Pi-DUAL: Using privileged information to distinguish clean from noisy labels

Ke Wang[1] Guillermo Ortiz-Jimenez[2 3] Rodolphe Jenatton[4 5] Mark Collier[6] Efi Kokiopoulou[6]
Pascal Frossard[1]

ICML 2024

Motivation

Noisy label methods :

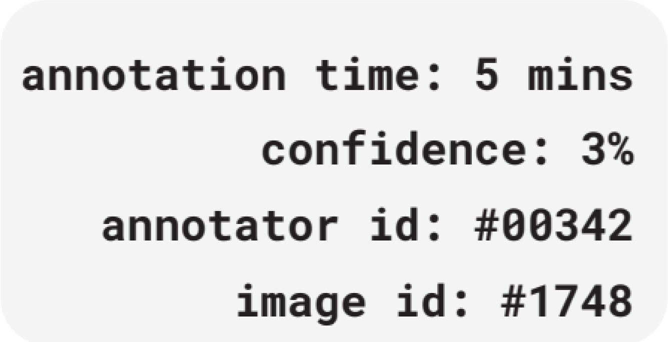
- Explicitly model the noise signal. Noise modeling techniques aim to learn the function that governs the noisy annotation process explicitly during training, inverting it during inference to obtain the clean labels.
- Rely on implicit network dynamics to correct or ignore the wrong labels. Implicit-dynamics based approaches operate under the assumption that wrong labels are harder to learn than the correct labels.

Introduction

Noisy label learning with privileged information (PI):

PI is defined as additional features available at training time but not at test time. PI may include information about the annotator, such as their ID or experience; or about the process itself, such as the annotation duration or confidence.

However, current PI methods can sometimes lag behind in performance with respect to no-PI baselines. The main reason is that these methods still try to learn the noise predictive distribution $\mathrm{p}(\tilde{y}|x)$ by marginalizing a in $\mathrm{p}(\tilde{y}|x,a)$, when they should aim to learn the clean distribution $\mathrm{p}(y|x)$ directly.



```
annotation time: 5 mins
         confidence: 3%
     annotator id: #00342
         image id: #1748
```

Privileged Information

Method

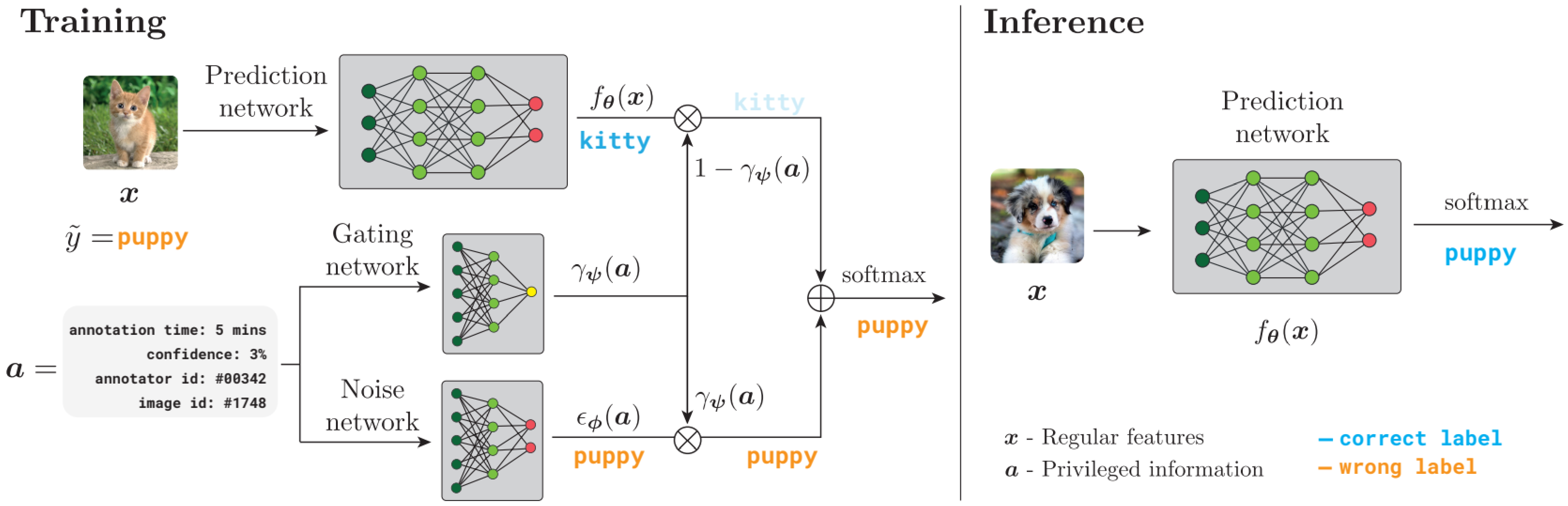


Figure 1. **Illustration of the architecture of Pi-DUAL.** (Left) During training, Pi-DUAL fits the noisy target label $\tilde{y}$ combining the output of a prediction network (which takes the regular features $\boldsymbol{x}$ as input) and a noise network (which takes the PI $\boldsymbol{a}$ as input). The outputs of these sub-networks are weighted based on the output of a gating network (which also has $\boldsymbol{a}$ as input) and then passed through a softmax operator to obtain the predictions. (Right) During inference, when only $\boldsymbol{x}$ is available, Pi-DUAL does not need access to PI and simply uses the prediction network to predict the clean target y.

Method

Method Description

During training, Pi-DUAL factorizes its output logits into two terms, i.e.,

$$h_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}}(\boldsymbol{x},\boldsymbol{a}) = [1-\gamma_{\boldsymbol{\psi}}(\boldsymbol{a})]f_{\boldsymbol{\theta}}(\boldsymbol{x}) + \gamma_{\boldsymbol{\psi}}(\boldsymbol{a})\epsilon_{\boldsymbol{\phi}}(\boldsymbol{a}),$$

Moreover, we augment the available PI features with a unique random identifier for each training sample to help the network explain away the missing factors of the noise using this identifier. During inference, when PI is not available, Pi-DUAL relies solely on f_{θ}(x) to predict the clean label y.

Previous methods tend to directly expose the no-PI term $\mathrm{f}_{\theta}(x)$ to the noisy labels, e.g., through $\mathrm{L}(\mathrm{f}_{\theta}(x), \tilde{y})$ which can thus lead to an overfitting to the noisy labels based on x. In contrast, Pi-DUAL instead solves

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}} \sum_{(\boldsymbol{x},\tilde{y};\boldsymbol{a})\in\mathcal{D}} \mathcal{L}\left(\mathrm{softmax}\left(h_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}}(\boldsymbol{x},\boldsymbol{a})\right), \tilde{y}\right),$$

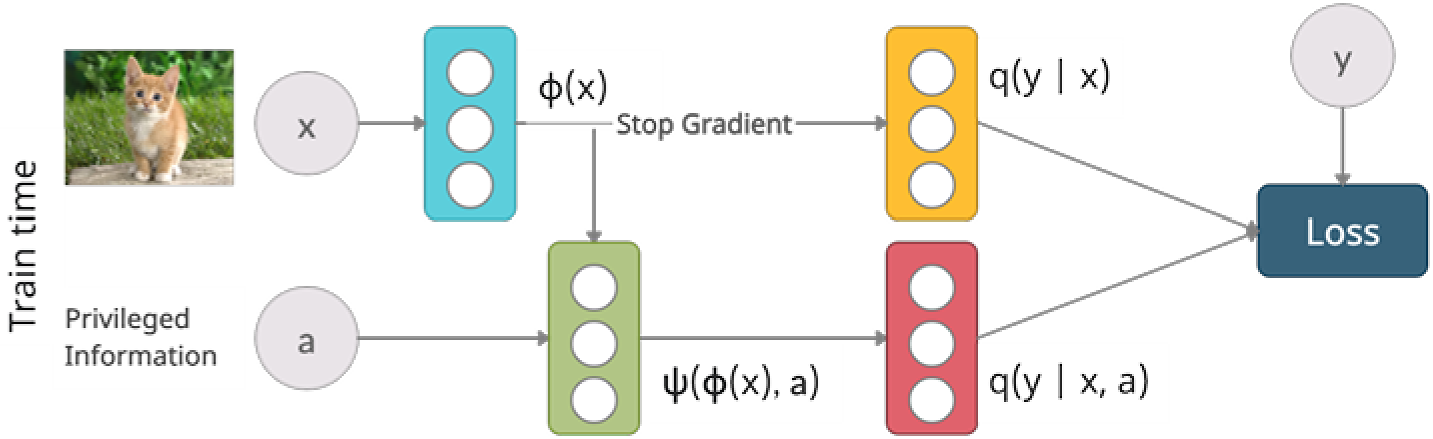
and never explicitly forces $\mathrm{f}_{\theta}(x)$ to fit all $\tilde{y}$’s.

Method

Method Description

Our design allows the model to predict clean label for all training samples without incurring loss penalty, as it can fit the residual noise signal with $\epsilon_{\phi}(\boldsymbol{a})$.

Another important advantage of Pi-DUAL is that it explicitly learns to model noise signal in training set. This makes it more interpretable than implicit-dynamics methods like TRAM, and puts it on par with state-of-the-art noise modeling methods. However, as Pi-DUAL can leverage PI to model noise signal, it exhibits a much better noise detection performance than no-PI methods, while at the same time allowing it to scale to datasets with millions of datapoints, as it does not require to store individual parameters for each sample in the training set to effectively learn the label noise.



TRAM

Method

Noise Detection

After training, we collect confidence estimates of the prediction network on the observed noisy labels $\tilde{y}$, i.e., with $soft\mathrm{max}(\mathrm{f}_{\theta}(x))[\tilde{y}]$, for the training samples and threshold the confidences to distinguish the correctly and incorrectly labeled examples.

If its confidence on an observed label is high, then it is highly likely that the sample is correctly labeled, i.e., $\tilde{y}$ = y; but if it is low, then probably the label $\tilde{y}$ is wrong.

Method

Theoretical Insights

To show that Pi-DUAL is a robust estimator in the presence of label noise as its risk depends less severely on the number of wrong labels, we study the theoretical behavior of the predictor $h_{\theta,\varphi,\psi}(x,a)$ within a simplified linear regression setting. More specifically, we consider the setting where the clean and noisy targets are respectively generated from two Gaussian distributions $\mathrm{N}(x^T w^*, \sigma^2)$ and $\mathrm{N}(a^T v^*, \sigma^2)$, for two weight vectors (w^*, v^*) parameterizing linearly their means.

We compare two estimators, Pi-DUAL and an ordinary least squares estimator (OLS) that ignores the side information a.

Consider n samples from the above Gaussian models with targets $\mathrm{y} = \gamma^* X w^* + (I - \gamma^*) A v^* + \varepsilon$, the contributions of the standard and PI features are respectively $Xw^* \in R^n$ and $Av^* \in R^n$, while $\gamma^* \in \{0,1\}^{n\times n}$ is a diagonal mask that indicates which contribution each entry in y corresponds to. Denoting $\delta^* = Xw^* - Av^*$, it can be shown that the risk of the OLS estimator has a bias term scaling with $\mathrm{O}((I - \gamma^*)\delta^*)$, while the risk of Pi-DUAL using an arbitrary diagonal mask $\gamma \in \{0,1\}^{n\times n}$ has a bias term that depends on $\mathrm{O}((\gamma^* - \gamma)\delta^*)$, which only scales with the number of disagreements with respect to the ground-truth γ^*.

Experiments

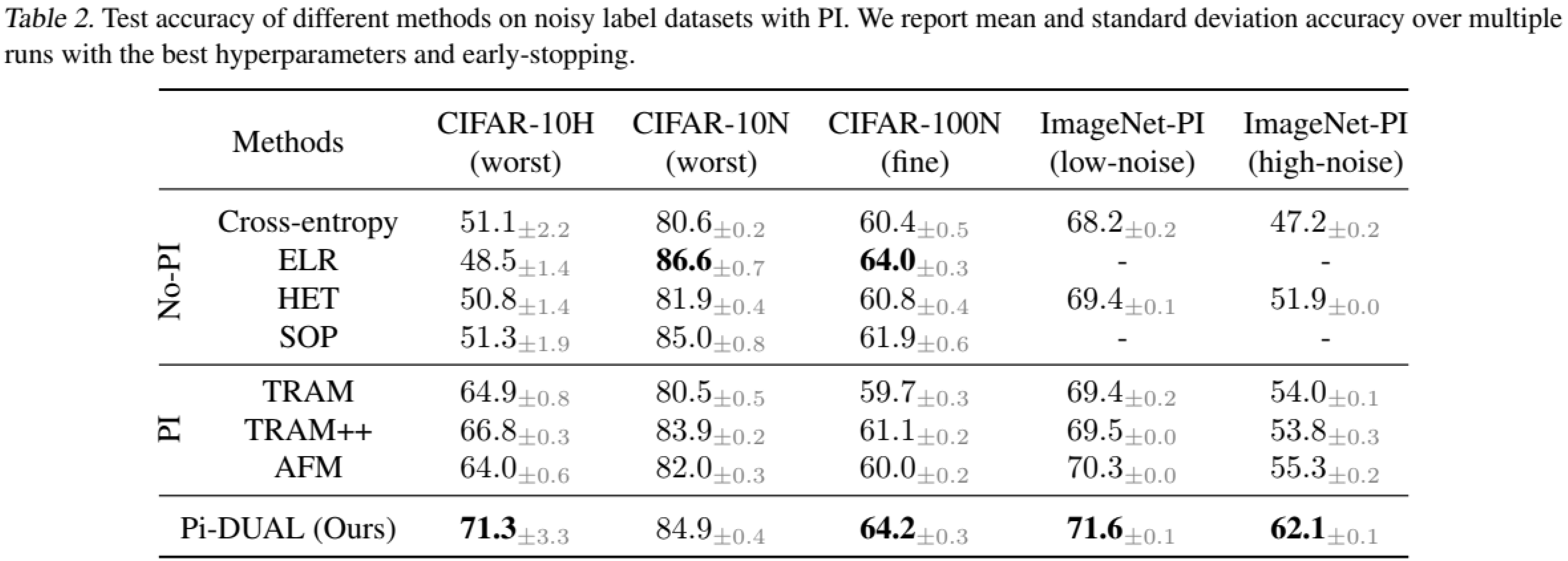
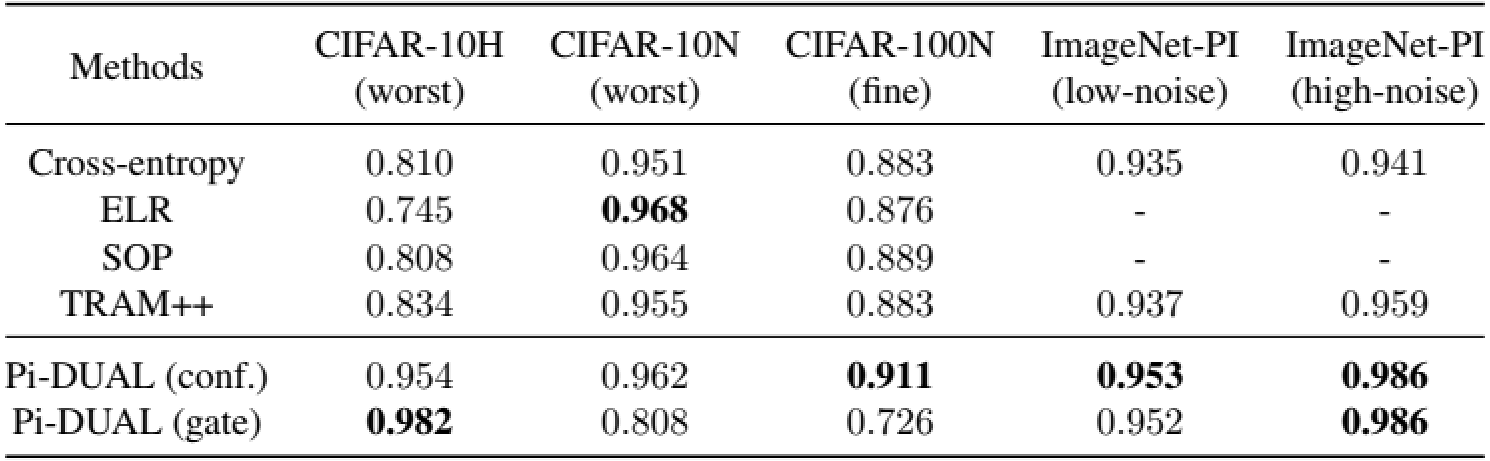


Table 2. Test accuracy of different methods on noisy label datasets with PI. We report mean and standard deviation accuracy over multiple runs with the best hyperparameters and early-stopping.

	Methods	CIFAR-10H (worst)	CIFAR-10N (worst)	CIFAR-100N (fine)	ImageNet-PI (low-noise)	ImageNet-PI (high-noise)
No-PI	Cross-entropy	51.1$_{\pm 2.2}$	80.6$_{\pm 0.2}$	60.4$_{\pm 0.5}$	68.2$_{\pm 0.2}$	47.2$_{\pm 0.2}$
	ELR	48.5$_{\pm 1.4}$	**86.6**$_{\pm 0.7}$	**64.0**$_{\pm 0.3}$	-	-
	HET	50.8$_{\pm 1.4}$	81.9$_{\pm 0.4}$	60.8$_{\pm 0.4}$	69.4$_{\pm 0.1}$	51.9$_{\pm 0.0}$
	SOP	51.3$_{\pm 1.9}$	85.0$_{\pm 0.8}$	61.9$_{\pm 0.6}$	-	-
PI	TRAM	64.9$_{\pm 0.8}$	80.5$_{\pm 0.5}$	59.7$_{\pm 0.3}$	69.4$_{\pm 0.2}$	54.0$_{\pm 0.1}$
	TRAM++	66.8$_{\pm 0.3}$	83.9$_{\pm 0.2}$	61.1$_{\pm 0.2}$	69.5$_{\pm 0.0}$	53.8$_{\pm 0.3}$
	AFM	64.0$_{\pm 0.6}$	82.0$_{\pm 0.3}$	60.0$_{\pm 0.2}$	70.3$_{\pm 0.0}$	55.3$_{\pm 0.2}$
	Pi-DUAL (Ours)	**71.3**$_{\pm 3.3}$	84.9$_{\pm 0.4}$	**64.2**$_{\pm 0.3}$	**71.6**$_{\pm 0.1}$	**62.1**$_{\pm 0.1}$

Experiments

Table 3. AUC of different noise detection methods based on confidence thresholding of the network predictions on noisy labels or thresholding of the gating network's output (for Pi-DUAL).



Methods	CIFAR-10H (worst)	CIFAR-10N (worst)	CIFAR-100N (fine)	ImageNet-PI (low-noise)	ImageNet-PI (high-noise)
Cross-entropy	0.810	0.951	0.883	0.935	0.941
ELR	0.745	**0.968**	0.876	-	-
SOP	0.808	0.964	0.889	-	-
TRAM++	0.834	0.955	0.883	0.937	0.959
Pi-DUAL (conf.)	0.954	0.962	**0.911**	**0.953**	**0.986**
Pi-DUAL (gate)	**0.982**	0.808	0.726	0.952	**0.986**

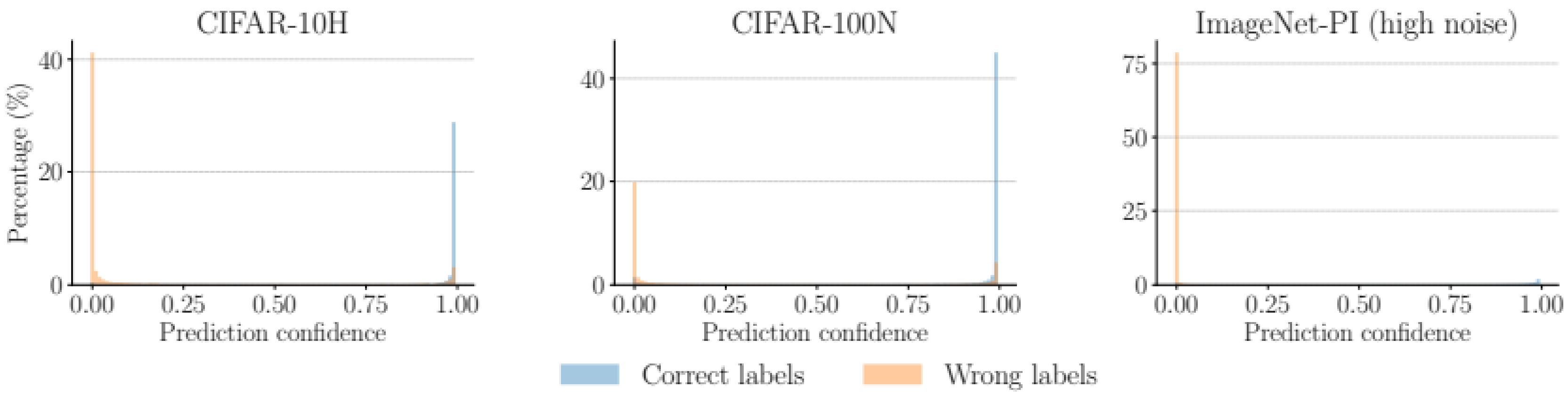


Figure 2. Distribution for the prediction network's confidence on the observed noisy labels for several datasets, separated by correctly and wrongly labeled samples.

Experiments

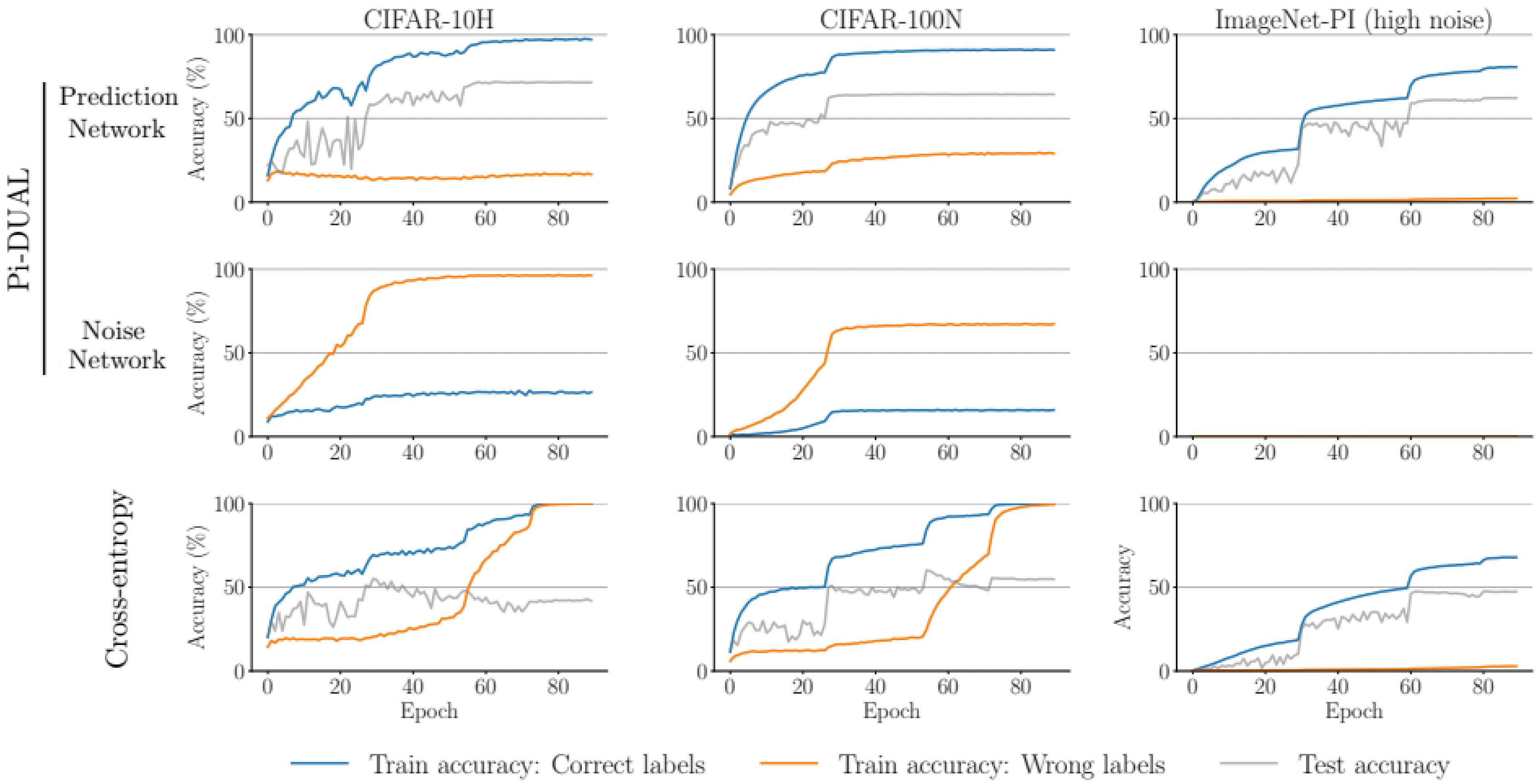


Figure 3. Training curves of Pi-DUAL and cross-entropy baseline on different datasets. The first two rows show the training dynamics of prediction network and noise network respectively.We plot separately the training accuracy on clean and wrong labels and test accuracy[3].

Experiments

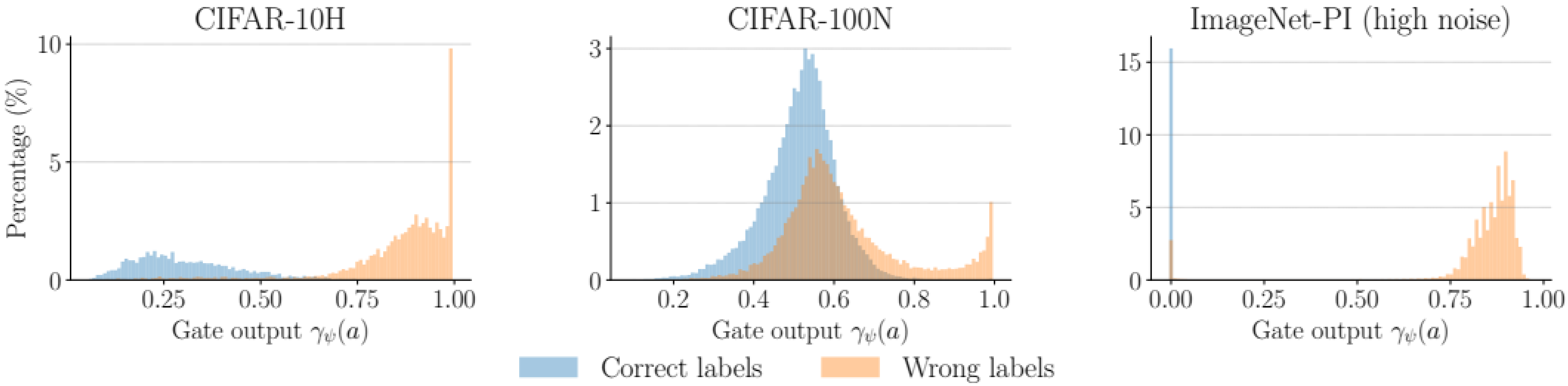


Figure 4. Distributions of $\gamma_{\boldsymbol{\psi}}(\boldsymbol{a})$ over training samples with correct and wrong labels on several datasets.

Experiments

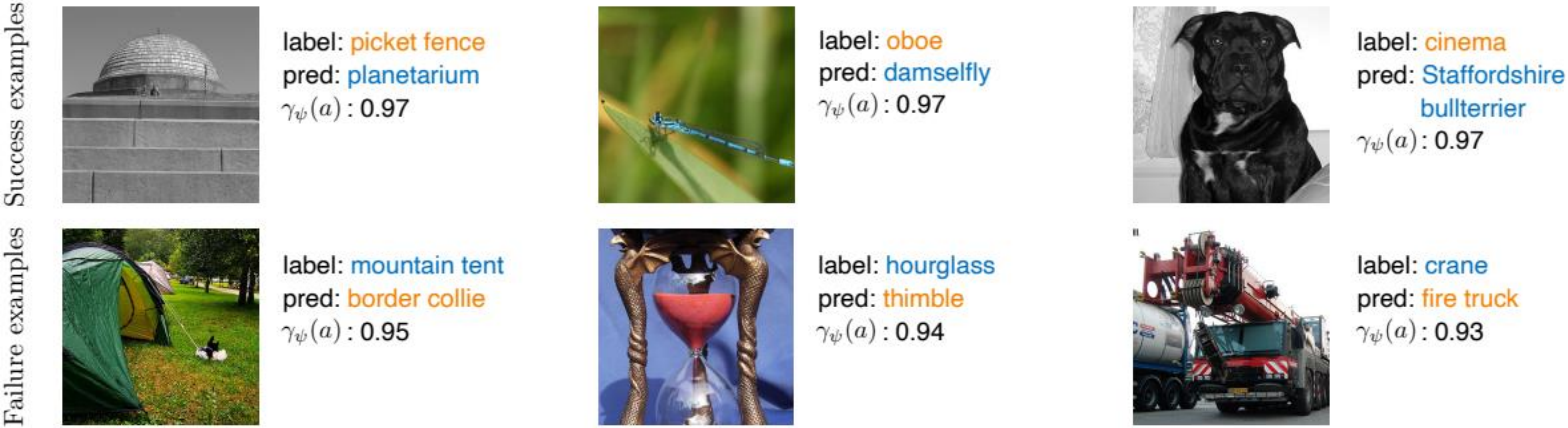


Figure 5. Examples of ImageNet-PI images that the gating network suggests are mislabeled. The first row shows samples with actually wrongly annotated labels, and the second row shows examples with correct labels but assumed to be wrong by the gating network. Here, “label" denotes the annotation label $\tilde{y}$ and “pred" the prediction by $f_{\boldsymbol{\theta}}$.

Ablation Study

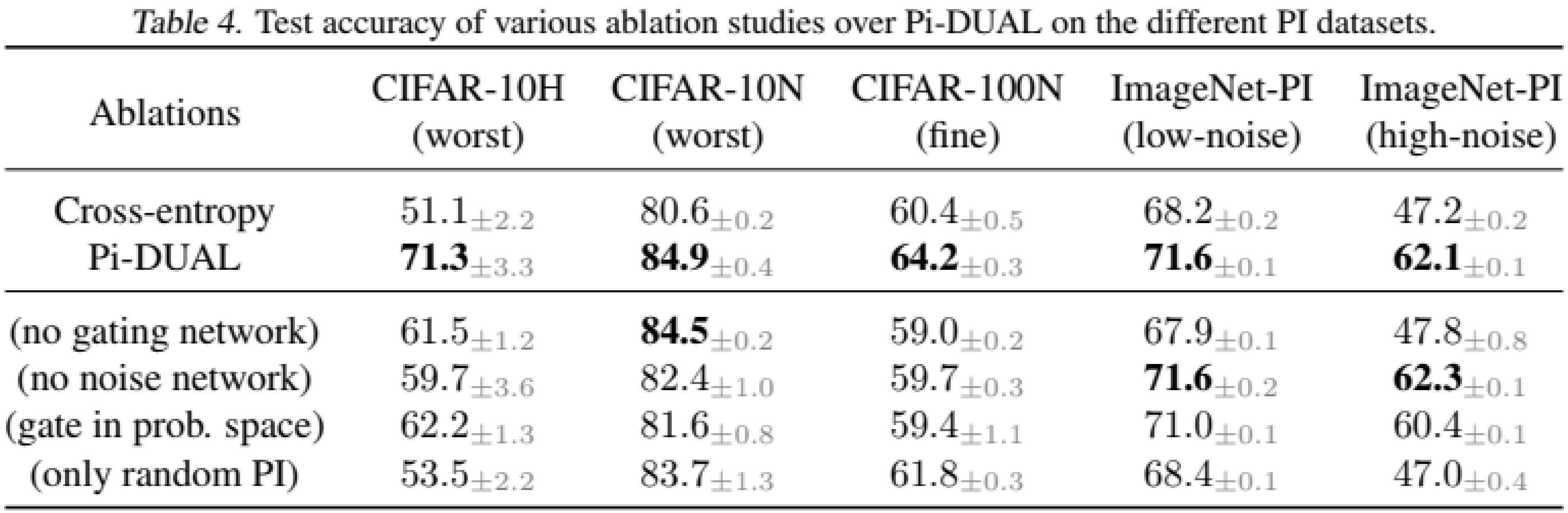


Table 4. Test accuracy of various ablation studies over Pi-DUAL on the different PI datasets.

Ablations	CIFAR-10H (worst)	CIFAR-10N (worst)	CIFAR-100N (fine)	ImageNet-PI (low-noise)	ImageNet-PI (high-noise)
Cross-entropy	$51.1_{\pm 2.2}$	$80.6_{\pm 0.2}$	$60.4_{\pm 0.5}$	$68.2_{\pm 0.2}$	$47.2_{\pm 0.2}$
Pi-DUAL	$\mathbf{71.3}_{\pm 3.3}$	$\mathbf{84.9}_{\pm 0.4}$	$\mathbf{64.2}_{\pm 0.3}$	$\mathbf{71.6}_{\pm 0.1}$	$\mathbf{62.1}_{\pm 0.1}$
(no gating network)	$61.5_{\pm 1.2}$	$\mathbf{84.5}_{\pm 0.2}$	$59.0_{\pm 0.2}$	$67.9_{\pm 0.1}$	$47.8_{\pm 0.8}$
(no noise network)	$59.7_{\pm 3.6}$	$82.4_{\pm 1.0}$	$59.7_{\pm 0.3}$	$\mathbf{71.6}_{\pm 0.2}$	$\mathbf{62.3}_{\pm 0.1}$
(gate in prob. space)	$62.2_{\pm 1.3}$	$81.6_{\pm 0.8}$	$59.4_{\pm 1.1}$	$71.0_{\pm 0.1}$	$60.4_{\pm 0.1}$
(only random PI)	$53.5_{\pm 2.2}$	$83.7_{\pm 1.3}$	$61.8_{\pm 0.3}$	$68.4_{\pm 0.1}$	$47.0_{\pm 0.4}$

Thanks