covering 14 different categories such as skirts, shirts, coats, shoes, etc. Images in the Clothing1M dataset typically do not have a fixed standard 954 resolution, as they are sourced from the internet. 955

Table 8: Summary of datasets used in the experiment
---

Dataset	# of training	# of testing	# of class	Image size
CIFAR-10	50,000	10,000	10	32x32
CIFAR-100	50,000	10,000	100	32x32
Clothing1M	120,000	10,000	14	not fixed

962 963

## 964 965

966

#### A.12 THE SIMULATION OF NOISE

As the datasets are clean, we follow Li et al. (2020) and simulate symmetric and asymmetric noise.

**968 Symmetric Noise.** Symmetric noise assumes that label errors occur randomly and are independent of 969 the true labels. For a classification problem with *C* classes, symmetric noise is typically defined as in 970 Eq. 9. The formula includes two things: 1) Each correct label remains unchanged with probability 971  $1 - \epsilon + \frac{\epsilon}{C}$ . 2) Each correct label *i* changes to one of the all *C* classes *j* with probability  $\epsilon$ . If the noise 971 is uniformly distributed, the probability of changing to any other class is  $\frac{\epsilon}{C}$ .  $\begin{cases} 1 - \epsilon + \frac{\epsilon}{C} & \text{if } i = j, \\ \frac{\epsilon}{C} & \text{otherwise.} \end{cases}$ (9)

Asymmetric Noise. Asymmetric noise reflects that label errors are more likely to occur between certain classes, typically based on practical scenarios where some classes are visually similar or commonly confused. The probability model for asymmetric noise typically depends on the specific task and data. A simple model might be that the label for a specific class i is incorrectly marked as class j with probability  $\epsilon_{ij}$ . The asymmetric noise can be formulated in Eq. 10 as follows.

$$P(\tilde{Y} = j \mid Y = i) = \epsilon_{ij},\tag{10}$$

where  $\epsilon_{ij}$  can be non-zero when  $i \neq j$ , reflecting specific mislabeling scenarios. where Y represents the true label,  $\tilde{Y}$  represents the potentially noisy label, and  $\epsilon$  is the noise ratio.

Overall, the symmetric noise assumes that the probability of mislabeling is uniform across all categories, whereas the asymmetric noise considers the similarity between categories. The choice of noise typically depends on the characteristics of the dataset and the specific requirements of the task.

## A.13 MORE EXPERIMENTS ON DIFFERENT NOISE TYPES

Although our experiments tested five types of noise in Sec. 4, the additional experiments of pairflip-45 and instance-dependent label noise (IDN) are illustrated here. Results are shown in Table 9 and Table 10 respectively. On pairflip noise, Jump-teaching also exhibits its strong performance, with an increase of 6.73% and 9.95% on CIFAR-10 and CIFAR-100, respectively. Meanwhile, our algorithm maintains a huge advantage in instance-dependent noise. The leading margins are 3.95% (CIFAR-10 IDN. 0.2), 6.74% (CIFAR-10 IDN. 0.4), 10.2% (CIFAR-10 IDN. 0.6), 4.87% (CIFAR-100 IDN. 0.2), 4.3% (CIFAR-100 IDN. 0.4) and 12.67% (CIFAR-100 IDN. 0.6). This indicates that our algorithm is also extremely robust under IDN settings. 

Table 9: Performance of methods on CIFAR-10 and CIFAR-100 with pairflip  $\epsilon = 0.45$  noise.

Methods	CIFAR-10	CIFAR-100
CE	$50.22\pm0.43$	$21.59\pm0.87$
Co-teaching	$52.99 \pm 0.16$	$33.22\pm0.66$
Co-teaching+	$55.19\pm0.27$	$29.26\pm0.15$
PENCIL	$54.58 \pm 2.26$	$24.97\pm0.57$
SPRL	$90.54\pm0.02$	$53.62 \pm 1.07$
FINE	$77.09\pm0.10$	$41.62 \pm 1.01$
Topofilter	$84.60\pm0.45$	$52.40 \pm 1.42$
Jump-teaching	$\textbf{91.33} \pm \textbf{0.23}$	$\textbf{62.35} \pm \textbf{0.74}$

Table 10: Performance of methods on CIFAR-10 and CIFAR-100 with varying IDN noise ratios.

Methods		CIFAR-10		CIFAR-100				
	0.2	0.4	0.6	0.2	0.4	0.6		
СЕ	$85.45 \pm 0.57$	$76.23 \pm 1.54$	$59.75 \pm 1.30$	$57.79 \pm 1.25$	$41.15\pm0.83$	25.68 ± 1.55		
Co-teaching	$88.87 \pm 0.24$	$73.00 \pm 1.24$	$62.51 \pm 1.98$	$43.30\pm0.39$	$23.21\pm0.57$	$12.58\pm0.51$		
Co-teaching+	$89.80 \pm 0.28$	$73.78 \pm 1.39$	$59.22\pm 6.34$	$41.71 \pm 0.78$	$24.45\pm0.71$	$12.58\pm0.51$		
PENCIL	$86.23 \pm 1.23$	$69.88 \pm 1.39$	$44.16 \pm 1.58$	$57.41 \pm 0.69$	$48.38\pm0.52$	$29.69\pm3.03$		
FINE	$90.85 \pm 1.21$	$84.53\pm0.77$	$61.24 \pm 2.02$	$60.14 \pm 0.015$	$43.05 \pm 1.43$	$28.88 \pm 2.01$		
Торо	$89.66 \pm 0.86$	$85.95 \pm 1.02$	$75.28 \pm 0.64$	$64.41 \pm 0.37$	$55.00\pm0.82$	$31.61 \pm 1.45$		
SPRL	$87.99 \pm 0.03$	$86.15\pm0.81$	$75.01\pm0.74$	$67.66 \pm 0.17$	$64.55\pm0.19$	$47.16 \pm 1.97$		
Jump-teaching	$\textbf{94.80} \pm \textbf{0.28}$	$\textbf{92.89} \pm \textbf{0.33}$	$\textbf{85.48} \pm \textbf{0.41}$	$\textbf{72.53} \pm \textbf{0.21}$	$\textbf{68.85} \pm \textbf{0.27}$	$\textbf{59.83} \pm \textbf{0.09}$		

# 1026 A.14 A BRIEF OVERVIEW OF COMPARED METHODS

We reimplemented Decoupling, Co-teaching, Co-teaching+, and PENCIL with the publicly available code<sup>T</sup> Similarly, FINE, SPRL, and TopoFilter were re-implemented by their respective official repositories<sup>TFIS</sup> Results in Table 3 are cited from (Xia et al., 2023). The outcomes for Decoupling, Co-teaching+, and PENCIL on *CIFAR-10* and *CIFAR-100* with symmetric noise, shown in Table 1, are derived from (Li et al., 2023). Additionally, baseline results featured in Table 5 for DivideMix, DivideMix with  $\theta_1$  test, and DivideMix without co-training are sourced from (Li et al., 2020).

1035 A.15 DATA AUGMENTATION

When the noise is severe, there are very few clean samples. Thus, the network struggles to acquire sufficient knowledge from a small number of available samples. Therefore, we adopt a data augmentation strategy following Cubuk et al. (2020) and linearly interpolate the augmented image with its weak view generated by random cropping and horizontal flipping. This enables the network to more easily focus on the key features of the samples. We follow Eq. [1] to combine weak and strong views.

$$\boldsymbol{x}_{aug} = \lambda \boldsymbol{x}_{weak} + (1 - \lambda) \boldsymbol{x}_{strong}, \text{ where } \lambda \sim \text{Beta}(\alpha, \alpha).$$
 (11)

#### 1044 1045 A.16 ANALYSIS OF THRESHOLD

1046 We test the sensitivity of Jump-teaching to the threshold  $\tau$ . We conduct experiments on CIFAR-10 with 1047 symmetric noise ratio  $\epsilon = 0.5$  and asymmetric noise ratio  $\epsilon = 0.4$ . As shown in Table 11, the results 1048 indicate a relatively stable performance with different thresholds. In other words, the effectiveness 1049 of Jump-teaching is not significantly influenced by the value of  $\tau$  and is more adaptable to various datasets. We attribute the success to two main factors. First, the Hadamard matrix for hashing 1050 codes exhibits uniformness in each column (Yang et al., 2015). Second, we utilize classification 1051 and auxiliary heads to collaboratively select samples. This guarantees that an adequate quantity of 1052 samples is always available for learning. Both of the two factors significantly enhance the stability 1053 and accuracy of the model across different datasets. 1054

Table 11: Test Accuracies(%) with different threshold settings.

au	0.001	0.005	0.01	0.05	0.1
Sym. $\epsilon = 0.5$	92.12	92.17	92.04	91.97	92.17
Asym. $\epsilon = 0.4$	90.71	90.66	90.79	90.82	90.70

1061 1062

1063

1055

1034

1042

1043

## A.17 THE CHOICE OF JUMP STEP

In the Table 12, we test the effect of jump steps in Jump-teaching. It is shown that performance with various step settings remains stable. In the Jump-update Strategy, we opt for a step size of 2, obviating the need for additional forward propagation. Furthermore, empirical results indicates a marginal decline in performance as step size increases. This evidence corroborates the assertions in Sec. 3.1 That is to say, delays in updating tables can detrimentally impact the network assimilating newly introduced clean samples.

1070

1072

## 1071 A.18 THE STRUCTURES OF BACKBONES AND THE AUXILIARY HEAD

The auxiliary head of Jump-teaching consists of three fully connected layers with ReLU activations and dropout layers between two fully connected layers, followed by a Tanh activation. For *CIFAR-10* and *Clothing1M*, the output dimension of the last fully connected layer is set to 32, whereas setting to 64 for *CIFAR-100*. The intermediate dimension is uniformly set to 512.

<sup>1077 \*</sup>https://github.com/JackYFL/DISC

<sup>1078 &</sup>lt;sup>†</sup>https://github.com/Kthyeon/FINE\_official

<sup>1079 &</sup>lt;sup>‡</sup>https://github.com/pxiangwu/TopoFilter

https://github.com/xsshi2015/Self-paced-Resistance-Learning

Table 12: Test accuracy(%) on *CIFAR-10* and *CIFAR-100* with symmetric and asymmetric noise among diverse noise settings.

$\label{eq:constraints} \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1083	Noise type and ratio/ St	ep 1	2	3	4	5	10	15	20	25	30	
$\frac{CTPAR:100 \text{ Sym. } \epsilon = 0.5}{66.91} \frac{90.71}{67.03} \frac{90.66}{66.27} \frac{90.38}{66.27} \frac{90.35}{65.37} \frac{90.36}{65.88} \frac{90.69}{65.46} \frac{90.69}{66.27} \frac{90.35}{65.88} \frac{90.69}{65.88} \frac{90.69}{66.27} \frac{90.36}{65.88} \frac{90.69}{65.88} \frac{90.69}{65.46} \frac{90.69}{66.27} \frac{90.36}{65.88} \frac{90.69}{65.88} \frac{90.69}{65.27} \frac{90.38}{65.88} \frac{65.46}{64.67}$ A.19 EXPERIMENTS OF THE JUMP-UPDATE STRATEGY ON THE Clothing IM DATASET In this section, we evaluate the performance of the jump-update strategy on the Clothing IM datas Specifically, we have integrated the jump-update strategy with Co-teaching and DivideMix. The experimental setup for DivideMix for the integration of the jump-update strategy results performance improvements of 2.88% for Co-teaching and 0.26% for DivideMix, demonstrating the reliability of the jump-update strategy in real-world scenarios. Table 13: Test accuracy(%) on Clothing IM. $\overline{\Delta councy} = 67.94 \qquad 70.82 \qquad 74.17 \qquad 74.43$	084	CIFAR-10 Sym. $\epsilon = 0$ .	.8 82.81	83.37	82.47	83.09	82.55	82.04	81.03	81.35	81.46	79.77	
$\frac{\text{CFAR-100 Sym }\epsilon = 0.5 \   66.91 \ 67.03 \ 66.85 \ 66.21 \ 66.27 \ 65.34 \ 65.87 \ 65.88 \ 65.46 \ 64.67}{\text{A.19 EXPERIMENTS OF THE JUMP-UPDATE STRATEGY ON THE Clothing IM DATASET}}$ A.19 EXPERIMENTS OF THE JUMP-UPDATE STRATEGY ON THE Clothing IM DATASET In this section, we evaluate the performance of the jump-update strategy on the Clothing IM datases the circular strategy of the control of the jump-update strategy on the Clothing IM datases to performance improvements of 2.48% for Co-teaching and DyideMix, IT experimental setup for Co-teaching adheres to Kia et al. (2023), and the setup for DivideMix follow Lie et al. (2020). As shown in Table []3, the integration of the jump-update strategy results is performance improvements of 2.48% for Co-teaching adheres to Kia et al. (2025) for DivideMix, demonstrating the eliability of the jump-update strategy in real-world scenarios. Table 13: Test accuracy(%) on Clothing1M. $Method   Co-teaching (2*ResNet18)   J-Co-teaching (1*ResNet18)   DivideMix   J-DivideMix / Accuracy   67.94 70.82   74.17 74.43 $	085	CIFAR-10 Asym. $\epsilon = 0$	0.4 90.91	90.77	90.66	90.78	90.86	90.42	90.35	90.60	90.69	90.23	
A.19 EXPERIMENTS OF THE JUMP-UPDATE STRATEGY ON THE Clothing/M DATASET In this section, we evaluate the performance of the jump-update strategy on the Clothing/M datast Specifically, we have integrated the jump-update strategy with Co-teaching and DyvideMix. Follow Let al.] (2020). As shown in Table [13], the integration of the jump-update strategy results in performance improvements of 2.288% for Co-teaching and DycK for DivideMix, demonstrating the reliability of the jump-update strategy in real-world scenarios. <b>Table 13:</b> Test accuracy(%) on Clothing1M. <u>Method [Co-teaching (2?ResNet18)] J-Co-teaching (1?ResNet18)] DivideMix J-DivideMix Accuracy 67.94 70.82 74.17 74.43 70.82 74.17 74.43</u>	086	CIFAR-100 Sym. $\epsilon = 0$	0.5 66.91	67.03	66.85	66.21	66.27	65.54	65.87	65.88	65.46	64.67	
A.19 EXPERIMENTS OF THE JUMP-UPDATE STRATEGY ON THE Clothing/M DATASET In this section, we evaluate the performance of the jump-update strategy on the Clothing/M datas specifically, we have integrated the jump-update strategy with Co-teaching and DivideMix. The experimental setup for Co-teaching adheres to [Xia et al.] (2023), and the setup for DivideMix of the performance improvements of 2.88% for Co-teaching and 0.26% for DivideMix, demonstrating the reliability of the jump-update strategy in real-world scenarios. <b>Fabe 13:</b> Test accuracy(%) on Clothing/M. Method [Co-teaching (2PResNet18)] JCO-teaching (1PREsNet18)] DivideMix - JDivideMix Accuracy 67.94 70.82 17.41.7 74.43	087												
A.19 EXPERIMENTS OF THE JUMP-UPDATE STRATEGY ON THE <i>Clohing1M</i> datase pecifically, we have integrated the jump-update strategy with Co-teaching and DivideMix. The experimental setup for Co-teaching adheres to <u>Via et all</u> (2023), and the setup for DivideMix follow <u>Li et all</u> (2020). As shown in Table [13], the integration of the jump-update strategy results in performance improvements of 2.38% for Co-teaching ad DivideMix, demonstrating the reliability of the jump-update strategy in real-world scenarios. Table 13: Test accuracy(%) on <i>Clothing1M</i> . <u>Method  Co-teaching (2*ResNet18)  JCo-teaching (1*ResNet18)  DivideMix - J-DivideMix</u> <u>Accuracy 67.94 70.82 174.17 74.43</u> Accuracy 67.94 70.82 174.17 74.43	088												
In this section, we evaluate the performance of the jump-update strategy on the Clothing IM datas Specifically, we have integrated the jump-update strategy with Co-teaching and DivideMix. The performance improvements of 2.88% for Co-teaching and 0.26% for DivideMix, demonstrating the liability of the jump-update strategy in real-world scenarios. <b>Fable 13:</b> Test accuracy(%) on Clothing1M. <u>Method [Co-teaching (2*ResNet18)]-Co-teaching (1*ResNet18)]DivideMix 1-DivideMix Accuracy</u> 67.94 70.82 74.17 74.43	1089 A.1	9 EXPERIMENTS OF	THE JUN	ip-Upi	DATE	STRAT	ΓEGY	ON TH	ie <i>Clo</i>	thing	$M \mathbf{D}$	ATASE	Т
$ \frac{1}{100} = \frac{1}{100} \frac$	1090 In th	his saction we evaluate	the parfo	rmana	a of th	o ium	n und	oto str	otogy	on the	Clath	ina 11	A datasat
$\frac{1}{10} = \frac{1}{10} \frac{1}{10}$	1091 III u Spec	cifically we have integr	rated the	iumn	undat	te stra	ip-upu teav v	ate su vith C	alegy	hing a	and $\mathbf{D}$	ung 11 ivideN	<i>I</i> ualaset
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	1092 Spec	erimental setup for Co-te	eaching a	dheres	toXi	a et al	1(202)	$\frac{3}{3}$ , and	the s	etun fo	or Div	ideMi	x follows
performance improvements of 2.88% for Co-teaching and 0.26% for DivideMix, demonstrating the reliability of the jump-update strategy in real-world scenarios.           Table 13: Test accuracy(%) on Clothing1M.           Method   Co-teaching (2*ResNet18) ]-Co-teaching (1*ResNet18) [DivideMix ]-DivideMix           Accuracy           67.94         70.82         74.17         74.43           Test         70.82         74.17         74.43	1093 Li e	et al. (2020). As show	n in Tat	le 13.	the i	ntegra	tion c	of the	jump-	updat	e stra	tegy 1	esults in
reliability of the jump-update strategy in real-world scenarios. Table 13: Test accuracy(%) on Clothing IM. $\hline Method \ Co-teaching (2*ResNet18) \ J-Co-teaching (1*ResNet18) \ DivideMix \ J-DivideMix \ Accuracy \ 67.94 \ 70.82 \ 74.17 \ 74.43 \ 74.43 \ 74.43 \ 74.43 \ 74.43 \ 76.75 $	1094 perf	ormance improvements	of 2.88%	6 for C	o-tead	ching a	and 0.	26% f	or Div	ideMi	ix, der	nonsti	rating the
Table 13: Test accuracy(%) on Clothing IM.MethodCo-teaching (2*ResNet18)J-Co-teaching (1*ResNet18)DivideMixJ-Couracy67.9470.8274.1774.43	<sup>1095</sup> relia	bility of the jump-upda	te strateg	y in re	al-wo	rld sce	enario	s.					U
Table 13: Test accuracy(%) on Clothing1M.         Method       Co-teaching (2*ResNet18)       I-Co-teaching (1*ResNet18)       I-DivideMix       I-DivideMix         Accuracy       67.94       70.82       74.17       74.43         Interview       67.94       70.82       74.17       74.43	1096												
Method         Co-teaching (2*ResNet18)         J-Co-teaching (1*ResNet18)         DivideMix         J-DivideMix           Accuracy         67.94         70.82         74.17         74.43	1097		Table 13	: Test	accura	ncy(%)	) on <i>C</i>	lothin	g1M.				
Accuracy         67.94         70.82         74.17         74.43           101	098	Mathad Co tagahi	ng ( <b>)*P</b> asi	Not19)	I Co. to	aching	(1* <b>D</b> oo	Not18)	Divide	Miy I	Divida	Mix	
Accuracy         07.94         70.62         7.4.17         74.43           100	1100		67.04	Net10)	J-C0-10		(1 · KCS.	inciro)			-Divide		
NULL         102         103         104         105         106         107         108         109         110         111         112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	100	Accuracy	67.94			/0.	.82		/4.	. /	74.43	<u> </u>	
102         103         104         105         106         107         108         109         110         111         112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	1101												
103         104         105         106         107         108         109         110         111         112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132	1102												
105         106         107         108         109         110         111         112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	104												
102         103         109         110         111         112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	105												
102         103         109         110         111         112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         120         121         122         123         124         125         126         127         128         129         121         122         123         131         132         133	106												
103         109         110         111         112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	107												
110         111         112         112         112         112         112         112         112         112         112         112         112         112         112         112         112         112         113         113         113         113         113         113         113         113         113         114         1	108												
110         111         112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	109												
111         112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	110												
112         113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	111												
113         114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	112												
114         115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	113												
115         116         117         118         119         120         121         122         123         124         125         126         127         128         129         120         130         131         132         133	114												
116         117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	115												
117         118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	116												
118         119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	117												
119         120         121         122         123         124         125         126         127         128         129         130         131         132         133	118												
120         121         122         123         124         125         126         127         128         129         130         131         132         133	119												
121         122         123         124         125         126         127         128         129         130         131         132         133	120												
122         123         124         125         126         127         128         129         130         131         132         133	121												
123         124         125         126         127         128         129         130         131         132         133	122												
124         125         126         127         128         129         130         131         132         133	123												
125 126 127 128 129 130 131 132	124												
126 127 128 129 130 131 132	125												
127 128 129 130 131 132 133	126												
128 129 130 131 132 133	127												
129 130 131 132 133	128												
130 131 132 133	129												
131 132 133	130												
132 133	131												
133	132												
	133												