DIRICHLET-BASED PER-SAMPLE WEIGHTING BY TRANSITION MATRIX FOR NOISY LABEL LEARNING

HeeSun Bae[1], Seungjae Shin[1], Byeonghu Na[1], Il—Chul Moon [1,2]

KAIST

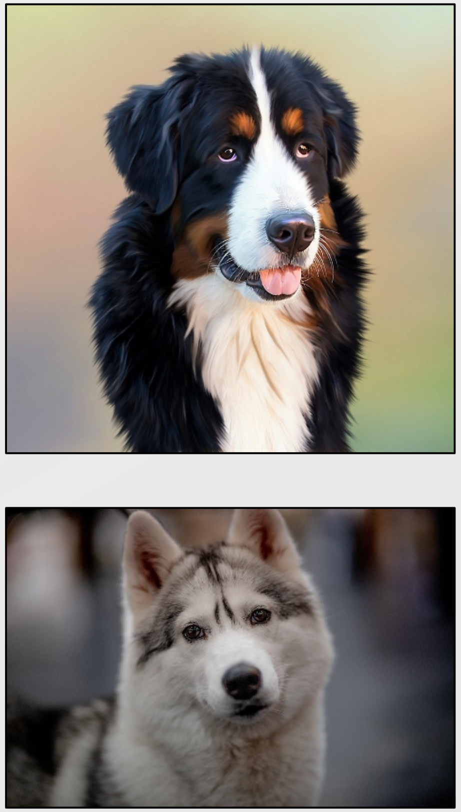
[Paper] [GitHub]

Introduction

- What is “noisy label”?
  - While collecting data, getting high quality annotation can be difficult and expensive => Noisy label
  - How to train the model robustly to the noisy label matters.
  - Example. All below images are labeled as “Cat”

Annotation = cat
True label = cat

(Clean)



Annotation = cat
True label = dog

(Wrong)

Introduction

- What is “noisy label”?
  - While collecting data, getting high quality annotation can be difficult and expensive => Noisy label
  - How to train the model robustly to the noisy label matters.

- Solutions:
  - Sample selection: filter (or remove) noisy sample
  - Label correction: change (or cleanse) noisy label
  - Robust loss modeling: a classifier will converge to the same optimal point with/without noisy label
  - **Transition matrix modeling**
  - …

Introduction

APPLIED ARTIFICIAL INTELLIGENCE LAB

- Solutions:
  - Sample selection: filter (or remove) noisy sample
  - Label correction: change (or cleanse) noisy label
  - Robust loss modeling: a classifier will converge to the same optimal point with/without noisy label
  - Transition matrix modeling
  - …

- What is "Transition matrix"?
  - Definition: The flipping probability of a clean label(Y) to noisy label($\tilde{Y}$)

$$p(\tilde{Y}|x) = \boldsymbol{T}p(Y|x) \text{ with } T_{jk} = p(\tilde{Y} = j|Y = k, x) \ \forall j, k = 1, \dots, C$$

  - Problem: We don't know what $\boldsymbol{T}$ is.
  - Previous methods have focused on **how to estimate $\boldsymbol{T}$** well.

Transition matrix for learning with noisy label

$$p(\tilde{Y}|x) = \boldsymbol{T} p(Y|x) \text{ if } T_{jk} = p(\tilde{Y} = j | Y = k, x) \; \forall j, k = 1, \ldots, C$$

- How to utilize the transition matrix is also important

1. Forward

$P(\tilde{y}|x) = TP(y|x)$ → $\boldsymbol{L}(\boldsymbol{T}\boldsymbol{f}(\boldsymbol{x}), \tilde{\boldsymbol{y}})$

   - Empirically, the classifier trained with forward loss can be different from true classifier

2. Backward

$\mathrm{T}^{-1}P(\tilde{y}|x) = P(y|x)$ → $\boldsymbol{T}^{-1}\boldsymbol{L}(\boldsymbol{f}(\boldsymbol{x}), \tilde{\boldsymbol{y}})$

   - Unstable performance

3. **Reweighting**

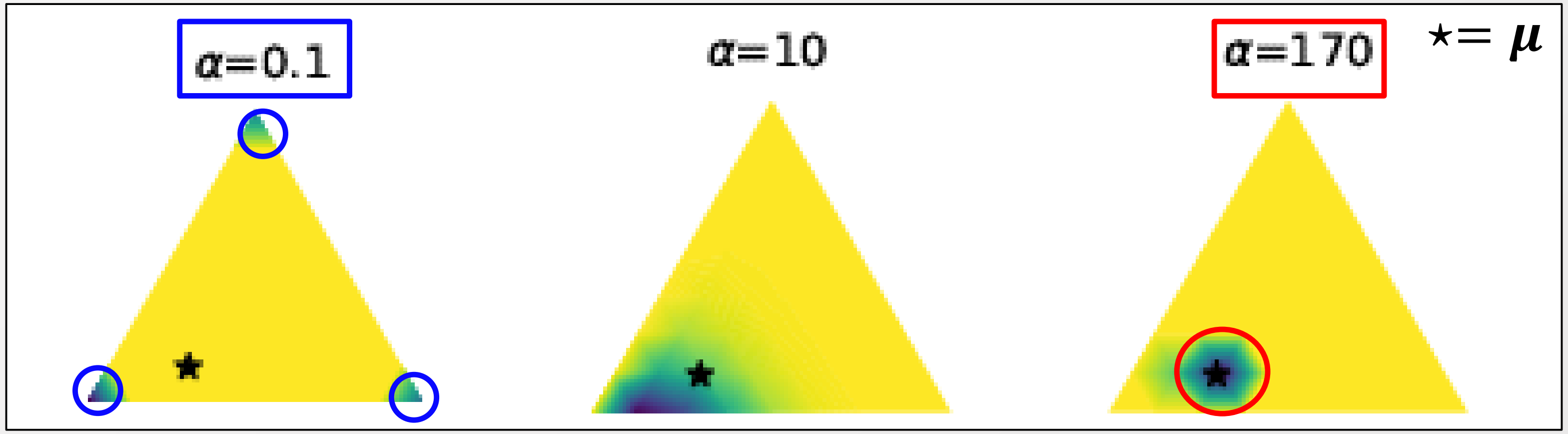
$P(\tilde{y}|x) = TP(y|x)$ → $\left(\frac{\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{x})}{\boldsymbol{T}\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{x})}\right) \cdot \boldsymbol{L}(\boldsymbol{f}(\boldsymbol{x}), \tilde{\boldsymbol{y}})$

   - The true weight $\left(\frac{\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{x})}{\boldsymbol{T}\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{x})}\right)$ is still inaccessible

- $\boldsymbol{L}$: Cross entropy
- $\boldsymbol{f}$: Model (Classifier), $\tilde{\boldsymbol{y}}$: (Sampled) noisy label. Noisy label data

DWS: Dirichlet-Based Per-sample Weight Sampling

- Dirichlet-based Weight Sampling
  - Properties of Dirichlet distribution
    - When $\boldsymbol{\alpha} \to \mathbf{0}$, the sampled vector is skewed to one specific dimension. E.g. [1,0,0]
    - When $\boldsymbol{\alpha} \to \infty$, vectors are sampled in the near region to the mean vector. E.g. [0.7,0.2,0.1]



[Density plot of $Dir(\alpha\boldsymbol{\mu})$ with different α. $\boldsymbol{\mu} = [\mathbf{0.7}, \mathbf{0.2}, \mathbf{0.1}]$]

DWS: Dirichlet-Based Per-sample Weight Sampling

f: Model (Classifier)
y: clean label
$\tilde{y}$: noisy label

- Dirichlet-based Weight Sampling
  - Properties of Dirichlet distribution
    - When $\boldsymbol{\alpha} \to \mathbf{0}$, the sampled vector is skewed to one specific dimension. E.g. [1,0,0]
    - When $\boldsymbol{\alpha} \to \infty$, vectors are sampled in the near region to the mean vector. E.g. [0.7,0.2,0.1]

- Suggest a loss function that can **integrate both reweighting and resampling**
  - Reweighting loss function$\left(R_{l,RW}^{emp}\right) := \frac{1}{\boldsymbol{N}}\sum_{i=1}^{\boldsymbol{N}} \frac{f_\theta(x_i)_{\tilde{y}_i}}{(Tf_\theta(x_i))_{\tilde{y}_i}} l(f_\theta(x_i), \tilde{y}_i)$
  - Resampling loss function$\left(R_{l,RENT}^{emp}\right) := \frac{1}{\boldsymbol{M}}\sum_{i=1}^{\boldsymbol{M}} l(f_\theta(x_i), \tilde{y}_i)$
    - Note the number of samples changed (sampling)
    - will be explained later in more details
  - Both reweighting and resampling can be expressed by modifying α value.

$$R_{l,DWS}^{emp} := \frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{N} w_i^j\, l(f_\theta(x_i), \tilde{y}_i), \qquad with\ \mathbf{w^j} \sim Dir(\alpha\boldsymbol{\mu})$$

DWS: Dirichlet-Based Per-sample Weight Sampling

APPLIED ARTIFICIAL INTELLIGENCE LAB

- Support explanations on why resampling is better than reweighting
  - Variance Analysis: Smaller α means variance increase with regard to the risk function
    - Variance increase can improve robustness for learning with noisy label

$$V\left(R_{l,DWS}^{emp}\right) = \frac{1}{M^2}\sum_{j=1}^{M}\left(\sum_{i=1}^{N} l(f_\theta(x_i),\tilde{y}_i)^2 V\left(w_i^j\right) + \sum_{k\neq i} Cov\left(w_i^j, w_k^j\right)\right), V\left(w_i^j\right) = \frac{\mu_i(1-\mu_i)}{\alpha+1} \text{ and } Cov\left(w_i^j, w_k^j\right) = -\frac{\mu_i\mu_k}{\alpha+1}$$

- Variance and Covariance are defined as such by the definition of the Dirichlet distribution.
- Since μ is a scalar value, it does not affect the variance.

DWS: Dirichlet-Based Per-sample Weight Sampling

- Support explanations on why resampling is better than reweighting
  - Distance from the true weight
    - Let $\widetilde{\mu_i}^* = \frac{p(Y=\tilde{y}_i|x_i)}{p(\tilde{Y}=\tilde{y}_i|x_i)}$ (true weight) and $\boldsymbol{\mu}^*$ = normalized vector of $\widetilde{\mu_i}^*$
    - While training, we cannot know $\boldsymbol{\mu}^*$ => It should be approximated from the output of the training classifier
    - $\boldsymbol{\mu}^*$ approximation error => the risk function statistical consistency is not approved
    - Smaller $\boldsymbol{\alpha}$ => smaller mahalanobis distance between $\boldsymbol{\mu}^*$ and $\frac{1}{M}\sum_{j=1}^{M}\boldsymbol{w}^j$

$$d_M\left(\boldsymbol{\mu}^*, \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{w}^j\right) = \sqrt{(\boldsymbol{\mu}^* - \boldsymbol{\mu})^{\boldsymbol{T}}\left(\frac{\Sigma}{M}\right)^{-1}(\boldsymbol{\mu}^* - \boldsymbol{\mu})} = \sqrt{M(\boldsymbol{\alpha} + 1)(\boldsymbol{\mu}^* - \boldsymbol{\mu})^{\boldsymbol{T}} S^{-1}(\boldsymbol{\mu}^* - \boldsymbol{\mu})}$$

- $S = (\alpha + 1)\Sigma$

DWS: Dirichlet-Based Per-sample Weight Sampling

- Support explanations on why resampling is better than reweighting
  - Noise injection impact
    - Injecting random noise to label **increases robustness** against label noise
    - $R_{l,DWS}^{emp}$ can be interpreted as injecting noise (following normal distribution) to label during training
    - With smaller $\boldsymbol{\alpha}$, the noise injection amount increases

$$\lim_{N \to \infty} R_{l,DWS}^{emp} = \sum_{i=1}^{N} \mu_i l(f_\theta(x_i), \tilde{y}_i) + \sum_{i=1}^{N} z_i l(f_\theta(x_i), \tilde{y}_i), z_i \sim \mathcal{N}(0, \frac{\mu_i(1-\mu_i)}{M(\boldsymbol{\alpha}+1)})$$

DWS: Dirichlet-Based Per-sample Weight Sampling

- RENT: RESAMPLE FROM NOISE TRANSITION => Importance Sampling based **Resampling technique**

Algorithm 1: REsampling utilizing the Noise Transition matrix (RENT)

Input: Dataset $\tilde{D} = \{x_i, \tilde{y}_i\}_{i=1}^N$, classifier f_θ, Transition matrix T, Resampling budget M

Output: Updated f_θ

while f_θ *not converge* **do**

  Get $\tilde{\mu}_i = f_\theta(x_i)_{\tilde{y}_i} / (T f_\theta(x_i))_{\tilde{y}_i}$ for all i

  Construct Categorical distribution $\pi_N = \text{Cat}(\frac{\tilde{\mu}_1}{\sum_{l=1}^N \tilde{\mu}_l}, \dots \frac{\tilde{\mu}_N}{\sum_{l=1}^N \tilde{\mu}_l})$

  Independently sample $(x_1, \tilde{y}_1), \dots, (x_M, \tilde{y}_M)$ from π_N

  Update f_θ by $\theta \leftarrow \theta - \nabla_\theta \frac{1}{M} \sum_{j=1}^M l(f_\theta(x_j), \tilde{y}_j)$

end

- Per-sample weight $\left(= \frac{\boldsymbol{P(y|x)}}{\boldsymbol{TP(y|x)}}\right)$ calculation
  - The true weight is inaccessible
  - $\boldsymbol{P(y_i|x_i)}$ is approximated as $\boldsymbol{f_\theta(x_i)}_{\tilde{y}_i}$
  - => $\tilde{\mu}_i = f_\theta(x_i)_{\tilde{y}_i} / (T f_\theta(x_i))_{\tilde{y}_i}$

DWS: Dirichlet-Based Per-sample Weight Sampling

- RENT: RESAMPLE FROM NOISE TRANSITION => Importance Sampling based **Resampling technique**

Algorithm 1: REsampling utilizing the Noise Transition matrix (RENT)

Input: Dataset $\tilde{D} = \{x_i, \tilde{y}_i\}_{i=1}^N$, classifier f_θ, Transition matrix T, Resampling budget M

Output: Updated f_θ

while f_θ *not converge* **do**

  Get $\tilde{\mu}_i = f_\theta(x_i)_{\tilde{y}_i} / (T f_\theta(x_i))_{\tilde{y}_i}$ for all i

  Construct Categorical distribution $\pi_N = \text{Cat}(\frac{\tilde{\mu}_1}{\sum_{l=1}^N \tilde{\mu}_l}, \dots \frac{\tilde{\mu}_N}{\sum_{l=1}^N \tilde{\mu}_l})$

  Independently sample $(x_1, \tilde{y}_1), \dots, (x_M, \tilde{y}_M)$ from π_N

  Update f_θ by $\theta \leftarrow \theta - \nabla_\theta \frac{1}{M} \sum_{j=1}^M l(f_\theta(x_j), \tilde{y}_j)$

end

- Categorical distribution (π_N) construction
  - Where the parameter of π_n is from?
    - $R_l(f_\theta) = \mathbb{E}_{(x,y)\sim p(X,Y)}[l(f_\theta(x), y)] = \mathbb{E}_{(x,y)\sim p(X,\tilde{Y})}\left[l(f_\theta(x), y)\frac{p(x,Y=\tilde{y})}{p(x,\tilde{Y}=\tilde{y})}\right]$ — Importance sampling

$$= \mathbb{E}_{(x,y)\sim p(X,\tilde{Y})}\left[l(f_\theta(x), y)\frac{p(Y=\tilde{y}|x)p(x)}{p(\tilde{Y}=\tilde{y}|x)p(x)}\right] = \mathbb{E}_{(x,y)\sim p(X,\tilde{Y})}\left[l(f_\theta(x), y)\frac{p(Y=\tilde{y}|x)}{p(\tilde{Y}=\tilde{y}|x)}\right]$$

$p(x)$ is same according to the problem setting

$$= \mathbb{E}_{(x,y)\sim p(X,\tilde{Y})}\left[\frac{p(Y=\tilde{y}|x)}{p(\tilde{Y}=\tilde{y}|x)} l(f_\theta(x), y)\right]$$

Per sample weight

DWS: Dirichlet-Based Per-sample Weight Sampling

- RENT: RESAMPLE FROM NOISE TRANSITION => Importance Sampling based **Resampling technique**

Algorithm 1: REsampling utilizing the Noise Transition matrix (RENT)

Input: Dataset $\tilde{D} = \{x_i, \tilde{y}_i\}_{i=1}^N$, classifier f_θ, Transition matrix T, Resampling budget M

Output: Updated f_θ

while f_θ *not converge* **do**

  Get $\tilde{\mu}_i = f_\theta(x_i)_{\tilde{y}_i} / (T f_\theta(x_i))_{\tilde{y}_i}$ for all i

  Construct Categorical distribution $\pi_N = \text{Cat}(\frac{\tilde{\mu}_1}{\sum_{l=1}^N \tilde{\mu}_l}, \dots \frac{\tilde{\mu}_N}{\sum_{l=1}^N \tilde{\mu}_l})$

  Independently sample $(x_1, \tilde{y}_1), \dots, (x_M, \tilde{y}_M)$ from π_N

  Update f_θ by $\theta \leftarrow \theta - \nabla_\theta \frac{1}{M} \sum_{j=1}^M l(f_\theta(x_j), \tilde{y}_j)$

end

- Resampling: From π_N, independently resample dataset
  - If $\tilde{\mu}_i = \tilde{\mu}_i^*$, $R_{l,RENT}^{emp}$ is statistically consistent to R_l

Experiment

- Classification performance
  - Training dataset include noisy label // Test on clean label dataset
  - SN/ASN = arbitrary noisy label included (%=noisy label ratio)
  - Base = How the transition matrix is estimated (CE is cross entropy. Not treating the noisy label)
  - w/XXX = How to utilize the transition matrix

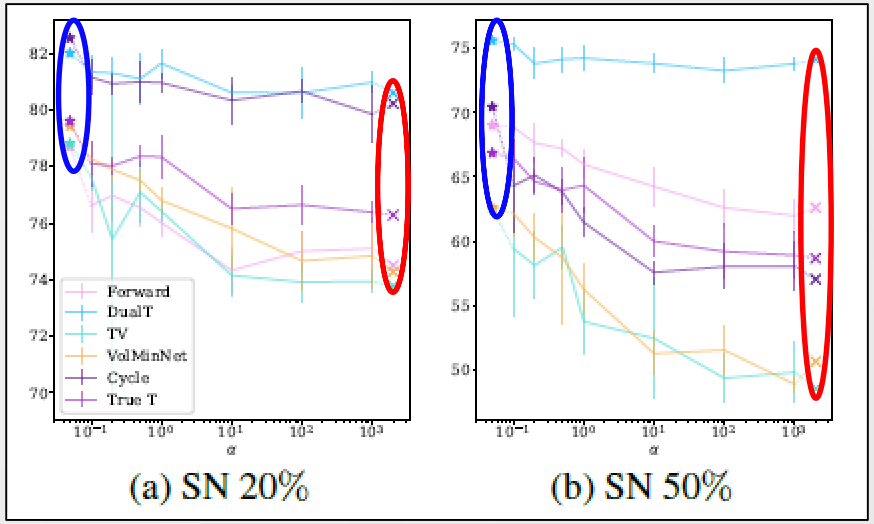
Base	Risk	CIFAR10 SN 20%	CIFAR10 SN 50%	CIFAR10 ASN 20%	CIFAR10 ASN 40%	CIFAR100 SN 20%	CIFAR100 SN 50%	CIFAR100 ASN 20%	CIFAR100 ASN 40%
CE	✗	73.4±0.4	46.6±0.7	78.4±0.2	69.7±1.3	33.7±1.2	18.5±0.7	36.9±1.1	27.3±0.4
Forward	w/ FL	73.8±0.3	58.8±0.3	79.2±0.6	74.2±0.5	30.7±2.8	15.5±0.4	34.2±1.2	25.8±1.4
	w/ RW	74.5±0.8	62.6±1.0	79.6±1.1	73.1±1.7	37.2±2.6	23.5±11.3	27.2±13.2	27.3±1.3
	w/ RENT	78.7±0.3	69.0±0.1	82.0±0.5	77.8±0.5	38.9±1.2	28.9±1.1	38.4±0.7	30.4±0.3
DualT	w/ FL	79.9±0.5	71.8±0.3	82.9±0.2	77.7±0.6	35.2±0.4	23.4±1.0	38.3±0.4	28.4±2.6
	w/ RW	80.6±0.6	74.1±0.7	82.5±0.2	77.9±0.4	38.5±1.0	12.0±12.5	38.5±1.6	24.0±11.6
	w/ RENT	82.0±0.2	74.6±0.4	83.3±0.1	80.0±0.9	39.8±0.9	27.1±1.9	39.8±0.7	34.0±0.4
TV	w/ FL	74.0±0.5	50.4±0.6	78.1±1.3	71.6±0.3	34.5±1.4	21.0±1.4	33.9±3.6	28.7±0.8
	w/ RW	73.7±0.9	48.5±4.1	77.3±2.0	70.2±1.0	32.3±1.0	17.8±2.0	32.0±1.5	23.2±0.9
	w/ RENT	78.8±0.8	62.5±1.8	81.0±0.4	74.0±0.5	34.0±0.9	20.0±0.6	34.0±0.2	25.5±0.4
VolMinNet	w/ FL	74.1±0.2	46.1±2.7	78.8±0.5	69.5±0.3	29.1±1.5	25.4±0.8	22.6±1.3	14.0±0.9
	w/ RW	74.2±0.5	50.6±6.4	78.6±0.5	70.4±0.8	36.9±1.2	24.4±3.0	34.9±1.3	26.5±0.9
	w/ RENT	79.4±0.3	62.6±1.3	80.8±0.5	74.0±0.4	35.8±0.9	29.3±0.5	36.1±0.7	31.0±0.8
Cycle	w/ FL	81.6±0.5	–	82.8±0.4	54.3±0.3	39.9±2.8	–	39.4±0.2	31.3±1.2
	w/ RW	80.2±0.2	57.0±3.4	78.1±0.9	70.6±1.1	37.8±2.7	30.2±0.6	38.1±1.6	29.3±0.6
	w/ RENT	82.5±0.2	70.4±0.3	81.5±0.1	70.2±0.7	40.7±0.4	32.4±0.4	40.7±0.7	32.2±0.6
True T	w/ FL	76.7±0.2	57.4±1.3	75.0±11.9	70.7±8.6	34.3±0.5	22.0±1.5	35.8±0.5	31.9±1.0
	w/ RW	76.2±0.3	58.6±1.2	–	–	35.0±0.8	21.8±0.8	21.3±16.6	21.6±10.4
	w/ RENT	79.8±0.2	66.8±0.6	82.4±0.4	78.4±0.3	36.1±1.1	24.0±0.3	34.4±0.9	27.2±0.6

Base	Risk	CIFAR-10N Aggre	CIFAR-10N Ran1	CIFAR-10N Ran2	CIFAR-10N Ran3	CIFAR-10N Worse	Clothing1M -
CE	✗	80.8±0.4	75.6±0.3	75.3±0.4	75.6±0.6	60.4±0.4	66.9±0.8
Forward	w/ FL	79.6±1.8	76.1±0.8	76.4±0.4	76.0±0.2	64.5±1.0	67.1±0.1
	w/ RW	80.7±0.5	75.8±0.3	76.0±0.5	75.8±0.6	63.9±0.7	66.8±1.1
	w/ RENT	80.8±0.8	77.7±0.4	77.5±0.4	77.2±0.6	68.0±0.9	68.2±0.6
DualT	w/ FL	81.9±0.2	79.4±0.4	79.3±1.0	79.4±0.4	72.1±0.9	68.2±1.0
	w/ RW	81.8±0.4	79.8±0.2	79.4±0.6	79.6±0.4	71.4±1.0	68.5±0.4
	w/ RENT	82.0±1.2	80.5±0.5	80.4±0.7	80.5±0.6	73.5±0.7	69.9±0.7
TV	w/ FL	80.5±0.7	76.4±0.4	76.2±0.5	76.1±0.1	60.2±5.2	66.7±0.3
	w/ RW	80.7±0.4	75.8±0.6	75.2±1.1	75.4±1.5	62.3±2.9	67.4±0.5
	w/ RENT	81.0±0.4	77.4±0.6	77.8±1.0	76.7±0.4	66.9±3.1	68.1±0.4
VolMinNet	w/ FL	80.9±0.3	76.3±0.5	75.9±0.7	75.9±0.6	61.8±1.3	65.0±0.1
	w/ RW	80.7±0.6	76.2±0.5	75.5±0.8	75.5±0.2	63.0±3.2	66.6±0.1
	w/ RENT	81.3±0.4	77.6±1.0	77.7±0.3	77.2±0.7	66.9±0.5	67.7±0.3

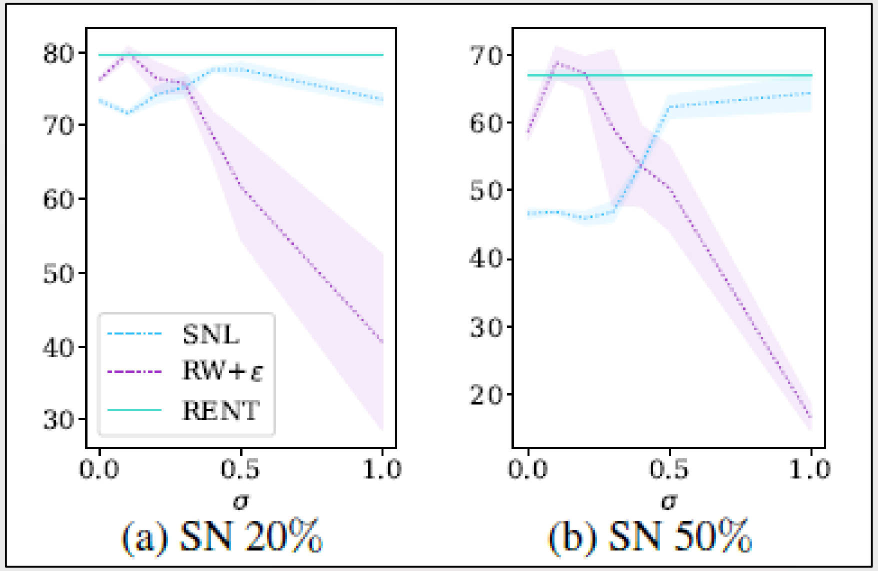
  - How to utilize the transition matrix is also important for the model performance, and RENT shows the best
  - RENT improves various baselines consistently

Experiment

- (DWS) α impact
  - ★(RENT) vs. ×(ReWeighting)
  - Lines are test accuracies with diverse α values.
  - Colors represent baselines to estimate the transition matrix.
  - ★ **shows the best performance**



- Noise injection impact of RENT
  - Risk functions
    - SNL $= \sum_{i=1}^{N} l(f_\theta(x_i), \tilde{y}_i) + \sigma \sum_{i=1}^{N} \sum_{k=1}^{C} z_{ik} l(f_\theta(x_i), k)$, $z_{ik} \sim \mathcal{N}(0,1)$
    - RW+ϵ $= \sum_{i=1}^{N} \mu_i l(f_\theta(x_i), \tilde{y}_i) + \sigma \sum_{i=1}^{N} \sum_{k=1}^{C} z_{ik} l(f_\theta(x_i), k)$, $z_{ik} \sim \mathcal{N}(0,1)$
    - RENT$= \sum_{i=1}^{N} \mu_i l(f_\theta(x_i), \tilde{y}_i) + \sum_{i=1}^{N} z_i l(f_\theta(x_i), \tilde{y}_i)$, $z_i \sim \mathcal{N}(0, \frac{\mu_i(1-\mu_i)}{M})$
  - RENT consistently shows better or comparable performance over SNL or RW+ϵ with regard to hyperparameter(σ)



Conclusion

- We first decompose the training procedure for noisy label classification with the label transition m atrix T as estimation and utilization, underscoring the importance of adequate utilization.

- We present an alternative utilization of the label transition matrix T by resampling, RENT.
  - RENT ensures the statistical consistency of risk function to the true risk for data resampling by utilizing T.
  - Yet it supports more robustness to noisy label (empirically shows good performance).

- We interpret resampling and reweighting in one framework through Dirichlet distribution-based pe r-sample Weight Sampling (DWS).
  - Integrating resampling and reweighting
  - analyzing the success of resampling over reweighting in learning with noisy label.

- Diverse experiments show consistent improvements over the existing T utilization methods.