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]

Abstract

Label noise is a pervasive problem in deep learning that often compromises the generalization performance of trained models. Recently, leveraging privileged information (PI) – information available only during training but not at test time – has emerged as an effective approach to mitigate this issue. Yet, existing PI-based methods have failed to consistently outperform their no-PI counterparts in terms of preventing overfitting to label noise. To address this deficiency, we introduce Pi-DUAL, an architecture designed to harness PI to distinguish clean from wrong labels. Pi-DUAL decomposes the output logits into a prediction term, based on conventional input features, and a noise-fitting term influenced solely by PI. A gating mechanism steered by PI adaptively shifts focus between these terms, allowing the model to implicitly separate the learning paths of clean and wrong labels. Empirically, Pi-DUAL achieves significant performance improvements on key PI benchmarks (e.g., $+6.8\%$ on ImageNet-PI), establishing a new state-of-the-art test set accuracy. Additionally, Pi-DUAL is a potent method for identifying noisy samples post-training, outperforming other strong methods at this task. Overall, Pi-DUAL is a simple, scalable and practical approach for mitigating the effects of label noise in a variety of real-world scenarios with PI.

1. Introduction

Many deep learning models are trained on large noisy datasets, as obtaining cleanly labeled datasets at scale can be expensive and time consuming (Snow et al., 2008; Sheng et al., 2008). However, the presence of label noise in the training set tends to damage generalization performance as it forces the model to learn spurious associations between the input features and the noisy labels (Zhang et al., 2017; Arpit et al., 2017). To mitigate the negative effects of label noise, recent methods have primarily tried to prevent overfitting to the noisy labels, often utilising the observation that neural networks tend to first learn the clean labels before memorizing the wrong ones (Maennel et al., 2020; Baldock et al., 2021). For instance, these methods include filtering out incorrect labels, correcting them, or enforcing regularization on the training dynamics (Han et al., 2018; Liu et al., 2020; Li et al., 2020a). Other works, instead, try to capture the noise structure in an input-dependent fashion (Patrini et al., 2017; Liu et al., 2022; Collier et al., 2022; 2023).

The above methods are however designed for a standard supervised learning setting, where models are tasked to learn an association between input features $\boldsymbol{x} \in \mathbb{R}^d$ and targets $y \in \{1, \dots, K\}$ (assuming K classes) from a training set of pairs $\{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i \in [n]}$ of features and (possibly) noisy labels $\tilde{y} \in \{1, \dots K\}$. As a result, they need to model the noise in the targets as a function of $\boldsymbol{x}$. Yet, in many practical situations, the mistakes introduced during the annotation process may not solely depend on the input $\boldsymbol{x}$, but rather be mostly explained by annotation-specific side information, such as the experience of the annotator or the attention they paid while annotating. For this reason, a recent line of work (Vapnik & Vashist, 2009; Collier et al., 2022; Ortiz-Jimenez et al., 2023) has proposed to use *privileged information* (PI) to mitigate the affects of label noise. PI is defined as additional features available at training time but not at test time. It can include annotation features such as the annotator ID, the amount of time to provide the label, or their experience.

Remarkably, having access to PI at training time, even when it is not available at test time, has been shown to be an effective tool for dealing with instance-dependent label noise. Most notably, Ortiz-Jimenez et al. (2023) showed that by exploiting PI it is possible to activate positive learning shortcuts to memorize, and therefore explain away, noisy training samples, thereby improving generalization. Nevertheless, and perhaps surprisingly, current PI-based methods do not systematically outperform no-PI baselines in the presence of label noise, making them a less competitive alternative in certain cases (Ortiz-Jimenez et al., 2023).

[1]École Polytechnique Fédérale de Lausanne (EPFL) [2]Google DeepMind [3]Work done while at EPFL [4]Bioptimus [5]Work done while at Google DeepMind [6]Google Research. Correspondence to: Ke Wang <k.wang@epfl.ch>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

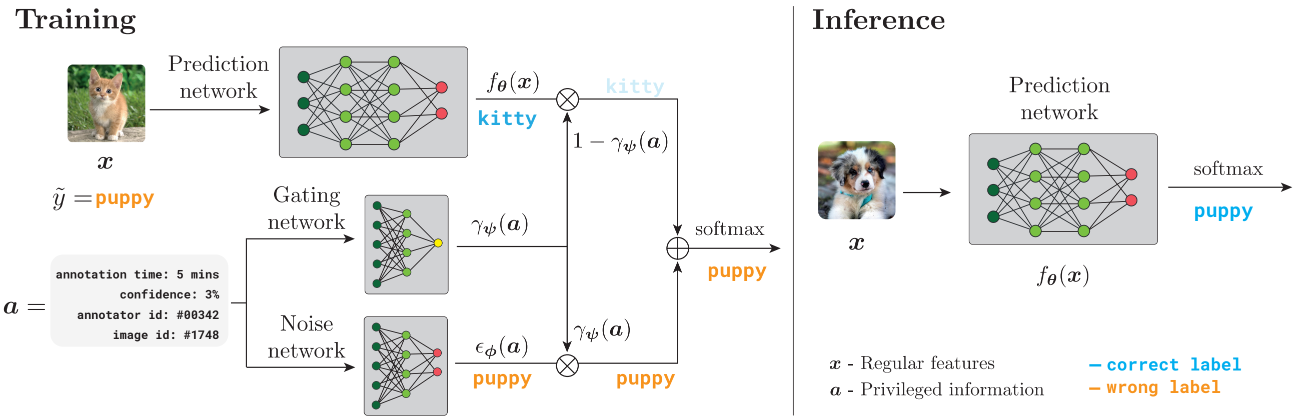

Figure 1. **Illustration of the architecture of Pi-DUAL.** (Left) During training, Pi-DUAL fits the noisy target label $\tilde{y}$ combining the output of a prediction network (which takes the regular features $\boldsymbol{x}$ as input) and a noise network (which takes the PI $\boldsymbol{a}$ as input). 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 $\mathrm{softmax}$ 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.

In this work, we aim to improve the performance of PI strategies by proposing a new PI-guided noisy label architecture: **Pi-DUAL**, a **P**rivileged **I**nformation network to **D**istinguish **U**ntrustworthy **A**nnotations and **L**abels. Specifically, during training, we propose to decompose the output logits into a weighted combination of a prediction term, that depends only on the regular features $\boldsymbol{x}$, and a noise-fitting term, that depends only on the PI features $\boldsymbol{a} \in \mathbb{R}^p$. Pi-DUAL toggles between these terms using a gating mechanism, also solely a function of $\boldsymbol{a}$, that decides if a sample should be learned primarily by the prediction network, or explained away by the noise network (see Fig. 1). This dual sub-network design adaptively routes the clean and wrong labels through the prediction and noise networks so that they are fit based on $\boldsymbol{x}$ or $\boldsymbol{a}$, respectively. This protects the prediction network from overfitting to the label noise. Pi-DUAL is simple to implement, effective, and can be trained end-to-end with minimal modifications to existing training pipelines. Unlike some previous methods Pi-DUAL also scales to training on very large datasets. Finally, in public benchmarks for learning with label noise, Pi-DUAL achieves state-of-the-art results on datasets with rich PI features ($+4.5\%$ on CIFAR-10H, $+1.3\%$ on ImageNet-PI (low-noise) and $+6.8\%$ on ImageNet-PI (high-noise)); and performs on par with previous methods on benchmarks with weak PI or no PI at all, despite not being specifically designed to work in these regimes.

Overall, the main contributions of our work are:

- We present Pi-DUAL, a novel PI method to combat label noise based on a dual path architecture that implicitly separates the noisy fitting path from the clean prediction path during training.
- We show that Pi-DUAL achieves strong performance on noisy label benchmarks without sacrificing scalability, outperforming previous state-of-the-art methods when given access to high-quality PI features.
- We provide a simple yet effective method to detect wrong labels in the training set using the prediction network of Pi-DUAL, achieving strong detection performance.

In summary, our work advances the state-of-the-art in noisy label learning by effectively leveraging privileged information through the novel Pi-DUAL architecture. Pi-DUAL can be easily integrated into any learning pipeline, requires minimal hyperparameters, and can be trained end-to-end in a single stage. Overall, Pi-DUAL is a scalable and practical approach for mitigating the effects of label noise in a variety of real-world scenarios with PI.

2. Related Work

Noisy label methods mostly fall into two broad categories: those that explicitly model the noise signal, and those that rely on implicit network dynamics to correct or ignore the wrong labels (Song et al., 2022). 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. Some methods model the annotation function using a transition matrix (Patrini et al., 2017); others model uncertainty via a heteroscedastic noise term (Collier et al., 2021; 2023); and recently, some works explicitly parameterize the label error signal as a vector for each sample in the training set (Tanaka et al., 2018; Yi & Wu, 2019; Liu et al., 2022). Implicit-dynamics

Table 1. Comparison of different representative methods to learn with label noise *vs* Pi-DUAL on several design axes: ability to leverage PI, ability to explicitly model the noise signal, parameter scalability, and whether training requires multiple models and training stages. Scalability indicates whether the number of parameters for the method remains constant regardless of the number of samples or the number of classes in the training set.

Methods	Leverage PI	Explicit noise modeling	Scalability w.r.t. num. of samples	Scalability w.r.t. num. of classes	Training complexity
Forward-T (Patrini et al., 2017)	✗	✓	✓	✗	1 model, 2 stages
Co-Teaching (Han et al., 2018)	✗	✗	✓	✓	2 models, 1 stage
Divide-Mix (Li et al., 2020a)	✗	✗	✓	✓	2 models, 1 stage
ELR (Liu et al., 2020)	✗	✗	✗	✗	1 model, 1 stage
SOP (Liu et al., 2022)	✗	✓	✗	✗	1 model, 1 stage
HET-XL (Collier et al., 2023)	✗	✓	✓	✓	1 model, 1 stage
Distill. PI (Lopez-Paz et al., 2015)	✓	✗	✓	✓	2 model, 2 stage
AFM (Collier et al., 2022)	✓	✗	✓	✓	1 model, 1 stage
TRAM++ (Ortiz-Jimenez et al., 2023)	✓	✗	✓	✓	1 model, 1 stage
Pi-DUAL (Ours)	✓	✓	✓	✓	1 model, 1 stage

based approaches, on the other hand, operate under the assumption that wrong labels are harder to learn than the correct labels (Zhang et al., 2017; Maennel et al., 2020). Using this intuition, different methods have come up with different heuristics to correct (Jiang et al., 2018; Han et al., 2018; Yu et al., 2019) or downweight (Liu et al., 2020; Menon et al., 2020; Bai et al., 2021) the influence of wrong labels during training. This has sometimes led to very complex methods that require multiple stages of training (Patrini et al., 2017; Bai et al., 2021; Albert et al., 2023; Wang et al., 2023), higher computational cost (Han et al., 2018; Jiang et al., 2018; Han et al., 2018; Yu et al., 2019), and many additional parameters that do not scale well to large datasets (Yi & Wu, 2019; Liu et al., 2020; 2022).

The introduction of privileged information (PI) offers an alternative dimension to tackle the noisy label problem (Hernández-Lobato et al., 2014; Lopez-Paz et al., 2015; Collier et al., 2022). In this regard, Ortiz-Jimenez et al. (2023) showed that most PI methods work as implicit-dynamics approaches. They rely on the use of PI to enable learning shortcuts, to avoid memorizing the incorrect labels using the regular features. Moreover, these approaches are attractive for their scalability, as they usually avoid the introduction of extra training stages or parameters. 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 $p(\tilde{y}|\boldsymbol{x})$ by marginalizing $\boldsymbol{a}$ in $p(\tilde{y}|\boldsymbol{x}, \boldsymbol{a})$, when they should actually aim to learn the clean distribution $p(y|\boldsymbol{x})$ directly. However, prior PI methods do not have an explicit mechanism to identify or correct the wrong labels.

Our proposed method, Pi-DUAL, tries to circumvent these issues by explicitly modeling the *clean distribution*, exploiting the ability of PI to distinguish clean and wrong labels. Our design allows Pi-DUAL to scale effectively across large datasets and diverse class distributions, while maintaining high performance and low training complexity as seen in Tab. 1. We further note that our design is reminiscent of mixtures of experts (MoE) that were shown to be a competitive architecture for language modeling (Shazeer et al., 2017) and computer vision (Riquelme et al., 2021). By analogy, we can see Pi-DUAL as an MoE containing a single MoE layer with two heterogeneous experts—the prediction and noise networks—located at the logits of the model and with a dense gating.

3. Pi-DUAL

3.1. Noise Modeling

In traditional supervised learning, we typically assume that there exists a groundtruth function $f^\star : \mathcal{X} \to \mathcal{Y}$ which maps input features $\boldsymbol{x} \in \mathcal{X}$ to labels $y \in \mathcal{Y}$ where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{1, \dots, K\}$. However, the labels in real-world scenarios are usually gathered via a noisy annotation process.

In this work, we model this annotation process as a function of some, possibly unknown, side information $\boldsymbol{a} \in \mathcal{A}$, which explains away the noise from the training labels. This side information could be anything, from the experience of the annotator, to their intrinsic motivation. The important modeling aspect is that given this side information one should be able to tell whether a label is incorrect or not, and the type of mistake that was made. We can model this process mathematically as a function $h : \mathcal{X} \times \mathcal{A} \to \mathcal{Y}$ that maps the input features and the side information to the noisy human label $\tilde{y}$. We assume that the mistakes in the annotation process depend only on $\boldsymbol{a}$, i.e.,

$$\tilde{y} = h(\boldsymbol{x}, \boldsymbol{a}) = [1 - \gamma(\boldsymbol{a})] f^\star(\boldsymbol{x}) + \gamma(\boldsymbol{a}) \epsilon(\boldsymbol{a}) \quad (1)$$

Here $\gamma : \mathcal{A} \to \{0, 1\}$ acts as a switch between clean and wrong labels, and $\epsilon : \mathcal{A} \to \mathcal{Y}$ models the incorrect labelling function. Consequently, the training dataset $\mathcal{D}$ consists of two types of training samples $\mathcal{D}_{\text{correct}} = \{(\boldsymbol{x}, \tilde{y}) \in \mathcal{D} \mid \tilde{y} = f^\star(\boldsymbol{x})\}$ and $\mathcal{D}_{\text{wrong}} = \{(\boldsymbol{x}, \tilde{y}) \in \mathcal{D} \mid \tilde{y} = \epsilon(\boldsymbol{a})\}$. In this regard, when training a network to map $\boldsymbol{x}$ to $\tilde{y}$ on $\mathcal{D} = \mathcal{D}_{\text{correct}} \cup \mathcal{D}_{\text{wrong}}$, we are effectively asking it to learn two different target functions, where only one of them depends on $\boldsymbol{x}$, which forces the network to memorize part of the training data and hurts its generalization (Zhang et al., 2017).

In practice, however, we will not have access to the exact side information, and we will be able to rely at most on meta-data, and PI, about the annotation process. That is, we consider a learning problem in which our training data consists of triplets $(\boldsymbol{x}, \tilde{y}; \boldsymbol{a})$ where $\boldsymbol{a} \in \mathbb{R}^p$ is a vector of PI features such as high latency features related to the annotation process, e.g., annotator experience, or even a randomly assigned unique vector introduced to model unobserved features (Ortiz-Jimenez et al., 2023). We present here our method that uses this setting to explicitly model h and learn effectively in the presence of large amounts of label noise in the training set.

3.2. Method Description

Based on the noise model, we propose Pi-DUAL, a novel PI-based architecture designed to mimic the generative noise model proposed in Eq. (1). Specifically, during training, Pi-DUAL factorizes its output logits into two terms, i.e.,

$$h_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{a}) = [1 - \gamma_{\boldsymbol{\psi}}(\boldsymbol{a})] f_{\boldsymbol{\theta}}(\boldsymbol{x}) + \gamma_{\boldsymbol{\psi}}(\boldsymbol{a}) \epsilon_{\boldsymbol{\phi}}(\boldsymbol{a}), \quad (2)$$

where $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^c$ represents a *prediction network* tasked with approximating the ground truth labelling function $f^\star$ and $\epsilon_{\boldsymbol{\phi}} : \mathcal{A} \to \mathbb{R}^c$ a *noise network*, modeling the noise signal ϵ. Here, $\gamma_{\boldsymbol{\phi}}$ denotes a *gating network* tasked with learning the switching mechanism γ, where we apply a sigmoid activation function to the output to restrict $\gamma_{\boldsymbol{\psi}}(\boldsymbol{a})$ to be within in $[0, 1]$. Moreover, following the recommendations of Ortiz-Jimenez et al. (2023), 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. The dimension of this vector, known as random PI length, is the only additional hyperparameter we tune for Pi-DUAL. During inference, when PI is not available, Pi-DUAL relies solely on $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ to predict the clean label y (see Fig. 1).

The dual gated logit structure of Pi-DUAL is reminiscent of sparsely-gated mixture of experts which also factorize its predictions at the logit level, albeit providing the same input $\boldsymbol{x}$ to each expert (Shazeer et al., 2017). Pi-DUAL instead provides $\boldsymbol{x}$ and $\boldsymbol{a}$ to different networks, which effectively decouples learning the task-specific samples and the noise-specific samples with different features. Indeed, assuming that the incorrect labels are independent of $\boldsymbol{x}$ and that the noise is only a function of the PI $\boldsymbol{a}$, there will always be a natural tendency by the network to use $\epsilon_{\boldsymbol{\phi}}(\boldsymbol{a})$ to explain away those labels that it cannot easily learn with $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. The gating network $\gamma_{\boldsymbol{\phi}}$ facilitates this separation by utilizing the discriminative power of the PI to guide this process. In Sec. 5.3, we ablate all these elements of the architecture to show that they all contribute to learning the clean labels.

Pi-DUAL has multiple advantages over previous PI methods like TRAM or AFM (Collier et al., 2022). Indeed, previous methods tend to directly expose the no-PI term $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ to the noisy labels, e.g., through $\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \tilde{y})$, which can thus lead to an overfitting to the noisy labels based on $\boldsymbol{x}$. In contrast, Pi-DUAL instead solves

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}} \sum_{(\boldsymbol{x},\tilde{y};\boldsymbol{a}) \in \mathcal{D}} \mathcal{L}\left(\text{softmax}\left(h_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{a})\right), \tilde{y}\right), \quad (3)$$

and never explicitly forces $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ to fit all $\tilde{y}$'s (we validate the loss design in Appendix C.7). 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 $\epsilon_{\boldsymbol{\phi}}(\boldsymbol{a})$. In Sec. 5.1 we analyze in detail these dynamics.

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.

3.3. Noise Detection

Finally, we provide a simple noise detection method based on Pi-DUAL: After training, we collect confidence estimates of the prediction network on the observed noisy labels $\tilde{y}$, i.e., with $\text{softmax}(f_\theta(\boldsymbol{x}))[\tilde{y}]$, for the training samples and threshold the confidences to distinguish the correctly and incorrectly labeled examples. Indeed, we show that because the prediction network $f_{\boldsymbol{\theta}}$ of Pi-DUAL only learns to confidently predict clean labels y during training without having to memorize the wrong labels, its confidence on the noisy labels is a very good proxy for a noise indicator: 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. In Sec. 4.3, we compare Pi-DUAL to other state-of-the-art methods to detect wrong labels.

3.4. Theoretical Insights

To further support the design of Pi-DUAL described in Eq. (2), we study the theoretical behavior of the predictor $h_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{a})$ 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 $\mathcal{N}(\boldsymbol{x}^\top \boldsymbol{w}^\star, \sigma^2)$ and $\mathcal{N}(\boldsymbol{a}^\top \boldsymbol{v}^\star, \sigma^2)$, for two weight vectors $(\boldsymbol{w}^\star, \boldsymbol{v}^\star)$ parameterizing linearly their means. In this tractable setting, we 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 compare two estimators, Pi-DUAL and an ordinary least squares estimator (OLS) that ignores the side information $\boldsymbol{a}$. We summarize below the main insights of our analysis, and provide the details in Appendix A.

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

We show with this theorem that in terms of their abilities to generalize on *clean targets*—as measured by their risks (Bach, 2021)—Pi-DUAL exhibits a more robust behavior. In particular, while the risk of OLS tends to be proportional to the number of wrong labels $|\mathcal{D}_{\text{wrong}}|$, Pi-DUAL has a risk that more gracefully scales with respect to the number of examples that the gates $\gamma_{\boldsymbol{\psi}}$ fail to identify. Our experiments in Sec. 5.2 show that, in practice, the gates learned by Pi-DUAL typically manage to identify the clean and wrong labels.

4. Experimental Results

We now validate the effectiveness of Pi-DUAL on several public noisy label benchmarks with PI and compare it extensively to other algorithms. We show that Pi-DUAL achieves (a) state-of-the-art results on clean test accuracy and noise detection tasks (especially when there is good PI available) and (b) scales up to datasets with millions of examples.

4.1. Experimental Settings

Our experimental settings follow the benchmarking practices laid out by Ortiz-Jimenez et al. (2023). In particular, we use the same architectures, training schedules and public codebase (Nado et al., 2021) to perform all our experiments. In terms of baseline choices, in order to achieve a fair comparison, we compare Pi-DUAL to our own implementations of the methods in Tab. 1 that use only one model and one stage of training[1]. Moreover, to further ensure fairness, we use on each dataset the same architecture and the same training strategy across all compared methods. For each result, we perform a grid search over hyperparameters. Notably, while other methods require tuning at least two additional hyperparameters on top of the cross-entropy baseline; Pi-DUAL only requires tuning the random PI length, making its tuning budget much smaller. We use a *noisy validation set*, held-out from the training set, to select the best hyperparameters and report results over the clean test set. We provide more details on our hyper-parameter tuning strategy and other experimental settings in Appendix B, and a computational cost analysis in Appendix D.

Pi-DUAL does not require to use early stopping to achieve strong results as it does not suffer from overfitting issues (see Fig. 3). However, early stopping is essential to achieve good performance for the other methods. Hence, we always report results at the epoch with the best accuracy on the noisy validation set. In Appendix C.4, we provide results for all methods without early stopping.

Our experiments are conducted on five noisy datasets with realistic label noise, either derived from a noisy human annotation process or produced by imperfect model predictions. A summary of the main features of each datasets is shown in Appendix B.1. Importantly, we note that CIFAR-10H (Peterson et al., 2019), ImageNet-PI (low-noise) and ImageNet-PI (high-noise) all have excellent-quality PI features at the sample level that seem to capture important information of the annotation process. On the other hand, CIFAR-10N and CIFAR-100N (Wei et al., 2022) provide aggregated PI, in the form of averages over batches of samples, which may not have enough resolution to distinguish clean and wrong labels at the sample level (Ortiz-Jimenez et al., 2023). Despite this, Pi-DUAL still performs comparatively to no-PI methods on those datasets.

We further provide in Appendix C.5 the results for Pi-DUAL+, a stronger version of Pi-DUAL boosted with advanced regularization techniques. It is competitive against state-of-art semi-supervised learning based methods, such as Divide-Mix (Li et al., 2020a) and SOP+ (Liu et al., 2022).

4.2. Predicting Clean Labels

Tab. 2 reports the test accuracy of Pi-DUAL compared to previous noisy label methods, averaged over 5 and 3 random seeds for CIFAR and ImageNet-PI, respectively. As

[1] We do not run ELR and SOP on ImageNet-PI as they require 1 billion extra parameters (see Appendix B.2).

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	CIFAR-10N	CIFAR-100N	ImageNet-PI	ImageNet-PI
		(worst)	(worst)	(fine)	(low-noise)	(high-noise)
No-PI	Cross-entropy	$51.1_{\pm2.2}$	$80.6_{\pm0.2}$	$60.4_{\pm0.5}$	$68.2_{\pm0.2}$	$47.2_{\pm0.2}$
	ELR	$48.5_{\pm1.4}$	**86.6**$_{\pm0.7}$	**64.0**$_{\pm0.3}$	-	-
	HET	$50.8_{\pm1.4}$	$81.9_{\pm0.4}$	$60.8_{\pm0.4}$	$69.4_{\pm0.1}$	$51.9_{\pm0.0}$
	SOP	$51.3_{\pm1.9}$	$85.0_{\pm0.8}$	$61.9_{\pm0.6}$	-	-
PI	TRAM	$64.9_{\pm0.8}$	$80.5_{\pm0.5}$	$59.7_{\pm0.3}$	$69.4_{\pm0.2}$	$54.0_{\pm0.1}$
	TRAM++	$66.8_{\pm0.3}$	$83.9_{\pm0.2}$	$61.1_{\pm0.2}$	$69.5_{\pm0.0}$	$53.8_{\pm0.3}$
	AFM	$64.0_{\pm0.6}$	$82.0_{\pm0.3}$	$60.0_{\pm0.2}$	$70.3_{\pm0.0}$	$55.3_{\pm0.2}$
	Pi-DUAL (Ours)	**71.3**$_{\pm3.3}$	$84.9_{\pm0.4}$	**64.2**$_{\pm0.3}$	**71.6**$_{\pm0.1}$	**62.1**$_{\pm0.1}$

we can see, Pi-DUAL achieves state-of-the-art performance on the three datasets with high quality PI. It improves by 4.5% over the most competitive PI baseline on CIFAR-10H and by 20 points over the best performing no-PI methods. It also achieves a 1.3 point and 6.8 point lead on ImageNet-PI low-noise and high-noise, respectively. These are remarkable results given the 1000 classes in ImageNet-PI and the scale of these datasets. Indeed, they show that Pi-DUAL can effectively leverage PI in these settings to distinguish between correct and wrong labels during training, while learning the clean labels with the prediction network.

On the other hand, on the two datasets with low quality PI, we observe that Pi-DUAL achieves better results than previous PI methods by more than 3 points on CIFAR-100N. It also performs comparatively with no-PI methods, even though the quality of the PI does not allow to properly distinguish between clean and wrong labels (see Sec. 5.2).

4.3. Detection of Wrong Labels

We validate the ability of Pi-DUAL to detect the wrong labels in the training set, allowing practitioners to relabel those instances, or filter them out in future runs.

Tab. 3 shows the area under the receiver operating characteristic curve (AUC) obtained by applying our confidence-based noise detection method described in Section 3.3, compared with different methods on all PI benchmarks. As we can see, Pi-DUAL achieves the best results by a large margin in all datasets except CIFAR-10N (where it performs comparatively to the best method). These performance gains are a clear sign that Pi-DUAL can effectively minimize the amount of overfitting of the prediction network to the noisy labels. In most cases, the prediction network has a very low confidence (near 0%) on the wrong labels, while having a very high confidence (near 100%) on the correct labels. We show the distribution of the prediction confidences in Fig. 2 for CIFAR-10H, CIFAR-100N, and ImageNet-PI (high noise) and two other datasets in Appendix C.3, where we observe that the prediction confidence is clearly separated over samples with correct and wrong labels.

In our experiments, we observe that the simple confidence thresholding is a strong detection method across all datasets. Meanwhile, we also evaluated the ability of the gating network γ_ψ in detecting the wrong labels. As shown in Tab. 3, thresholding the gate's outputs is also an effective method for noise detection, which can even outperform confidence thresholding on certain datasets, i.e., CIFAR-10H. However, we observe that the performance of gate thresholding suffers more than confidence thresholding on datasets with low-resolution PI. As we will see in Sec. 5.2, this is due to the fact that, in those datasets, the gating network cannot exploit the PI to discriminate easily between correct and wrong labels. Still, this does not prevent the prediction network from learning the clean distribution, and thus its detection ability does not suffer as much. Choosing which of the two methods to use is, in general, a dataset-dependent decision: If there is good PI available gate thresholding achieves the best results, but confidence thresholding performs well overall, so we recommend it as a default choice.

5. Further Analysis

In this section, we provide further analysis on the training dynamics of Pi-DUAL, the distribution of the learned gates and several ablations on our method. Overall, we show that Pi-DUAL behaves as expected from its design, and that all pieces of its architecture contribute to its good performance.

5.1. Training Dynamics

To verify that Pi-DUAL effectively decouples the learning paths of samples with correct and wrong labels, we study the

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**

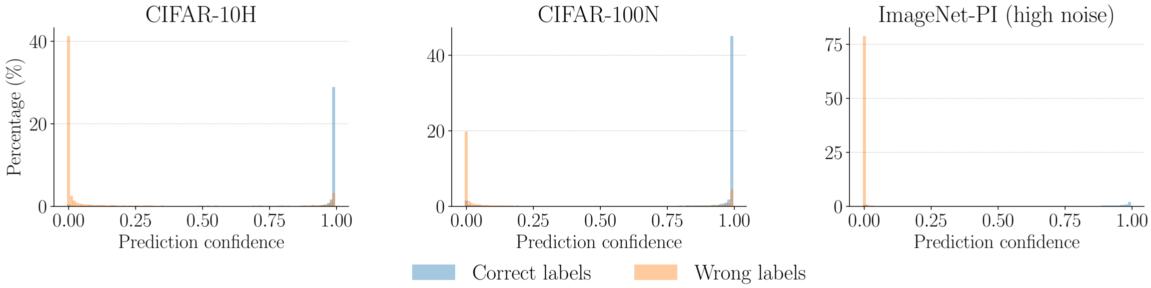

Figure 2. Distribution for the prediction network's confidence on the observed noisy labels for several datasets, separated by correctly and wrongly labeled samples.

training dynamics of the prediction and noise networks on each of these sets of samples, in comparison to the training dynamics of cross-entropy baseline. We observe in Fig. 3 that the prediction network of Pi-DUAL mostly fits the correct labels, as its training accuracy on samples with wrong labels is always very low on all datasets[4]. Meanwhile, the noise network shows the opposite behavior and mostly fits the wrong labels on CIFAR-10H and CIFAR-100N. Interestingly, we observe that the noise network does not fit any samples on ImageNet-PI. We attribute this behavior to the fact that ImageNet has more than a million samples and 1000 classes, so fitting the noise is very hard. Indeed, as shown on the bottom row of Fig. 3, the cross-entropy baseline also ignores the samples with wrong labels. However, the cross-entropy baseline has lower training accuracy on the correct labels than Pi-DUAL as it cannot effectively separate the two distributions, and therefore achieves worse test accuracy.

In all datasets, we see that the test accuracy of Pi-DUAL grows gradually and steadily with training and that overfitting to the wrong labels does not hurt its performance as these are mostly fit by the noise network $\epsilon_{\boldsymbol{\phi}}$. Meanwhile, we observe that on CIFAR-10H and CIFAR-100N, the test accuracy of the cross-entropy baseline starts degrading as the accuracy on samples with wrong labels starts to grow. This is a clear sign that Pi-DUAL effectively leverages the PI to learn shortcuts that protect the feature extraction of $f_{\boldsymbol{\theta}}$ and therefore does not require to use early-stopping to achieve its best results.

5.2. Analysis of the Gating Network Predictions

In our model, the gating network γ_ψ is tasked with learning the binary indicator signal γ, which tells whether a sample belongs to $\mathcal{D}_{\text{correct}}$ or $\mathcal{D}_{\text{wrong}}$. To show that the model works as intended, we plot in Fig. 4 the distribution of $\gamma_\psi(\boldsymbol{a})$ separately for samples with correct and wrong labels after training on different datasets[5]. As expected, in the two datasets with high-quality PI – CIFAR-10H and ImageNet-PI – the gate distribution achieves a separation between the two distributions (cf. Tab. 3). And even in the case of CIFAR-100N, where the PI is not very informative, the gate output still separates a big portion of the wrong labels.

To give a better intuition of what Pi-DUAL learns, we provide some visual examples of both success and failures cases of the gating network when training on ImageNet-PI (high-noise). As shown in Fig. 5, the gating network can often detect blatantly wrong annotations which are further corrected by the prediction network. Interestingly, we observe

[4] Results for other datasets are shown in Appendix C.1.

[5] Results for other datasets are shown in Appendix C.2.

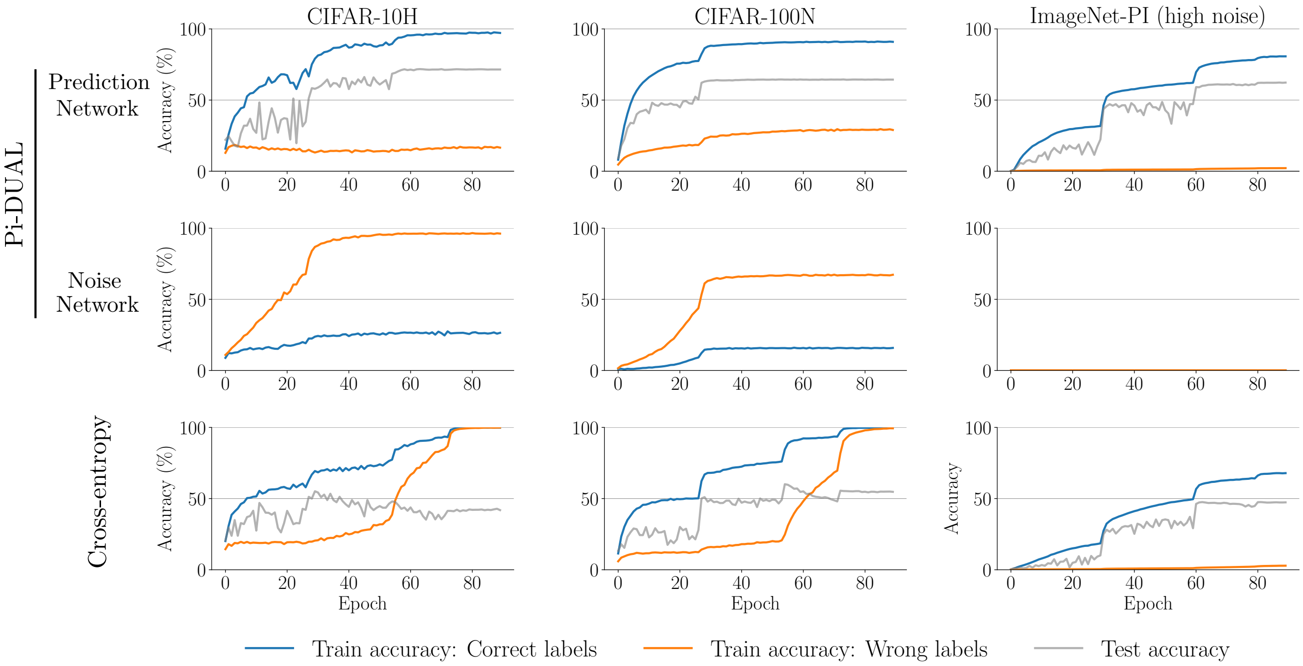

Figure 3. Training curves of Pi-DUAL and cross-entropy baseline on different datasets. The first two rows show the training dynamics of prediction network and noise network respectively.We plot separately the training accuracy on clean and wrong labels and test accuracy[3].

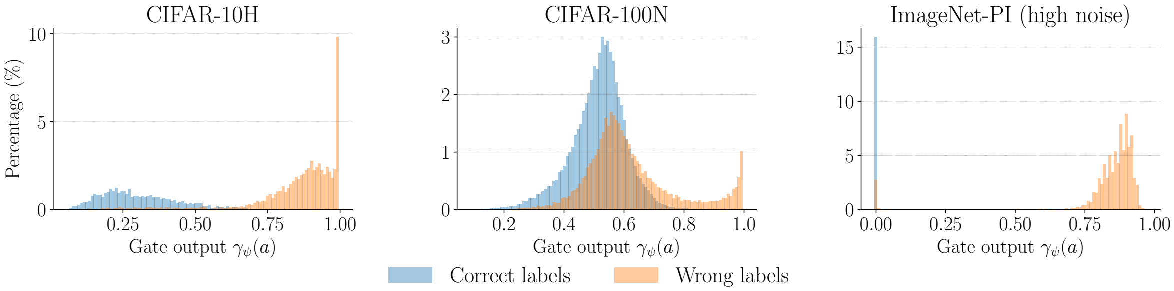

Figure 4. Distributions of $\gamma_{\boldsymbol{\psi}}(\boldsymbol{a})$ over training samples with correct and wrong labels on several datasets.

that in the few cases where the gating network makes a mistake, the predicted clean label is not so far from what many humans would suggest – like in the crane picture on the bottom right which is recognized by Pi-DUAL as a fire truck.

5.3. Ablation Studies

We finally present various ablation studies analysing the contributions of different components of Pi-DUAL in Tab. 4.

Architecture ablation. Pi-DUAL gives $\boldsymbol{a}$ as input to both the noise network $\epsilon_{\boldsymbol{\phi}}$ and the gating network $\gamma_{\boldsymbol{\psi}}$. As shown in Tab. 4, removing either of the two elements from the architecture generally results in lower performance gains than with the full architecture. Interestingly, on ImageNet-PI, the noise network does not seem to be critical. We attribute this behavior to the fact that, on these datasets, Pi-DUAL does not need to overfit to the noisy labels to achieve good performance (cf. Fig. 3). Indeed, just using the gating mechanism to toggle on-off the fitting of the noisy labels seems sufficient to achieve good performance in a dataset with so many classes. We provide more ablation studies on the model architecture in Appendix C.6, including the model backbone for the prediction network and the network structure for the noise and gating network. Pi-DUAL delivers consistent, high performance with different architectures.

Gating in probability space. In Sec. 3.2 we chose to parameterize Pi-DUAL in the logit space. An alternative is to parameterize the gating mechanism in the probability space.

[3] The training accuracy for the noise network on ImageNet-PI (high noise) is 0.07% and 0.1% respectively for correct and wrong labels.

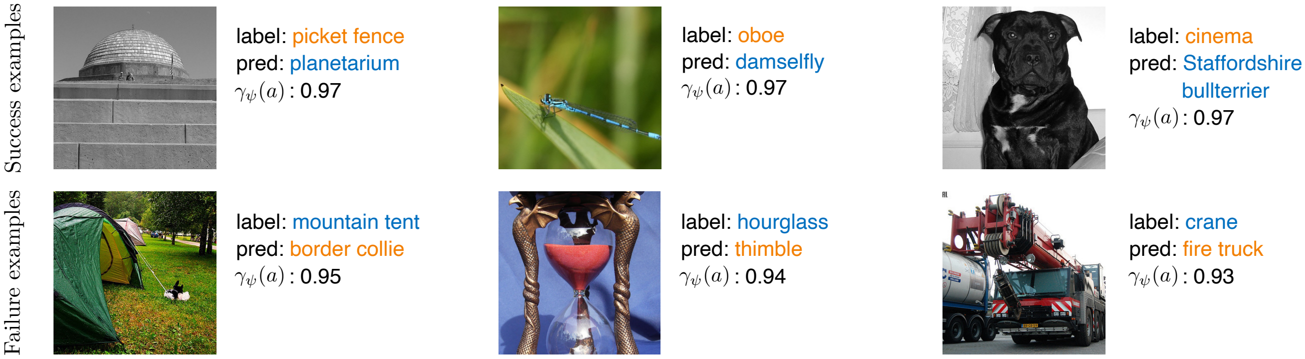

Figure 5. Examples of ImageNet-PI images that the gating network suggests are mislabeled. The first row shows samples with actually wrongly annotated labels, and the second row shows examples with correct labels but assumed to be wrong by the gating network. Here, “label" denotes the annotation label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

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)	$53.5_{\pm 2.2}$	$83.7_{\pm 1.3}$	$61.8_{\pm 0.3}$	$68.4_{\pm 0.1}$	$47.0_{\pm 0.4}$

However, although the probabilistic version of Pi-DUAL performs better than the cross-entopy baseline in most cases, it underperforms compared to the logit space version.

Performance without PI. We argued before that Pi-DUAL performs better on datasets with high-quality PI as this permits to wield the most power from its structure. For completeness, we now test the performance of Pi-DUAL without access to dataset-specific PI features. That is, having only access to the random PI sample-identifier proposed by Ortiz-Jimenez et al. (2023). We see that without access to PI features, Pi-DUAL still can perform better than the cross-entropy baseline, but its performance deteriorates significantly, i.e., having access to good PI is fundamental for Pi-DUAL's success.

6. Conclusion

In this paper, we have presented Pi-DUAL, a new method that utilizes PI to combat label noise by introducing a dual network structure designed to model the generative process of the noisy annotations. Experimental results have demonstrated the effectiveness of Pi-DUAL in learning to both fit the clean label distribution and detect noisy samples. Pi-DUAL sets a new state-of-the-art accuracy in datasets with high-quality PI features. We have performed extensive ablation studies and thorough analysis, both empirical and theoretical, to provide insights into how Pi-DUAL works. Importantly, Pi-DUAL is very easy to implement and can be plugged into any training pipeline. Unlike competing approaches, it gracefully scales up to datasets with millions of examples and thousands of classes. Moving forward, it will be interesting to study extensions of Pi-DUAL that can also tackle other problems with PI beyond supervised classification.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning.

Overall, we do not see any special ethical concerns stemming directly from our work. In particular we note that although annotator IDs are part of the PI used in our experiments, none of our results require the use of personally identifiable annotator IDs. In fact, cryptographically safe IDs in the form of hashes work equally well as PI. In this regard, we do not think that there are serious concerns about possible identity leakages stemming from the proposed framework if the proper anonymization protocols are followed.

Acknowledgments

We thank the anonymous reviewers for helpful feedback and comments.

References

Albert, P., Arazo, E., Krishna, T., O'Connor, N. E., and McGuinness, K. Is your noise correction noisy? pls: Robustness to label noise with two stage detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2023.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

Bach, F. *Learning Theory from First Principles*. (draft), 2021.

Bai, Y., Yang, E., Han, B., Yang, Y., Li, J., Mao, Y., Niu, G., and Liu, T. Understanding and improving early stopping for learning with noisy labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Baldock, R. J. N., Maennel, H., and Neyshabur, B. Deep learning through the lens of example difficulty. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. 2019.

Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., and Liu, Y. Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*, 2020.

Collier, M., Mustafa, B., Kokiopoulou, E., Jenatton, R., and Berent, J. Correlated input-dependent label noise in large-scale image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Collier, M., Jenatton, R., Kokiopoulou, E., and Berent, J. Transfer and marginalize: Explaining away label noise with privileged information. In *International Conference on Machine Learning (ICML)*, 2022.

Collier, M., Jenatton, R., Mustafa, B., Houlsby, N., Berent, J., and Kokiopoulou, E. Massively scaling heteroscedastic classifiers. In *International Conference on Learning Representations (ICLR)*, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. 2018.

Hernández-Lobato, D., Sharmanska, V., Kersting, K., Lampert, C. H., and Quadrianto, N. Mind the nuisance: Gaussian process classification using privileged noise. In *Advances in Neural Information Processing Systems (NeurIPS*, 2014.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning (ICML)*, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Li, J., Socher, R., and Hoi, S. C. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020a.

Li, J., Socher, R., and Hoi, S. C. H. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2020b.

Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Liu, S., Zhu, Z., Qu, Q., and You, C. Robust training under label noise by over-parameterization. In *International Conference on Machine Learning (ICML)*, 2022.

Lopez-Paz, D., Bottou, L., Schölkopf, B., and Vapnik, V. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Maennel, H., Alabdulmohsin, I. M., Tolstikhin, I. O., Baldock, R. J. N., Bousquet, O., Gelly, S., and Keysers, D. What do neural networks learn when trained with random labels? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Menon, A. K., Rawat, A. S., Reddi, S. J., and Kumar, S. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations (ICLR)*, 2020.

Nado, Z., Band, N., Collier, M., Djolonga, J., Dusenberry, M. W., Farquhar, S., Feng, Q., Filos, A., Havasi, M., Jenatton, R., et al. Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

Ortiz-Jimenez, G., Collier, M., Nawalgaria, A., D'Amour, A., Berent, J., Jenatton, R., and Kokiopoulou, E. When does privileged information explain away label noise? In *International Conference on Machine Learning (ICML)*, 2023.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Peterson, J. C., Battleday, R. M., Griffiths, T. L., and Russakovsky, O. Human uncertainty makes classification more robust. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34: 8583–8595, 2021.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Sheng, V. S., Provost, F., and Ipeirotis, P. G. Get another label? improving data quality and data mining using multiple, noisy labelers. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence, 2020.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Tanaka, D., Ikami, D., Yamasaki, T., and Aizawa, K. Joint optimization framework for learning with noisy labels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Vapnik, V. and Vashist, A. A new learning paradigm: Learning using privileged information. *Neural Networks*, 2009.

Wang, H., Xiao, R., Dong, Y., Feng, L., and Zhao, J. ProMix: combating label noise via maximizing clean sample utility. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2023.

Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., and Liu, Y. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021.

Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., and Liu, Y. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations (ICLR)*, 2022.

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yi, K. and Wu, J. Probabilistic end-to-end noise correction for learning with noisy labels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., and Sugiyama, M. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning (ICML)*, 2019.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

A. Theoretical Insights: Risk Analysis

Model and notations. We assume the following regression setting

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_1 \boldsymbol{w}^\star \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{A}_2 \boldsymbol{v}^\star \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \in \mathbb{R}^n$$

where we have $n = n_1 + n_2$ observations such that

- $\boldsymbol{y}_1 = \boldsymbol{X}_1 \boldsymbol{w}^\star + \varepsilon_1 \in \mathbb{R}^{n_1}$ with $\boldsymbol{X}_1 \in \mathbb{R}^{n_1 \times d}$ and $\varepsilon_1 \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$,
- $\boldsymbol{y}_2 = \boldsymbol{A}_2 \boldsymbol{v}^\star + \varepsilon_2 \in \mathbb{R}^{n_2}$ with $\boldsymbol{A}_2 \in \mathbb{R}^{n_2 \times m}$ and $\varepsilon_2 \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$.

The vector $\boldsymbol{y}_1$ corresponds to the clean targets that depend on the features $\boldsymbol{X}_1$ while $\boldsymbol{y}_2$ corresponds to the noisy targets that are explained by the privileged information (PI) represented by $\boldsymbol{A}_2$.

We use the matrix forms $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2] \in \mathbb{R}^{n \times d}$, $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2] \in \mathbb{R}^{n \times m}$ and $\varepsilon = [\varepsilon_1, \varepsilon_2] \in \mathbb{R}^n$. Moreover, we consider the diagonal mask matrix $\boldsymbol{\gamma}^\star \in \{0, 1\}^{n \times n}$ such that

$$\boldsymbol{\gamma}^* \boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{0} \end{bmatrix} \text{ and } (\boldsymbol{I} - \boldsymbol{\gamma}^*)\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{A}_2 \end{bmatrix}.$$

We list below some notation that we will repeatedly use

- The covariance matrices $\boldsymbol{Q} = \boldsymbol{X}^\top \boldsymbol{X}$ and $\boldsymbol{Q}_1 = \boldsymbol{X}_1^\top \boldsymbol{X}_1$
- The difference between the contributions of the standard features and the PI features
$$\boldsymbol{\delta}^\star = \boldsymbol{X}\boldsymbol{w}^\star - \boldsymbol{A}\boldsymbol{v}^\star \in \mathbb{R}^n$$
- The orthogonal projector onto the span of the columns of $\boldsymbol{X}$:
$$\boldsymbol{\Pi}_x = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \in \mathbb{R}^{n \times n}.$$
- For any diagonal mask matrix $\boldsymbol{\gamma} \in \{0, 1\}^{n \times n}$, we define the diagonal matrix that records the differences with respect to the reference $\boldsymbol{\gamma}^\star$
$$\boldsymbol{\Delta}_\gamma = \boldsymbol{\gamma}^\star - \boldsymbol{\gamma} \in \{-1, 0, 1\}^{n \times n}.$$

The rest of our exposition follows the structure of Collier et al. (2022).

A.1. Definition of the Risk

To compare different predictors, we will consider their *risks*, that is, their ability to generalize. We focus on the *fixed* design analysis (Bach, 2021), i.e., we study the errors only due to resampling the additive noise ε. In our context, we are more specifically interested in *the performance of the predictors on the clean targets* (with predictors having been trained on both clean and noisy targets).

Formally, given a predictor $\boldsymbol{\theta}$ based on the training quantities $(\boldsymbol{X}, \boldsymbol{A}, \varepsilon)$, we consider

$$\boldsymbol{y}_1' = \boldsymbol{X}_1 \boldsymbol{w}^\star + \varepsilon_1'$$

where the prime $'$ is to show the difference with the training quantities without prime, and we define the risk of $\boldsymbol{\theta}$ as

$$\mathcal{R}(\boldsymbol{\theta}) = \mathbb{E}_{\varepsilon_1' \sim p(\varepsilon_1')} \left\{ \frac{1}{n_1} \|\boldsymbol{y}_1' - \boldsymbol{X}_1 \boldsymbol{\theta}\|^2 \right\}. \tag{4}$$

A simple expansion of the square with $\mathbb{E}_{\varepsilon'}[\|\varepsilon_1'\|^2] = n_1 \sigma^2$ leads to the standard expression

$$\mathcal{R}(\boldsymbol{\theta}) = \frac{1}{n_1} \|\boldsymbol{X}_1(\boldsymbol{\theta} - \boldsymbol{w}^\star)\|^2 + \sigma^2 = \frac{1}{n_1} \|\boldsymbol{\gamma}^\star \boldsymbol{X}(\boldsymbol{\theta} - \boldsymbol{w}^\star)\|^2 + \sigma^2. \tag{5}$$

To obtain the final expression of the risk, we eventually take a second expectation $\mathbb{E}_{\varepsilon \sim p(\varepsilon)}[\mathcal{R}(\boldsymbol{\theta})]$ with respect to the training quantity ε (Bach, 2021).

A.2. Main Result

We state below our main result and discuss its implications.

Proposition 2. *Consider some diagonal mask matrix $\boldsymbol{\gamma} \in \{0,1\}^{n\times n}$ and the masked versions of $\boldsymbol{X}$ and $\boldsymbol{A}$ which we refer to as $\bar{\boldsymbol{X}} = \boldsymbol{\gamma}\boldsymbol{X}$ and $\bar{\boldsymbol{A}} = (\boldsymbol{I} - \boldsymbol{\gamma})\boldsymbol{A}$.*

Let us assume that $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$, $\bar{\boldsymbol{X}}^\top \bar{\boldsymbol{X}} \in \mathbb{R}^{d\times d}$, $\bar{\boldsymbol{A}}^\top \bar{\boldsymbol{A}} \in \mathbb{R}^{m\times m}$ and

$$\begin{bmatrix} \bar{\boldsymbol{X}}^\top \bar{\boldsymbol{X}} & \bar{\boldsymbol{X}}^\top \bar{\boldsymbol{A}} \\ \bar{\boldsymbol{A}}^\top \bar{\boldsymbol{X}} & \bar{\boldsymbol{A}}^\top \bar{\boldsymbol{A}} \end{bmatrix} \in \mathbb{R}^{(d+m)\times(d+m)} \tag{6}$$

are all invertible. Let us define by $\boldsymbol{w}_0$ the ordinary-least-squares predictor (see Eq. (8)). Similarly, let us define by $\boldsymbol{w}_1$ the Pi-DUAL predictor, using $\boldsymbol{\gamma}$ as (pre-defined) gates (see Eq. (10)).

It holds that the risk $\mathbb{E}[\mathcal{R}(\boldsymbol{w}_0)]$ of $\boldsymbol{w}_0$ is larger than the risk $\mathbb{E}[\mathcal{R}(\boldsymbol{w}_1)]$ of $\boldsymbol{w}_1$ if and only if

$$\|\gamma^\star \boldsymbol{\Pi}_x(\boldsymbol{I} - \gamma^*)\boldsymbol{\delta}^\star\|^2 + \sigma^2 \mathrm{tr}(\boldsymbol{Q}^{-1}\boldsymbol{Q}_1) > \|\gamma^\star \boldsymbol{X}\bar{\boldsymbol{H}}\boldsymbol{\Delta}_\gamma \boldsymbol{\delta}^\star\|^2 + \sigma^2 \mathrm{tr}(\bar{\boldsymbol{Q}}_a^{-1}\boldsymbol{Q}_1) \tag{7}$$

where the matrices $\bar{\boldsymbol{H}}$ and $\bar{\boldsymbol{Q}}_a$ are defined in Section A.4.

The proofs of the risk expressions can be found in Sections A.3 and A.4.

A.2.1. Discussion

The condition in Eq. (7) brings into play the bias terms and the variance terms of the risks of $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$.

As intuitively expected, the variance term corresponding to $\boldsymbol{w}_0$ is smaller than that of $\boldsymbol{w}_1$. Indeed, Pi-DUAL requires to learn more parameters (both $\boldsymbol{w}$ and $\boldsymbol{v}$) than in the case of the standard ordinary least squares. More precisely, if the spans of the columns $\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{X}}$ are close to be orthogonal to each other (as suggested by the invertibility condition for Eq. (6)), we approximately have

$$\mathrm{tr}(\boldsymbol{Q}^{-1}\boldsymbol{Q}_1) \approx d\frac{n_1}{n} < \mathrm{tr}(\bar{\boldsymbol{Q}}_a^{-1}\boldsymbol{Q}_1) \approx d\frac{n_1}{\bar{n}_1}$$

where $\bar{n}_1 = \boldsymbol{1}^\top \boldsymbol{\gamma}\boldsymbol{1}$ stands for the number of examples selected by the gate $\boldsymbol{\gamma}$ (with $\bar{n}_1 < n$).

When looking at the bias terms, we see how Pi-DUAL can compensate for a larger variance term to achieve a lower risk overall. We first recall the definition of $\boldsymbol{\delta}^\star = \boldsymbol{X}\boldsymbol{w}^\star - \boldsymbol{A}\boldsymbol{v}^\star$ that computes the difference between the contributions of the standard features $\boldsymbol{X}$ and the PI features $\boldsymbol{A}$. If the level of noise explained by $\boldsymbol{A}_2$ has a large contribution compared with the signal from $\boldsymbol{X}_2$, the second part $\boldsymbol{\delta}_2^\star$ of $\boldsymbol{\delta}^\star$ can contain large entries. While $\boldsymbol{w}_0$ has a bias term scaling with $\mathcal{O}((\boldsymbol{I} - \boldsymbol{\gamma}^*)\boldsymbol{\delta}^\star)$—that is, proportional to the number n_2 of noisy examples captured by $\boldsymbol{\delta}_2^\star$—we can observe that $\boldsymbol{w}_1$ has a more robust scaling. Indeed, it depends on $\mathcal{O}(\boldsymbol{\Delta}_{\boldsymbol{\gamma}}\boldsymbol{\delta}^\star)$ that only scales with the number of disagreements between the reference gate $\boldsymbol{\gamma}^\star$ and that used for training $\boldsymbol{\gamma}$.

A.3. Proof: Risk of Ordinary Least Squares

We assume that $\boldsymbol{Q}$ is invertible. We focus on the solution of

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 \tag{8}$$

that is given by

$$\begin{aligned}
\boldsymbol{w}_0 &= \boldsymbol{Q}^{-1}\boldsymbol{X}^\top \boldsymbol{y} \\
&= \boldsymbol{Q}^{-1}\boldsymbol{X}^\top(\gamma^*\boldsymbol{X}\boldsymbol{w}^\star + (\boldsymbol{I} - \gamma^*)\boldsymbol{A}\boldsymbol{v}^\star + \varepsilon) \\
&= \boldsymbol{Q}^{-1}\boldsymbol{X}^\top(-(\boldsymbol{I} - \gamma^*)\boldsymbol{\delta}^\star + \boldsymbol{X}\boldsymbol{w}^\star + \varepsilon) \\
&= -\boldsymbol{Q}^{-1}\boldsymbol{X}^\top(\boldsymbol{I} - \gamma^*)\boldsymbol{\delta}^\star + \boldsymbol{w}^\star + \boldsymbol{Q}^{-1}\boldsymbol{X}^\top \varepsilon.
\end{aligned}$$

Plugging into Eq. (5), we obtain

$$\mathcal{R}(\boldsymbol{w}_0) = \frac{1}{n_1}\|\boldsymbol{\gamma}^\star \boldsymbol{\Pi}_x(\boldsymbol{I} - \boldsymbol{\gamma}^*)\boldsymbol{\delta}^\star - \boldsymbol{\gamma}^\star \boldsymbol{\Pi}_x \varepsilon\|^2 + \sigma^2.$$

Expanding the square and using that $\mathrm{tr}(\boldsymbol{\gamma}^*\boldsymbol{\Pi}_x(\boldsymbol{\gamma}^*\boldsymbol{\Pi}_x)^\top) = \mathrm{tr}(\boldsymbol{\gamma}^*\boldsymbol{\Pi}_x) = \mathrm{tr}(\boldsymbol{Q}^{-1}\boldsymbol{Q}_1)$, the final risk expression is

$$\begin{aligned}\mathbb{E}[\mathcal{R}(\boldsymbol{w}_0)] &= \frac{1}{n_1}\|\boldsymbol{\gamma}^\star\boldsymbol{\Pi}_x(\boldsymbol{I}-\boldsymbol{\gamma}^*)\boldsymbol{\delta}^\star\|^2 + \frac{1}{n_1}\mathbb{E}[\|\boldsymbol{\gamma}^\star\boldsymbol{\Pi}_x\varepsilon\|^2] + \sigma^2 \\ &= \frac{1}{n_1}\|\boldsymbol{\gamma}^\star\boldsymbol{\Pi}_x(\boldsymbol{I}-\boldsymbol{\gamma}^*)\boldsymbol{\delta}^\star\|^2 + \frac{\sigma^2}{n_1}\mathrm{tr}(\boldsymbol{Q}^{-1}\boldsymbol{Q}_1) + \sigma^2. \end{aligned} \tag{9}$$

A.4. Proof: Risk of Pi-DUAL

We focus on the solution of

$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\boldsymbol{v}\in\mathbb{R}^m} \frac{1}{2n}\|\boldsymbol{y} - (\boldsymbol{\gamma}\boldsymbol{X}\boldsymbol{w} + (\boldsymbol{I}-\boldsymbol{\gamma})\boldsymbol{A}\boldsymbol{v})\|^2 \tag{10}$$

to construct an estimator. Here, $\boldsymbol{\gamma}$ refers to a diagonal mask matrix of size $n \times n$ which we use as (pre-defined) gates for Pi-DUAL. We introduce the notations:

- The masked versions of $\boldsymbol{X}$ and $\boldsymbol{A}$: $\bar{\boldsymbol{X}} = \boldsymbol{\gamma}\boldsymbol{X}$ and $\bar{\boldsymbol{A}} = (\boldsymbol{I}-\boldsymbol{\gamma})\boldsymbol{A}$
- The projector onto the span of the columns of $\bar{\boldsymbol{A}}$:

$$\bar{\boldsymbol{\Pi}}_a = \bar{\boldsymbol{A}}(\bar{\boldsymbol{A}}^\top\bar{\boldsymbol{A}})^{-1}\bar{\boldsymbol{A}}^\top \in \mathbb{R}^{n\times n}$$

- The projection $\bar{\boldsymbol{X}}_a = (\boldsymbol{I}-\bar{\boldsymbol{\Pi}}_a)\bar{\boldsymbol{X}} \in \mathbb{R}^{n\times d}$ of $\bar{\boldsymbol{X}}$ onto the orthogonal of the span of the columns of $\bar{\boldsymbol{A}}$, and the matrices

$$\bar{\boldsymbol{H}} = (\bar{\boldsymbol{X}}_a^\top\bar{\boldsymbol{X}}_a)^{-1}\bar{\boldsymbol{X}}_a^\top \in \mathbb{R}^{d\times n} \text{ and } \bar{\boldsymbol{Q}}_a = \bar{\boldsymbol{X}}_a^\top\bar{\boldsymbol{X}}_a \in \mathbb{R}^{d\times d}.$$

We can reuse Lemma I.3 from Collier et al. (2022), with $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{A}})$ in lieu of $(\boldsymbol{X}, \boldsymbol{A})$. The solution of $\boldsymbol{w}$ is thus given by

$$\begin{aligned}\boldsymbol{w}_1 &= \bar{\boldsymbol{H}}\boldsymbol{y} \\ &= \bar{\boldsymbol{H}}(\boldsymbol{\gamma}^*\boldsymbol{X}\boldsymbol{w}^\star + (\boldsymbol{I}-\boldsymbol{\gamma}^*)\boldsymbol{A}\boldsymbol{v}^\star + \varepsilon) \\ &= \bar{\boldsymbol{H}}((\boldsymbol{\Delta}_\gamma + \boldsymbol{\gamma})\boldsymbol{X}\boldsymbol{w}^\star + (\boldsymbol{I} - (\boldsymbol{\Delta}_\gamma + \boldsymbol{\gamma}))\boldsymbol{A}\boldsymbol{v}^\star + \varepsilon) \\ &= \bar{\boldsymbol{H}}(\boldsymbol{\Delta}_\gamma\boldsymbol{\delta}^\star + \bar{\boldsymbol{X}}\boldsymbol{w}^\star + \bar{\boldsymbol{A}}\boldsymbol{v}^\star + \varepsilon) \\ &= \bar{\boldsymbol{H}}\boldsymbol{\Delta}_\gamma\boldsymbol{\delta}^\star + \boldsymbol{w}^\star + \boldsymbol{0} + \bar{\boldsymbol{H}}\varepsilon. \end{aligned}$$

where in the last line, we have used that $\bar{\boldsymbol{H}}\bar{\boldsymbol{X}} = (\bar{\boldsymbol{X}}_a^\top\bar{\boldsymbol{X}}_a)^{-1}\bar{\boldsymbol{X}}_a^\top\bar{\boldsymbol{X}}_a = \boldsymbol{I}$ (because $\boldsymbol{I} - \bar{\boldsymbol{\Pi}}_a = (\boldsymbol{I} - \bar{\boldsymbol{\Pi}}_a)^2$) and $(\boldsymbol{I} - \bar{\boldsymbol{\Pi}}_a)\bar{\boldsymbol{A}} = \boldsymbol{0}$.

Plugging into Eq. (5), we obtain

$$\mathcal{R}(\boldsymbol{w}_1) = \frac{1}{n_1}\|\boldsymbol{\gamma}^\star\boldsymbol{X}\bar{\boldsymbol{H}}\boldsymbol{\Delta}_\gamma\boldsymbol{\delta}^\star + \boldsymbol{\gamma}^\star\boldsymbol{X}\bar{\boldsymbol{H}}\varepsilon\|^2 + \sigma^2.$$

Expanding the square and using that $\mathrm{tr}(\boldsymbol{\gamma}^\star\boldsymbol{X}\bar{\boldsymbol{H}}(\boldsymbol{\gamma}^\star\boldsymbol{X}\bar{\boldsymbol{H}})^\top) = \mathrm{tr}(\boldsymbol{\gamma}^\star\boldsymbol{X}\bar{\boldsymbol{Q}}_a^{-1}\boldsymbol{X}^\top\boldsymbol{\gamma}^\star) = \mathrm{tr}(\bar{\boldsymbol{Q}}_a^{-1}\boldsymbol{Q}_1)$, the final risk expression is

$$\begin{aligned}\mathbb{E}[\mathcal{R}(\boldsymbol{w}_1)] &= \frac{1}{n_1}\|\boldsymbol{\gamma}^\star\boldsymbol{X}\bar{\boldsymbol{H}}\boldsymbol{\Delta}_\gamma\boldsymbol{\delta}^\star\|^2 + \frac{1}{n_1}\mathbb{E}[\|\boldsymbol{\gamma}^\star\boldsymbol{X}\bar{\boldsymbol{H}}\varepsilon\|^2] + \sigma^2 \\ &= \frac{1}{n_1}\|\boldsymbol{\gamma}^\star\boldsymbol{X}\bar{\boldsymbol{H}}\boldsymbol{\Delta}_\gamma\boldsymbol{\delta}^\star\|^2 + \frac{\sigma^2}{n_1}\mathrm{tr}(\bar{\boldsymbol{Q}}_a^{-1}\boldsymbol{Q}_1) + \sigma^2. \end{aligned} \tag{11}$$

B. Experimental Details

We now report the main details of all our experiments. All our experiments, including the reimplementation of other noisy label methods, are built on the open-source `uncertainty_baselines` codebase (Nado et al., 2021) and follow as much as possible the benchmarking practices of Ortiz-Jimenez et al. (2023).

B.1. Datasets

We use the following PI datasets to evaluate the performance of Pi-DUAL and other methods:

CIFAR-10H (Peterson et al., 2019) is a relabeled version for CIFAR-10 (Krizhevsky et al., 2009) test set with 10,000 images. However, as proposed by Collier et al. (2022) we use CIFAR-10H as a training set so we use the standard CIFAR-10 training set as our test set. Following Ortiz-Jimenez et al. (2023), we use the noisiest version of CIFAR-10H (denoted as "worst") in our experiments. It has a noise rate (defined as the percentage of the labels that disagree with the original CIFAR-10 dataset) of approximately 64.6%. The PI of CIFAR-10H consists of annotator IDs, annotator experiences and the time taken for the annotations.

CIFAR-10N and CIFAR-100N (Wei et al., 2022) are relabeled versions of CIFAR-10 and CIFAR-100 with noisy human annotations. In our experiments, we use the noisiest version of these two datasets, known as CIFAR-10N (worst) and CIFAR-100N (fine), which both have a 40.2% noise rate. The PI on these datasets consist on annotator IDs and annotator experience. It is worth noting that compared to CIFAR-10H, and as reported by Ortiz-Jimenez et al. (2023), the PI on these two datasets is of a much lower quality. In general, it is much less predictive of the presence of a label mistake on a specific sample, as the PI features are only provided as averages over batches of samples.

ImageNet-PI (Ortiz-Jimenez et al., 2023) is a relabeled version of the ImageNet ILSVRC12 dataset (Deng et al., 2009). In contrast to the human-relabeled datasets described above, the labels of ImageNet-PI are provided by 16 different deep neural networks pre-trained on the original ImageNet. The PI for this dataset contains the annotator confidence, the annotator ID, the number of parameters of the model and its accuracy. In our experiments, we use both the high-noise (83.8% noise rate) and low-noise version (48.1% noise rate) of ImageNet-PI.

A summary of the features of these datasets is given in Tab. 5.

Table 5. Summary of the main features of each of the datasets used in our experiments.

	CIFAR-10H (worst)	CIFAR-10N (worst)	CIFAR-100N (fine)	ImageNet-PI (low-noise)	ImageNet-PI (high-noise)
Training set size	10k	50k	50k	1.28M	1.28M
PI quality	High	Low	Low	High	High
Annotators	Humans	Humans	Humans	Models	Models
Noise rate	64.6%	40.2%	40.2%	48.1%	83.8%

B.2. Baselines

In our experiments, we compare the performance of Pi-DUAL on different tasks against several baselines selected to provide a fair comparison and good coverage of different methods in the literature. Specifically, we restrict ourselves to methods that only require training a single model on a single stage. We discard, therefore, methods that need multiple stages of training, such as Forward-T (Patrini et al., 2017) or Distillation PI (Lopez-Paz et al., 2015); or multiple models, such as co-teaching (Han et al., 2018) and DivideMix (Li et al., 2020a), as these are more computationally demanding, harder to tune, and in general harder to scale to the large-scale settings we are interested in. Also, most of these methods have been compared against more recent strategies like SOP (Liu et al., 2022) or ELR (Liu et al., 2020), and shown to perform worse than these baselines.

A short description of each of the baselines we compare to is provided below:

Cross-entropy Conventional training strategy consisting in the direct minimization of the cross-entropy loss between the model's predictions and the noisy labels.

SOP (Liu et al., 2022) A noise-modeling method which models the label noise as an additive sparse signal. During training

SOP uses the implicit bias of a custom overparameterized formulation to drive the learning of this sparse components. SOP needs $\mathcal{O}(n \times K)$ extra parameters over the cross-entropy baseline, where n is the number of training samples, and K the number of training classes.

ELR (Liu et al., 2020) This method adds an extra regularization term to the cross-entropy loss to bias the model's predictions towards their value in the early stages of training. To that end, it requires storing a moving average of the model predictions at each training iteration which adds $\mathcal{O}(n \times K)$ extra parameters over the cross-entropy baseline.

HET (Collier et al., 2021) Another noise-modeling method which models the uncertainty in the predictions as heteroscedastic per-sample Gaussian component in the logit space. The original version scales poorly with the number of classes, but the more recent HET-XL version (Collier et al., 2023), allows to scale this modeling approach to datasets with thousands of classes with only $\mathcal{O}(1)$ extra parameters coming from the small network that parameterizes the covariance of the noise. The standard version of HET achieves similar performance to HET-XL on ImageNet and can be run efficiently on this dataset. In our experiments, we thus use HET instead of HET-XL as a baseline.

TRAM (Collier et al., 2022) A PI-method which uses two heads, one with access to PI and one without it, to learn $p(\tilde{y}|\boldsymbol{x}, \boldsymbol{a})$ and $p(\tilde{y}|\boldsymbol{x})$ respectively. However, the feature extraction network leading to these heads is only trained using the gradients coming from the PI head. During inference, only the no-PI head is used. TRAM only requires $\mathcal{O}(1)$ extra parameters for the additional PI head.

TRAM++ (Ortiz-Jimenez et al., 2023) On top of TRAM, TRAM++ augments the PI features with random sample-identifier to encourage the model to use the PI as a learning shortcut to memorize the noisy labels.

AFM (Collier et al., 2022) Another PI-method that during training learns to approximate $p(\tilde{y}|\boldsymbol{x}, \boldsymbol{a})$ and during inference uses approximate marginalization based on the independence assumption $p(\boldsymbol{a}|\boldsymbol{x}) \approx p(\boldsymbol{a})$ and Monte-Carlo sampling to marginalize over $\boldsymbol{a}$. AFM only requires $\mathcal{O}(1)$ extra parameters to accomodate for the PI in the last layers.

B.3. Hyperparameter Tuning Strategy

As mentioned in Sec. 4, to ensure a fair comparison of the different methods, we apply the same hyperparameter tuning strategy in all our experiments and for all methods. In particular, we use a noisy validation set taken from the training set to select the best hyperparameters of a grid search. On CIFAR-10H, we randomly select 4% of the samples; on CIFAR-10N and CIFAR-100N, 2%; and on ImageNet-PI, 1% of all the samples in the training set. In all the experiments presented in the main text we use early stopping to select the best epoch to evaluate each method. Early stopping is also performed over the noisy validation set, although the reported accuracies are given over the clean test set.

B.4. Training Details for CIFAR

General settings We use a WideResNet-10-28 architecture in our CIFAR experiments. We train all models for 90 epochs, with the learning rate decaying multiplicatively by 0.2 after 36, 72 and 96 epochs. We use a batch size of 256 in all experiments, and train the models with an SGD optimizer with 0.9 Nesterov momentum. In our grid searches, we sweep over the initial learning rate $\{0.01, 0.1\}$ and weight decay strength $\{10^{-4}, 10^{-3}\}$. We always we use random crops combined with random horizontal flips as data augmentation.

Method-specific settings For ELR (Liu et al., 2020), we additionally sweep over the temporal ensembling parameter β of $\{0.5, 0.7, 0.9\}$ and the regularization coefficient λ of $\{1, 3, 7\}$. For SOP (Liu et al., 2022), we sweep over the learning rate for u_i of $\{1, 10, 100\}$, as well as the learning rate for v_i of $\{1, 10, 100, 1000\}$. We refer to the original papers for ELR (Liu et al., 2020) and SOP (Liu et al., 2022) for detailed illustration of the hyperparameters.

For TRAM and AFM, we set the PI tower width to 1024 following the settings in Collier et al. (2022). For TRAM++, we tune the PI tower width over a range of $\{512, 1024, 2048, 4096\}$. Additionally, we tune the random PI length of $\{8, 14, 28\}$, and no-PI loss weight over $\{0.1, 0.5\}$ for TRAM++.

For HET, we tune the heteroscedastic temperature over a range of $\{0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 2.0, 3.0, 5.0\}$. For CIFAR-10H and CIFAR-10N, the number of factors for the low-rank component of the heteroscedastic covariance matrix is set to 3, while we set it to 6 for CIFAR-100N experiments.

For Pi-DUAL, we set the width of the noise network and gating network to 1024 in the CIFAR-10H experiments, and to 2048 for the experiments with CIFAR-10N and CIFAR-100N. We use a three-layer MLP with ReLU activations for the noise network. The gating network shares the first layer with the noise network followed by another two fully-connected layers with ReLU activations and a sigmoid activation at its output. We additionally search the random PI length over {4,8,12,16}. We do not apply weight decay regularization on the gating network and noise network for experiments on CIFAR for Pi-DUAL.

We highlight that, compared with the competing methods (except the cross-entropy baseline), Pi-DUAL requires the smallest budget for hyperparameter tuning as it requires to tune only one additional hyperparameter, i.e., the random PI length; while the other methods require tuning at least two additional method-specific hyperparameters.

B.5. Training Details for ImageNet-PI

General settings. For all experiments with ImageNet-PI, we use a ResNet-50 architecture and SGD optimizer with Nesterov momentum of 0.9. The models are trained for 90 epochs in total with a batch size of 2048, with the learning rate decaying multiplicatively by 0.1 after 30, 60 and 80 epochs. The initial learning rate is set to 0.1, and we search over $\{10^{-5}, 10^{-4}\}$ for the weight decay strength. Random crop and random horizontal flip are used for data augmentation.

Method-specific settings. For all PI-related baselines (TRAM, TRAM++, AFM), we set the PI tower width to 2048. We set the no-PI loss weight to 0.5 and set the random PI length of 30 for TRAM++. For HET, we set the number of factors for the low-rank component of the heteroscedastic covariance matrix to 15, and we set the heteroscedastic temperature to 3.0.

For Pi-DUAL, we set the random PI length to 30. The weight decay regularization on the gating network and noise network is the same as the prediction network. The architecture of the noise and gating network is the same as the one of the CIFAR experiments with a width of 2048.

C. Additional Results

C.1. Training Dynamics on CIFAR-10N and ImageNet-PI (low noise)

Here we provide the training dynamics for CIFAR-10N and ImageNet-PI (low noise) in Fig. 6 with the same findings as in Sec. 5.1.

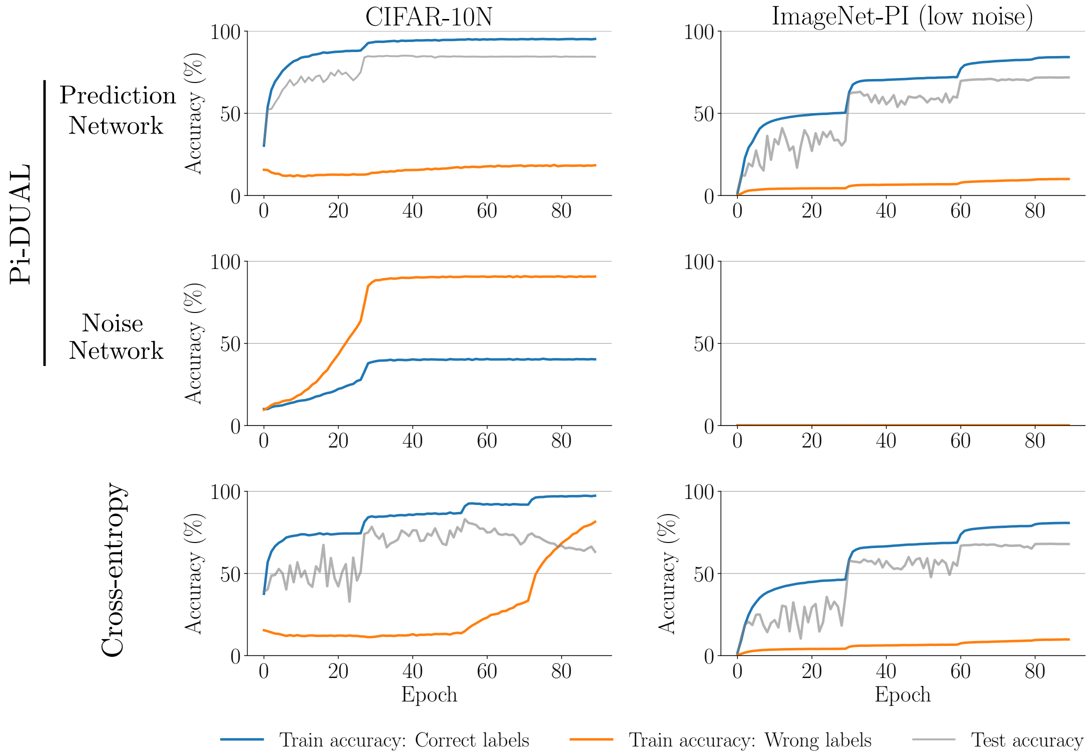

Figure 6. Training accuracy dynamics on correct and mislabled samples, respectively for the prediction and noise sub-networks on CIFAR-10N (worst) and ImageNet-PI (low noise).

C.2. Distribution of the Gating Network Predictions on CIFAR-10N and ImageNet-PI (low noise)

We show the distribution of the predictions of the gating network for CIFAR-10N and ImageNet-PI (low noise) in Fig. 7 with the same findings as in Sec. 5.2.

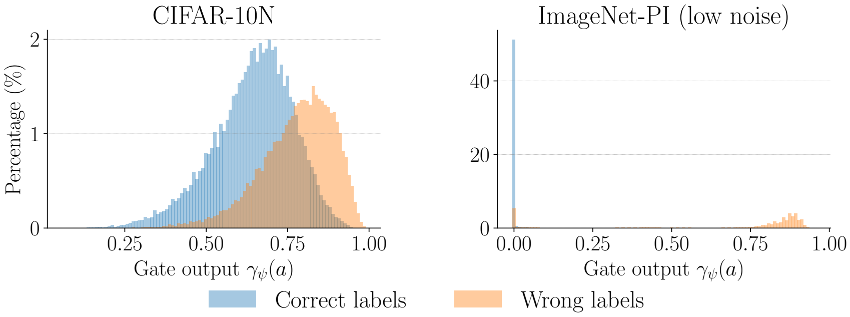

Figure 7. Distribution of $\gamma_{\boldsymbol{\psi}}(\boldsymbol{a})$, on correct training examples with correct and wrong labels for CIFAR-10N and ImageNet-PI (low noise).

C.3. Distribution of the Prediction Network Confidence on CIFAR-10N and ImageNet-PI (low noise)

We present the distribution of the prediction confidence of the prediction network on observed labels for CIFAR-10N and ImageNet-PI (low noise) in Fig. 8 complementing the findings of Sec. 4.3. From the figure, we see that the confidence of the prediction network is clearly separated over samples with clean and wrong labels.

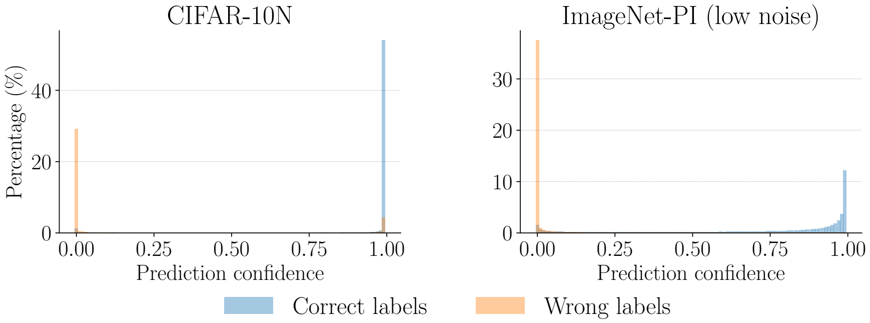

Figure 8. Distribution for the prediction network's confidence on the observed noisy labels, separated by correctly and wrongly labeled samples.

C.4. Results without Early-stopping

In the main text, as it is standard practice in the literature, we always provided results using early stopping. However, as mentioned before, early stopping is a key ingredient to achieve good performance by other methods, not Pi-DUAL. Indeed, Pi-DUAL still barely overfits to the incorrect labels using the prediction network, and thus it does not require early-stopping to achieve good results. To demonstrate this, Tab. 6 reports the results of the same experiments as in Tab. 2, but without using early stopping. From the table, we observe the performance of Pi-DUAL does not degrade in any of the datasets, while for other methods it suffers a heavily.

The same applies in the case of the noise detection results, where in Tab. 7 we see that Pi-DUAL can still detect the noisy labels equally well as in Tab. 3 without the use of early stopping. The other methods on the other hand perform worse when applied to the last training epoch than to the early stopped one.

C.5. Augmenting Pi-DUAL with State-of-the-art Regularization Techniques

In main text, we follow the common practice in literature and only compare Pi-DUAL with methods which do not incorporate techniques from semi-supervised learning, which greatly increases the computational cost and complexity. Here we propose

Table 6. Test accuracy (without early stopping) on CIFAR-10H, CIFAR-10N, CIFAR-100N, and ImageNet-PI, comparing Pi-DUAL with previous methods (grouped by PI-based methods and No-PI methods). Mean and standard deviation are reported over 5 individual runs for CIFAR experiments and 3 runs for ImageNet experiments.

	Methods	CIFAR-10H (worst)	CIFAR-10N (worst)	CIFAR-100N (fine)	ImageNet-PI (low-noise)	ImageNet-PI (high-noise)
No-PI	Cross-entropy	42.4$_{\pm0.2}$	67.7$_{\pm0.6}$	55.8$_{\pm0.2}$	68.2$_{\pm0.2}$	47.4$_{\pm0.4}$
	ELR	49.2$_{\pm1.2}$	84.3$_{\pm0.4}$	**63.8**$_{\pm0.3}$	-	-
	SOP	50.3$_{\pm0.7}$	**86.0**$_{\pm0.3}$	61.1$_{\pm0.2}$	-	-
PI	TRAM	59.2$_{\pm0.2}$	67.0$_{\pm0.4}$	56.4$_{\pm0.3}$	69.6$_{\pm0.1}$	54.0$_{\pm0.0}$
	TRAM++	64.7$_{\pm0.6}$	82.3$_{\pm0.1}$	60.6$_{\pm0.2}$	69.5$_{\pm0.0}$	54.1$_{\pm0.1}$
	AFM	61.2$_{\pm0.7}$	69.8$_{\pm0.5}$	58.9$_{\pm0.3}$	70.3$_{\pm0.0}$	55.3$_{\pm0.2}$
	Pi-DUAL (Ours)	**73.8**$_{\pm0.3}$	84.9$_{\pm0.3}$	**64.2**$_{\pm0.3}$	**71.7**$_{\pm0.1}$	**62.3**$_{\pm0.1}$

Table 7. AUC of different noise detection methods without using early-stopping.

Methods	CIFAR-10H (worst)	CIFAR-10N (worst)	CIFAR-100N (fine)	ImageNet-PI (low-noise)	ImageNet-PI (high-noise)
Cross-entropy	0.558	0.676	0.666	0.935	0.941
ELR	0.660	0.839	0.843	-	-
SOP	0.743	0.932	0.793	-	-
TRAM++	0.887	0.946	0.890	0.937	0.959
Pi-DUAL (conf.)	0.972	**0.960**	**0.910**	**0.953**	**0.987**
Pi-DUAL (gate)	**0.983**	0.815	0.729	**0.953**	0.986

an extension of Pi-DUAL, Pi-DUAL+, which boosts the performance of Pi-DUAL with additional regularization techniques.

Prior works in the literature have shown that the noisy-label training methods can be boosted with techniques from semi-supervised learning domain (Li et al., 2020b; Liu et al., 2020; 2022), at the cost of extra complexity costs and complexity. Pi-DUAL + adds to Pi-DUAL two regularization techniques, label smoothing and prediction consistency regularizer.

Label smoothing Label smoothing is a regularization technique that was introduced to mitigate overconfidence during training by replacing hard labels with smoothed soft labels (Szegedy et al., 2016). It has become a widely-used method to improve model generalization performance in classification tasks.

Consistency regularizer Prediction consistency regularizer is commonly used in both the semi-supervised learning (Berthelot et al., 2019; Sohn et al., 2020; Xie et al., 2020) and learning with label noise literature (Cheng et al., 2020; Liu et al., 2022). It encourages the prediction consistency of the model across different input views. In Pi-DUAL+, we add a consistency regularizer $\mathcal{L}_C$ on the generalization term. Specifically, $\mathcal{L}_C$ is defined as the Kullback-Leibler divergence between softmax prediction from images with the default augmentation in Sec. B.4 and softmax predictions from the corresponding images augmented by Unsupervised Data Augmentation (Xie et al., 2020):

$$\mathcal{L}_C = \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(\mathrm{softmax}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) \parallel \mathrm{softmax}(f_{\boldsymbol{\theta}}(\mathrm{UDA}(\boldsymbol{x}_i)))) \tag{12}$$

We use a hyper-parameter λ_C to control the strength of the consistency regularizer.

For Pi-DUAL+, we sweep the label smoothing over {0, 0.4}, and λ_C over {0.5, 1}. We train the Pi-DUAL+ for 300 epochs with a batch size of 128. The learning rate is set as 0.1 and decays with a cosine annealing schedule (Loshchilov & Hutter, 2016). Additionally, we sweep the random-pi length over {4, 8} and set the $l2$ regularization strength to $1e^{-4}$.

We compare Pi-DUAL+ against several semi-supervised learning pipeline methods, including Divide-Mix (Li et al., 2020a), CORES* (Cheng et al., 2020), PES(semi) (Bai et al., 2021), ELR+ (Liu et al., 2020) and SOP+ (Liu et al., 2022). The results are compared in three datasets, CIFAR-10H, CIFAR-10N and CIFAR-100N, as shown in Tab. 8.

Table 8. Test accuracy on CIFAR-10H, CIFAR-10N and CIFAR-100N, comparing Pi-DUAL+ against state-of-the-art methods which combine noisy labels techniques with semi-supervised learning methods. The results of baseline methods on CIFAR-10N and CIFAR-100N are taken from (Wei et al., 2021; Liu et al., 2022)

	CIFAR-10H	CIFAR-10N	CIFAR-100N
CE	$51.10_{\pm 2.20}$	$80.60_{\pm 0.20}$	$60.40_{\pm 0.50}$
Divide-Mix	$71.68_{\pm 0.27}$	$92.56_{\pm 0.42}$	$\mathbf{71.13}_{\pm 0.48}$
PES(semi)	$71.16_{\pm 1.78}$	$92.68_{\pm 0.22}$	$70.36_{\pm 0.33}$
ELR+	$54.46_{\pm 0.50}$	$91.09_{\pm 1.60}$	$66.72_{\pm 0.07}$
CORES*	$57.80_{\pm 0.57}$	$91.66_{\pm 0.09}$	$55.72_{\pm 0.42}$
SOP+	$66.02_{\pm 0.06}$	$\mathbf{93.24}_{\pm 0.21}$	$67.81_{\pm 0.23}$
Pi-DUAL+	$\mathbf{83.23}_{\pm 0.26}$	$\mathbf{93.31}_{\pm 0.21}$	$67.99_{\pm 0.08}$

From the table, we observe that, Pi-DUAL+ outperforms the other methods by a margin of over 11.0 points on CIFAR-10H, while it performs on par with the state-of-the-art on the other two CIFAR-*N datasets. It demonstrates again the importance of good-quality PI features to maximize the performance of Pi-DUAL/Pi-DUAL+, and also demonstrates that Pi-DUAL can be effectively boosted by semi-supervised learning methods.

C.6. Ablations over Model Structures

C.6.1. Ablation for Prediction Network Backbone

In all our CIFAR-level experiments (for both Pi-DUAL and other baselines methods), we used a WideResNet-10-28 as model backbone. Here we replace the WideResNet-10-28 with a ResNet-34 and present the performance comparison between Pi-DUAL and CE baseline on Tab. 9.

Table 9. Accuracy comparison between Pi-DUAL and CE on three datasets, using two different model backbones: WideResNet-10-28 and ResNet34.

	WideResNet-10-28		ResNet34	
Dataset \Method	CE	Pi-DUAL	CE	Pi-DUAL
CIFAR-10H	$51.1_{\pm 2.2}$	$\mathbf{71.3}_{\pm 3.3}$	$51.4_{\pm 2.1}$	$\mathbf{69.3}_{\pm 3.1}$
CIFAR-10N	$80.6_{\pm 0.2}$	$\mathbf{84.9}_{\pm 0.4}$	$80.7_{\pm 1.0}$	$\mathbf{84.5}_{\pm 0.4}$
CIFAR-100N	$60.4_{\pm 0.5}$	$\mathbf{64.2}_{\pm 0.3}$	$56.9_{\pm 0.4}$	$\mathbf{62.2}_{\pm 0.3}$

From the results, we observe that Pi-DUAL maintains its performance improvement over CE with a different model backbone, exceeding the performance of CE baseline by a notable margin in all three datasets.

C.6.2. Ablations on Structure for Noise and Gating Networks

Ablations on the width In the paper we set the width of the PI-related modules (for both noise network and gating network) by default to 1024 for CIFAR-10H, and 2048 for all other experiments, without fine-tuning. Here we provide the results for using different widths for the PI networks on three CIFAR datasets in Tab. 10.

From the table we observe that the performance of Pi-DUAL benefits from larger network width for the PI-related modules.

Ablations on the depth In the paper we set the depth of the PI-related modules by default to 3 by default for all experiments, without fine-tuning. Here we provide the results for using different depths for the PI networks on three CIFAR datasets in Tab. 11. Note that the width of the PI-related modules are set to the default value when tuning the depth.

From the table we observe that the depth of the PI-related modules have to be at least of 3 layers to maximize the performance

Table 10. Accuracy of Pi-DUAL on three datasets, varying the width of noise and gating networks of Pi-DUAL.

Dataset \ Model width	512	1024	2048	4096
CIFAR-10H	**72.8**$_{\pm 2.9}$	**71.3**$_{\pm 3.3}$	**71.4**$_{\pm 3.6}$	**71.2**$_{\pm 3.8}$
CIFAR-10N	83.8$_{\pm 0.4}$	83.6$_{\pm 0.7}$	**84.9**$_{\pm 0.4}$	**85.3**$_{\pm 0.2}$
CIFAR-100N	62.3$_{\pm 0.2}$	63.7$_{\pm 0.4}$	**64.2**$_{\pm 0.3}$	**64.4**$_{\pm 0.2}$

Table 11. Accuracy of Pi-DUAL on three datasets, varying the depth of noise and gating networks of Pi-DUAL.

Dataset \ Model depth	2	3	4
CIFAR-10H	**68.9**$_{\pm 2.0}$	**71.3**$_{\pm 3.3}$	**70.1**$_{\pm 3.5}$
CIFAR-10N	83.4$_{\pm 0.4}$	**84.9**$_{\pm 0.4}$	**85.2**$_{\pm 0.3}$
CIFAR-100N	59.4$_{\pm 1.0}$	**64.2**$_{\pm 0.3}$	**64.1**$_{\pm 0.2}$

of Pi-DUAL.

C.6.3. Ablations on Modeling of Noise Signals

For Pi-DUAL, we chose to the model the label noise signal as a function of PI features $\boldsymbol{a}$. We show here that this modeling effectively prevents the model from memorizing the noise signal with input image features $\boldsymbol{x}$.

We show in Tab. 12 that the performance of Pi-DUAL deteriorates significantly if the noise signal is modeled as a function of both PI features $\boldsymbol{a}$ and input image features $\boldsymbol{x}$. This is because in this unconstrained setup, the model would directly memorize the label noise using the input images. This result demonstrates again that our modeling of label noise effectively decouples the learning of clean labels and the overfitting to the wrong labels.

Table 12. Ablation for modeling of the noise signal.

Method \ Dataset	CIFAR-10H	CIFAR-10N	CIFAR-100N
Cross-entropy	51.1$_{\pm 2.2}$	80.6$_{\pm 0.2}$	60.4$_{\pm 0.5}$
Pi-DUAL (model label noise with $\boldsymbol{a}$)	**71.3**$_{\pm 3.3}$	**84.9**$_{\pm 0.4}$	**64.2**$_{\pm 0.3}$
Pi-DUAL (model label noise with both $\boldsymbol{x}$ and $\boldsymbol{a}$)	57.8$_{\pm 1.9}$	82.5$_{\pm 0.2}$	43.6$_{\pm 14.4}$

C.7. Training Pi-DUAL with Loss Function of TRAM

While both TRAM and Pi-DUAL utilize PI features to combat label noise, Pi-DUAL uses a much simpler loss function than TRAM, which implements a weighted combination of two functions to train its two heads (Collier et al., 2022). To validate the importance of the loss design of Pi-DUAL, which prevents the prediction network from overfitting to label noise, here we present the results for Pi-DUAL if we train it with the loss function of TRAM in Tab. 13.

C.8. Performance of Pi-DUAL with Corrupted PI Features

We have emphasized in the main text the importance of the quality of PI features to the performance of Pi-DUAL. In this section, we perform an ablation study where we gradually corrupt the PI features of CIFAR-10H, and train Pi-DUAL with the dataset with corrupted PI features.

For each experiment, we randomly corrupt the PI features of a percentage of samples in the train set, where the corrupted PI features will be replaced by randomly generated PI features. Specifically, we substitute the annotator ID by a new random integer, and we substitute all other continual PI features by a random Gaussian vector with the same mean and standard deviation as the distribution of those features in the training set. We vary gradually the percentage of corrupted samples and report the performance for Pi-DUAL trained correspondingly in Tab. 14.

From the table, we observe that the accuracy of Pi-DUAL decreases as there are more noise in the PI features of the training set, which demonstrates again the importance for high-quality PI features to maximize the performance of Pi-DUAL.

Table 13. Ablation for training Pi-DUAL using the loss function of TRAM.

Dataset \ Method	CE	Pi-DUAL	Pi-DUAL with TRAM loss
CIFAR-10H	$51.1_{\pm 2.2}$	**71.3**$_{\pm 3.3}$	$61.6_{\pm 4.8}$
CIFAR-10N	$80.6_{\pm 0.2}$	**84.9**$_{\pm 0.4}$	$81.6_{\pm 0.9}$
CIFAR-100N	$60.4_{\pm 0.5}$	**64.2**$_{\pm 0.3}$	$59.0_{\pm 0.6}$

Table 14. Performance of Pi-DUAL with different levels of corruption in the PI features for the train set.

PI Corruption percentage	No corrupt	25% corrupt	50% corrupt	75% corrupt	100% corrupt
Accuracy	**71.3**$_{\pm 3.3}$	$66.4_{\pm 2.7}$	$63.9_{\pm 0.7}$	$52.6_{\pm 4.0}$	$47.2_{\pm 3.7}$

C.9. Importance of Individual PI Features

We study here the importance of each individual PI feature, by performing an ablation experiment on three CIFAR datasets where we remove one of the PI features while training Pi-DUAL.

The results are presented in Tab. 15. It suggest that annotator ID is generally an important PI feature for these datasets, which suggests that the quality of the annotation varies from the annotators.

Table 15. Ablation for the importance of each PI feature.

Method \ Dataset	CIFAR-10H	CIFAR-10N	CIFAR-100N
Cross-entropy	$51.1_{\pm 2.2}$	$80.6_{\pm 0.2}$	$60.4_{\pm 0.5}$
Pi-DUAL (with all PI)	**71.3**$_{\pm 3.3}$	**84.9**$_{\pm 0.4}$	**64.2**$_{\pm 0.3}$
(w/o annotator ID)	$55.9_{\pm 1.0}$	$84.0_{\pm 0.2}$	$62.9_{\pm 0.3}$
(w/o annotator times)	**73.6**$_{\pm 0.2}$	**84.7**$_{\pm 0.2}$	**64.2**$_{\pm 0.1}$
(w/o random PI)	**71.6**$_{\pm 2.9}$	$82.1_{\pm 0.3}$	$61.5_{\pm 0.4}$
(w/o trial index)	**74.4**$_{\pm 0.1}$	NA	NA

D. Computational Cost in Large-scale Datasets

In this paper, we used a TPU V3 with 8 cores for experiments on ImageNet-PI, and A100 (40G) for experiments on CIFAR.

Here we provide a computational cost analysis for Pi-DUAL on ImageNet-PI, comparing it with other baseline methods, with respect to both the number of parameters and the training time in Tab. 16. Note that in the table we do not have run time for SOP and ELR as these two methods are very hard to scale to ImageNet-PI, where they require over 1 billion parameters.

From the table, we can see that Pi-DUAL is a scalable method with almost the same training time as the cross-entropy baseline. Importantly, its parameters do not scale with neither the number of classes nor the number of samples, making it scalable to very large datasets.

Table 16. Computational cost analysis on ImageNet-PI in terms of number of parameters and running time, comparing Pi-DUAL with baseline methods.

	CE	TRAM++	Pi-DUAL	HET	SOP	ELR
Number of parameters	26M	32M	36M	58M	>1B	>1B
Run time per step	0.510s	0.541s	0.566s	0.575s	-	-