南京航空航天大學

Nanjing University of Aeronautics and Astronautics

Estimating Instance-dependent Bayes-label Transition Matrix using a Deep Neural Network

Shuo Yang [1] Erkun Yang [2] Bo Han [3] Yang Liu [4] Min Xu [1] Gang Niu [5] Tongliang Liu [6]

ICML 2022

Transition matrix

Symmetric Noise 0.4

60%	8%	8%	8%	8%	8%
8%	60%	8%	8%	8%	8%
8%	8%	60%	8%	8%	8%
8%	8%	8%	60%	8%	8%
8%	8%	8%	8%	60%	8%
8%	8%	8%	8%	8%	60%

Asymmetric Noise 0.4

100%	0%	0%	0%	0%	0%
0%	60%	0%	0%	40%	0%
40%	0%	60%	0%	0%	0%
0%	0%	0%	100%	0%	0%
0%	0%	0%	0%	100%	0%
0%	0%	40%	0%	0%	60%

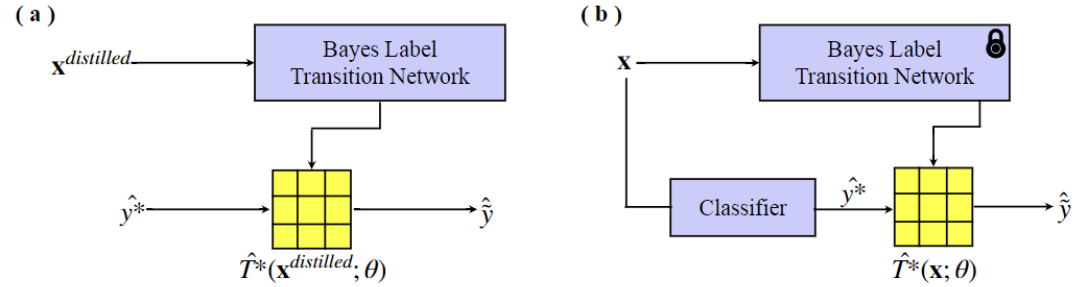


Figure 2. (a) The Bayes Label Transition Network is used to predict *Bayes-label transition matrix* for each input instance, it is trained in a supervised way by employing the collected Bayes optimal labels. (b) The learned Bayes Label Transition Network is *fixed* to train the classifier by leveraging the loss correction approach (Patrini et al., 2017).

Class Conditional Noise:

$$\Pr(\bar{Y} \mid Y, X) \quad = \Pr(\bar{Y} \mid Y) \text{=====>} T_{i,j} \quad = \{\Pr(\bar{Y} \quad = j \mid Y \quad = i)\}$$

Instance Dependent Noise:

$$T_{i,j}(x) \quad = \{\Pr(\bar{Y} \quad = j \mid Y \quad = i, \mathrm{x})\}$$

$$P(\bar{Y} = j \mid \mathbf{X} = \mathbf{x}) = \sum_{i=1}^{K} T_{ij}(\mathbf{x}) P(Y = i \mid \mathbf{X} = \mathbf{x})$$

Collecting Bayes Optimal Labels

Learning with Bounded Instance- and Label-dependent Label Noise ICML 2020

Bayes optimal labels can be inferred from the noisy class posterior probabilities (Confident Examples)

Theorem 2. *Denote by* $\tilde{\eta}(X)$ *the conditional probability* $P_{D_\rho}(Y=+1|X)$. $\forall X_i \in \mathcal{X}$, *given that* $UB(\rho_{\pm 1}(X_i))$ *is an upper bound of* $\rho_{\pm 1}(X_i)$, *we have*

$$\tilde{\eta}(X_i) < \frac{1-UB(\rho_{+1}(X_i))}{2} \implies (X_i, Y_i = -1) \text{ is distilled;}$$
$$\tilde{\eta}(X_i) > \frac{1+UB(\rho_{-1}(X_i))}{2} \implies (X_i, Y_i = +1) \text{ is distilled.}$$

Corollary 1. $\forall X_i \in \mathcal{X}$, *we have*

$$\tilde{\eta}(X_i) < \frac{1-\rho_{+1\max}}{2} \implies (X_i, Y_i = -1) \text{ is distilled;}$$
$$\tilde{\eta}(X_i) > \frac{1+\rho_{-1\max}}{2} \implies (X_i, Y_i = +1) \text{ is distilled.}$$

The noisy class posterior probability η can be estimated by several probabilistic classification methods

Initialize the distilled dataset $\mathcal{S}^* = \{\}$;
Learn $\hat{\tilde{\eta}}$ on the noisy dataset $\tilde{S}$;
for $(\mathbf{x}_i, \tilde{y}_i)$ *in* $\tilde{S}$ **do**
  for y *in* $\mathcal{Y}$ **do**
    if $\hat{\tilde{\eta}}_y(\mathbf{x}_i) > \frac{1+\rho_{max}}{2}$ **then**
      $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \{(\mathbf{x}^{distilled} = \mathbf{x}_i, \tilde{y} = \tilde{y}_i, \hat{y^*} = y)\}$
    end
  end
end

Bayes Label Transition Network

With the collected distilled examples $(\mathbf{x}^{distilled}, \tilde{y}, \hat{y^*})$

$$\hat{T^*_{i,j}}(\mathbf{x}^{distilled}; \theta) = P(\tilde{Y} = j | Y^* = i, \mathbf{x}^{distilled}; \theta)$$

Bayes Label Transition Network:

$\hat{R}_1(\theta) = -\frac{1}{m}\sum_{i=1}^{m} \tilde{\mathbf{y}}_i \, log\left(\hat{\mathbf{y}}_\mathbf{i}^* \cdot \hat{T}^*\left(\mathbf{x}_i^{distilled}; \theta\right)\right)$ to learn IDTM(X)

Classifier Training with Forward Correction:

$$P(\tilde{Y} = j \mid \mathbf{x}; w, \theta)$$
$$= \sum_{i=1}^{k} P(\tilde{Y} = j \mid Y^* = i, \mathbf{x}; \theta) P(Y^* = i \mid \mathbf{x}; w)$$

$$\hat{R}_2(w) = -\frac{1}{n}\sum_{i=1}^{n} \tilde{\boldsymbol{y}}_{\boldsymbol{i}} \log(f(\mathbf{x}_i; w) \cdot \hat{T^*}(\mathbf{x}_i; \theta))$$

Instance-dependent Bayes-label Transition Matrix

Algorithm 1 Instance-dependent Label-noise Learning with Bayes Label Transition Network.

Input: Noisily-labeled dataset $\tilde{S} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{n}$

Required: the noise rate upper bound ρ_{max}, random initialized Bayes label transition network $\hat{T}^*(\cdot;\theta)$, random initialized classification network $f(\cdot; w)$

// Section. 4.2: Collecting Bayes Optimal Labels

Initialize the distilled dataset $\mathcal{S}^* = \{\}$;

Learn $\hat{\tilde{\eta}}$ on the noisy dataset $\tilde{S}$;

for $(\mathbf{x}_i, \tilde{y}_i)$ *in* $\tilde{S}$ **do**

  for y *in* $\mathcal{Y}$ **do**

    if $\hat{\tilde{\eta}}_y(\mathbf{x}_i) > \frac{1+\rho_{max}}{2}$ **then**

      $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \{(\mathbf{x}^{distilled} = \mathbf{x}_i, \tilde{y} = \tilde{y}_i, \hat{y^*} = y)\}$

    end

  end

end

// Section. 4.3: Training Bayes Label Transition Network

Minimize the $\hat{R}_1(\theta)$ in Eq. 3 on $\mathcal{S}^*$ to learn the Bayes label transition network's parameter θ.

// Section. 4.4: Training Classifier with Forward Correction

Fix the learned θ and minimize the $\hat{R}_2(w)$ in Eq. 5 on $\tilde{S}$ to learn the classifier's parameter w.

Output: The classifier $f(\cdot; w)$

Classifier Training with Forward Correction

$$\hat{R}_2(w) = -\frac{1}{n}\sum_{i=1}^{n} \tilde{\boldsymbol{y}}_{\boldsymbol{i}} \log(f(\mathbf{x}_i; w) \cdot \hat{T}^*(\mathbf{x}_i; \theta))$$

Extract confident clean examples using example distillation method:

Initialize the distilled dataset $\mathcal{S}^* = \{\}$;

Learn $\hat{\tilde{\eta}}$ on the noisy dataset $\tilde{S}$;

for $(\mathbf{x}_i, \tilde{y}_i)$ *in* $\tilde{S}$ **do**

  for y *in* $\mathcal{Y}$ **do**

    if $\hat{\tilde{\eta}}_y(\mathbf{x}_i) > \frac{1+\rho_{max}}{2}$ **then**

      $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \{(\mathbf{x}^{distilled} = \mathbf{x}_i, \tilde{y} = \tilde{y}_i, \hat{y^*} = y)\}$

    end

  end

end

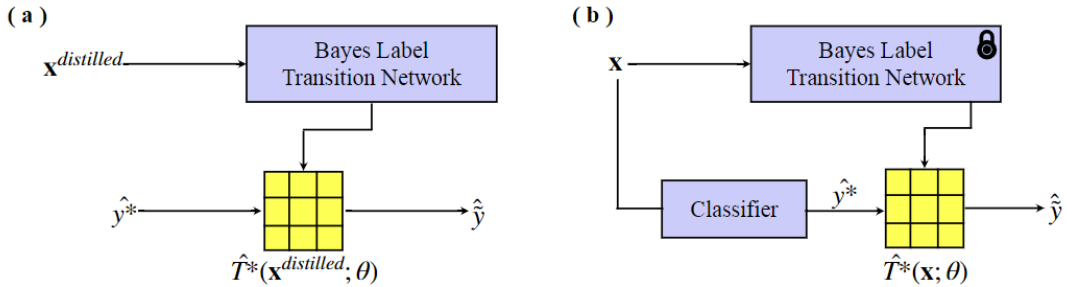


Figure 2. (a) The Bayes Label Transition Network is used to predict *Bayes-label transition matrix* for each input instance, it is trained in a supervised way by employing the collected Bayes optimal labels. (b) The learned Bayes Label Transition Network is *fixed* to train the classifier by leveraging the loss correction approach (Patrini et al., 2017).

$$T^*_{i,j}\left(\mathbf{x}^{\boldsymbol{distilled}};\theta\right) = P\left(\tilde{Y} = j \mid Y^* = i, \mathbf{x}^{\boldsymbol{distilled}};\theta\right)$$

$$\hat{R}_1(\theta) = -\frac{1}{m}\sum_{i=1}^{m} \tilde{\mathbf{y}}_i \, log\left(\hat{\mathbf{y}}^*_{\mathbf{i}} \cdot \hat{T}^*\left(\mathbf{x}_i^{distilled};\theta\right)\right)$$ to learn IDTM(X)

Experiments

Table 3. Means and standard deviations (percentage) of classification accuracy on *SVHN* with different label noise levels. '-V' indicates matrix revision (Xia et al., 2019).

	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	90.39 ± 0.13	89.04 ± 1.32	85.65 ± 1.84	79.94 ± 2.71	61.01 ± 5.41
GCE	90.82 ± 0.15	89.35 ± 0.94	86.43 ± 0.63	81.66 ± 1.58	54.77 ± 0.25
APL	71.78 ± 0.76	89.48 ± 1.67	83.46 ± 2.17	77.90 ± 2.31	55.25 ± 3.77
Decoupling	90.55 ± 0.83	88.74 ± 0.77	85.03 ± 1.63	83.36 ± 2.73	56.76 ± 1.87
MentorNet	90.28 ± 0.52	89.09 ± 0.95	85.89 ± 0.73	82.63 ± 1.73	55.27 ± 4.14
Co-teaching	91.05 ± 0.33	89.56 ± 1.77	87.75 ± 1.37	84.92 ± 1.59	59.56 ± 2.34
Co-teaching+	92.83 ± 0.87	90.73 ± 1.39	86.37 ± 1.66	75.24 ± 3.77	54.58 ± 3.46
Joint	88.39 ± 0.62	85.37 ± 0.44	81.56 ± 0.43	78.98 ± 2.98	59.14 ± 3.22
DMI	92.11 ± 0.49	91.63 ± 0.87	86.98 ± 0.36	81.11 ± 0.68	63.22 ± 3.97
Forward	90.01 ± 0.78	89.77 ± 1.54	86.70 ± 1.44	80.24 ± 2.77	57.57 ± 1.45
Reweight	91.06 ± 0.19	92.01 ± 1.04	87.55 ± 1.71	83.79 ± 1.11	55.08 ± 1.25
S2E	92.70 ± 0.51	92.02 ± 1.54	88.77 ± 1.77	83.06 ± 2.19	65.39 ± 2.77
T-Revision	93.07 ± 0.79	92.67 ± 0.88	88.49 ± 1.44	82.43 ± 1.77	67.64 ± 2.57
PTD	93.77 ± 0.33	92.59 ± 1.07	89.64 ± 1.98	83.56 ± 2.21	71.57 ± 3.32
BLTM	96.05 ± 0.32	94.97 ± 0.58	93.99 ± 1.24	87.67 ± 1.29	78.13 ± 4.62
BLTM-V	**96.37 ± 0.77**	**95.12 ± 0.40**	**94.69 ± 0.24**	**88.13 ± 3.23**	**78.71 ± 4.37**

Table 4. Classification accuracy on *Clothing1M*. In the experiments, only noisy samples are exploited to train and validate the deep model.

CE	Decoupling	MentorNet	Co-teaching	Co-teaching+	Joint	DMI
68.88	54.53	56.79	60.15	65.15	70.88	70.12
Forward	Reweight	T-Revision	PTD	PTD-V	BLTM	BLTM-V
69.91	70.40	70.97	70.07	70.26	**73.33**	**73.39**

南京航空航天大學

Nanjing University of Aeronautics and Astronautics

Instance-Dependent Label-Noise Learning with Manifold-Regularized Transition Matrix Estimation

De Cheng[1], Tongliang Liu[2], Yixiong Ning[1], Nannan Wang[1]*, Bo Han[3], Gang Niu[4], Xinbo Gao[5], Masashi Sugiyama[4,6].

[1] Xidian University, [2] TML Lab, The University of Sydney, [3] Hong Kong Baptist University, [4] RIKEN, [5] Chongqing University of Posts and Telecommunications, [6] The University of Tokyo.

{dcheng,nnwang}@xidian.edu.cn, tongliang.liu@sydney.edu.au, yxning@stu.xidian.edu.cn, bhanml@comp.hkbu.edu.hk, gang.niu.ml@gmail.com,gaoxb@cqupt.edu.cn, sugi@k.u-tokyo.ac.jp

CVPR 2022

Overview

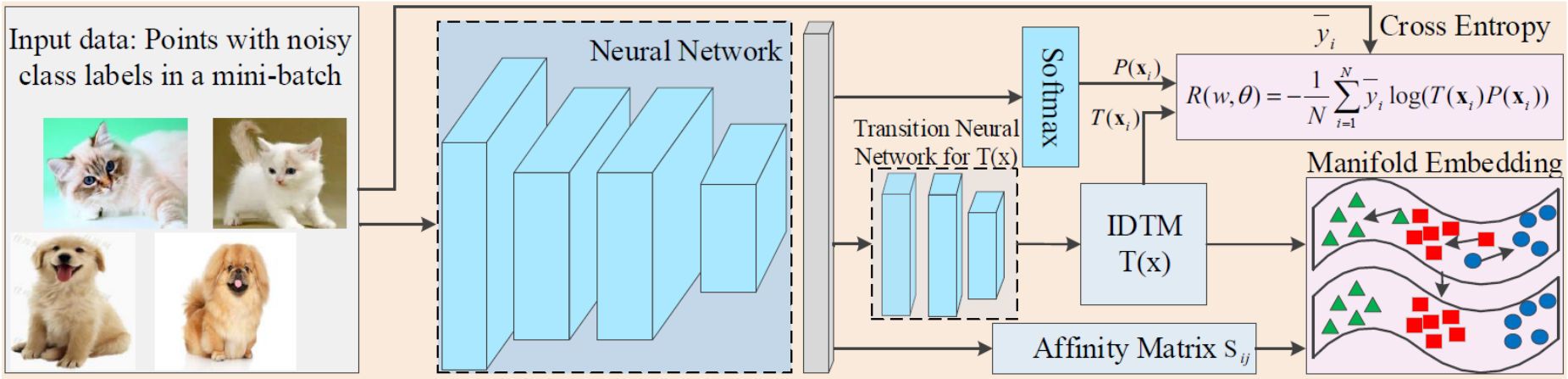


Figure 1. The proposed instance-dependent label-noise learning framework. We train a classifier in a statistically consistent manner through the proposed IDTM $T(\mathbf{x})$, where $T(\mathbf{x}_i) \in R^{K\times K}$ is estimated by the transition neural network. It is regularized by the manifold embedding to reduce the degree of freedom of $T(\mathbf{x})$ and make it estimable in practice. In the manifold embedding $\mathcal{L}$, the affinity matrix S_{ij} is obtained by finding the k-nearest neighbors in the instance feature space. Finally, we use the cross-entropy to train the classifier assisted by $T(\mathbf{x})$.

$$\min_{\mathbf{w},\theta} R(\mathbf{w},\theta) = -\frac{1}{N}\sum_{i=1}^{N} \bar{y}_i \log(T(\mathbf{x}_i;\theta) f(\mathbf{x}_i;\mathbf{w})).$$

Learn a Transition Neural Network regularized by the manifold embedding

Manifold Regularized Transition Matrix

the closer two instances are, the more similar their corresponding transition matrices should be

within-manifold regularization:

$$\mathcal{M}_I = \sum_{i,j=1}^{N} S_{ij}^{(I)} ||T(\mathbf{x}_i) - T(\mathbf{x}_j)||^2,$$

$$S_{ij}^{(I)} = \begin{cases} 1, & \text{if } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i, k_1) \text{ and } \bar{y}_i = \bar{y}_j, \\ 0, & \text{else}, \end{cases}$$

between manifolds:

$$\mathcal{M}_B = \sum_{i,j=1}^{N} S_{ij}^{(B)} ||T(\mathbf{x}_i) - T(\mathbf{x}_j)||^2,$$

$$S_{ij}^{(B)} = \begin{cases} 1, & \text{if } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i, k_2) \text{ and } \bar{y}_i \neq \bar{y}_j, \\ 0, & \text{else}, \end{cases}$$

the overall proposed manifold-regularization:

$$\mathcal{M}(\theta) = \mathcal{M}_I - \mathcal{M}_B.$$

Algorithm

Algorithm 1: Instance-dependent Label-Noise Learning Algorithm

Input: Noisy training dataset $\bar{\mathcal{D}} = \{\mathbf{x}_i, \bar{y}_i\}_{i=1}^{N}$

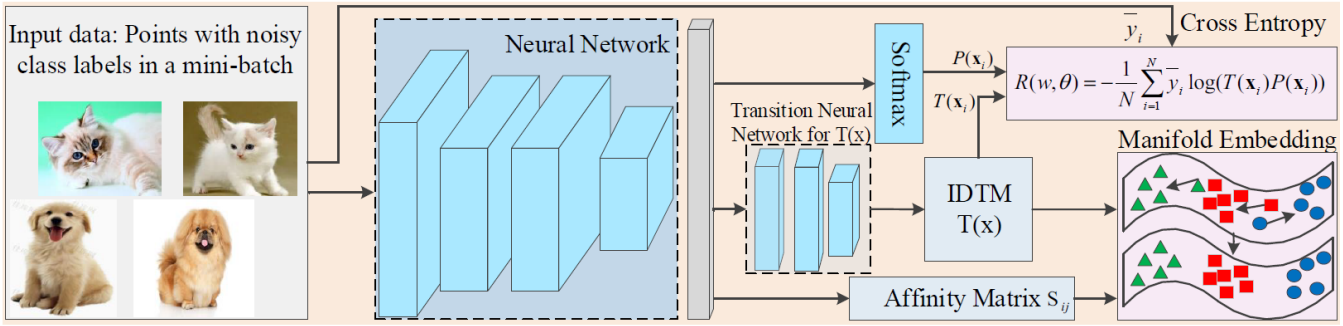
Output: The final classifier $f(\mathbf{x}; \mathbf{w})$ and the transition matrix $T(\mathbf{x}; \theta)$.

Warmup: Train the DNN on the noisy dataset $\bar{\mathcal{D}}$ with the early-stop strategy to obtain the initial classifier $f(\mathbf{x}; \mathbf{w})$;

while *Number of training epoch* $\leq$ *Max-Epoch* **do**

- Extract confident clean examples using example distillation method [50] with current classifier $f(\mathbf{x}; \mathbf{w})$ to form the sub-dataset $\bar{\mathcal{D}}^s = \{\mathbf{x}_i^s, \bar{y}_i\}_{i=1}^{N^s}$;
- Input the extracted confident clean examples into the backbone network;
- Compute the affinity graph matrix $S_{ij}^{(I)}$ and $S_{ij}^{(B)}$ based on current instance features according to Eq. (5) and (7) or Eq. (9) and (10);
- Optimize the DNN based on the loss function shown in Eq. (11) to obtain new classifier $f(\mathbf{x}; \mathbf{w})$ and transition matrix $T(\mathbf{x}; \theta)$.

end



Overall Objective Function

$$\min_{\mathbf{w},\theta} \mathcal{L}(\mathbf{w}, \theta) = R(\mathbf{w}, \theta) + \lambda \mathcal{M}(\theta)$$

Experiments

Table 1. Comparison with state-of-the-art methods on F-MNIST and CIFAR-10 datasets. The mean and standard deviation computed over five runs are presented. "IDN-xx%" means the noise rate is xx% and noise type is "IDN".

Method	F-MNIST					CIFAR-10				
	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE (baseline)	87.73± 1.25	87.63± 1.11	85.25± 0.57	75.00 ±0.25	65.42± 1.59	88.86± 0.23	86.93± 0.17	82.42± 0.44	76.68± 0.23	58.93± 1.54
GCE [54]	90.24 ±0.16	88.71± 0.17	85.90± 0.23	76.78± 0.37	67.67± 0.58	90.82± 0.05	88.89± 0.08	82.90± 0.51	74.18± 3.10	58.93± 2.67
DMI [48]	90.14± 0.22	88.13± 0.47	85.90± 0.23	76.22± 0.71	64.84± 1.28	91.43± 0.18	**89.99**± 0.15	86.87± 0.34	80.74± 0.44	63.92± 3.92
Forward [32]	90.78± 0.30	89.01± 0.44	86.51± 1.20	78.17± 0.32	68.31± 1.07	**91.71**± 0.08	89.62± 0.14	**86.93**± 0.15	80.29± 0.27	**65.91**± 1.22
CoTeaching [11]	90.54± 0.35	88.53± 0.09	87.37± 0.14	78.36± 0.82	67.81± 1.02	90.80± 0.05	88.43± 0.08	86.40± 0.41	80.85± 0.97	62.63± 1.51
CoTeaching++ [52]	90.67±0.49	88.52± 0.44	87.33± 0.87	79.85± 1.03	68.86± 1.39	91.47± 0.59	89.78± 0.34	85.72± 0.35	**81.00**± 0.82	61.46± 1.36
JoCor [43]	**91.48**± 0.11	89.24± 0.09	86.50± 0.10	77.15±1.04	67.85±0.84	91.42± 0.11	89.30± 0.27	85.54± 0.82	80.87± 0.91	64.11± 2.57
PeerLoss [23]	90.76±0.41	87.06± 0.74	84.40± 0.93	73.95± 2.37	65.79± 2.49	90.89± 0.07	89.21± 0.63	85.70± 0.56	78.51± 1.23	59.08± 1.05
TMDNN [50]	91.33± 0.27	89.70± 0.14	87.63± 1.28	78.40± 3.69	66.55± 7.52	90.45± 0.72	88.14± 0.66	84.55±0.48	79.71± 0.95	63.33± 2.75
PartT [44]	91.27± 0.38	**89.78**± 0.43	**88.30**± 0.51	**80.75**± 2.86	**72.22**± 4.22	90.32± 0.15	89.33± 0.70	85.33±1.86	80.59± 0.41	64.58± 2.86
MEIDTM (Ours)	91.78± 0.87	90.49± 0.35	88.74± 0.25	84.21± 0.52	73.67± 3.76	92.17± 0.21	91.38± 0.34	87.68± 0.26	82.63± 0.24	72.17± 1.51
kMEIDTM (Ours)	**91.96**±0.08	**90.83**± 0.05	**89.61**± 0.65	**85.81**± 0.44	**76.43**± 4.88	**92.91**± 0.07	**92.26**± 0.25	**90.73**± 0.34	**85.94**± 0.92	**73.77**± 0.82

Table 2. Comparison with state-of-the-art methods on SVHN and CIFAR-100 datasets. The mean and standard deviation computed over five runs are presented. "IDN-xx%" means the noise rate is xx% and noise type is "IDN".

Method	SVHN					CIFAR-100				
	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE (baseline)	90.47±0.27	89.85±0.16	86.31±0.79	80.59±0.56	64.93±2.03	66.55±0.23	63.94±0.51	61.97±1.16	58.70±0.56	56.63±0.69
GCE [54]	90.82±0.12	89.48±0.66	86.92±0.24	81.95±1.45	63.20±2.75	**69.18**±0.14	**68.35**±0.33	**66.35**±0.13	62.09±0.09	56.68±0.75
DMI [48]	92.66±0.58	91.88±0.42	88.44±0.85	82.27±1.54	68.72±2.32	67.06±0.46	64.72±0.64	62.8±1.46	60.24±0.63	56.52±1.18
Forward [32]	92.01±1.10	90.67±0.27	86.04±0.40	83.18±0.95	70.72±2.00	67.81±0.48	67.23±0.29	65.42±0.63	**62.18**±0.26	58.61±0.44
CoTeaching [11]	91.11±0.16	90.88±0.17	88.21±0.62	86.46±1.33	70.04±1.05	67.91±0.34	67.40±0.44	64.13±0.43	59.98±0.28	57.48±0.740
CoTeaching++ [52]	92.64±0.43	91.59±0.43	87.55±1.26	87.69±1.06	72.36±1.39	68.67±0.25	68.30±0.69	65.77±0.30	61.75±0.53	**57.94**±0.15
JoCor [43]	93.52±0.47	93.47±0.40	89.47±1.04	**88.56**±1.28	73.70±1.92	68.48±0.49	67.87±0.80	65.73±0.55	61.64±0.54	57.75±0.80
PeerLoss [23]	92.59±0.56	91.67±0.72	89.86±0.67	85.44±0.97	73.91±2.30	65.64±1.07	63.83±0.48	61.64±0.67	58.30±0.80	55.41±0.28
TMDNN [50]	95.51±0.13	**94.83**±0.64	92.43±0.91	86.91±1.17	76.53±2.15	68.42±0.42	66.62±0.85	64.72±0.64	59.38±0.65	55.68±1.43
PartT [44]	**95.56**±0.45	94.19±0.20	**92.56**±0.83	88.13±1.56	**77.04**±2.56	67.33±0.33	65.33±0.59	64.56±1.55	59.73±0.76	56.80±1.32
MEIDTM (Ours)	95.72±0.40	95.48±0.01	94.23±0.27	92.00±0.10	78.25±0.35	68.19±0.32	67.21±0.38	66.06±0.77	62.34±0.18	57.69±0.51
kMEIDTM (Ours)	**96.38**±0.07	**95.66**±0.02	**94.68**±0.17	**92.20**±0.23	**80.22**±2.00	**69.88**±0.45	**69.16**±0.16	**66.76**±0.30	**63.46**±0.48	**59.18**±0.16

Ablation study

Table 3. Classification accuracy (%) on the Clothing1M dataset. (*) indicates that the implementation is based on the authors' code.

Methods	CE (Baseline)	GCE [54]	SL [42]	Co-teaching [11]	JointOpt [39]	L_{DMI} [48]	PTD-R-V [44]	ERL [21]
Accuracy	68.94	69.75	71.02	69.21	72.16	72.46	71.67	72.87
Methods	ForwardT [32]	JoCor [43]	CORES [5]	CAL [55]	DivideMix* [18]	MEIDTM(Ours)	kMEIDTM(Ours)	kMEIDTM (+DivideMix)
Accuracy	69.84	70.30	73.24	74.17	74.67	73.05	73.34	**74.82**

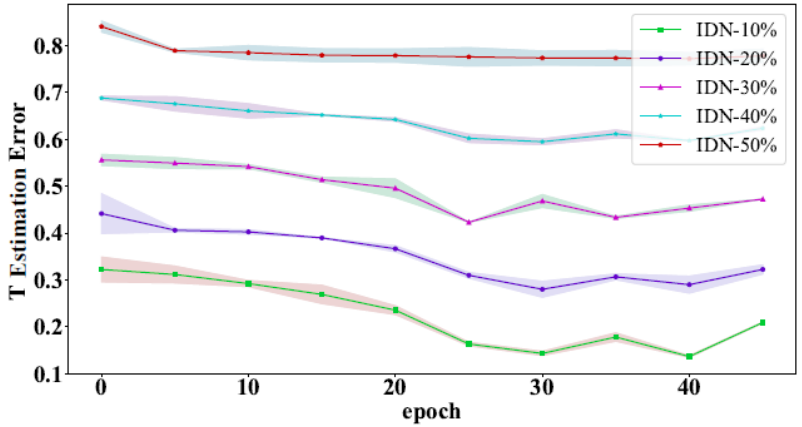


Figure 2. shows the transition matrix estimation error varying with the number of epoches during model training, under five different noise rates, on CIFAR-10 dataset.

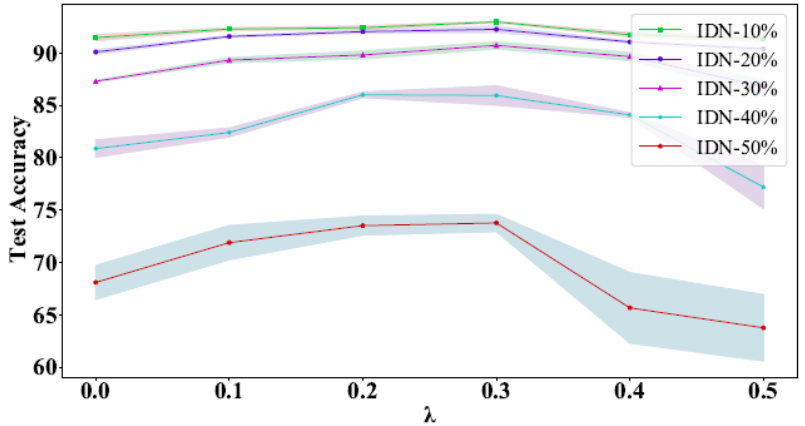


Figure 3. shows the classification accuracy varying with hyper-parameter λ under five different noise rates on CIFAR-10 dataset.

THANKS