SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

February 2026, Vol. 69, Iss. 2, 122105:1–122105:17
https://doi.org/10.1007/s11432-025-4576-9

TransTS: an adaptive post-hoc method for probability calibration under label noise

Yuefei WU[1,2,4], Bin SHI[1,2,4*], Bo DONG[2,3,4] & Qinghua ZHENG[1,2,4]

[1]*School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China*
[2]*Ministry of Education Key Laboratory of Intelligent Networks and Network Security, Xi'an 710049, China*
[3]*School of Distance Education, Xi'an Jiaotong University, Xi'an 710049, China*
[4]*Shaanxi Province Key Laboratory of Big Data Knowledge Engineering, Xi'an 710049, China*

Received 12 May 2025/Revised 15 July 2025/Accepted 28 August 2025/Published online 4 January 2026

Abstract Probability calibration aims to align the classifier's confidence with the true likelihood of correctness (i.e., empirical frequencies). Existing calibration methods perform well when labels are clean. However, when labels are corrupted, calibration deterioration is almost inevitable. In this paper, we study probability calibration under label noise. Specifically, we first observe that existing calibration methods struggle to maintain calibration quality in the presence of label noise. Second, we reveal that label noise leads to calibration deterioration due to the failure of the calibration equation, which further results in over-calibration for temperature scaling (TS). To address this issue, we propose adaptive transitional temperature scaling (TransTS), which adaptively scales the logits according to the noise level. TransTS constructs a consistent calibrator that can converge to its counterpart trained on clean data, and we provide theoretical justifications for this property. As a general post-hoc method, TransTS can be easily integrated with any pre-trained model. Results on a variety of experimental cases show that TransTS outperforms five built-in methods and eleven post-hoc methods, as well as several widely used learning with noisy label (LNL) methods.

Keywords probability calibration, label noise, uncertainty representation, trustworthy decision

Citation Wu Y F, Shi B, Dong B, et al. TransTS: an adaptive post-hoc method for probability calibration under label noise. Sci China Inf Sci, 2026, 69(2): 122105, https://doi.org/10.1007/s11432-025-4576-9

1 Introduction

Deep neural networks (DNNs) have achieved great success in many fields, including natural language processing [1], computer vision [2], and data mining [3], among others. For real-world applications, however, DNN classifiers are expected not only to be accurate but also to know when they may be wrong [4, 5]. For example, when a medical diagnostic system makes predictions with low confidence, it should defer to human doctors for takeover. Unfortunately, it has been shown that modern DNN classifiers tend to be grossly overconfident [6–8]. This hinders the deployment of DNNs in high-risk scenarios, such as medical diagnostics [9], autonomous driving [10], and financial fraud detection [11].

Given the importance of this issue, probability calibration has been proposed. Probability calibration provides statistical guarantees on the predictions, ensuring that the predicted confidence aligns with the true likelihood of correctness [12–14]. For example, if a classifier is well calibrated, among 100 predictions assigned a 70% confidence level, 70 of them should be correctly classified. Existing calibration methods fall broadly into two categories: (1) post-hoc methods, which scale the predicted probabilities after training [12, 15–19]; (2) built-in methods, which calibrate the network during training through designed loss and regularization terms [6–8, 20–23]. These methods have achieved impressive calibration performance when the labels are clean. However, the assumption for clean labels may not always hold in real-world practice. Due to the challenges in collecting high-quality data, annotations are often obtained through substandard methods such as web scraping, crowdsourcing, or surveys, making label noise inevitable. Existing literature shows that the proportion of label noise in real-world datasets ranges from 8.0% to 38.5% [24]. Such label noise could potentially harm probability calibration, further posing challenges for deploying DNN classifiers in real-world applications.

A possible solution is to leverage existing learning with noisy label (LNL) techniques [2, 25–29]. For instance, one can first apply LNL methods to denoise the dataset and subsequently perform probability calibration. Although

* Corresponding author (email: shibin@xjtu.edu.cn)

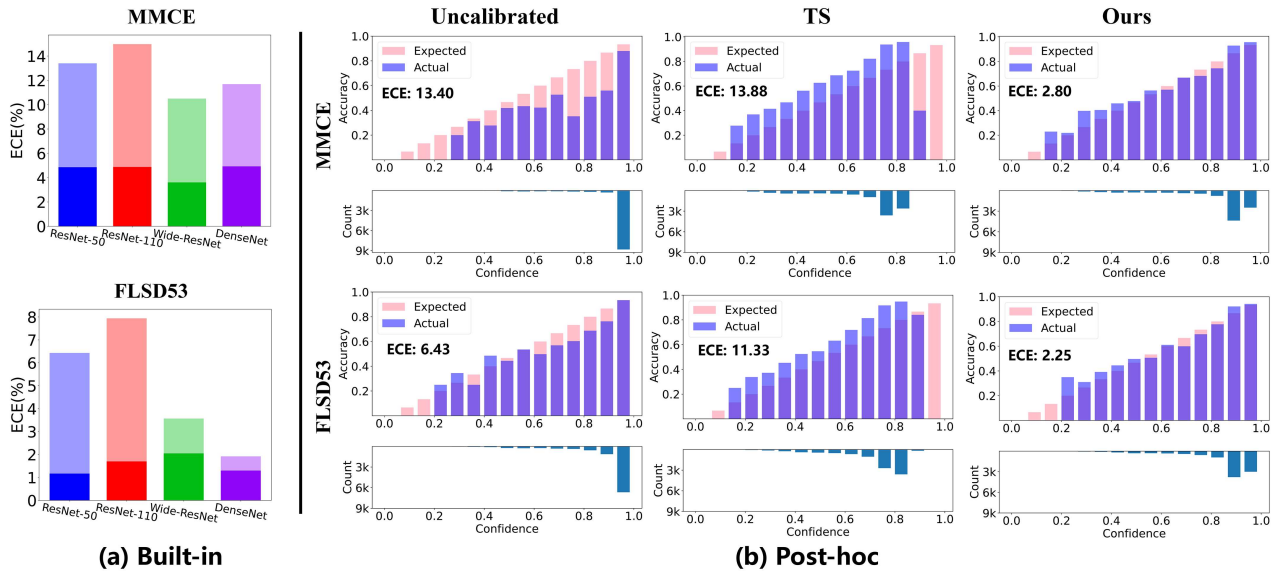

Figure 1 (Color online) Label noise causes existing calibration methods to fail. (a) Four network architectures are trained on CIFAR-10 with built-in calibration methods. The smaller the ECE, the better. Dark for clean, light for noisy. (b) Reliability diagram for post-hoc calibration methods. All cases are run on CIFAR-10 with 20% symmetry noise.

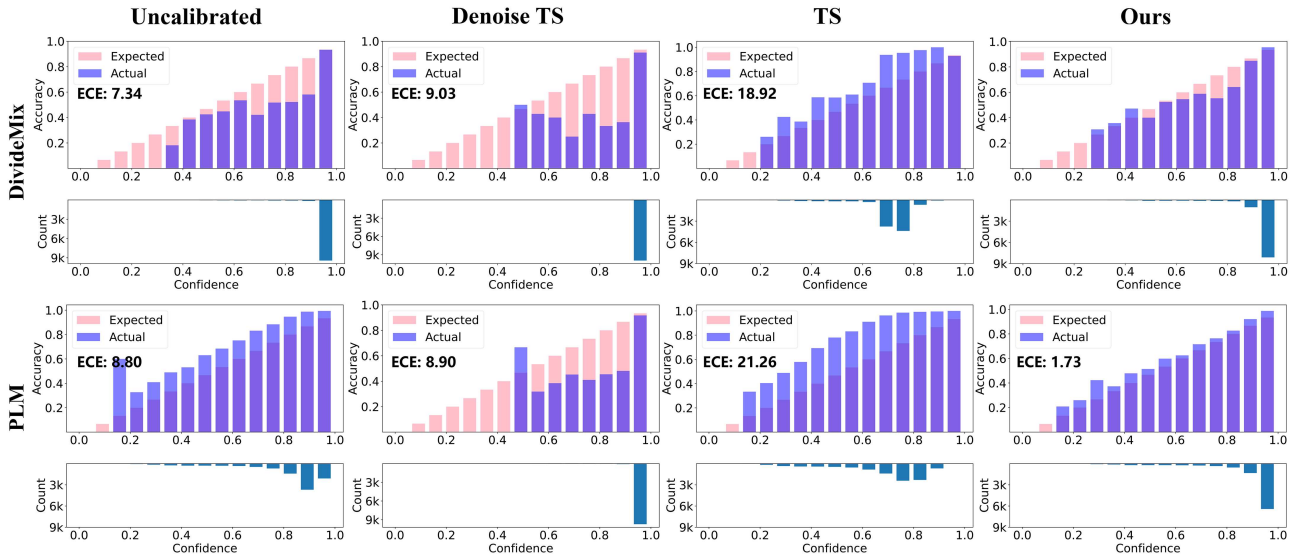

Figure 2 (Color online) Reliability diagram for denoise methods: DivideMix (Acc: 94.83%) and PLM (Acc: 90.40%). Denoise TS refers to denoising the validation set firstly and then applying TS. All cases are run on CIFAR-10 with 20% symmetry noise.

label noise can be reduced through LNL, it is rarely completely eliminated. Residual label noise can still hinder the calibration methods, making calibration poor. To investigate how label noise impacts calibration, we conducted an empirical study. Figure 1(a) shows that calibration performance significantly deteriorates under label noise for two built-in calibration methods. When temperature scaling (TS) [12]—a popular post-hoc method—is used, it can even shift the model from over-confidence to under-confidence, as shown in Figure 1(b). Moreover, Figure 2 shows that even the widely used LNL methods also fail to effectively calibrate the model. Notably, the Denoise TS is even counterintuitively more over-confident. This occurs because Denoise TS, which aims to minimize the empirical risk between output probabilities and their predicted labels, pushes the output probabilities further toward a one-hot distribution, leading to increased over-confidence. Compared to well-calibrated confidence, under-confidence requires additional caution or backup plans, which will lose user trust. In downstream decision-making, both overconfident and underconfident predictions are problematic [8]. This motivates the central question addressed in this work: how to calibrate DNN classifiers under label noise?

In this paper, we investigate the open problem of probability calibration under label noise. Specifically, we observe

that label noise can deteriorate the calibration performance of both built-in and post-hoc methods. Furthermore, we argue that label noise compels TS to learn an overly aggressive scaling factor T. This aggressive scaling factor T increases the probabilities assigned to incorrect labels, thereby minimizing the negative log-likelihood (NLL) loss. As a result, under label noise, TS shifts the predicted distribution from being excessively sharp (over-confidence) to overly smooth (under-confidence)—a phenomenon we refer to as over-calibration.

To calibrate DNN classifiers under label noise, we propose a simple yet effective post-hoc calibration method called adaptive transitional temperature scaling (TransTS). First, TransTS introduces a noise transition matrix, which adaptively scales the output probabilities, thereby sharing part of the burden from the temperature scaling factor. This dual-parameter mechanism enables the temperature scaling factor to be moderate, alleviating over-calibration while preserving model performance. Second, we also reveal that TransTS essentially constructs a consistent calibrator that converges to its counterpart learned on clean data. Finally, we show through experiments on a variety of image classification datasets and noisy settings that TransTS is better calibrated than various built-in and post-hoc baselines, as well as several widely used LNL methods. Moreover, on clean datasets, TransTS maintains performance comparable to existing calibration methods. Our main contributions can be summarized as follows.

(1) We analyze the limitations of existing calibration methods under label noise. Specifically, we theoretically show that widely used TS suffers from over-calibration in the presence of noisy labels.

(2) We propose a TransTS method, supported by theoretical analysis. We derive a theoretical upper bound for TransTS, showing that it converges to the calibrator learned on a clean dataset.

(3) Extensive experiments demonstrate that our method achieves superior calibration performance across various noise types and noise levels.

2 Preliminaries

2.1 Problem formulation

The scope we study in this paper is supervised multiclass classification with DNNs. Let $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote a dataset consisting of N samples drawn from a joint distribution $\pi(X, Y) = \pi(Y|X)\pi(X)$, where each sample $\mathbf{x}_i \in X \subseteq \mathbb{R}^d$ is the input and $y_i \in Y = \{1, 2, \ldots, K\}$ is the given class label. However, the class labels may be corrupted in the real-world. In this paper, we denote the joint distribution as $\pi(X, \bar{Y}) = \pi(\bar{Y}|X)\pi(X)$ and the noisy labels by $\bar{y}_i \in \bar{Y} = \{1, 2, \ldots, K\}$. Let $f : \mathbb{R}^d \to \mathbb{R}^n$ denote a non-probabilistic K-way classifier parameterized by θ and $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$ be the logits that f predicts on a given input $\mathbf{x}_i$, where $\mathbf{z}_i = [z_{i1}, z_{i2}, \ldots, z_{ik}]^\top$. Then, the vector $\mathbf{p}_i = \mathrm{softmax}(\mathbf{z}_i)$ represents the predicted probability distribution for input $\mathbf{x}_i$, i.e., $\mathbf{p}_i = [p_{i1}, p_{i2}, \ldots, p_{iK}]^\top$. The class that f predicts for $\mathbf{x}_i$ is computed as $\hat{y}_i = \mathrm{argmax}_{\bar{y} \in \bar{Y}}\ \mathbf{p}_i$, and the predicted confidence as $\hat{p}_i = \max_{\bar{y} \in \bar{Y}}\ \mathbf{p}_i$.

In this paper, following the general setting in learning with noisy labels [2,25,26], the training set and validation set contain noisy labels and are assumed to be i.i.d. from the same distribution $\pi(X, \bar{Y})$, whereas the test set is clean and comes from distribution $\pi(X, Y)$. We have $\pi^{tr}(X, \bar{Y}) = \pi^{val}(X, \bar{Y}) \neq \pi^{te}(X, Y)$. Note that the test set serves as the ground truth and gold standard to ensure accurate and unbiased evaluation. This constitutes a general evaluation paradigm in machine learning. If this gold standard is corrupted (e.g., due to mislabeled data), the evaluation results (e.g., accuracy, ECE) will be incorrect. In other words, the test set may contain samples with noisy features to reflect real-world conditions, but the labels must remain correct to accurately assess how well the model performs. Therefore, the test set used in this work is required to be free of label noise.

2.2 Probability calibration

Calibration equation. A DNN classifier is said to be perfectly calibrated if, for each sample $(\mathbf{x}, y) \in D$, the predicted confidence p equals the model accuracy $P(\hat{y} = \tilde{y}|\hat{p})$ on samples with that confidence p. The formal definition of perfect calibration is given by [5,8,12]

$$P(\hat{y} = \tilde{y}|\hat{p} = p) = p, \quad \forall p \in [0, 1], \tag{1}$$

where $\tilde{y}$ represents the given class label, and p denotes predicted confidence. The $\tilde{y}$ may be a noisy label in this paper. The $P(\hat{y} = \widetilde{y}|\hat{p} = p)$ on the left side of the equal sign represents the proportion of samples whose predicted label $\hat{y}$ is equal to the given label $\widetilde{y}$ in the samples with confidence of exactly p, whose statistical interpretation is empirical frequency. Eq. (1) aligns the confidence of the DNN classifier and the true correctness probability (i.e., empirical frequency), which is used to guide the model to achieve perfect calibration.

Temperature scaling (TS). TS [12], one of the most user-friendly and effective post-hoc calibration methods that preserves accuracy under the clean-label assumption, has attracted sustained research attention in the latest important literature [30, 31]. TS adjusts the sharpness of the output probability distribution via the temperature scaling factor T in the Softmax function

$$\hat{p}_{ij}^{c} = \frac{\exp(z_{ij}/T^*)}{\sum_{k=1}^{K}\exp(z_{ik}/T^*)}. \tag{2}$$

The T^* value is obtained by minimizing the NLL loss between the logit vectors $\mathbf{z}$ scaled by T and the given class label on the held-out validation set

$$T^* = \arg\min_{T} \mathbb{E}_{x,y\sim\pi^{val}(x,\bar{y})}\ \mathcal{L}_{\mathrm{NLL}}\left(\sigma(\mathbf{z}/T), \bar{y}\right), \tag{3}$$

where $\sigma(\cdot)$ is the softmax function.

Expected calibration error (ECE). The ECE [32] is defined as the expected absolute difference between the model's confidence and its accuracy, i.e., $\mathbb{E}_{\hat{p}}\big[\,|\mathbb{P}(\hat{y}=y|\hat{p})-\hat{p}|\,\big]$. However, in discrete classification tasks, this continuous definition cannot be directly computed due to finite samples. In practice, predictions are partitioned into M equally spaced confidence bins, and the ECE is estimated as the weighted average of the differences between accuracy and confidence across these bins

$$\mathrm{ECE} = \sum_{m=1}^{M}\frac{|B_m|}{n}\left|A_m - C_m\right|, \tag{4}$$

where $A_m = \frac{1}{|B_m|}\sum_{j\in B_m}1\left(\hat{y}_j = y_j\right)$ and $C_m = \frac{1}{|B_m|}\sum_{j\in B_m}\hat{p}_j$ denote the prediction accuracy and average confidence level within the m-th bin, respectively.

2.3 Noise transition matrix

The noise transition matrix $T(\mathbf{x})$ serves as a bridge from the noisy class posterior $P(\bar{Y}|\mathbf{x})$ to the clean class posterior $P(Y|\mathbf{x})$ [33–35]. Specifically, the ij-th element of the transition matrix, $T_{ij}(\mathbf{x}) = P(\bar{Y}=j|Y=i, X=\mathbf{x})$, denotes the probability that a sample $\mathbf{x}$ with clean label $Y=i$ has a noisy label $\bar{Y}=j$. The diagonal mean of the noise transition matrix can reflect the noise level. When the noise transition matrix appears as a whole, it can be written as $T(\mathbf{x})$. The most popular setting in existing studies [36, 37] assumes the transition matrix is class-dependent and instance-independent, i.e., $T_{ij} = P(\bar{Y}=j|Y=i, X=\mathbf{x}) = P(\bar{Y}=j|Y=i)$. That is, the noisy label $\bar{Y}$ only depends on the true label Y. The transition matrix and noisy class posterior $P(\bar{Y}|X)$ can be estimated from the noisy data, which allows inferring the clean class posterior $P(Y|X)$. The relationship between the three is $P(\bar{Y}|\mathbf{x}) = T(\boldsymbol{x})^{\top}P(Y|\mathbf{x})$, eventually ensuring generalization of the model trained on the corrupted labeled training set. Inspired by this, we hope to achieve calibrated generalization of models trained on corrupted datasets with the help of a noise transition matrix.

3 How does label noise cause calibration to worsen?

Proposition 1. Eq. (1) does not simultaneously contain the $P_{clean}(\cdot)$ trained on clean dataset and clean labels Y. When it is forced to be satisfied, the calibration becomes even worse. Consequently, Eq. (1) cannot guide existing methods to well calibrate the model.

For noisy data, the calibration equation should be

$$P_{noise}(\hat{y}=\bar{y}|\hat{p}=p) = p, \tag{5}$$

where P_{noise} denotes the model trained on noisy data. And for clean data, the calibration equation is transferred to

$$P_{clean}(\hat{y}=y|\hat{p}=p) = p, \tag{6}$$

where P_{clean} denotes the model trained on clean data. When labels are corrupted, the calibrator is trained on noisy data that contains incorrect labels. As a result, the condition required by the calibration equation—ground truth—is not satisfied. The calibration equation actually executed is

$$P_{noise}(\hat{y}=y|\hat{p}=p) \neq p. \tag{7}$$

Table 1 Quantitative empirical support for (8). Comparison of average predicted probabilities for correct and incorrect labels.

		CIFAR-10		CIFAR-100		CIFAR-10		CIFAR-100	
		Sym-20%	Sym-40%	Sym-20%	Sym-40%	Asym-20%	Asym-40%	Asym-20%	Asym-40%
MMCE	$\mathbb{E}(p_{i\bar{y}})$	0.0212	0.0428	0.0060	0.0066	0.0210	0.0448	0.0041	0.0063
	$\mathbb{E}(p_{iy})$	0.8192	0.5800	0.5423	0.3488	0.8272	0.6060	0.5671	0.3681
FLSD53	$\mathbb{E}(p_{i\bar{y}})$	0.0259	0.0458	0.0055	0.0066	0.0202	0.0429	0.0046	0.0063
	$\mathbb{E}(p_{iy})$	0.7811	0.6088	0.5136	0.3324	0.7905	0.5931	0.5033	0.3270

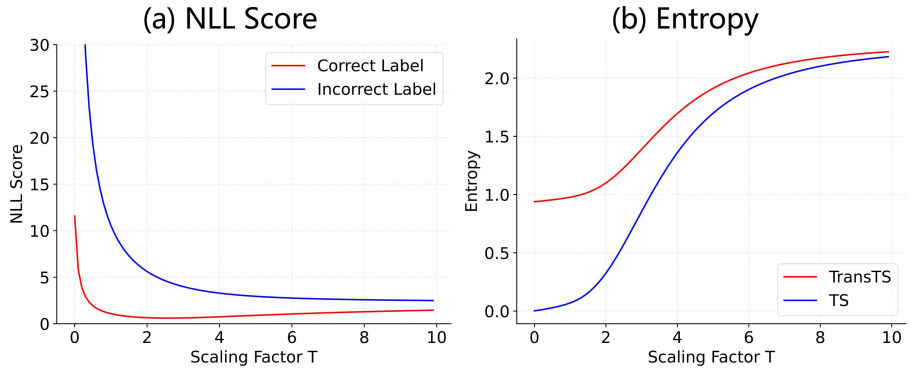

Figure 3 (Color online) Qualitative observations for temperature scaling. ResNet-50 is trained on CIFAR-10 with 20% symmetric noise using cross entropy. Incorrect label denotes samples with an incorrect label, while correct label denotes the samples with the true label.

Eq. (7) should be viewed as invalid. However, it is still used to guide the calibrator to perform probability calibration, inevitably resulting in poor calibration. Next, we analyze why TS methods tend to over-calibrate.

Remark 1. Over-calibration refers to a situation where calibration methods adjust output logits so that the output probabilities become overly smoothed. This results in the model being under-confident, meaning it assigns lower probabilities to its predictions than the true likelihood of correctness.

Assumption 1. Even if the label noise rate is $r\%$, the model can still be determined by $(1-r)\%$ clean labels. The existing literature refers to this as identifiability [33]. We conclude

$$\mathbb{E}(\hat{p}_{i\bar{y}}) < \mathbb{E}(\hat{p}_{iy}), \tag{8}$$

where $\hat{p}_{iy} = \sigma(\mathbf{z}_i/T)[y]$ and $\hat{p}_{i\bar{y}} = \sigma(\mathbf{z}_i/T)[\bar{y}]$ denote the probability corresponding to the correct label or incorrect label, respectively. Obviously, in an output probability distribution, this shows that the probability corresponding to the correct label is relatively large. We provide quantitative empirical support for (8), as shown in Table 1.

Proposition 2. Given a model $f_\theta(\bar{y}|\mathrm{x}_i)$ trained on noisy data, TS learns the scaling factor T^* on noisy validation set by minimizing the $\mathbb{E}_{x,\bar{y}\sim\pi^{val}(x,\bar{y})}\ \mathcal{L}_{\mathrm{NLL}}\left(\sigma(\mathbf{z}/T),\bar{y}\right)$. However, the learned scaling factor T^* tends to be overly aggressive compared to its counterpart from the clean validation set, resulting in an overly smoothed output distribution—a phenomenon known as over-calibration.

Assuming a noise rate of r and sample size of N, the following formula is established:

$$-\sum_{i=N-rN+1}^{N} \log\ \hat{p}_{iy} < -\sum_{i=N-rN+1}^{N} \log\ \hat{p}_{i\bar{y}}, \tag{9}$$

where $[N-rN+1:N]$ represents samples with incorrect label. Figure 3(a) gives the qualitative empirical observation: Eq. (9) holds during temperature scaling optimization.

How does label noise affect TS? First, as an accuracy-preserving method, TS preserves the relative magnitudes of the model's output probability distribution. As a result, the highest probability value remains the highest after scaling. According to Assumption 1, the probability of the incorrect label is smaller than that of the correct label. When the label is incorrect, TS tend to learn a larger scaling factor T to make the logits smoother. The smoother logits can increase $p_{i\bar{y}}$, thus contributing to the minimization of the NLL loss. Compared to the clean validation set, the learned scaling factor T becomes overly large. We also provide quantitative empirical support for the aggressive scaling factor T^*, as shown in Table 2.

Table 2 Quantitative observations for aggressive T. The model is trained on ResNet-50 on CIFAR-10. Noisy TS and clean TS denote TS executed on noisy/clean validation dataset, respectively.

Loss	Method	Sym-20%	Sym-40%	Asym-20%	Asym-40%	IDN-20%	IDN-40%
MMCE	Noisy TS	3.1	2.8	3.3	3.0	3.7	3.0
	Clean TS	2.3	1.6	2.3	1.6	2.6	1.8
	Ours	2.3	1.5	2.4	1.5	2.7	1.7
FLSD53	Noisy TS	1.6	2.3	1.6	2.3	1.5	1.9
	Clean TS	1.1	1.2	1.1	1.2	1.1	1.1
	Ours	1.2	1.1	1.2	1.1	1.0	1.1

4 Adaptive transitional temperature scaling (TransTS)

Overview. We propose a simple yet effective method, termed TransTS. First, we introduce a noise estimator to learn the noise transition matrix, where the noise level can be reflected by the mean of its diagonal elements. Second, the predicted probabilities are scaled incorporating the estimated noise transition matrix $T(x)$, enabling the temperature scaling factor to adapt to label noise. As a result, a consistent calibrator is learned on the noisy validation set that closely matches the one learned on the clean validation set.

Noise transition matrix estimation. In this paper, we study the class-dependent and instance-independent transition matrix, i.e., $P(\bar{Y}=j|Y=i,X=x)=P(\bar{Y}=j|Y=i)=T_{ij}$. Following existing studies [2,38–40], we estimate the noise transition matrix based on the anchor point assumption. Formally, a sample $x^i \in X$ is defined as an anchor point of the i-th clean class if $P(Y=i|X=x^i)=1$. The transition matrix can be estimated via the noisy class posterior of the anchor points. Given an anchor point of the i-th class if $P(Y=i|X=x^i)=1$, when $k\neq i$, $P(Y=k|X=x^i)=0$, then the transition matrix can be estimated

$$\begin{aligned} &P(\bar{Y}=j|X=x^i)\\ &=\sum_{k=1}^{K} P(\bar{Y}=j|Y=k,X=x^i)P(Y=k|X=x^i)\\ &=P(\bar{Y}=j|Y=i,X=x^i)\\ &=P(\bar{Y}=j|Y=i)\\ &=T_{ij}, \end{aligned} \tag{10}$$

where the second equation holds because when $k\neq i$, $P(Y=k|X=x)=0$. The third equation holds because the transition matrix is class-dependent and instance-independent. However, perfect anchor points are usually difficult to obtain when clean labels are unavailable. To address this, we follow [39] and derive approximate anchor points solely from noisy data, subsequently estimating the noise transition matrix:

$$\widehat{\mathcal{X}}^i=\text{top-}k\left\{\left\{x\in X \mid P\left(\bar{Y}=i \mid X=x^i\right)\geqslant \tau_i\right\}, P\left(\bar{Y}=i \mid X=x^i\right)\right\}, \tag{11}$$

where $\hat{\mathcal{X}}^i$ denotes the approximate anchor point set, τ_i denotes a percentile, and top-k aims to replace a single anchor point and reduces the variance within the anchor point [2,38–40]. In this paper, we set the k to 10.

$$\hat{T}_{ij}=\frac{1}{|\hat{\mathcal{X}}^i|}\sum_{x^i\in\mathcal{X}^i} P(\bar{Y}=j \mid X=x^i), \tag{12}$$

where $\hat{T}_{ij}$ denotes the ij-th element of estimated noise transition matrix. The time cost for obtaining the noise transition matrix is proportional to a single forward pass, with a time complexity of $O(k^2\cdot|X|)$ [39]. The required time is 18.23 s for CIFAR-10, 19.07 s for CIFAR-100, and 104.23 s for ANIMAL-10N, respectively, which is acceptable. For all datasets, we train the transition matrix estimator using ResNet-50 and cross entropy loss, and select the epoch with the best accuracy on the noisy validation set as the final transition matrix estimator.

For instance-dependent noise, the estimated noise transition matrix estimated serves as a global statistic approximating the dataset's average noise level. Although not explicitly modeling per-instance variations, it captures dominant noise level. This global estimation also reduces the burden of the temperature factor T (see Table 2), while empirical results demonstrate its effectiveness for instance-dependent noise (see Tables 3–5).

Adaptive transitional calibration. TransTS aims to learn a calibrator from a held-out noisy validation set $D_{val}(x,\bar{y})$:

$$\mathbf{p}^c=g_T(\bar{y}|\mathbf{z}), \tag{13}$$

Table 3 Main results: the ECE (%) on Sym, Asym and IDN noise. ResNet-50 and ViT are trained on CIFAR-10/100. Un-post denotes no post-hoc method is used, the same below. The best scores are bold.

Built-in	Post-hoc	ResNet-50												Vision Transformer			
		CIFAR-10						CIFAR-100						CIFAR-10		CIFAR-100	
		Sym-20	Sym-40	Asym-20	Asym-40	IDN-20	IDN-40	Sym-20	Sym-40	Asym-20	Asym-40	IDN-20	IDN-40	Sym-40	Asym-40	Sym-40	Asym-40
Dual focal	Un-post	13.83	30.72	14.16	30.72	13.51	24.26	12.29	15.25	17.76	15.25	16.73	16.05	21.74	21.79	12.12	12.48
	Vector	13.92	20.72	13.13	20.76	13.44	19.83	14.08	20.14	11.28	20.01	11.27	18.24	28.39	28.69	21.81	21.35
	Spline	17.52	22.36	15.33	22.52	17.24	24.56	12.86	19.04	12.32	18.93	12.15	18.52	25.98	27.07	15.90	14.99
	CPCS	14.55	21.05	13.87	21.16	14.79	22.30	11.76	23.39	11.58	23.36	11.95	22.77	26.55	27.36	16.92	16.43
	TS	13.29	20.99	13.70	21.23	14.41	19.82	14.44	19.53	19.57	20.80	12.79	19.21	29.17	29.42	22.50	22.47
	CTS	13.73	20.41	13.16	20.52	13.45	19.79	14.04	19.91	11.18	19.83	11.09	18.45	28.25	28.58	22.17	21.59
	ETS	14.71	20.74	14.03	20.99	14.97	21.47	13.58	19.04	11.47	19.19	11.89	18.85	21.74	21.79	12.12	12.48
	PTS	13.73	20.76	12.99	20.80	13.46	19.53	14.44	20.41	11.61	20.38	12.09	18.43	28.33	28.87	23.26	22.91
	STCL	14.55	21.72	13.72	21.84	14.58	20.60	15.19	21.20	12.51	21.15	12.93	19.40	29.43	29.91	24.04	23.54
	TvA	14.02	18.60	13.20	18.68	14.38	21.26	17.13	22.21	14.64	22.15	15.65	22.40	26.51	27.46	16.85	16.05
	HB	17.22	21.87	16.36	21.75	17.61	24.12	12.59	15.58	15.06	15.55	13.05	14.58	26.17	27.16	11.10	11.29
	BBQ	17.50	23.04	16.35	22.63	18.03	25.17	17.07	21.34	16.65	21.23	15.96	21.34	26.24	27.23	16.15	15.01
	Ours	**3.86**	**7.38**	**3.81**	**7.38**	**2.45**	**3.38**	**7.40**	**7.99**	**7.67**	**7.99**	**8.04**	**9.51**	**2.12**	**7.45**	**1.90**	**2.07**
AdaFocal	Un-post	11.18	19.35	11.75	20.13	10.80	20.75	10.59	7.19	11.45	6.30	10.67	7.52	27.16	26.05	12.93	11.28
	Vector	12.49	24.30	13.61	24.48	12.60	19.54	10.14	18.99	10.14	21.86	9.01	18.26	29.76	28.68	22.56	21.29
	Spline	17.82	26.58	16.94	26.82	17.65	24.33	11.35	18.18	11.82	20.25	11.29	19.59	27.70	25.77	16.07	14.42
	CPCS	13.23	24.87	14.45	25.09	14.02	22.21	10.55	20.63	10.30	22.72	10.46	22.21	27.96	26.76	17.00	15.79
	TS	12.39	25.68	14.38	24.87	13.47	18.67	10.79	19.90	10.23	23.11	10.59	18.35	30.91	29.77	23.40	21.59
	CTS	12.42	23.98	13.42	24.26	12.45	19.47	10.11	18.98	9.91	21.74	8.90	18.09	29.66	28.58	22.79	21.68
	ETS	13.79	25.02	14.27	24.84	13.89	22.44	11.33	19.23	10.87	20.94	11.04	19.73	27.16	26.05	12.93	11.28
	PTS	12.71	24.62	13.52	24.61	12.58	18.87	10.08	19.35	10.34	22.36	9.33	18.96	29.85	28.82	23.78	22.21
	STCL	13.60	25.93	14.33	25.53	13.75	19.61	11.02	20.17	11.23	23.10	10.14	19.74	30.74	29.86	24.52	22.89
	TvA	13.33	24.91	14.46	25.22	14.20	22.83	12.45	20.53	12.72	22.89	12.96	22.24	27.87	26.45	16.81	15.41
	HB	16.65	25.43	17.07	25.98	17.28	23.35	12.93	12.32	12.96	14.41	13.27	13.46	27.91	26.60	10.72	11.44
	BBQ	17.27	26.27	17.00	26.63	18.27	24.43	17.21	20.83	16.60	22.20	16.98	21.69	27.87	26.08	16.30	14.64
	Ours	**3.49**	**3.67**	**3.01**	**8.21**	**2.67**	**4.81**	**4.96**	**6.46**	**5.30**	**6.29**	**4.80**	**7.52**	**3.64**	**4.04**	**2.12**	**1.84**
MMCE	Un-post	13.40	19.65	13.45	20.27	13.52	23.29	**9.06**	**8.95**	12.16	**9.31**	9.61	**9.13**	19.65	20.57	9.92	11.55
	Vector	13.96	21.75	13.46	22.22	13.62	16.65	15.38	21.14	12.45	21.59	14.87	19.79	28.14	28.09	17.57	20.77
	Spline	16.33	22.18	16.98	22.79	16.66	20.73	12.50	18.80	12.29	20.33	13.56	17.94	26.16	26.00	11.55	14.84
	CPCS	14.65	22.12	14.15	22.61	14.99	19.19	11.71	21.82	11.05	23.62	13.24	22.30	26.66	26.54	12.86	16.23
	TS	13.88	21.75	13.00	22.90	13.12	16.92	16.68	20.68	12.55	22.62	15.60	20.57	27.35	28.25	18.90	21.59
	CTS	13.50	21.91	13.49	22.18	13.43	19.46	15.17	20.69	12.52	21.78	13.37	12.95	28.06	27.99	17.71	21.20
	ETS	15.00	22.72	14.00	21.74	14.21	18.91	13.59	18.90	12.44	20.74	15.11	12.45	19.65	20.57	9.92	11.55
	PTS	13.86	22.27	13.57	22.34	13.51	20.13	15.61	21.40	12.98	22.18	13.87	11.67	28.16	27.93	18.32	21.78
	STCL	14.73	23.37	14.45	23.27	14.62	21.18	16.26	22.23	13.69	22.95	14.57	12.74	29.24	28.92	19.04	22.41
	TvA	14.65	22.47	14.40	23.09	14.33	20.31	18.16	22.56	16.04	23.69	16.46	12.56	26.63	26.40	12.37	16.01
	HB	16.30	21.68	17.08	22.67	16.12	19.69	11.89	12.56	13.97	13.93	11.85	7.53	26.48	26.22	9.15	10.90
	BBQ	16.95	23.08	17.10	23.82	17.97	20.92	18.01	19.21	17.16	20.28	17.56	8.99	26.19	26.05	11.90	15.08
	Ours	**2.80**	**5.94**	**2.80**	**8.08**	**2.87**	**11.11**	**9.06**	**8.95**	**7.06**	**9.31**	**8.6**	**9.13**	**3.40**	**3.21**	**1.07**	**1.55**
FLSD53	Un-post	6.43	10.54	5.95	10.54	4.35	8.85	3.19	6.00	**2.84**	**4.80**	3.61	**5.64**	32.38	33.86	14.48	14.85
	Vector	12.73	25.47	13.21	24.51	12.18	18.34	10.77	19.33	11.46	18.82	10.09	16.47	27.71	28.88	21.66	22.12
	Spline	16.32	26.27	17.90	22.79	15.79	21.79	11.40	18.17	11.96	17.58	11.30	18.73	25.49	25.15	15.12	14.04
	CPCS	12.57	23.32	13.20	23.46	12.35	19.17	9.96	19.91	10.63	19.56	10.04	20.48	25.12	25.77	15.77	15.10
	TS	11.33	25.59	14.66	24.00	10.57	17.79	11.23	18.12	11.65	20.56	9.80	17.50	29.02	30.43	23.78	24.02
	CTS	12.70	24.96	12.95	24.51	11.80	17.63	10.46	19.59	11.39	19.25	10.15	16.72	27.96	29.32	22.14	22.37
	ETS	12.53	24.46	13.76	24.36	12.49	19.45	12.42	19.99	12.20	18.67	11.78	20.16	32.38	33.86	14.48	14.85
	PTS	12.72	24.84	13.08	24.92	11.74	16.85	11.08	20.40	11.69	19.24	10.46	17.53	27.84	28.93	23.05	23.05
	STCL	13.48	26.03	13.83	25.89	12.53	17.83	11.91	21.38	12.64	20.10	11.38	18.44	28.81	29.78	23.74	23.70
	TvA	13.05	22.41	13.34	22.91	12.87	19.47	12.25	20.18	12.78	19.35	12.45	20.79	24.99	25.52	15.42	14.67
	HB	16.01	25.46	16.44	24.92	15.64	21.73	12.76	13.50	12.03	11.51	11.91	11.77	25.47	26.45	13.08	12.18
	BBQ	16.82	26.51	17.25	25.43	16.45	22.32	17.39	21.68	17.02	21.04	17.22	20.86	25.48	25.40	15.32	14.36
	Ours	**2.25**	**7.42**	**2.45**	**7.30**	**2.53**	**5.79**	**3.19**	**5.40**	**2.84**	**4.80**	**3.60**	5.68	**4.81**	**3.50**	**4.48**	**4.64**
MbLs	Un-post	11.28	25.31	12.17	26.95	10.45	22.81	9.55	8.24	9.83	8.99	**7.77**	**8.44**	20.80	21.36	15.70	14.55
	Vector	14.90	23.03	13.77	22.10	15.17	20.99	11.36	20.52	11.58	25.50	11.26	19.02	21.88	23.06	17.98	17.10
	Spline	17.65	25.77	16.41	24.62	17.80	23.59	12.39	19.22	12.34	25.49	12.43	19.19	20.07	22.84	12.97	12.25
	CPCS	16.02	23.69	14.77	22.78	16.74	23.19	14.27	23.99	13.71	28.65	14.58	24.76	20.47	22.73	15.10	13.54
	TS	15.58	23.08	14.79	22.16	14.47	20.76	11.10	22.46	10.50	26.24	13.68	19.47	20.80	24.79	18.71	17.50
	CTS	14.88	22.83	13.72	21.86	15.13	21.27	11.34	20.15	11.45	25.32	11.16	19.03	21.95	23.03	18.25	17.43
	ETS	15.65	24.20	14.96	22.82	16.45	23.08	12.99	21.05	12.86	26.08	13.71	21.14	20.80	21.36	15.70	14.55
	PTS	15.15	23.19	13.95	22.19	15.28	20.58	11.68	20.95	11.62	25.99	11.78	19.69	21.58	22.96	19.05	17.50
	STCL	15.93	23.92	14.84	23.17	16.30	21.50	12.55	21.86	12.50	26.74	12.62	20.55	22.06	23.78	19.73	18.17
	TvA	16.02	23.22	14.84	22.02	16.90	23.33	15.03	22.71	14.53	27.75	15.58	23.27	20.30	22.87	14.68	13.18
	HB	17.40	24.27	16.40	23.39	17.16	22.87	12.80	13.74	13.51	21.64	13.01	14.97	21.17	22.45	11.20	10.96
	BBQ	17.43	25.19	16.98	24.46	18.03	23.94	17.48	21.62	18.05	28.64	18.89	21.70	20.11	22.61	13.37	12.47
	Ours	**3.23**	**5.58**	**4.43**	**5.97**	**3.11**	**4.38**	**6.83**	**7.22**	**5.11**	**4.70**	**7.77**	**8.44**	**6.96**	**8.07**	**1.21**	**1.41**

where T is the calibrator parameter, i.e., the scaling factor; $\mathbf{z}$ denotes the output logits; $\bar{y}$ denotes the given noisy label; and $\mathbf{p}^c$ denotes the calibrated probability distribution. If the noise transition matrix $\hat{T}(\mathbf{x})$ has been estimated, the optimization objective of TransTS is as follows:

$$T^* = \arg\min_{T} \mathbb{E}_{x,\bar{y}\sim D_{val}(x,\bar{y})} \mathcal{L}_{\text{NLL}}\left(\hat{T}(\mathbf{x})^\top \sigma(\mathbf{z}/T), \bar{y}\right), \tag{14}$$

where $\hat{T}(\mathbf{x})^\top$ denotes the transpose of the noise transition matrix. This means the solution (T^*) of (14) is conditional on the noise level. Different noise levels will result in different T^*, which gives TransTS the adaptability to the noise

Table 4 Main results: the AdaECE (%) on Sym, Asym and IDN noise. ResNet-50 and ViT are trained on CIFAR-10/100. Un-post denotes no post-hoc method is used, the same below. The best scores are bold.

Built-in	Post-hoc	ResNet-50												Vision Transformer			
		CIFAR-10						CIFAR-100						CIFAR-10		CIFAR-100	
		Sym-20	Sym-40	Asym-20	Asym-40	IDN-20	IDN-40	Sym-20	Sym-40	Asym-20	Asym-40	IDN-20	IDN-40	Sym-40	Asym-40	Sym-40	Asym-40
Dual focal	Un-post	13.83	30.69	14.11	30.69	13.45	24.25	12.20	15.03	17.68	15.03	16.71	16.02	21.74	21.80	12.09	12.48
	Vector	13.90	20.72	13.09	20.75	13.38	19.84	14.09	19.89	11.37	19.78	11.25	18.27	28.39	28.69	21.81	21.35
	Spline	17.52	22.36	15.33	22.52	17.24	24.56	12.87	19.04	12.32	18.93	12.15	18.52	25.98	27.08	15.90	15.00
	CPCS	14.53	21.04	13.87	21.15	14.76	22.27	11.69	23.34	11.60	23.30	12.10	22.73	26.55	27.36	16.92	16.43
	TS	13.27	20.98	13.70	20.98	14.38	19.72	14.47	19.49	11.56	19.49	12.92	19.08	29.17	29.42	22.50	22.47
	CTS	13.72	20.41	12.95	20.52	13.41	19.79	14.25	19.75	11.27	19.68	11.12	18.45	28.25	28.58	22.17	21.59
	ETS	14.70	20.73	14.04	20.98	14.95	21.42	13.58	19.05	11.44	19.19	11.89	18.85	21.74	21.80	12.09	12.48
	PTS	13.71	20.75	12.97	20.79	13.44	19.41	14.50	20.32	11.63	20.29	12.29	18.40	28.33	28.87	23.26	22.91
	STCL	14.53	21.71	13.72	21.83	14.55	20.53	15.25	21.12	12.47	21.06	13.05	19.28	29.43	29.91	24.04	23.54
	TvA	14.01	18.60	13.20	18.68	14.36	21.20	17.15	22.15	14.52	22.08	15.63	22.36	26.51	27.46	16.85	16.05
	HB	16.25	21.77	15.75	21.64	16.69	24.12	12.52	14.87	13.64	14.86	12.88	14.47	26.17	27.15	11.10	11.29
	BBQ	16.94	22.94	15.98	22.67	17.54	25.17	17.12	21.37	16.92	21.38	15.62	21.69	26.24	27.23	16.15	15.01
	Ours	**4.95**	**7.36**	**4.45**	**7.36**	**3.62**	**3.77**	**7.49**	**7.94**	**7.70**	**7.94**	**8.00**	**9.51**	**2.15**	**7.45**	**2.00**	**2.25**
AdaFocal	Un-post	11.18	19.32	11.75	20.11	10.76	20.70	10.50	6.99	11.32	6.47	10.63	7.48	27.16	26.05	12.93	11.28
	Vector	12.49	24.28	13.61	24.47	12.59	19.50	10.13	18.84	10.25	21.54	9.16	18.23	29.76	28.68	22.56	21.30
	Spline	17.82	26.58	16.94	26.82	17.65	24.33	11.36	18.18	11.82	20.26	11.28	19.60	27.70	25.78	16.07	14.42
	CPCS	13.13	24.84	14.41	25.01	13.96	22.19	10.51	20.61	10.29	22.61	10.49	22.11	27.96	26.76	17.00	15.79
	TS	12.28	25.64	14.34	24.81	13.42	18.66	10.75	19.87	10.15	23.03	10.59	18.34	30.91	29.77	23.40	21.59
	CTS	12.42	23.97	13.42	24.24	12.44	19.47	10.11	18.86	9.98	21.53	8.97	18.16	29.67	28.58	22.78	21.68
	ETS	13.69	25.00	14.23	24.79	13.84	22.41	11.33	19.23	10.86	20.94	11.04	19.73	27.16	26.05	12.93	11.28
	PTS	12.60	24.62	13.47	24.53	12.57	18.86	10.09	19.32	10.33	22.23	9.47	18.95	29.85	28.82	23.78	22.21
	STCL	13.50	25.92	14.29	25.43	13.71	19.60	10.98	20.14	11.12	23.02	10.24	19.70	30.74	29.86	24.52	22.89
	TvA	13.24	24.86	14.43	25.14	14.15	22.79	12.38	20.51	12.62	22.77	12.74	22.14	27.87	26.45	16.81	15.41
	HB	15.69	25.43	16.59	26.00	16.85	23.35	12.95	12.46	12.82	14.67	13.18	13.56	27.90	26.60	10.72	11.44
	BBQ	17.19	26.27	16.89	26.62	17.37	24.43	17.00	20.84	16.74	22.33	16.62	22.01	27.87	26.08	16.30	14.64
	Ours	**4.05**	**4.25**	**4.05**	**8.18**	**3.08**	**4.82**	**4.92**	**6.18**	**5.22**	**6.47**	**4.82**	**7.48**	**3.84**	**4.20**	**2.55**	**1.85**
MMCE	Un-post	13.39	19.55	13.44	20.04	13.50	23.06	**9.11**	**8.83**	12.18	9.27	9.69	**9.08**	19.64	20.57	9.92	11.53
	Vector	13.82	22.05	13.56	22.44	13.57	16.65	15.41	20.61	12.69	21.46	14.78	19.75	28.14	28.08	17.57	20.77
	Spline	16.62	22.74	17.27	23.63	17.03	20.12	13.33	18.07	12.24	20.02	14.03	18.26	26.14	26.00	11.56	14.84
	CPCS	14.54	22.41	14.24	22.85	15.06	19.21	12.01	21.85	11.13	23.06	13.14	21.85	26.65	26.54	12.86	16.23
	TS	13.84	21.75	12.98	22.88	13.06	16.89	16.71	20.58	12.67	22.53	15.66	20.50	27.34	28.25	18.90	21.59
	CTS	13.50	21.91	13.45	22.18	13.60	16.54	15.24	20.68	12.54	21.61	14.66	20.23	28.05	27.99	17.71	21.20
	ETS	14.98	22.72	13.99	21.74	14.65	18.68	13.59	18.90	12.44	20.74	14.45	19.94	19.64	20.57	9.92	11.53
	PTS	13.81	22.27	13.55	22.34	13.67	16.25	15.56	21.24	13.14	22.11	15.68	20.82	28.15	27.93	18.32	21.78
	STCL	14.69	23.37	14.43	23.27	14.69	17.09	16.26	22.04	13.82	22.88	16.34	21.70	29.23	28.92	19.04	22.41
	TvA	14.60	22.47	14.39	23.09	14.84	19.79	18.19	22.41	16.10	23.65	19.13	23.67	26.63	26.40	12.37	16.01
	HB	15.90	21.62	16.37	22.66	17.59	20.12	12.30	12.73	13.38	14.01	12.38	13.53	26.45	26.22	9.15	10.90
	BBQ	16.54	23.04	16.52	23.80	17.45	21.50	18.06	19.29	17.00	20.68	18.19	20.25	26.19	26.05	11.90	15.08
	Ours	**3.06**	**5.79**	**3.92**	**5.48**	**3.35**	**6.70**	**9.11**	**8.83**	**7.19**	**9.31**	**8.38**	**9.08**	**3.44**	**3.19**	**1.47**	**1.80**
FLSD53	Un-post	6.31	10.52	5.94	10.53	4.25	8.79	**3.25**	5.84	**2.91**	**4.88**	3.73	5.58	32.38	33.86	14.48	14.85
	Vector	12.77	25.18	13.12	24.57	11.99	18.15	10.44	19.60	11.57	19.24	10.26	16.58	27.71	28.88	21.66	22.12
	Spline	16.01	25.38	15.71	24.99	16.11	22.42	11.80	18.70	11.36	17.76	11.27	18.90	25.50	25.15	15.12	14.04
	CPCS	12.71	22.87	13.07	23.12	12.43	18.67	10.00	20.70	10.33	19.50	10.15	20.52	25.12	25.77	15.77	15.10
	TS	11.13	25.61	14.57	25.69	10.50	17.52	11.22	22.11	11.63	20.53	10.68	17.42	29.02	30.43	23.78	24.02
	CTS	12.64	24.96	12.92	24.48	11.75	17.63	10.45	19.53	11.39	19.25	10.24	16.49	27.96	29.32	22.14	22.37
	ETS	12.35	24.45	13.73	24.28	12.44	19.12	12.42	19.99	12.20	18.67	11.78	20.16	32.38	33.86	14.48	14.85
	PTS	12.54	24.83	12.99	24.81	11.67	16.69	11.03	20.40	11.69	19.24	10.46	17.42	27.84	28.93	23.05	23.05
	STCL	13.32	26.02	13.74	25.84	12.50	17.58	11.91	21.38	12.61	20.08	11.38	18.34	28.81	29.78	23.74	23.70
	TvA	12.88	22.39	13.24	22.75	12.83	19.13	12.23	20.18	12.78	19.35	12.45	20.68	24.99	25.52	15.42	14.67
	HB	15.97	25.44	16.37	24.92	15.61	21.73	12.76	13.57	11.78	11.79	11.66	12.01	25.46	26.45	13.08	12.18
	BBQ	16.89	26.51	17.31	25.42	16.37	22.32	17.65	21.74	17.10	21.08	17.25	20.98	25.48	25.40	15.32	14.36
	Ours	**2.91**	**7.38**	**2.81**	**7.30**	**2.98**	**5.72**	**3.25**	**5.68**	**2.91**	6.01	**3.73**	**5.76**	**4.85**	**3.93**	**4.47**	**4.65**
MbLs	Un-post	11.27	25.30	12.17	26.95	10.45	22.81	9.37	8.23	9.90	8.97	**7.57**	8.26	18.91	18.23	14.59	14.62
	Vector	14.90	23.03	13.77	22.09	15.17	20.98	11.36	20.46	11.58	25.50	11.32	18.89	22.79	22.51	19.26	19.13
	Spline	17.65	25.77	16.41	24.62	17.80	23.59	12.39	19.22	12.32	25.38	12.43	19.19	20.32	19.67	13.52	12.88
	CPCS	16.02	23.68	14.77	22.78	16.74	23.19	14.27	23.99	13.71	28.61	14.58	24.76	20.82	20.54	15.05	14.07
	TS	15.59	23.07	14.79	22.16	14.48	20.77	11.06	22.46	10.47	26.19	13.68	19.47	22.18	21.48	20.48	20.27
	CTS	14.88	22.82	13.72	21.86	15.08	21.27	11.35	20.10	11.45	25.32	11.30	18.89	22.84	22.61	19.77	19.44
	ETS	15.65	24.19	14.96	22.83	16.45	23.09	12.99	21.05	12.86	26.04	13.71	21.14	18.91	18.23	14.59	14.62
	PTS	15.15	23.18	13.96	22.19	15.28	20.58	11.66	20.95	11.62	25.94	11.75	19.69	23.01	22.54	20.71	20.02
	STCL	15.93	23.91	14.84	23.17	16.30	21.50	12.55	21.86	12.50	26.69	12.61	20.55	23.48	23.37	21.36	20.60
	TvA	16.02	23.22	14.84	22.02	16.90	23.32	15.03	22.71	14.53	27.72	15.58	23.27	20.55	20.03	14.75	13.94
	HB	16.68	24.28	15.80	23.39	17.21	22.87	13.15	13.77	13.19	21.28	13.76	15.04	21.77	21.01	11.20	11.87
	BBQ	17.38	25.19	16.53	24.46	17.48	24.01	17.46	21.62	18.05	28.65	18.70	22.18	20.56	19.88	13.82	13.24
	Ours	**3.77**	**5.85**	**4.38**	**6.23**	**3.28**	**4.25**	**6.85**	**7.08**	**5.12**	**4.66**	**7.57**	**8.24**	**8.31**	**8.85**	**5.51**	**5.87**

level. The optimal temperature scaling factor T^* is applied to adjust the logits. Given an input $\mathbf{x}_i \sim \pi^{te}(X, Y)$, we compute the calibrated probability as follows:

$$\hat{p}_{ij}^{c} = \frac{\exp(z_{ij}/T^*)}{\sum_{k=1}^{K} \exp(z_{ik}/T^*)}, \tag{15}$$

where $\hat{p}_{ij}^{c}$ represents the calibrated prediction confidence. In this section, our method adaptively operates on the noisy dataset, without any prior knowledge about the noise (e.g., noise rates, types, or access to clean samples).

Assumption 2. The noise transition matrix is a doubly stochastic matrix, i.e., $\sum_i T_{ij} = 1$ and $\sum_j T_{ij} = 1$ for all

Table 5 Main results: the ClasswiseECE (%) on Sym, Asym and IDN noise. ResNet-50 and ViT are trained on CIFAR-10/100. Un-post denotes no post-hoc method is used, the same below. The best scores are bold. The "–" denotes that Classwise calculation is not supported.

Built-in	Post-hoc	ResNet-50												Vision Transformer			
		CIFAR-10						CIFAR-100						CIFAR-10		CIFAR-100	
		Sym-20	Sym-40	Asym-20	Asym-40	IDN-20	IDN-40	Sym-20	Sym-40	Asym-20	Asym-40	IDN-20	IDN-40	Sym-40	Asym-40	Sym-40	Asym-40
Dual focal	Un-post	2.81	6.29	2.87	6.29	2.74	4.95	0.31	0.38	0.41	0.38	0.39	0.40	5.07	4.98	0.39	0.39
	Vector	3.41	4.85	3.30	4.85	3.23	5.23	0.37	0.41	0.33	0.42	0.33	0.41	6.38	6.37	0.54	0.53
	Spline	–	–	–	-	–	–	–	–	–	-	–	–	–	–	–	–
	CPCS	3.54	5.09	3.52	5.11	3.53	5.56	0.34	0.44	0.35	0.44	0.33	0.46	6.07	6.10	0.46	0.45
	TS	3.28	5.09	3.48	5.09	3.45	5.09	0.38	0.40	0.35	0.40	0.34	0.42	6.59	6.54	0.55	0.55
	CTS	3.35	4.91	3.29	4.94	3.25	5.05	0.38	0.42	0.34	0.42	0.33	0.40	6.38	6.35	0.55	0.53
	ETS	3.56	5.06	3.54	5.09	3.57	5.44	0.37	0.43	0.36	0.43	0.34	0.43	5.07	4.98	0.39	0.39
	PTS	3.37	5.06	3.33	5.06	3.27	5.03	0.38	0.41	0.35	0.41	0.34	0.41	6.42	6.42	0.56	0.56
	STCL	3.54	5.18	3.49	5.20	3.48	5.23	0.39	0.42	0.36	0.42	0.35	0.42	6.64	6.64	0.57	0.57
	TvA	3.43	4.73	3.38	4.74	3.44	5.37	0.41	0.43	0.38	0.43	0.38	0.46	3.43	4.73	3.38	4.74
	HB	2.81	6.29	2.87	6.29	2.74	4.95	0.31	0.38	0.41	0.38	0.39	0.40	2.81	6.29	2.87	6.29
	BBQ	2.81	6.29	2.87	6.29	2.74	4.95	0.31	0.38	0.41	0.38	0.39	0.40	2.81	6.29	2.87	6.29
	Ours	**1.10**	**1.83**	**1.18**	**1.83**	**1.08**	**1.80**	**0.26**	**0.29**	**0.26**	**0.29**	**0.26**	**0.31**	**1.38**	**1.63**	**0.29**	**0.30**
AdaFocal	Un-post	2.31	4.00	2.43	4.08	2.21	4.28	0.29	0.29	0.30	**0.29**	0.29	**0.29**	6.09	5.90	0.39	0.41
	Vector	3.09	5.50	3.37	5.55	3.10	5.34	0.33	0.43	0.33	0.45	0.32	0.43	6.61	6.38	0.55	0.54
	Spline	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	CPCS	3.22	5.68	3.54	5.74	3.39	5.71	0.35	0.45	0.34	0.46	0.34	0.48	6.24	6.04	0.46	0.48
	TS	3.05	5.82	3.53	5.71	3.30	5.08	0.35	0.45	0.34	0.47	0.34	0.43	6.86	6.66	0.57	0.56
	CTS	3.07	5.60	3.34	5.64	3.06	5.10	0.34	0.44	0.34	0.46	0.32	0.43	6.61	6.37	0.56	0.55
	ETS	3.33	5.73	3.51	5.70	3.37	5.75	0.36	0.45	0.35	0.46	0.36	0.46	6.09	5.90	0.39	0.41
	PTS	3.12	5.62	3.38	5.66	3.12	5.12	0.34	0.44	0.34	0.46	0.33	0.44	6.64	6.47	0.57	0.57
	STCL	3.29	5.87	3.52	5.83	3.34	5.25	0.36	0.45	0.35	0.47	0.34	0.45	6.82	6.68	0.58	0.58
	TvA	3.24	5.68	3.54	5.76	3.43	5.82	0.37	0.45	0.37	0.47	0.37	0.48	3.24	5.68	3.54	5.76
	HB	2.31	4.00	2.43	4.08	2.21	4.28	0.29	0.29	0.30	0.29	0.29	0.29	2.31	4.00	2.43	4.08
	BBQ	2.31	4.00	2.43	4.08	2.21	4.28	0.29	0.29	0.30	0.29	0.29	0.29	2.31	4.00	2.43	4.08
	Ours	**1.13**	**1.51**	**1.14**	**1.84**	**0.93**	**1.96**	**0.26**	**0.28**	**0.25**	**0.29**	**0.25**	**0.29**	**1.30**	**1.45**	**0.29**	**0.33**
MMCE	Un-post	2.76	4.10	2.75	4.14	2.76	4.82	**0.28**	**0.30**	0.32	**0.30**	**0.27**	**0.31**	4.73	4.94	0.38	0.37
	Vector	3.27	4.99	3.33	5.16	3.28	4.69	0.37	0.39	0.35	0.42	0.36	0.41	6.45	6.43	0.45	0.51
	Spline	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	CPCS	3.42	5.15	3.47	5.26	3.61	5.12	0.35	0.40	0.33	0.41	0.34	0.43	6.18	6.18	0.41	0.45
	TS	3.28	5.02	3.19	5.26	3.23	4.67	0.40	0.39	0.35	0.41	0.38	0.42	6.32	6.53	0.49	0.52
	CTS	3.22	5.07	3.29	5.12	3.31	4.48	0.38	0.40	0.35	0.41	0.37	0.41	6.43	6.42	0.47	0.52
	ETS	3.50	5.33	3.41	5.26	3.53	5.05	0.38	0.41	0.36	0.43	0.38	0.44	4.73	4.94	0.38	0.37
	PTS	3.27	5.13	3.32	5.17	3.35	4.54	0.39	0.40	0.35	0.41	0.38	0.42	6.49	6.46	0.48	0.53
	STCL	3.45	5.32	3.50	5.34	3.53	4.71	0.40	0.40	0.36	0.41	0.39	0.43	6.71	6.67	0.49	0.53
	TvA	3.43	5.16	3.49	5.30	3.56	5.23	0.42	0.40	0.39	0.42	0.42	0.45	3.43	5.16	3.49	5.30
	HB	2.76	4.10	2.75	4.14	2.76	4.82	0.28	0.30	0.32	0.30	0.27	0.31	2.76	4.10	2.75	4.14
	BBQ	2.76	4.10	2.75	4.14	2.76	4.82	0.28	0.30	0.32	0.30	0.27	0.31	2.76	4.10	2.75	4.14
	Ours	**1.13**	**1.69**	**1.13**	**1.64**	**1.12**	**2.04**	**0.28**	**0.30**	**0.25**	**0.30**	**0.27**	**0.31**	**1.44**	**1.36**	**0.31**	**0.29**
FLSD53	Un-post	1.45	2.39	1.28	2.25	1.08	2.37	0.25	0.31	**0.26**	**0.30**	**0.26**	0.30	7.39	7.53	0.42	0.44
	Vector	3.03	5.38	3.00	5.54	2.80	4.33	0.33	0.47	0.36	0.46	0.35	0.44	6.35	6.43	0.53	0.55
	Spline	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	CPCS	3.09	4.97	3.02	5.20	2.97	4.72	0.33	0.49	0.35	0.47	0.35	0.49	5.90	5.82	0.44	0.44
	TS	2.81	5.56	3.31	5.76	2.61	4.51	0.35	0.51	0.37	0.48	0.36	0.45	6.72	6.78	0.57	0.58
	CTS	3.01	5.36	2.99	5.50	2.75	4.24	0.35	0.48	0.37	0.46	0.35	0.44	6.46	6.50	0.54	0.56
	ETS	3.06	5.71	3.23	5.80	3.02	4.81	0.37	0.48	0.38	0.45	0.37	0.48	7.39	7.53	0.42	0.44
	PTS	3.05	5.39	3.00	5.57	2.84	4.34	0.35	0.49	0.37	0.47	0.35	0.45	6.48	6.48	0.57	0.57
	STCL	3.21	5.64	3.15	5.79	2.98	4.52	0.36	0.50	0.38	0.48	0.37	0.46	6.68	6.65	0.57	0.58
	TvA	3.12	4.88	3.05	5.13	3.03	4.82	0.36	0.49	0.39	0.47	0.38	0.50	3.12	4.88	3.05	5.13
	HB	1.45	2.39	1.28	2.26	1.08	2.37	0.25	0.31	0.26	0.30	0.26	0.30	1.45	2.39	1.28	2.26
	BBQ	1.45	2.39	1.28	2.26	1.08	2.37	0.25	0.31	0.26	0.30	0.26	0.30	1.45	2.39	1.28	2.26
	Ours	**1.07**	**1.87**	**0.93**	**1.75**	**0.98**	**2.13**	**0.25**	**0.29**	**0.26**	**0.30**	**0.26**	**0.29**	**1.65**	**1.37**	**0.31**	**0.34**
MbLs	Un-post	2.33	5.15	2.50	5.51	2.17	4.70	0.27	0.30	0.28	0.27	**0.26**	**0.29**	5.10	4.73	0.43	0.44
	Vector	3.53	5.49	3.39	5.28	3.64	5.46	0.33	0.42	0.33	0.55	0.33	0.42	5.73	5.53	0.49	0.49
	Spline	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	CPCS	3.75	5.59	3.58	5.56	3.89	5.81	0.38	0.46	0.37	0.60	0.37	0.48	5.45	5.22	0.44	0.43
	TS	3.66	5.50	3.59	5.45	3.47	5.34	0.34	0.44	0.33	0.56	0.36	0.42	5.73	5.39	0.51	0.51
	CTS	3.57	5.45	3.38	5.36	3.59	5.34	0.35	0.43	0.34	0.54	0.33	0.42	5.71	5.55	0.50	0.49
	ETS	3.67	5.67	3.62	5.57	3.85	5.80	0.37	0.44	0.36	0.57	0.37	0.45	5.10	4.73	0.43	0.44
	PTS	3.57	5.51	3.43	5.45	3.63	5.30	0.34	0.42	0.34	0.55	0.33	0.42	5.89	5.60	0.51	0.51
	STCL	3.73	5.62	3.60	5.63	3.81	5.48	0.35	0.44	0.36	0.56	0.35	0.43	5.98	5.77	0.51	0.51
	TvA	3.75	5.52	3.60	5.42	3.92	5.83	0.39	0.44	0.38	0.58	0.39	0.46	3.75	5.52	3.60	5.42
	HB	2.33	5.15	2.50	5.51	2.17	4.70	0.27	0.30	0.28	0.27	0.26	0.29	2.33	5.15	2.50	5.51
	BBQ	2.33	5.15	2.50	5.51	2.17	4.70	0.27	0.30	0.28	0.27	0.26	0.29	2.33	5.15	2.50	5.51
	Ours	**1.11**	**1.50**	**1.15**	**1.60**	**1.20**	**1.95**	**0.25**	**0.29**	**0.26**	**0.26**	**0.26**	**0.29**	**2.32**	**2.22**	**0.32**	**0.34**

i, j, which includes well-known cases such as symmetric and asymmetric label noise.

Proposition 3. The TransTS introduces the noise transition matrix $T(x)$ to scale the output probabilities $\mathbf{p}_i = \text{softmax}(\mathbf{z}_i)$ adaptively, share the effect of the temperature scaling factor T, prevent T from being too aggressive, thereby alleviating the over-calibration.

Multiply the output probability distribution P by the noise transition matrix to obtain the new probability distribution P', where $p'_i = \sum_j p_j T_{ji}$[1)]. We aim to show that the noise transition matrix can smooth the output

1) For clarity, we denote the probability distribution as $P = \{p_1, p_2, \ldots, p_k\}$ and denote noise transition matrix as $T(x)$ rather than $\hat{T}(x)$ in this proof.

probabilities, i.e., $H(P') > H(P)$. We first introduce the concave entropy function $f(x) = -x\log x$. Applying Jensen's inequality yields

$$-\left(\sum_j p_j T_{ji}\right)\log\left(\sum_j p_j T_{ji}\right) \geqslant \sum_j T_{ji}(-p_j \log p_j). \tag{16}$$

Sum on all i

$$\sum_i -\left(\sum_j p_j T_{ji}\right)\log\left(\sum_j p_j T_{ji}\right) \geqslant \sum_i \sum_j T_{ji}(-p_j \log p_j). \tag{17}$$

By exchanging the order of summation and using $\sum_i T_{ji} = 1$, the right side can be simplified to

$$\sum_i \sum_j T_{ji}(-p_j \log p_j) = \sum_j (-p_j \log p_j)\sum_i T_{ji} = \sum_j -p_j \log p_j. \tag{18}$$

Now

$$\sum_i -\left(\sum_j p_j T_{ji}\right)\log\left(\sum_j p_j T_{ji}\right) \geqslant \sum_j -p_j \log p_j. \tag{19}$$

It is

$$H(P') \geqslant H(P). \tag{20}$$

Figure 3(b) shows that TransTS makes the entropy of the probability distribution larger, which confirms (20). Further, the noise transition matrix shares part of the scaling effect of the scaling factor T, preventing T from being too large. Table 2 shows that the scaling factor T learned by TransTS is usually moderate.

Proposition 4. Suppose that the noise transition matrix $T(x)$ is invertible, TransTS achieves consistent calibration: the calibrator learned on the noisy validation set $D_{val}(x, \bar{y})$ converges to the one learned on the clean validation set $D_{val}(x, y)$.

We consider TransTS to be a single-parameter model with learnable scaling factor T. It takes $\mathbf{z}$ as input and outputs the calibrated output probabilities. The relationship between the noisy class posteriors $P(\bar{y} = j|\mathbf{z})$ and the clean class posteriors $P(y = i|\mathbf{z})$ is

$$\begin{aligned} P(\bar{y} = j|\mathbf{z}) &= \sum_{i=1}^{K} P(\bar{y} = j|y = i, \mathbf{z}) P(y = i|\mathbf{z}) \\ &= \sum_{i=1}^{K} T_{ij}(\mathbf{z}) P(y = i|\mathbf{z}) \\ &= \sum_{i=1}^{K} T_{ij}(\mathbf{x}) P(y = i|\mathbf{z}). \end{aligned} \tag{21}$$

Since the transition matrix is class-dependent and instance-independent, $T_{ij}(\mathbf{z})$ and $T_{ij}(\mathbf{x})$ are equivalent. Note that Proposition 4 requires the noise transition matrix to be invertible, which means that the clean class posterior derived from the noisy class posterior has a unique solution [28, 39, 40]:

$$\operatorname*{argmin}_P \mathbb{E}_{\mathbf{z},\bar{y}} \ell(\bar{y}, P(\mathbf{z})) = \operatorname*{argmin}_P \mathbb{E}_{\mathbf{z},y} \ell(y, P(\mathbf{z})), \tag{22}$$

where the $\ell(\cdot)$ denotes the loss function. In practice, the noise transition matrix is almost certainly invertible. Even when the noise transition matrix exhibits a certain degree of non-invertibility, its condition number can be improved by mixing identity matrix, thereby approximately satisfying invertibility [39]. Therefore, we can call (21) calibrator consistent, which means that the calibrator learned on the noisy validation set can converge to the one learned on the clean validation set. Further, for the calibration equation, we can get

$$\begin{aligned} &P(\bar{y} = j|\hat{p} = p) \\ &= \sum_{i=1}^{K} P(\bar{y} = j|y = i, \hat{p} = p) P(y = i|\hat{p} = p) \\ &= \sum_{i=1}^{K} T_{ij}(x) P(y = i|\hat{p} = p). \end{aligned} \tag{23}$$

Science TransTS can converge to the clean label domain, what TransTS actually implements is $P(y = i|\hat{p} = p) \rightarrow P(\hat{y} = y|\hat{p} = p) = p$, which aligns with the ideal calibration behavior under clean labels. This effectively resolves the issue identified in Proposition 1. Thanks to TransTS's consistent calibration property, Eq. (7) becomes valid even in the presence of label noise. This ensures that the calibration equation remains effective in guiding the calibration process for DNN classifiers, just as it would with clean data. Proposition 4 shows that TransTS theoretically approaches the performance upper bound achievable with a clean validation set. Importantly, this performance is achieved relying only on noisy data. Compared with single-matrix methods (such as matrix scaling), TransTS can adaptively adjust the temperature scaling factor according to the estimated noise level. Furthermore, TransTS can achieve consistent calibration, which single-matrix methods (such as matrix scaling) cannot do.

5 Experiments

5.1 Experimental setup

Dataset and network architecture. In this paper, we use the CIFAR-10, CIFAR-100 [41]. CIFAR-10 and CIFAR-100 both contain 50000 training images and 10000 test images, with 10 classes and 100 classes, respectively. The number of images per class is approximately equal. We also use a real-world noisy dataset: ANIMAL-10 [42], which contains 5 pairs of confusing animal categories and a total of 55000 images. The training dataset contains 50000 images and the test dataset contains 5000 images. The noise rate is about 8%. Two network architectures are considered in this paper: ResNet-50 [43] and ViT-Base [44]. The ViT is designed to process input images of size 32x32 pixels, dividing them into 4x4 patches. The model uses a 512-dimensional embedding space, a transformer encoder with 6 layers, and 8 attention heads. The feed-forward network within the transformer has a hidden dimension of 512. In addition, we performed supplementary analyses on ResNet-110 [43], Wide-ResNet-26-10 [45], and DenseNet-121 [46]. Following existing studies [2,38–40], we employ τ values of 94% for CIFAR-10/ANIMAL-10 and 99.6% for CIFAR-100 from a 10% held-out validation set.

Baselines. For built-in baselines, we use MMCE [47], FLSD53 [6], dual focal [8], AdaFocal [7], and MbLs [20]. For post-hoc baselines, we use HB [48], BBQ [32], vector scaling [49], TS [12], Splines [17], CPCS [19], CTS [50], ETS [6], PTS [16], STCL [51], and TvA [18]. Additionally, we include the following LNL baselines: DivideMix [29], Dual-T [40], LogitsClip [52], PLM [2], and RENT [53].

Metrics. Three metrics are used in this paper: ECE [32], adaptive ECE (AdaECE) [54], and ClasswiseECE [55]. For ECE, AdaECE and ClasswiseECE, the number of bins is set to 15.

Noisy setting. Following existing studies [2, 26, 29, 52, 53], we artificially corrupt the dataset to obtain noisy datasets with noise rates of 20% and 40%. We evaluate both symmetric and asymmetric label noise, which are generated by randomly flipping a portion of labels in the training and validation sets to any other class. Although our method is designed for class-dependent and instance-independent label noise, it is also evaluated under instance-dependent noise settings, see Subsection 5.2. For clarity, we abbreviate the three noise types as follows: symmetric noise (Sym), asymmetric noise (Asym), and instance-dependent noise (IDN).

Model training. In our experiments, the training settings are consistent with existing studies [6–8]. For training networks on CIFAR-10, CIFAR-100, and ANIMAL-10N, we use SGD with a momentum of 0.9 as our optimizer, and train the networks for 350 epochs. The learning rate is set according to epoch, and the learning rate changes at 150th, 250th, and 350th are 0.1, 0.01, 0.001, respectively. The training batch size is set to 128. All experiments are conducted on Ubuntu 22.10. The computing device used is NVIDIA GeForce RTX 4090. The software platform versions are as follows: CUDA version 12.10, PyTorch version 1.7.1, and Python version 3.7.5. Since our method is a post-hoc method, the results will remain consistent across multiple runs for an already-trained model. Therefore, we did not report variances. For temperature scaling, the temperature scaling factor ranges from 0.1 to 100 with a step size of 0.1, and the loss function is the NLL. For the LNL methods, we strictly follow their respective official implementations.

5.2 Main results

Synthetic noisy dataset. We evaluate our method under symmetric, asymmetric, and instance-dependent label noise on CIFAR-10 and CIFAR-100 with label noise introduced at rates of 20% and 40%, as shown in Table 3. In this experiment, we employ ResNet-50 and ViT as backbones. In all cases, the performance of all post-hoc baselines degrades under label noise, while our method consistently outperforms them. The reason is that label noise can also lead to over-calibration in other post-hoc methods, whereas TransTS mitigates this by adaptively scaling the logits to obtain a moderate scaling factor T. To support this, we provide a visual analysis in Figure 4.

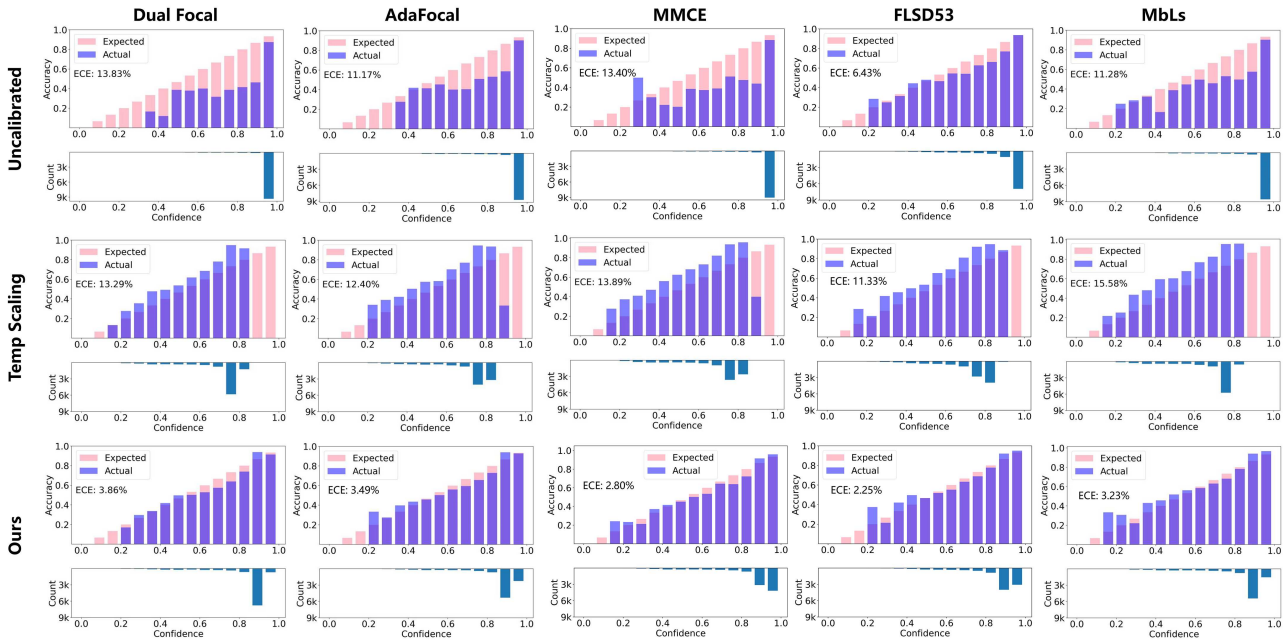

Figure 4 (Color online) Over-calibration and its mitigation on the CIFAR-10. ResNet-50 model trained under Sym-20% noise.

The results show that the temperature scaling method still suffers from severe over-calibration, while our method significantly alleviates it and achieves better calibration results. We also provide quantitative empirical support for the moderate scaling factor T^* in Table 2. For the built-in baseline methods, our method achieves the best ECE in the vast majority of cases. For example, when combined with DualFocal and AdaFocal losses, our method significantly improved calibration performance, yielding the best calibration results. Our method achieved the best ECE across most experimental settings when integrated with MMCE, FLSD53, and MbLs. However, in a few cases on CIFAR-100, our method performed similarly to built-in methods such as MMCE. We argue that, when the number of classes is large, the validation data is partitioned into highly imbalanced subsets by post-hoc calibration methods, which lack sufficient examples for reliable calibration [18]. For ViT, when all built-in methods perform poorly, our method achieves the best performance in all cases, indicating that our method remains effective with the ViT backbone. The results of AdaECE and ClasswiseECE scores are shown in Tables 4 and 5, respectively.

Real-world noisy dataset. We evaluate our method on the real-world noisy dataset ANIMAL-10N. Table 6 shows that our method achieves the best calibration performance in all cases. Take ECE scores as an example, under MMCE, FLSD53, DualFocal, AdaFocal, and MbLs, our method consistently outperforms other approaches, irrespective of the post-hoc calibration method used. Without combining with post-hoc baselines, each built-in method performs differently. For example, FLSD53 demonstrates moderate performance, indicating a certain level of effectiveness on real-world noisy datasets. However, when combined with our method, it achieves significant improvements. Moreover, we provide a visual analysis in Figure 5. The results show that, when faced with real-world noisy datasets, TS also suffers from severe over-scaling, while our method effectively mitigates this issue. The results of AdaECE and ClasswiseECE scores are also shown in Table 6.

5.3 TransTS can effectively calibrate LNL method

We evaluate our method combined with LNL methods, as shown in Table 7. The calibration of LNL baselines is suboptimal as they are not specifically designed for calibration. Among these baselines, DivideMix performs slightly better, whereas LogitsClip exhibits the worst calibration performance. When denoising strategies are applied (i.e., first using an LNL method to denoise the validation set, then combining with TS), the calibration performance tends to degrade. For example, DivideMix achieves 94.83% accuracy with 20% symmetric noise. Applying it to denoise the validation set reduces label noise by 14.83%. TS is applied to the output probabilities of DivideMix and its predicted labels. Due to the use of argmax in label prediction, the output probabilities align exactly with the predicted class labels, which naturally minimizes the empirical classification risk. But TS, which aims to further minimize empirical risk, pushes the predicted probabilities toward a one-hot distribution, leading to over-confidence. We provide reliability diagrams in Figure 6 to demonstrate that denoising strategies lead to over-confidence. In contrast, our method achieves the best results in all cases. When RENT is used, we observe a 48.24%

Table 6 Main results: evaluation on the real-world noisy dataset ANIMAL-10N. The ECE, AdaECE and ClasswiseECE scores trained on ResNet-50 are shown. The best scores are bold. The "–" denotes that Classwise calculation is not supported.

ECE (%)													
Built-in	Un-post	Vector	Spline	CPCS	TS	CTS	ETS	PTS	STCL	TvA	HB	BBQ	Ours
MMCE	10.16	3.66	5.90	4.68	4.47	3.40	5.24	4.04	5.04	6.17	6.08	6.94	**2.03**
FLSD53	3.01	4.25	4.94	4.67	2.96	3.92	4.67	4.00	5.07	4.62	5.09	5.41	**1.85**
DualFocal	10.65	4.17	6.27	5.15	3.69	4.09	5.55	4.22	5.12	6.52	8.16	8.28	**1.48**
AdaFocal	8.87	3.33	5.08	4.46	3.89	3.19	4.72	3.35	4.50	5.43	7.78	8.65	**0.93**
MbLs	8.78	3.46	5.75	5.25	3.90	3.35	5.25	3.67	4.32	5.34	8.10	8.68	**2.54**
AdaECE (%)													
Built-in	Un-post	Vector	Spline	CPCS	TS	CTS	ETS	PTS	STCL	TvA	HB	BBQ	Ours
MMCE	6.45	3.78	4.92	4.66	**3.37**	3.54	5.14	4.07	5.07	6.16	6.32	7.42	3.61
FLSD53	3.05	4.11	5.57	4.53	3.14	4.00	4.53	3.79	4.90	4.39	5.60	5.24	**1.72**
Dual	10.62	4.17	6.13	5.17	3.93	4.02	5.55	4.36	5.15	6.52	7.90	8.12	**2.84**
Ada	8.87	3.07	5.15	4.30	3.77	3.06	4.58	3.21	4.33	5.25	7.66	9.07	**2.08**
MbLs	8.77	3.31	5.42	4.96	3.44	2.92	4.96	3.22	4.00	5.07	8.81	8.65	**3.02**
ClasswiseECE (%)													
Built-in	Un-post	Vector	Spline	CPCS	TS	CTS	ETS	PTS	STCL	TvA	HB	BBQ	Ours
MMCE	1.44	1.20	–	4.57	1.18	1.19	1.51	1.27	1.42	1.61	1.44	1.44	**1.10**
FLSD53	1.29	1.08	–	4.67	1.36	**1.10**	1.58	1.47	1.64	1.57	1.29	1.29	1.29
Dual	2.23	1.28	–	5.15	1.32	1.30	1.60	1.38	1.52	1.74	2.23	2.23	**0.97**
Ada	1.87	1.11	–	4.46	1.38	1.20	1.50	1.28	1.44	1.60	1.87	1.87	**0.99**
MbLs	1.82	1.21	–	5.25	1.46	1.31	1.70	1.42	1.55	1.71	1.82	1.82	**1.01**

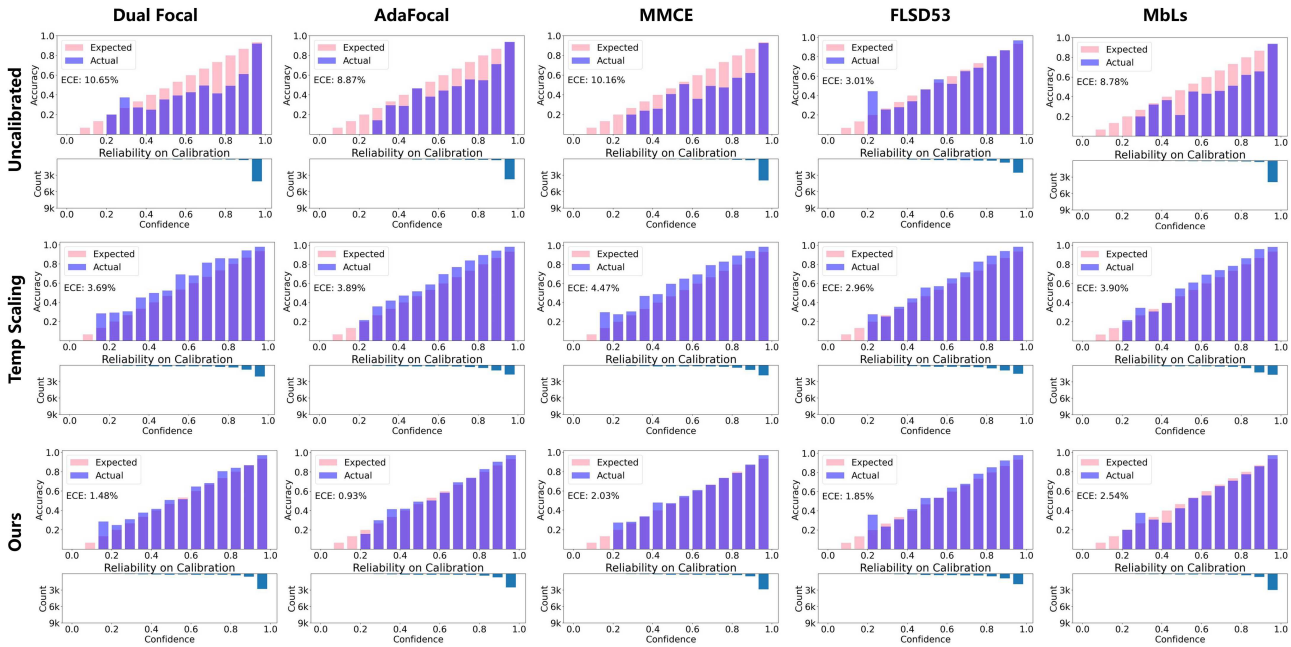

Figure 5 (Color online) Over-calibration and its mitigation on the real-world noisy dataset ANIMAL-10N. ResNet-50 model trained under Sym-20% noise.

ECE improvement in CIFAR-100 with 40% symmetric noise. This further indicates that our method, as a general post-hoc method, can be combined with any pre-trained model, regardless of how it was trained.

5.4 TransTS has adaptability on clean data

On clean datasets, TransTS maintains performance comparable to existing calibration methods. According to the anchor point assumption (i.e., a sample $x \in X$ is considered to be an anchor point of the i-th clean class if $P(Y = i|X = x) = 1$), clean labels do not transfer to noisy labels, implying that the transition matrix is an identity matrix. For a well-trained model on a clean dataset, the confidence of anchor points obtained from it is close to 1.

Table 7 Evaluation on the LNL methods. The ECE, AdaECE and ClasswiseECE scores are shown. The best scores are bold.

LNL	Type	ECE (%)				AdaECE (%)				ClasswiseECE (%)			
		CIFAR-10		CIFAR-100		CIFAR-10		CIFAR-100		CIFAR-10		CIFAR-100	
		Sym-20	Sym-40	Sym-20	Sym-40	Sym-20	Sym-40	Sym-20	Sym-40	Sym-20	Sym-40	Sym-20	Sym-40
DivideMix	Un-post	7.34	3.63	12.34	9.99	7.34	3.61	12.34	9.98	1.53	0.80	0.31	0.27
	TS	18.92	38.83	15.11	33.53	18.91	38.82	15.05	33.53	4.01	7.82	0.41	0.74
	Denoise TS	9.03	4.76	20.66	21.29	9.04	4.73	20.66	21.29	1.75	0.94	0.42	0.44
	Ours	**1.80**	**2.47**	**4.42**	**8.14**	**2.35**	**2.41**	**4.51**	**8.05**	**1.19**	**0.62**	**0.24**	**0.29**
Dual-T	Un-post	12.25	18.25	13.78	10.02	12.24	18.24	13.78	10.01	2.60	3.76	0.42	0.38
	TS	20.52	32.34	15.41	21.46	20.53	32.34	15.41	21.45	4.24	6.80	0.40	0.46
	Denoise TS	25.75	31.18	52.33	59.03	25.75	31.17	52.34	59.03	5.19	6.22	1.09	1.24
	Ours	**1.35**	**1.91**	**2.69**	**6.22**	**1.64**	**1.85**	**2.80**	**6.23**	**0.89**	**0.98**	**0.27**	**0.28**
LogitsClip	Un-post	34.13	29.02	34.60	44.46	34.13	29.03	47.98	43.34	7.54	6.67	0.72	0.46
	TS	21.73	29.02	14.52	22.11	21.70	29.03	20.14	21.06	5.10	6.67	0.47	0.46
	Denoise TS	16.20	19.88	16.07	42.88	16.19	19.88	41.11	44.07	3.28	4.02	0.86	0.93
	Ours	**3.76**	**6.02**	**3.90**	**12.86**	**5.10**	**5.99**	**5.43**	**12.92**	**1.42**	**1.64**	**0.31**	**0.36**
PLM	Un-post	8.80	13.64	8.81	18.95	8.80	13.64	8.79	18.95	2.16	3.32	0.32	0.50
	TS	21.26	33.95	14.70	30.61	21.26	33.95	14.70	30.61	4.63	7.30	0.41	0.71
	Denoise TS	8.90	13.62	30.38	33.74	8.86	13.62	30.38	33.73	1.81	2.77	0.64	0.71
	Ours	**1.73**	**3.89**	**3.40**	**2.88**	**1.74**	**3.64**	**3.46**	**2.91**	**0.91**	**1.15**	**0.24**	**0.25**
RENT	Un-post	10.98	17.56	43.87	50.70	2.63	4.91	40.74	49.77	1.30	1.64	0.93	1.13
	TS	41.16	33.54	38.29	50.70	65.66	64.96	37.97	49.77	11.00	0.28	0.03	1.13
	Denoise TS	17.03	21.01	58.60	64.54	17.77	21.91	58.39	65.72	3.60	4.37	1.02	1.04
	Ours	**1.52**	**4.57**	**2.17**	**2.46**	**1.84**	**2.46**	**1.65**	**2.73**	**1.36**	**1.67**	**0.30**	**0.31**

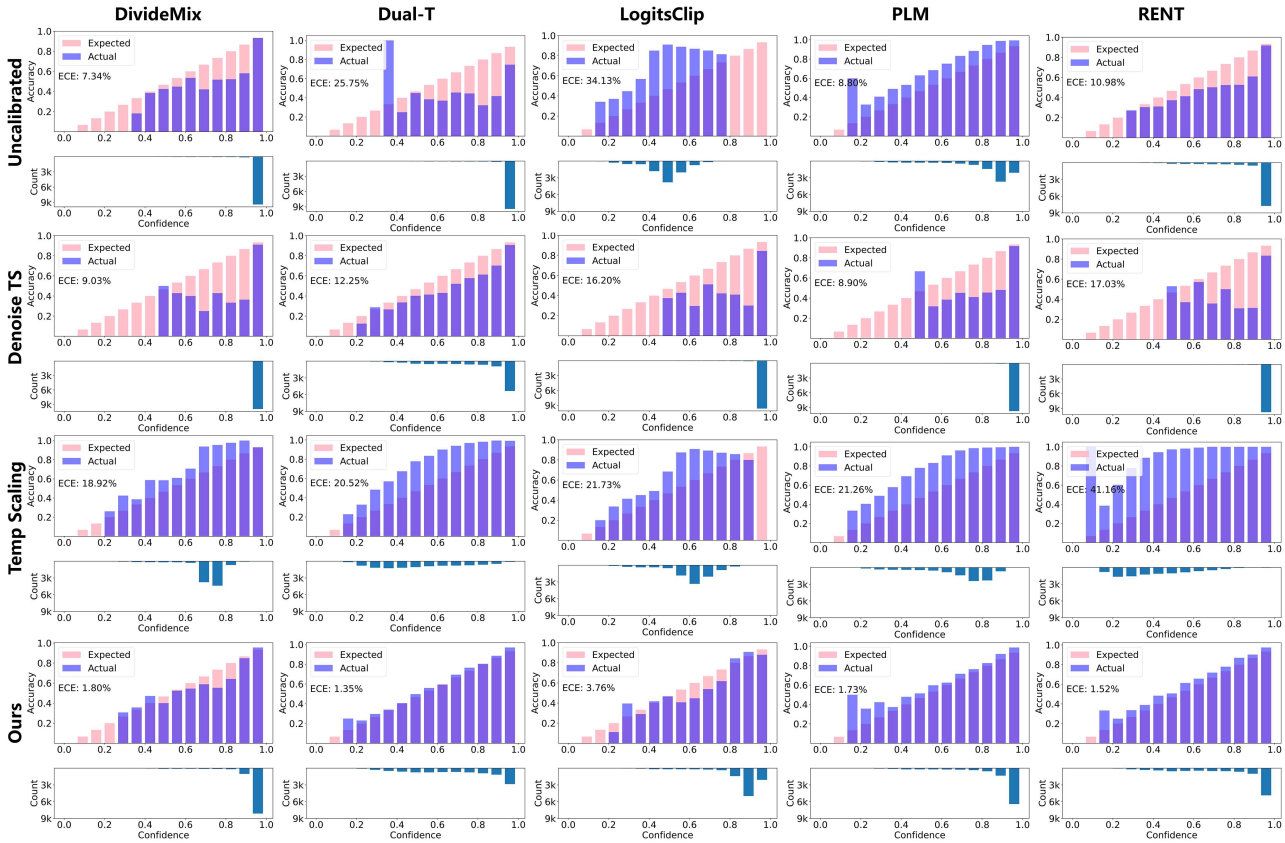

Figure 6 (Color online) Reliability diagram of LNL methods. ResNet-50 model trained on CIFAR-10 under Sym-20% noise.

Figure 7(a) confirms this: the estimated transfer matrix is an approximate identity matrix. Our method scales the logits by the approximate identity matrix, which is nearly equivalent to applying no scaling at all. Consequently,

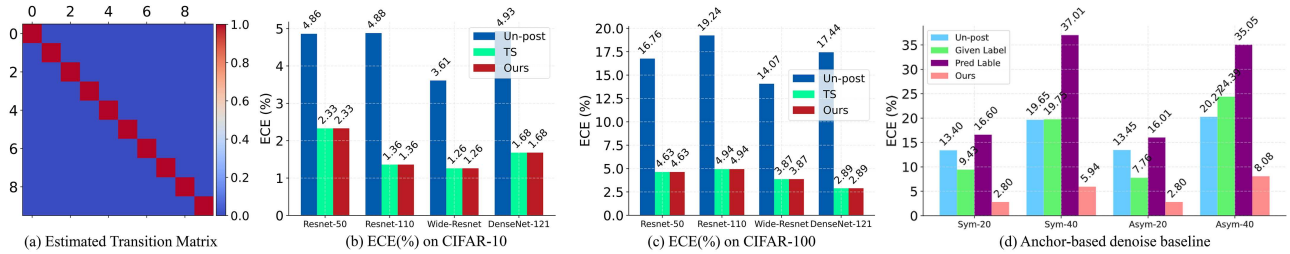

Figure 7 (Color online) TransTS's adaptability to clean data. (a) shows the estimated transition matrix from the clean dataset, and is diagonal mean is almost equal to 1. (b) and (c) present the ECE scores under MMCE. These results demonstrate that when the dataset is clean, our method adaptively reduces to TS without compromising its calibration performance. In addition, (d) is the result of anchor-based denoising baseline on CIFAR-10.

Table 8 Parameter analysis of τ. The τ represents the percentile in noise transition matrix estimation, defined in (11). The $T(x)$ error denotes the transition matrix error, defined as the difference between the average diagonal value of the estimated transition matrix and that of the true transition matrix. The best score is bold.

τ	Un-post	90%	91%	92%	93%	94%	95%	96%	97%	98%	99%	100%
$T(x)$ error	–	−0.652	−0.491	−0.185	−0.060	−0.007	+0.018	+0.034	+0.048	+0.061	+0.078	+0.138
ECE	13.4	12.32	11.47	9.02	4.46	**2.80**	4.07	4.7	4.72	5.83	6.88	9.44

our method adaptively degrades to the original TS. As shown in Figures 7(b) and (c), our method performs the same as the original TS. This demonstrates that our method has strong adaptability to clean datasets.

5.5 Analysis on estimation of noise transition matrix

In this section, we conduct more analysis on the estimation of the noise transition matrix. First, we introduced a baseline using an anchor-derived validation set to further demonstrate the advantages of our method, see Figure 7(d). In this section, the selected anchor points $\hat{\mathcal{X}}$ are regarded as anchor-driven clean data (approximation). Among them, the $x_i \in \hat{\mathcal{X}}$, but y_i has two options: predicted label or given noisy label. We evaluated them separately. The results reveal that the original TS fails to achieve effective calibration, as it remains only approximately clean without guaranteed noise elimination. Moreover, minimizing the NLL between predicted labels and logits amplifies overconfidence. These findings collectively underscore our method's efficacy in calibrating DNN classifiers across both noisy and clean environments. Furthermore, we also performed a parameter analysis to present the impact of imperfect noise transition matrix on the calibration, see Table 8. The results show that an accurate noise transition matrix provides a direct benefit for the calibration. For example, when τ is 94, the $T(x)$ error is only -0.007, which means that the estimated noise transition matrix is closer to the true noise transition matrix, and the ECE score is the best at this time. When τ is 90, the $T(x)$ error is -0.652, and the ECE score is the worst at this time, only 12.32, which is only slightly better than the uncalibrated one (Un-post). Fortunately, TransTS can accurately estimate the noise transition matrix of noisy data and thus achieve better calibration performance.

6 Related work

Probability calibration for DNNs. Existing calibration methods can be broadly categorized into two classes: built-in methods and post-hoc methods. Built-in methods calibrate the model during the training process through the designed learning objectives, such as maximum mean calibration error (MMCE) [47], sample-dependent focal loss (FLSD53) [6, 8], adaptive focal loss (AdaFocal) [7], dual focal loss (DualFocal) [8], class adaptive label smoothing (CALS) [56], and margin-based label smoothing (MbLs) [20]. However, the built-in methods require modification of the training process, which is difficult to decouple from label noise. Post-hoc methods aim to calibrate confidence for already-trained models, which decouple accuracy optimization and calibration. These methods include histogram binning (HB) [48], Bayesian binning into quantiles (BBQ) [32], vector scaling (VS) [49], TS [12], splines [17], calibrated prediction with covariate shift (CPCS) [19], class-based temperature scaling (CTS) [50], ensemble temperature scaling (ETS) [6], parameterized temperature scaling (PTS) [16], scaling of class-wise training losses (SCTL) [51], and top-versus-all (TvA) [18]. Due to their simplicity and no need to retrain the classifier, the post-hoc methods show more promise for calibrating DNN classifiers under label noise. Therefore, in this paper, we aim to study probability calibration under label noise by combining existing post-hoc calibration methods.

Learning with noisy labels. LNL aims to mitigate the impact of noisy label data during training, thereby improving model accuracy. Among these methods, sample selection and loss correction have attracted widespread attention. Sample selection methods optimize training by dynamically removing noisy labeled data. For example, Ref. [29] divided the data into labeled and unlabeled subsets for semi-supervised training; Ref. [57] minimized the KL divergence between the outputs of two networks to align their predictions more closely with the true labels. Ref. [58] employed evidence deep learning to select reliable samples based on prediction uncertainty. Loss correction methods use a noise transition matrix to adjust the loss function, ensuring the classifier converges toward the true data distribution. For instance, Ref. [53] proposed a resampling strategy to better utilize transition matrices; Ref. [2] improved noise-aware class posterior estimation through a partial label learning framework; Ref. [40] estimated transition matrices in two stages to improve accuracy; and Ref. [52] introduced a logit clipping mechanism to enhance robustness against noisy labels. It is worth noting that the pursuit of improved accuracy in LNL represents only one aspect of the problem. The complementary challenge—calibrating DNN classifiers under label noise—also warrants further investigation.

7 Conclusion

This paper focuses on probability calibration under label noise, a critical aspect for deploying DNNs in real-world applications. Specifically, we observe that label noise can deteriorate model calibration in both built-in and post-hoc methods. Furthermore, we argue that label noise compels temperature scaling to learn an aggressive scaling factor T. This aggressive factor leads to over-calibration. To address this issue, we propose the adaptive transitional calibration method (TransTS). TransTS introduces a noise transition matrix, which adaptively scales the output probabilities, thereby obtaining a moderate scaling factor T and alleviating over-calibration. We also show that TransTS essentially builds a consistent calibrator that can converge to its counterpart learned on clean data. Finally, we show that TransTS achieves better calibration through experiments on a variety of cases. Finally, we validate the effectiveness of TransTS through experiments across multiple datasets, demonstrating improved calibration performance in terms of metrics such as ECE, AdaECE, and ClasswiseECE. Future work will explore building an end-to-end framework that simultaneously optimizes for both accuracy and calibration under label noise, eliminating the need for post-hoc calibration.

Acknowledgements This work was partially supported by Key Research and Development Project in Shaanxi Province (Grant No. 2023GXLH-024), National Natural Science Foundation of China (Grant Nos. 62476215, 62302380, 62037001, 62137002, 62192781), and China Postdoctoral Science Foundation (Grant No. 2023M742789).

References

1 Zhang Z, Shi B, Zhang H, et al. Nerco: a contrastive learning based two-stage Chinese ner method. In: Proceedings of IJCAI, 2023
2 Zhao R, Shi B, Ruan J, et al. Estimating noisy class posterior with part-level labels for noisy label learning. In: Proceedings of CVPR, 2024
3 Peng Z, Luo M, Li J, et al. A deep multi-view framework for anomaly detection on attributed networks. IEEE Trans Knowl Data Eng, 2022, 34: 2539–2552
4 Tao L, Zhu Y, Guo H, et al. A benchmark study on calibration. In: Proceedings of ICLR, 2024
5 Blasiok J, Nakkiran P. Smooth ECE: principled reliability diagrams via kernel smoothing. In: Proceedings of ICLR, 2024
6 Mukhoti J, Kulharia V, Sanyal A, et al. Calibrating deep neural networks using focal loss. In: Proceedings of NeurIPS, 2020
7 Ghosh A, Schaaf T, Gormley M. Adafocal: calibration-aware adaptive focal loss. In: Proceedings of NeurIPS, 2022
8 Tao L, Dong M, Xu C. Dual focal loss for calibration. In: Proceedings of ICML, 2023
9 Cao C, Liu F, Tan H, et al. Deep learning and its applications in biomedicine. Genomics Proteomics BioInf, 2018, 16: 17–32
10 Huang X, Kroening D, Ruan W, et al. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. Comput Sci Rev, 2020, 37: 100270
11 Ozbayoglu A M, Gudelek M U, Sezer O B. Deep learning for financial applications: a survey. Appl Soft Comput, 2020, 93: 106384
12 Guo C, Pleiss G, Sun Y, et al. On calibration of modern neural networks. In: Proceedings of ICML, 2017
13 Chidambaram M, Ge R. On the limitations of temperature scaling for distributions with overlaps. In: Proceedings of ICLR, 2024
14 Wang T, Wang Y, Zhou J, et al. From aleatoric to epistemic: exploring uncertainty quantification techniques in artificial intelligence. ArXiv:2501.03282
15 Zhang J, Kailkhura B, Han T Y J. Mix-n-match: ensemble and compositional methods for uncertainty calibration in deep learning. In: Proceedings of ICML, 2020
16 Tomani C, Cremers D, Buettner F. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In: Proceedings of ECCV, 2022
17 Gupta K, Rahimi A M, Ajanthan T, et al. Calibration of neural networks using splines. ArXiv:2006.12800
18 Coz A L, Herbin S, Adjed F. Confidence calibration of classifiers with many classes. In: Proceedings of NeurIPS, 2024
19 Park S, Bastani O, Weimer J, et al. Calibrated prediction with covariate shift via unsupervised domain adaptation. In: Proceedings of AISTATS, 2020
20 Liu B, Ben Ayed I, Galdran A, et al. The devil is in the margin: margin-based label smoothing for network calibration. In: Proceedings of CVPR, 2022
21 Cheng J, Vasconcelos N. Calibrating deep neural networks by pairwise constraints. In: Proceedings of CVPR, 2022
22 Hui L, Belkin M. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. In: Proceedings of ICLR, 2021
23 Neo D, Winkler S, Chen T. Maxent loss: constrained maximum entropy for calibration under out-of-distribution shift. In: Proceedings of AAAI, 2024

24 Song H, Kim M, Park D, et al. Learning from noisy labels with deep neural networks: a survey. IEEE Trans Neural Netw Learn Syst, 2023, 34: 8135–8153
25 Wei Q, Feng L, Sun H, et al. Fine-grained classification with noisy labels. In: Proceedings of CVPR, 2023
26 Huang Z, Zhang J, Shan H. Twin contrastive learning with noisy labels. In: Proceedings of CVPR, 2023
27 Li S, Xia X, Zhang H, et al. Estimating noise transition matrix with label correlations for noisy multi-label learning. In: Proceedings of NeurIPS, 2022
28 Xia X, Liu T, Wang N, et al. Are anchor points really indispensable in label-noise learning? In: Proceedings of NeurIPS, 2019
29 Li J, Socher R, Hoi S C. Dividemix: learning with noisy labels as semi-supervised learning. In: Proceedings of ICLR, 2020
30 Wang D B, Zhang M L. Rethinking calibration of deep neural networks: do not be afraid of overconfidence. In: Proceedings of NIPS, 2021
31 Wang D B, Zhang M L. Calibration bottleneck: over-compressed representations are less calibratable. In: Proceedings of ICML, 2024
32 Naeini M P, Cooper G, Hauskrecht M. Obtaining well calibrated probabilities using Bayesian binning. In: Proceedings of AAAI, 2015
33 Xia X, Han B, Wang N, et al. Extended: learning with mixed closed-set and open-set noisy labels. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 3047–3058
34 Cheng D, Liu T, Ning Y, et al. Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In: Proceedings of CVPR, 2022
35 Yang S, Yang E, Han B, et al. Estimating instance-dependent Bayes-label transition matrix using a deep neural network. In: Proceedings of ICML, 2022
36 Han B, Yao J, Niu G, et al. Masking: a new perspective of noisy supervision. In: Proceedings of NeurIPS, 2018
37 Han B, Yao Q, Yu X, et al. Co-teaching: robust training of deep neural networks with extremely noisy labels. In: Proceedings of NeurIPS, 2018
38 Menon A, van Rooyen B, Ong C S, et al. Learning from corrupted binary labels via class-probability estimation. In: Proceedings of International Conference on Machine Learning, 2015. 125–134
39 Patrini G, Rozza A, Krishna Menon A, et al. Making deep neural networks robust to label noise: a loss correction approach. In: Proceedings of CVPR, 2017
40 Yao Y, Liu T, Han B, et al. Dual T: reducing estimation error for transition matrix in label-noise learning. In: Proceedings of NeurIPS, 2020
41 Krizhevsky A, Hinton G, et al. Learning multiple layers of features from tiny images. 2009. https://api.semanticscholar.org/CorpusID:18268744
42 Song H, Kim M, Lee J G. SELFIE: refurbishing unclean samples for robust deep learning. In: Proceedings of ICML, 2019
43 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of CVPR, 2016
44 Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv:2010.11929
45 Zagoruyko S, Komodakis N. Wide residual networks. ArXiv:1605.07146
46 Huang G, Liu Z, van der Maaten L, et al. Densely connected convolutional networks. In: Proceedings of CVPR, 2017
47 Kumar A, Sarawagi S, Jain U. Trainable calibration measures for neural networks from kernel mean embeddings. In: Proceedings of ICML, 2018
48 Zadrozny B, Elkan C. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: Proceedings of ICML, 2001
49 Zou Y, Deng W, Zheng L. Adaptive calibrator ensemble: navigating test set difficulty in out-of-distribution scenarios. In: Proceedings of ICCV, 2023
50 Frenkel L, Goldberger J. Network calibration by class-based temperature scaling. In: Proceedings of the 29th European Signal Processing Conference (EUSIPCO), 2021
51 Jung S, Seo S, Jeong Y, et al. Scaling of class-wise training losses for post-hoc calibration. In: Proceedings of ICML, 2023
52 Wei H, Zhuang H, Xie R, et al. Mitigating memorization of noisy labels by clipping the model prediction. In: Proceedings of ICML, 2023
53 Bae H, Shin S, Na B, et al. Dirichlet-based per-sample weighting by transition matrix for noisy label learning. In: Proceedings of ICLR, 2024
54 Nguyen K, O'Connor B. Posterior calibration and exploratory analysis for natural language processing models. ArXiv:1508.05154
55 Kull M, Perello Nieto M, Kängsepp M, et al. Beyond temperature scaling: obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In: Proceedings of NeurIPS, 2019
56 Liu B, Rony J, Galdran A, et al. Class adaptive network calibration. In: Proceedings of CVPR, 2023
57 Wei H, Feng L, Chen X, et al. Combating noisy labels by agreement: a joint training method with co-regularization. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020
58 Zong C C, Wang Y W, Xie M K, et al. Dirichlet-based prediction calibration for learning with noisy labels. In: Proceedings of AAAI, 2024