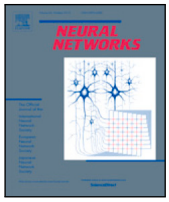
Neural Networks 172 (2024) 106122

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

Bayesian DivideMix++ for Enhanced Learning with Noisy Labels

Bhalaji Nagarajan [a,*], Ricardo Marques [a,c], Eduardo Aguilar [a,b,c], Petia Radeva [a,c]

[a] Dept. de Matemàtiques i Informàtica, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain
[b] Dept. de Ingeniería de Sistemas y Computación, Universidad Católica del Norte, Avenida Angamos 0610, 1270709, Antofagasta, Chile
[c] Computer Vision Center, Cerdanyola (Barcelona), Spain

ARTICLE INFO

Keywords:
Learning with noisy labels
Neural network memorization
Data augmentation
Self-supervised pre-training
Label uncertainty
Monte-Carlo dropouts

ABSTRACT

Leveraging inexpensive and human intervention-based annotating methodologies, such as crowdsourcing and web crawling, often leads to datasets with noisy labels. Noisy labels can have a detrimental impact on the performance and generalization of deep neural networks. Robust models that are able to handle and mitigate the effect of these noisy labels are thus essential. In this work, we explore the open challenges of neural network memorization and uncertainty in creating robust learning algorithms with noisy labels. To overcome them, we propose a novel framework called "Bayesian DivideMix++" with two critical components: (i) DivideMix++, to enhance the robustness against memorization and (ii) Monte-Carlo MixMatch, which focuses on improving the effectiveness towards label uncertainty. DivideMix++ improves the pipeline by integrating the warm-up and augmentation pipeline with self-supervised pre-training and dedicated different data augmentations for loss analysis and backpropagation. Monte-Carlo MixMatch leverages uncertainty measurements to mitigate the influence of uncertain samples by reducing their weight in the data augmentation MixMatch step. We validate our proposed pipeline using four datasets encompassing various synthetic and real-world noise settings. We demonstrate the effectiveness and merits of our proposed pipeline using extensive experiments. Bayesian DivideMix++ outperforms the state-of-the-art models by considerable differences in all experiments. Our findings underscore the potential of leveraging these modifications to enhance the performance and generalization of deep neural networks in practical scenarios.

1. Introduction

Deep Neural Networks (DNNs) have achieved remarkable advancements in tackling complex and challenging problems such as image classification, outperforming human-level performances (Schmarje, Santarossa, Schröder, & Koch, 2021). One key contributing factor to the high performance of DNNs is the availability of large-scale training datasets, like ImageNet-21K (Ridnik, Ben-Baruch, Noy, & Zelnik-Manor, 2021), OpenImages (Kuznetsova et al., 2020), and JFT-3B (Zhai, Kolesnikov, Houlsby, & Beyer, 2022). However, curating such extensive datasets with high precision poses significant challenges. Data collection is labour-intensive and expensive in both sample accumulation and sample labelling (Liao, Kar, & Fidler, 2021). Emerging research has explored several cost-effective alternatives, like web crawling, human crowdsourcing, and creating annotations using semi-supervised models, thereby enabling the creation of large-scale training datasets. However, these methods and processes carry an inherent introduction of noise in the assigned labels (Northcutt, Jiang, & Chuang, 2021; Oyen, Kucer, Hengartner, & Singh, 2022; Wei et al., 2021). As shown in the literature, real-world datasets exhibit considerable label noise estimated between 8.0% to 38.5 Training with noisy labels impacts the generalization of the DNNs due to their over-parameterization (Allen-Zhu, Li, & Liang, 2019) and strong memorization capability (Zhang, Bengio, Hardt, Recht, & Vinyals, 2021).

Learning with Noisy Labels (LNL) has been a long-studied problem (Angluin & Laird, 1988) due to its importance in creating models that are robust towards noisy labels. Recent advances in LNL algorithms use a variety of strategies (Song, Kim, Park, Shin, & Lee, 2022), such as modifying the loss functions (Wu et al., 2021; Zhang, Niu, & Sugiyama, 2021), using more robust loss functions (Ma et al., 2020; Wang et al., 2019), adding regularization (Chen, Hu, Shen, Ai, & Suykens, 2022; Menon, Rawat, Reddi, & Kumar, 2020), sample re-weighting (Ren, Zeng, Yang, & Urtasun, 2018), and label aggregation (Wu et al., 2023). A more recent and promising class of LNL algorithms is **sample selection**, which relies on selecting clean samples for training the models. These algorithms work on the intuition that less noisy data leads to more robust DNNs (Arpit et al., 2017). Using small-loss for selecting samples is a well-established technique (Arazo, Ortego, Albert, O'Connor, & McGuinness, 2019; Li, Socher, & Hoi, 2020). Earlier

* Corresponding author.
E-mail address: bhalaji.nagarajan@ub.edu (B. Nagarajan).

https://doi.org/10.1016/j.neunet.2024.106122
Received 2 August 2023; Received in revised form 4 December 2023; Accepted 9 January 2024
Available online 10 January 2024

sample selection-based methods suffered a significant challenge of error accumulation. Using two *co-operative* 'peer' networks reduced the error flow introduced by the noisy labels (Jiang, Zhou, Leung, Li, & Fei-Fei, 2018; Malach & Shalev-Shwartz, 2017). Co-teaching (Han et al., 2018) and Co-teaching+ (Yu et al., 2019) use clean samples of one network to train the other. On the other hand, JoCoR (Wei, Feng, Chen, & An, 2020) train two networks using a joint loss for each training example.

Advances in Semi-supervised learning have helped to reduce the need for labelled data by leveraging unlabelled data. MixMatch (Berthelot et al., 2019) integrated several dominant semi-supervised approaches into a single framework. The success of MixMatch, combined with the effectiveness of training 'peer' networks, led the way for the emergence of a prominent LNL benchmark algorithm - **DivideMix** (Li et al., 2020). DivideMix leveraged the per-sample loss distribution and modelled it using a Gaussian Mixture Model (GMM) to divide the samples into clean samples and noisy samples dynamically. Subsequently, DivideMix used the clean samples as a labelled set and noisy samples as an unlabelled set and employed an improved version of MixMatch for semi-supervised training. Ever since, DivideMix has served as the foundation for numerous LNL algorithms, such as Contrast2Divide (Zheltonozhskii, Baskin, Mendelson, Bronstein, & Litany, 2022), AugDesc (Nishi, Ding, Rich, & Hollerer, 2021), Probabilistic Noise Perception (Sun et al., 2022), ProMix (Wang, Xiao, Dong, Feng, & Zhao, 2022) and SplitNet (Kim, Ryoo, Cho, & Kim, 2022).

A typical behaviour of any DNN is to learn the patterns first and then gradually *memorize* all the samples (Arpit et al., 2017). In the case of datasets with noisy labels, the DNNs would be able to learn the clean labels first, followed by the noisy ones. Sample selection algorithms choose the small-loss samples as clean samples for subsequent training steps (Wei et al., 2020). The rationale behind this strategy is that small-loss samples are assumed to have a lower likelihood of being affected by label noise, thus making them more reliable for training the model. However, this approach to sample selection is arguable (Xia et al., 2021b). The reason is that small-loss samples might not always represent the true underlying data distribution or capture the challenging patterns present in the dataset. Also, this might not lead towards a generalized performance (Ji, Oh, Hyun, Kwon, & Park, 2021). Moreover, in most cases, the low-loss criterion uses the losses by the current prediction, making the estimation of the noisy class posterior unstable (Yao et al., 2020). Additionally, not all noisy samples have high losses. When a model encounters samples that are not so easy to learn, such as those in non-dominant classes, it would have a high loss (Xia et al., 2021b). Misleading predictions due to uncertainty can also impact the identification of clean samples (Huang, Bai, Zhao, Bai, & Wang, 2022), which, in turn, can affect the quality of training data for the subsequent steps.

Our proposal. With this understanding of LNL algorithms, we investigate the open challenges surrounding the memorization of noisy samples and the uncertainty of clean-noisy samples split that can affect the learning performance in the presence of noisy labels. We propose **Bayesian DivideMix++** using two novel components, DivideMix++ and Monte-Carlo MixMatch, to tackle these issues in the LNL algorithms. Specifically, we study the popular DivideMix algorithm for these challenges and improve the training pipeline, which, as shown by our results, leads to substantial improvements in the learning process. On this front, our main contributions to this work are outlined as follows:

- **DivideMix++:** DNN memorization has been a well-established challenge to robust LNL algorithms. We study the influence of pre-training and augmentation strategies in the challenge of memorization and establish a promising model called Bayesian DivideMix++, modifying both these phases of the training pipeline.
- **Monte-Carlo MixMatch:** We analyse the epistemic uncertainty present in the training data and improve the MixMatch component, which is one of the core components of the training process. We study the effectiveness of this improvement with respect to label uncertainty and show how using uncertainty is beneficial in sample selection.
- We evaluate the efficacy of our proposed pipeline using different benchmarks and real-world LNL datasets. We perform an extensive analysis of the results and show the improvements with respect to several state-of-the-art methods. We also present an ablation study to verify the importance of each component used in our proposal.

The rest of the paper is structured as follows: We present a comprehensive review of relevant literature in Section 2. We give the background information and rationale in Section 3. We explain the details of our proposed method in Section 4 and present the results used to validate our approach in Section 5. We provide detailed analysis in Section 6 and then present the concluding remarks in Section 7.

2. Related work

In this section, we provide an overview of the latest literature that is highly relevant to our work. We first present the works focussed on LNL algorithms and later highlight works based on Uncertainty quantification.

2.1. Learning with noisy labels

LNL has garnered significant research interest in recent years, and several works have been proposed using deep learning algorithms to tackle the problem of label noise (Song et al., 2022). LNL algorithms fall into different families of algorithms based on their mode of operation.

Loss correction methods. Loss correction methods modify the loss of samples using a noise transition matrix (Zhang, Niu, & Sugiyama, 2021). The transition matrices are constructed using the probability that a clean label flips into a noisy label (Patrini, Rozza, Menon, Nock, & Qu, 2016). Class2Simi (Wu et al., 2021) instead used a noise label similarity transformation to simplify the LNL problem. Dual-T (Yao et al., 2020) factorized the transition matrix into a product of two easy-to-estimate transition matrices. However, in these methods, estimating the label transition matrix is quite challenging as it is hard to calculate this probability accurately.

Regularization-based methods. Regularization-based methods find wide usage, as they limit the memorization of labels and make algorithms more robust. Using Data Augmentation (DA) strategies such as Mixup (Zhang, Cisse, Dauphin, & Lopez-Paz, 2017), adding dropouts (Chen et al., 2022), and using gradient clipping (Menon et al., 2020) have proved to be effective in LNL solutions. Advanced regularization techniques such as Progressive Early Stopping (Bai et al., 2021), Neighbour Consistency Regularization (Iscen, Valmadre, Arnab, & Schmid, 2022), Multi-Objective Interpolation Training (Ortego, Arazo, Albert, O'Connor, & McGuinness, 2021), k-nearest neighbour-based filtering (Bahri, Jiang, & Gupta, 2020), Robust early-learning (Xia et al., 2021a), Sparse over-parameterization (Liu, Zhu, Qu, & You, 2022) and Scalable Penalized Regression (Wang, Sun, & Fu, 2022) have been successful in creating robust LNL algorithms. Importantly, these methods often entail complex architectures and sensitive hyperparameters, making it difficult during training.

Noise-robust loss functions and architectures. Another category of LNL algorithms uses noise-robust loss functions. Mean Absolute Error (Ghosh, Kumar, & Sastry, 2017) was more robust than the standard cross-entropy loss. Symmetric cross-entropy learning (Wang et al., 2019) used a reverse cross-entropy to boost the cross-entropy loss in handling the overfitting class labels. Active Passive Loss (Ma et al., 2020) combined two robust loss functions mutually helping each other. Reweighting samples (Ren et al., 2018), making robust architectures such as

noise adaptation layer (Goldberger & Ben-Reuven, 2017), and using distillation (Li, Yang, et al., 2017) are some of the other methods that have increased the robustness of LNL algorithms. Recently, contrastive learning algorithms such as Sel-CL+ (Li, Xia, Ge, & Liu, 2022), supervised contrastive learning for label correction (Huang, Lin, & Xu, 2022), initializing with representations learned by contrastive learning (Ghosh & Lan, 2021) have proven successful in LNL training.

Sample selection methods. Of late, sample selection techniques have become very popular amongst the LNL community. These methods rely on selecting a possible clean subset of data and providing different training strategies for the clean and noisy subsets. The challenge is finding the criterion used in selecting clean or noisy samples. Several strategies are used in sample selection, such as clustering training data (Huang, Qu, Jia, & Zhao, 2019) and progressively removing the noisy samples from subsequent training steps. A common strategy in sample selection relies on per-sample loss, designating samples with minimal loss as clean samples (Jiang et al., 2018). Following this, a two-component beta-mixture model (Arazo et al., 2019) was employed to fit the loss distribution and split the training samples into clean and noisy samples. Alternatively, DivideMix (Li et al., 2020) used a GMM to model the loss distribution. GMMs have since been widely used in several methods. Semi-supervised approach (Ding, Wang, Fan, & Gong, 2018) based on a two-stage framework was used to identify samples that could possibly mislead the learning process. UNICON (Karim, Rizve, Rahnavard, Mian, & Shah, 2022), Gradient Switching Strategy (Yu et al., 2023), and SSS-Net (Cai, Zhang, Pedrycz, & Miao, 2023) are recent methods that focus on selecting clean samples. In general, sample selection methods work well. However, they suffer from accumulated errors caused by incorrect selection. With this notion, several studies have employed multiple DNNs (Han et al., 2018; Malach & Shalev-Shwartz, 2017; Tan, Xia, Wu, & Li, 2021), thereby reducing the confirmation bias.

Multi-network learning. Decoupling (Malach & Shalev-Shwartz, 2017) updated the model parameters using only instances on which the predictions of two networks are different. MentorNet (Jiang et al., 2018) employed curriculum-based learning to guide the student network. Co-teaching (Han et al., 2018) trained two DNNs simultaneously, allowing them to teach each other based on small-loss samples and Co-teaching++ (Yu et al., 2019) improved co-teaching by adapting the disagreement strategy. JoCoR (Wei et al., 2020) used co-regularization to bring the prediction of networks closer to each other, while JoSRC (Yao et al., 2021) used Jensen–Shannon divergence to estimate the likelihood of a sample 'being' clean. Recent advancements in various fields have spurred the emergence of innovative approaches that combine sample selection techniques with other novel concepts, resulting in improved performance compared to the existing methods. SELF (Nguyen et al., 2019) used gradual filtering of noisy labels during the training using running averages of predictions as ensembles. Similarly, SELC (Lu & Selc, 2022) worked on gradually correcting the noisy labels. Co-learning (Tan et al., 2021) instead used a shared encoder with two heads, one self-supervised and another supervised and maximized the agreement between them.

The DivideMix family. One of the most popular LNL multi-network benchmark algorithms in the family of sample selection is DivideMix (Li et al., 2020). DivideMix used two networks to train each other by employing GMMs to split the training data into clean and noisy samples. The GMMs are modelled based on the per-sample loss distribution. DivideMix used an improved MixMatch (Berthelot et al., 2019) to train the networks. Different stages in the DivideMix algorithm were further improved so as to mitigate the potential drawbacks in the training pipeline. The importance of the warm-up stage in LNL algorithms was studied in Contrast2Divide (Zheltonozhskii et al., 2022), which proposed using self-supervised pre-training to create better feature extractors. AugDesc (Nishi et al., 2021) explored DA techniques in LNL algorithms and proposed using two different augmentations — one for analysing the loss and the other for backpropagation. CCLM (Tatjer, Nagarajan, Marques, & Radeva, 2023) studied the per-sample loss modelling of DivideMix and proposed a class-conditional approach to split the clean and noisy samples. A dynamic class-conditional weighting focused on less-learned classes was used to improve the balance of losses in learning samples (Nagarajan, Marques, Mejia, & Radeva, 2022). Several methods adapt the training pipeline of DivideMix. Probabilistic Noise Perception (Sun et al., 2022) used different optimization objectives for the specific samples based on two networks — one to predict the category label and the other to predict the noise type. ProMix (Wang, Xiao, et al., 2022) combined the small-loss sample selection with a high confidence-based selection. SplitNet (Kim et al., 2022) used an additional complementary learning module to predict the clean-noisy label.

Sample selection algorithms focus on avoiding confirmation bias and try to use the entire noisy data to create robust LNL models. The main challenge faced by these algorithms revolves around DNN memorization. Given the importance of DivideMix in solving the LNL problem, we investigate the training pipeline of DivideMix with the notion of tackling this issue. Importantly, we delve into the role of warm-up and DAs, which are of prime importance during the training step. Our results highlight how the proposed pipeline addresses the issue of DNN memorization.

2.2. Uncertainty quantification

DNNs typically contain millions of trainable parameters, far more than the number of training samples. Despite their vast capacity, these models often encounter challenges in generalization (Zhang, Bengio, et al., 2021). Different uncertainty estimation techniques such as Deep Ensembles (Lakshminarayanan, Pritzel, & Blundell, 2017) and MC-Dropouts (Gal & Ghahramani, 2016) are employed to find the prediction confidence of the models. Bayesian probability theory provides tools to reason about model uncertainty (Gal & Ghahramani, 2016). Bayesian approaches offer a principled framework to capture uncertainty by introducing a distribution over model parameters and then sampling multiple sets of parameters to form a predictive distribution (Maddox, Izmailov, Garipov, Vetrov, & Wilson, 2019). Bayesian modelling allows capturing and modelling two distinct types of uncertainty: *aleatoric uncertainty*, which captures the noise in observations and *epistemic uncertainty*, which accounts for uncertainty in model parameters (Der Kiureghian & Ditlevsen, 2009). Epistemic uncertainty reflects the lack of knowledge of the model parameters and often arises due to limited data. Exact Bayesian inference is intractable in DNNs and, therefore, usually approximated; Bayesian approximation methods fall majorly into (1) Sampling-based, (2) Variational Inference, or (3) Hybrid approaches (Lange, Benjamin, Haefner, & Pitkow, 2022; Wang & Yeung, 2020). Abdar et al. (2021) presents an in-depth exploration of different uncertainty quantification techniques. By leveraging the confidence of model outputs, it is possible to handle the uncertain inputs and the special cases explicitly. Uncertainty measurements have been incorporated into various LNL algorithms previously. An iterative label noise filtering process using ensembles and MC-Dropouts (Köhler, Autenrieth, & Beluch, 2019) was introduced based on the assumption that the predictive uncertainties of clean and noisy data follow two different distributions. Uncertain Aware Co-Training (Ji et al., 2021) studied the relationship between uncertainty and clean labels using MC-Dropouts and used it as a criterion to choose the samples along with the small-loss trick. Entropy-based Debiasing framework (Oh, Lee, Byun, & Shin, 2022) used predictive uncertainty to identify minority group samples while preventing the models from learning the noisy labels. The reliability of uncertainty estimation methods in the presence of label noise was critically analysed (Pan, Yuan, Zhou, & Yao, 2022). Specifically, Goel and Chen (2021) performed an in-depth analysis of the effectiveness of using MC-Dropouts in the presence of label noise. Combats Noisy Labels by Concerning Uncertainty (CNLCU) algorithm (Xia

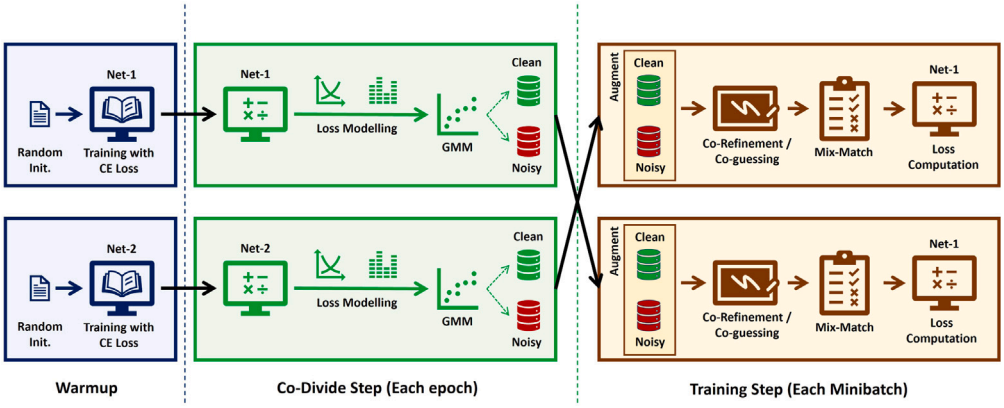


Fig. 1. DivideMix uses co-teaching of two models (Net-1 and Net-2). During each epoch, the loss distribution of each model is used to divide the dataset into clean and noisy data (Co-Divide), which is used to train the other model. During each mini-batch of the training step, the models are trained in a semi-supervised manner using MixMatch.

et al., 2021b) used the uncertainty of losses by adopting interval estimation instead of point estimation. CNLCU works on large loss samples, distinguishing between mislabelled or underrepresented samples. Uncertainty-aware Label Correction framework (Huang, Bai, et al., 2022) studied the impact of label noise on imbalanced datasets and used a combination of epistemic uncertainty and aleatoric uncertainty to boost the performance of the LNL algorithm.

Following the importance and the role of uncertainty in improving and interpreting DNNs, particularly in LNL problems, we analyse the epistemic uncertainty of the training data by converting the deterministic model into a Bayesian one. We identify the role of MixMatch as an important one in the training pipeline, where improving this step would provide a better robust learning pipeline. In this regard, we employ the measured uncertainty in order to create an improved version of MixMatch for data augmentation, which we call Monte-Carlo MixMatch. This step differs from the other techniques, which only use uncertainty computations to modify the loss function.

3. Background and rationale

First, we provide a concise overview of the baseline algorithm for LNL, DivideMix, which provides the background information for understanding the LNL algorithm pipeline. Later, we outline the key challenges we tackle using our proposed method.

3.1. DivideMix

Small-loss-based sample selection methods rely on the hypothesis that clean samples are faster to learn than noisy samples, leading to a lower loss for clean samples (Arpit et al., 2017). Formally, let $\tilde{D} = \{(x_i, \tilde{y}_i)\}_{i=1}^{N}$ denote a noisy training set of N samples, where x_i is the ith image and $\tilde{y}_i \in \{0,1\}^C$ is the one-hot label over C classes. The goal is to develop a robust learning algorithm capable of effectively handling and mitigating the impact of label noise, ultimately ensuring reliable and accurate predictions on unseen data. DivideMix algorithm (Li et al., 2020) trains two homogeneous models simultaneously, thereby avoiding the confirmation bias (Tarvainen & Valpola, 2017). Different random components, such as parameter initialization and batch sequences, make the models diverge from each other. We show an overview of the DivideMix algorithm in Fig. 1. Before employing the learning steps of the algorithm, a **warm-up** (sapphire-coloured (■) in Fig. 1) is performed for both models. During the warm-up phase, the models are trained for a few epochs using the standard cross-entropy loss. The warm-up allows the model to achieve initial convergence and enable better loss separations and feature representations.

Given a model with parameters θ, the cross-entropy loss $l(\theta)$ is defined as follows:

$$l(\theta) = \{l_i\}_{i=1}^{N} = \left\{ -\sum_{c=1}^{C} \tilde{y}_i{}^c log\left(p_{model}^c(x_i;\theta)\right) \right\}_{i=1}^{N} \quad (1)$$

where p_{model}^c is the model's output softmax probability for class c.

Once the model is capable of producing an informed loss, DivideMix exploits the above-mentioned hypothesis. To this end, a two-component GMM, g, is fit to the loss, $l(\theta)$, using the Expectation–Maximization algorithm. The resulting GMM is then used to produce a probability density function $p(\cdot)$, such that $p(x_i \mid l_i; g)$ yields the posterior probability of x_i being clean, given its loss l_i. It has been shown by Li et al. (2020) that using two different GMMs, each associated with a model, mitigates the confirmation bias. The GMM of one model is used to split the training data, and subsequently, the other model is trained on this data. This division strategy, named **Co-Divide** (jade-coloured (■) in Fig. 1), is determined by a threshold τ over the probability density function $p(\cdot)$, to split the training set into a labelled set ($\mathcal{X}$) and an unlabelled set ($\mathcal{U}$).

Next, DivideMix incorporates semi-supervised learning to improve the model's performance. During each training step (highlighted in saddle brown (■) in Fig. 1), an improved **MixMatch** (Berthelot et al., 2019) is applied on both the labelled set and the unlabelled set. MixMatch uses DAs on both labelled and unlabelled sets. A set of augmentations, M, is created for every sample in the batch. In the case of the labelled set, the model predictions are averaged across augmentations and refined using a temperature-sharpening step. For the unlabelled set, the model predictions are averaged across the augmentations to *guess* a pseudo-label for each sample.

Using a combined mini-batch of $\widehat{\mathcal{X}}$ and $\widehat{\mathcal{U}}$, the MixMatch algorithm uses MixUp (Zhang et al., 2017) to interpolate two random samples to produce a modified labelled set, $\mathcal{X}'$ and an unlabelled set, $\mathcal{U}'$. The sum of the cross-entropy loss for the modified labelled set ($\mathcal{X}'$), the mean-squared error for the unlabelled set ($\mathcal{U}'$), and a regularization term (as used by Arazo et al. (2019), Tanaka, Ikami, Yamasaki, and Aizawa (2018)) constitutes the total training loss. DivideMix improves MixMatch by: (i) Label co-refinement for the labelled set ($\widehat{\mathcal{X}}$): Combining ground-truth labels with the model's prediction and clean probability of the other model; (ii) Co-guessing for the unlabelled data ($\widehat{\mathcal{U}}$): Ensemble both models to predict unlabelled data.

3.2. The challenge of uncertainty

The interpretation of predictive probabilities as model confidence on classification tasks is often inaccurate (Gal & Ghahramani, 2016). Model confidence is crucial in capturing uncertain inputs and effectively handling those cases. Bayesian probability theory offers a solid

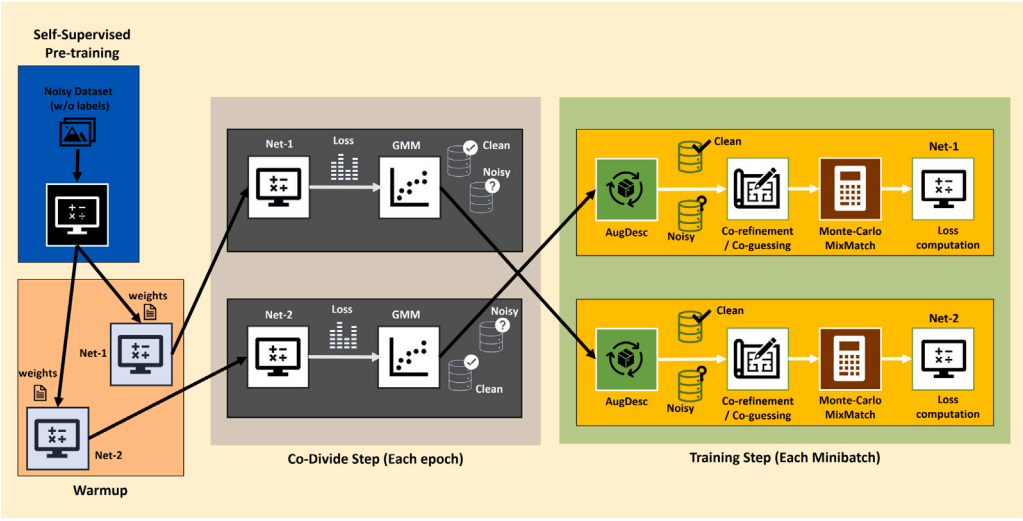


Fig. 2. Bayesian DivideMix++ uses co-teaching of two models (Net-1 and Net-2), initialized with **Self-supervised pre-trained weights** (■). During each epoch, the loss distribution of each model is used to divide the dataset into clean and noisy data (Co-Divide), which in turn is used to train the other. During each mini-batch of the training step, **AugDesc** (■) is used to augment the data splits followed by **Monte-Carlo MixMatch** (■) to train in a semi-supervised way.

mathematical framework to effectively reason about model uncertainty (Jospin, Laga, Boussaid, Buntine, & Bennamoun, 2022) and is robust to over-fitting. In the context of LNL, uncertainty estimation methods ensure a delayed memorization effect of noisy labels and contribute to improved generalization on the test set (Goel & Chen, 2021). By embracing the principles of Bayesian inference, we claim to achieve a more robust estimation of uncertainty, leading to enhanced model performance for LNL.

3.3. The challenge of DNN memorization

In a dataset with noisy labels, the DNNs learn to fit on the easy clean samples during the early learning stages. Later, the samples with noisy labels are memorized (Arpit et al., 2017). This phenomenon is well-studied (Zhang, Bengio, et al., 2021) and exploited to make the models more robust to label noise. Early learning and memorization have been fundamental to any high-dimensional learning task (Liu, Niles-Weed, Razavian, & Fernandez-Granda, 2020). With this regard, we dive into the role of *warm-up* and *data augmentations* in making the LNL algorithm more robust. The *warm-up* phase of DivideMix ensures two critical purposes: enhanced loss separability for the follow-up stages and better feature extraction. It is a well-analysed and documented procedure to initialize the training procedures using clean labelled datasets such as ImageNet (Li et al., 2020; Li, Wong, Zhao, & Kankanhalli, 2019; Patrini, Rozza, Krishna Menon, Nock, & Qu, 2017; Tanaka et al., 2018). However, the benefits of supervised pre-training are generally inconsistent on LNL algorithms, where the influence of factors such as noise level and domain gap plays a critical role in achieving high-quality features (Zheltonozhskii et al., 2022). The primary goal of any *data augmentation* technique is to enhance the generalization capability of the dataset. However, it is crucial to ensure that the use of DA does not negatively impact loss modelling and the convergence of the algorithms. LNL algorithms such as DivideMix typically employ simple augmentations such as random flips and crops (called *weak augmentations*). However, using the basic set of augmentations during the learning step is suboptimal (Nishi et al., 2021) as it limits the generalization abilities of the models. Incorporating additional *strong* augmentations such as AutoAugment (Cubuk, Zoph, Mane, Vasudevan, & Le, 2019) and RandAugment (Cubuk, Zoph, Shlens, & Le, 2020) is highly advantageous, especially in high-noise settings. The additional variations and transformations introduced by these augmentations contribute to a more diverse training set, effectively reducing overfitting and increasing the model's ability to generalize well.

4. Bayesian DivideMix++

In this section, we present our proposed **Bayesian DivideMix++** with a comprehensive explanation of each of the two novel components, uncertainty-aware DivideMix++ and Monte-Carlo MixMatch, and their role in enhancing the performance of LNL algorithms by tackling the above-listed challenges. We show the overview of our proposed method in Fig. 2.

4.1. Bayesian formulation of DivideMix

Training DNNs with dropouts can be interpreted as an approximate Bayesian inference of the weight's posterior (Gal & Ghahramani, 2016). Dropouts have been widely used in several applications to measure uncertainty and have been very effective. Integrating models with Monte-Carlo (MC) Dropout helps in reducing the decline in classification performance when data is more challenging or ambiguous (Goel & Chen, 2021).

In this work, we take advantage of the uncertainty information captured using MC-Dropouts to improve the MixMatch step of DivideMix. Let d_l denote a dropout at the lth layer, where $d_l \sim Bernoulli(p_b)$, with p_b being the dropout probability. Let us perform at inference time K forward passes of the LNL algorithm; hence, we obtain a distribution of K logits and predictions for each input data. Here, we use Shannon entropy (Lin, 1991) as a measure of uncertainty. For any sample x_i, the Shannon entropy $\mathcal{H}$ is calculated as:

$$\mathcal{H}(x_i) = -\sum_{c=1}^{C} p_c(x_i) log(p_c(x_i))$$

where $p_c(x_i)$ is the output softmax probability for class c.

Epistemic uncertainty affects the identification of clean samples from noisy samples (Huang, Bai, et al., 2022). The robustness property and ease of implementation of MC-Dropouts indicate that it is an effective and practical solution against noisy labels.

Monte-Carlo MixMatch. Here, we propose a new variant of MixMatch, called Monte-Carlo MixMatch (MC-MixMatch) (highlighted in saddle brown (■) in Fig. 2), based on the uncertainty of samples. MixMatch combines two samples randomly by interpolating them using a weighted combination. MixMatch uses a random parameter, λ, which is derived from a Beta distribution to combine the two samples. However, in our proposed component, MC-MixMatch, the combination is determined by the parameter λ_{unc}, which is derived from a Beta distribution

and weighted using the uncertainty of the samples $\mathcal{H}$. Given a pair of samples (x_1, x_2) and their corresponding labels $(\tilde{y_1}, \tilde{y_2})$, the mixed sample (x', y') is computed by:

$$\lambda \sim Beta(\alpha, \alpha) \tag{2}$$

$$\lambda' = max(\lambda, 1-\lambda) \tag{3}$$

$$\lambda_{unc} = 0.5 \times (\mathcal{H}(x_1) * \lambda' + 1)$$

$$x' = \lambda_{unc}\, x_1 + (1-\lambda_{unc})\, x_2$$

$$y' = \lambda_{unc}\, y_1 + (1-\lambda_{unc})\, y_2$$

where $\mathcal{H}(x_1)$ is the uncertainty of sample x_1. The λ_{unc} is computed, only using the uncertainty of the first sample $\mathcal{H}(x_1)$, with the notion that the second sample also undergoes a similar learning process during some part of the training step. The importance of the sample is determined using the uncertainty of the samples. Highly uncertain samples are weighted more in the interpolation, thereby giving more importance to those samples. Meanwhile, highly certain samples are weighted comparatively less, helping the model to generalize better. Note that by Eq. (3), the weight of sample x_1 is always higher than 0.5.

4.2. Diversity-increasing DivideMix++

In our proposed DivideMix++, we tackle the challenge of DNN memorization by leveraging the advantages of self-supervised pre-training and dedicated augmentation strategies within the training pipeline. First, we use self-supervised pre-training weights (shown in navy blue (■) in Fig. 2) to initialize the model rather than starting with random initializations or with supervised pre-training. By this step, we discard the presence of labels during the warm-up phase, which in turn leads to removing the influence of label noise. The pre-trained features are thus agnostic to noisy labels, which results in more effective and reliable feature representations (Zheltonozhskii et al., 2022).

The benefits of self-supervised pre-training are twofold. First, the model becomes more robust to the presence of noisy labels during subsequent supervised training, improving the model's ability to generalize and make accurate predictions. Second, the pre-training enhances the model's discriminative power by extracting more informative and representative features. Self-supervised pre-training thus provides a robust starting phase for subsequent training steps, laying a solid foundation for the model's feature representation.

The next improvement, we make is to modify each training batch by incorporating two separate sets of augmentations, one for loss analysis and another for gradient descent, instead of a single augmentation step (highlighted in birch green (■) in Fig. 2). This strategy allows us to leverage the benefits of diverse transformations and expand the augmentation space. By incorporating two different sets of augmentations, we introduce additional variations and increase the diversity within the training data. By expanding the augmentation set, we empower the model to learn from a richer set of training samples. However, applying augmentations during warm-up negatively impacts the loss convergence (Nishi et al., 2021). Hence, we perform this addition only to the training step and not during the warm-up phase. Self-supervised pre-training aids in better loss separation of clean and noisy samples and the modified augmentation strategy positively impacts the loss modelling during the training step.

4.3. Overall training

We train the proposed Bayesian DivideMix++ similar to the DivideMix algorithm, retaining the loss function:

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_u \mathcal{L}_{\mathcal{U}} + \lambda_r \mathcal{L}_{reg} \tag{4}$$

where $\mathcal{L}_{\mathcal{X}}$ is the cross-entropy loss over the labelled set, $\mathcal{L}_{\mathcal{U}}$ is the mean-squared error over the unlabelled set, $\mathcal{L}_{reg}$ is the regularization term, λ_u and λ_r are used to control the strength of the loss terms. Incorporating self-supervised pre-training during warm-up and AugDesc during training constitutes the DivideMix++, whereas MC-MixMatch is added to DivideMix++ to enhance the model generalization, thereby comprising the proposed Bayesian DivideMix++.

5. Experiments

In this section, we describe the experiment settings used to validate our approach. We evaluate the performance on two synthetic noise datasets - CIFAR-10 (Krizhevsky, Hinton, et al., 2009) and CIFAR-100 (Krizhevsky et al., 2009) at different noise rates, as well as on two real noise datasets — WebVision (Li, Wang, Li, Agustsson, & Van Gool, 2017) and Clothing1M (Xiao, Xia, Yang, Huang, & Wang, 2015). Comparisons with state-of-the-art models demonstrate the effectiveness of our approach.

5.1. Datasets

Both the **CIFAR datasets** consist of 50,000 training samples and 10,000 testing samples where each image is of size 32 × 32. We follow common practices in synthetic noise benchmarks and vary the amount of injected noise. We study two types of synthetic label noise - *Symmetric* and *Asymmetric*. Symmetric noise is introduced by replacing the labels of a percentage of samples with random labels (all possible classes) drawn from a uniform distribution. Asymmetric noise is closer to real-world label noise, where the labels are only replaced by similar classes. The noise is injected, following Patrini et al. (2017). The labels of CIFAR-10 are flipped as Truck→Automobile, Bird→Airplace, Deer→Horse, Dog↔Cat. For CIFAR-100, the classes are grouped into 20 super-classes of five (e.g. Aquatic Mammals contain Beaver, Dolphin, Otter, Seal and Whale), and the noise flips each class into the next circularly. Following the previous works (Li et al., 2020; Nishi et al., 2021; Zheltonozhskii et al., 2022), we test our method on 20%, 50%, 80% and 90% symmetric noise. We also test our approach on 40% asymmetric noise on the CIFAR-100 dataset.

WebVision 1.0 contains 2.4 million images from the 1000 ILSVRC12 ImageNet classes (Russakovsky et al., 2015). The dataset is collected from Flickr and Google images and is estimated to have a label noise of ≈20 Following previous work (Li et al., 2020), we use the first 50 classes of the Google image subset for comparisons. **Clothing1M** is a real-world apparel dataset consisting of 1 million images of 256 × 256 split into 14 categories. The cloth classes are based on online shopping website categories. The dataset is estimated to have a label noise of 38.5%. To ensure comparability, we maintain the dataset pre-processing the same as those in Li et al. (2020, 2019). We resize the images to 256 × 256, crop the middle 224 × 224, and perform normalization. We use the standard train/val/test split for all experiments. Both WebVision and Clothing1M include clean validation and evaluation sets.

5.2. Implementation details

For all experiments, we follow the setup similar to those in Li et al. (2020), Nishi et al. (2021) and Zheltonozhskii et al. (2022). For CIFAR datasets, we use 18-layer PreAct ResNet (He, Zhang, Ren, & Sun, 2016) as the backbone architecture. For self-supervised pre-training, we use SimCLR (Chen, Kornblith, Norouzi, & Hinton, 2020) trained weights. Following Zheltonozhskii et al. (2022), we keep a batch size of 64 and increase the number of epochs to 360 for all our experiments. We set the initial learning rate of 0.02 and reduced it by a factor of 10 after 150 epochs. Strong pre-trained weights reduce the warm-up duration, and we use a warm-up period of 5 epochs for both CIFAR datasets following Zheltonozhskii et al. (2022) (DivideMix uses 10 and 30 epochs for CIFAR-10 and CIFAR-100 respectively). The number of augmentations ($M = 2$) is fixed, as in other methods (Nagarajan et al., 2022; Nishi et al., 2021; Zheltonozhskii et al., 2022) for fair comparisons. The GMM

Table 1
Summary of **Experiment Settings.**

Parameter/Dataset	CIFAR	WebVision	Clothing1M
Architecture	18-layer PreAct ResNet	ResNet-50	ResNet-50
Batch Size	64	32	32
Epochs	360	100	80
Initial LR	0.02	0.01	0.002
LR Reduction Epochs	150	50	40
LR Reduction Factor	10	10	10
(Bayesian DivideMix++ Specific Parameters)			
Warm-up (Epochs)	5	1	1
GMM Threshold (τ)	0.5	0.5	0.5
Beta Parameter (α)	4	0.5	0.5
Sharpening Temperature (T)	0.5	0.5	0.5
Number of Augmentations (M)	2	2	2

Table 2
Unsupervised loss weights (λ_u **in Eq.** (4)) for CIFAR datasets.

Dataset	20%	50%	80%	90%
CIFAR-10	0	25	25	50
CIFAR-100	25	150	500	500

threshold (τ) is maintained as 0.5 for all noise ratios, except for the 90% noise ratio, where it is set as 0.6, following the original DivideMix implementations. The beta distribution parameter of MixMatch (α in Eq. (2)) is set to 4 and the sharpening temperature (T) is maintained at 0.5. We present a summary of experiment settings in Table 1.

The only parameter that has a considerable effect on the algorithm is the unsupervised loss weight (λ_u in Eq. (4)) and is fixed as in Table 2. For higher noise settings of CIFAR-100 (80% and 90%), we follow the values of Zheltonozhskii et al. (2022), whereas the rest are following Li et al. (2020). We use these values following the experiments of Zheltonozhskii et al. (2022), where higher λ_u benefited the training of models. For the Asymmetric 40% noise of CIFAR-100, we set λ_u to 150. We report the sensitivity of Bayesian DivideMix++ for hyperparameters in Section 6.4.3. We find the results to be as per the parameters reported by Zheltonozhskii et al. (2022), and therefore, we maintain the same for fair comparisons.

For WebVision and Clothing1M, we use ResNet-50 as the backbone architecture. For both WebVision and Clothing1M, we use the same hyperparameters: $\alpha = 0.5$, $\lambda_u = 0$, $M = 2$, $\tau = 0.5$, $T = 0.5$. We use a warm-up period of 1 epoch. For WebVision, we train the models for 100 epochs, with an initial learning rate of 0.01, reduced by a factor of 10 after 50 epochs. For Clothing1M, we train the models for 80 epochs, with an initial learning rate of 0.002, reduced by a factor of 10 after 40 epochs. For all experiments (CIFAR and real-noise), we use SGD with a momentum of 0.9 and weight decay of 5e-4. For CIFAR experiments, we train on an NVIDIA 2080 Ti GPU, whereas for the real noise dataset, we use an NVIDIA 3090 GPU.

Augmentation Policies: For all experiments, we follow Nishi et al. (2021) and use the AugDesc-WS augmentation policy. This augmentation policy corresponds to applying weak augmentations (Resize, Random crop, Flip) for the loss analysis and strong augmentations (including ImageNet policies and CIFAR policies of AutoAugment (Cubuk et al., 2019) along with the weak augmentations) for the optimization steps. For CIFAR and WebVision experiments, we use weak augmentations during the warm-up phase, whereas for Clothing1M experiments, we use strong augmentations during the warm-up phase. We do this change to adapt to the complex noise structure of Clothing1M following Nishi et al. (2021).

Measuring Uncertainty: Following the works of Boluki, Ardywibowo, Dadaneh, Zhou, and Qian (2020) and Rizve, Duarte, Rawat, and Shah (2021), we use $K = 10$ forward passes to calculate the uncertainty. For the CIFAR experiments, we use one dropout layer after the 4th ResNet block and one before the Fully Connected layer with a dropout ratio (p_b) of 0.1. We explain with experiments the decision of the placement of the dropout layer and dropout ratio in Section 6.4. We use max-scaling to normalize the uncertainty values obtained using Shannon entropy. As with DivideMix, the λ_{unc} is maintained $\in [0.5, 1]$. For high noise settings (80% and 90% symmetric noise of both CIFAR datasets), we use a settling period of 150 epochs before using uncertainty measurements in the training pipeline. For WebVision and Clothing1M experiments, we use a dropout ratio (p_b) of 0.2. We keep the settling period as 50 epochs for WebVision and 40 for Clothing1M experiments.

5.3. Results

We first provide an overview of the different state-of-the-art methods we compare against our proposed method. Later, we highlight the performance of our approach across various datasets. We discuss the results in detail and provide insights on the improvements.

5.3.1. Comparison methods

We compare the results with state-of-the-art LNL models: DivideMix (Li et al., 2020) proposes a semi-supervised learning framework based on per-sample loss distribution modelling; AugDesc (Nishi et al., 2021) presents DA strategies for LNL algorithms; Contrast2Divide (Zheltonozhskii et al., 2022) employs self-supervised pre-training to tackle the problem of warm-up; Sel-CL+ (Li et al., 2022) is a confidence-based selection method for supervised contrastive LNL; UNICON (Karim et al., 2022) uses Jensen–Shannon divergence-based sample selection replacing probabilistic modelling; ULC (Huang, Bai, et al., 2022) exploits uncertainty measurements in label correction for imbalanced LNL problems; LongReMix (Cordeiro et al., 2023) utilizes oversampling of smaller selected high-confidence clean sets to build a robust LNL algorithm; Lipschitz Regularization (Miao et al., 2023) uses Bayes' rule to detect clean samples and combines FixMatch and MixUp to improve the semi-supervised learning step. Note that all mentioned methods above use DivideMix as the baseline method. Also, note that all comparing methods use the same backbone for all the reported datasets as used by the proposed Bayesian DivideMix++.

5.3.2. Synthetic noise datasets

We show the results for CIFAR-10 experiments in Table 3 and for CIFAR-100 in Table 4 with different levels of **symmetric noise**. Following the comparison methods, we report the **best** test accuracy (%) over all epochs and the average of the last 10 epochs (**last**). We report the mean and standard deviation of the best and the last test accuracy computed over five runs. Note that we use the settling period described in Section 5.2 to obtain the results for high noise settings (80% and 90%). Bayesian DivideMix++ outperforms the state-of-the-art methods by a considerable margin across all noise ratios, which can be attributed to bringing together the self-supervised pre-training and augmentation strategies, thereby reducing the degree of memorization. Also, by including uncertainty measurements using MC-MixMatch in the learning step, the proposed Bayesian DivideMix++

Table 3
CIFAR-10 — Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods on different symmetric noise settings.

Method / Noise ratio		20%	50%	80%	90%
DivideMix (ICLR, 2020)	Best	96.1	94.6	93.2	76.0
Li et al. (2020)	Last	95.7	94.4	92.9	75.4
DM + AugDesc (CVPR, 2021)	Best	96.3	95.4	93.8	91.9
Nishi et al. (2021)	Last	96.2	95.1	93.6	91.8
DM + C2D (WACV, 2022)	Best	**96.43±0.07**	95.32±0.12	94.40±0.04	93.57±0.09
Zheltonozhskii et al. (2022)	Last	96.23±0.09	95.15±0.16	94.30±0.12	93.42±0.09
Sel-CL+ (CVPR, 2022)	Best	95.5	93.9	89.2	81.9
Li et al. (2022)	Last	–	–	–	–
UNICON (CVPR, 2022)	Best	96.0	95.6	93.9	90.8
Karim et al. (2022)	Last	–	–	–	–
ULC (AAAI, 2022)	Best	96.1	95.2	94.0	86.4
Huang, Bai, et al. (2022)	Last	95.9	94.7	93.2	85.8
LongReMix (PR, 2023)	Best	96.3±0.1	95.1±0.1	93.8±0.2	79.9±2.7
Cordeiro, Sachdeva, Belagiannis, Reid, and Carneiro (2023)	Last	96.0±0.1	94.8±0.1	93.8±0.2	79.1±3.1
Lipschitz Reg. (PR, 2023)	Best	96.2	95.2	93.4	85.0
Miao, Wu, Xu, Zuo, and Meng (2023)	Last	95.7	94.8	93.1	84.3
Bayesian DivideMix++	Best	96.39±0.06	**95.68±0.09**	**95.25±0.08**	**94.46±0.15**
	Last	96.13±0.07	95.40±0.11	94.97±0.02	94.20±0.12

Table 4
CIFAR-100 — Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods on different symmetric noise settings.

Method / Noise ratio		20%	50%	80%	90%
DivideMix (ICLR, 2020)	Best	77.3	74.6	60.2	31.5
Li et al. (2020)	Last	76.9	74.2	59.6	31.0
DM + AugDesc (CVPR, 2021)	Best	79.5	77.2	66.4	41.2
Nishi et al. (2021)	Last	79.2	77.0	66.1	40.9
DM + C2D (WACV, 2022)	Best	78.69±0.17	76.43±0.25	67.78±0.30	58.7±0.31
Zheltonozhskii et al. (2022)	Last	78.32±0.35	76.07±0.41	67.43±0.30	58.45±0.30
Sel-CL+ (CVPR, 2022)	Best	76.5	72.4	59.6	48.8
Li et al. (2022)	Last	–	–	–	–
UNICON (CVPR, 2022)	Best	78.9	77.6	63.9	44.8
Karim et al. (2022)	Last	–	–	–	–
ULC (AAAI, 2022)	Best	77.3	74.9	61.2	34.5
Huang, Bai, et al. (2022)	Last	77.1	74.3	60.8	34.1
LongReMix (PR, 2023)	Best	77.9±0.2	75.5±0.2	62.3±0.5	34.7±0.3
Cordeiro et al. (2023)	Last	77.5±0.2	74.9±0.2	61.7±0.5	30.7±5.9
Lipschitz Reg. (PR, 2023)	Best	78.8	75.3	62.6	37.5
Miao et al. (2023)	Last	77.4	74.3	61.0	34.7
Bayesian DivideMix++	Best	**80.02±0.03**	**78.31±0.14**	**70.01±0.23**	**61.15±0.34**
	Last	79.56±0.13	77.71±0.13	69.55±0.22	60.70±0.42

can perform better, especially in high-noise settings where the improvements are substantial. In the 90% noise setting, we improve by 0.27% in CIFAR-10 and 0.85% in CIFAR-100 experiments, respectively (details in Section 6.4). In particular, Bayesian DivideMix++ outperforms the state-of-the-art methods by large margins in CIFAR-100 experiments, especially in the higher noise settings, where the improvements are more significant (2.23% gain in 80% and 2.45% gain in 90% symmetric noise settings over the best comparison methods). This improvement can be attributed not only to better warm-up and augmentation but also to the higher levels of uncertainty in samples, which is directly related to the noisy labels. Bayesian DivideMix++ has on-par performances with DM + C2D in CIFAR-10 20% noise. The reason could be that the models show higher performances already, making it difficult to surpass using Bayesian DivideMix++. We present the results of the **asymmetric noise** setting on CIFAR-100 in Table 5. We report only on 40% because over 50% makes the classes indistinguishable (Li et al., 2020). Similar to the symmetric cases, the proposed method outperforms the comparison methods by a substantial margin, proving the effectiveness of reducing the effect of memorization and reducing uncertainty during the training process.

Table 5
CIFAR-100 — Asymmetric 40% Noise Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods.

Method		Test accuracy
DM + C2D (WACV, 2022)	Best	75.48±0.16
Zheltonozhskii et al. (2022)	Last	75.06±0.16
Sel-CL+ (CVPR, 2022)	Best	74.20
Li et al. (2022)	Last	–
UNICON (CVPR, 2022)	Best	74.80
Karim et al. (2022)	Last	–
LongReMix (PR, 2023)	Best	59.80±0.1
Cordeiro et al. (2023)	Last	54.90±0.4
Bayesian DivideMix++	Best	**76.52±0.12**
	Last	76.06±0.13

5.3.3. Real noise datasets

We show the results of Real Noise datasets in Table 6 for WebVision and in Table 7 for Clothing1M. Following the literature, we report the top-1 and top-5 accuracy of both WebVision and ILSVRC12 validation sets for WebVision experiments and test accuracy for the Clothing1M experiments. As in CIFAR experiments, we report the mean and standard deviation computed over five runs.

Performance on WebVision: As can be seen in Table 6, we observe a Top-1 accuracy gain of 0.70% compared to the best comparison method (DM + C2D) in the WebVision validation set. However, there is no significant improvement in Top-5 accuracy, which indicates that the model is able to focus on making precise and accurate predictions. However, a lower Top-5 accuracy shows that the model lacks a higher range of possible predictions. Similarly, in the case of the ImageNet validation set, our proposed method is on par or better than the

Table 6
WebVision — Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods. Results are reported for both WebVision and ILSVRC12 (ImageNet) validation sets. († - *results reported in DM + C2D (Zheltonozhskii et al., 2022). DivideMix results are reported only on InceptionResNet-V2*).

Validation Set / Method	WebVision		ILSVRC12	
	Top-1	Top-5	Top-1	Top-5
DivideMix (ICLR, 2020) † Li et al. (2020)	76.32±0.36	90.65±0.16	74.42±0.29	91.21±0.12
DM + C2D (WACV, 2022) Zheltonozhskii et al. (2022)	79.42±0.34	92.32±0.33	**78.57±0.37**	93.04±0.10
UNICON (CVPR, 2022) Karim et al. (2022)	77.60	**93.44**	75.29	**93.72**
Bayesian DivideMix++	**80.12±0.28**	92.40±0.30	78.51±0.28	92.67±0.42

Table 7
Clothing1M — Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods.

Method	Test accuracy
DivideMix (ICLR, 2020) Li et al. (2020)	74.38[a]
DM + AugDesc (CVPR, 2021) Nishi et al. (2021)	74.57[a]
DM + C2D (WACV, 2022) Zheltonozhskii et al. (2022)	73.96[a]
ULC (AAAI, 2022) Huang, Bai, et al. (2022)	**74.9±0.2**
LongReMix (PR, 2023) Cordeiro et al. (2023)	74.38
Lipschitz Reg. (PR, 2023) Miao et al. (2023)	74.87
DivideMix++ (Deterministic)	74.09
DivideMix++ (with Dropouts)	74.13
Bayesian DivideMix++	74.81±0.09

[a] Denotes results acquired by us using published source code.

performance of other models on Top-1 accuracy. However, it does not obtain higher Top-5 accuracy.

Performance on Clothing1M: We reproduce the experiments of Li et al. (2020), Nishi et al. (2021), Zheltonozhskii et al. (2022) and observe an interesting phenomenon. DivideMix++ uses the dataset itself for self-supervised pre-training. However, the results of deterministic DivideMix++ and DivideMix++ with MC-Dropouts indicate that the warm-up gain obtained in other experiments is not present in this case. These results also follow the individual experiments of using only self-supervised pre-training and only augmentation strategies, where the experiment with respect to augmentation was initialized with ImageNet pre-trained weights. This performance could be attributed to the complicated noise structure in Clothing1M as observed by Zheltonozhskii et al. (2022). We, therefore, initialize the subsequent addition of the Bayesian DivideMix++ experiment using ImageNet weights. The proposed method has a test accuracy gain of 0.43% and 0.24% compared with DivideMix (Li et al., 2020) and AugDesc (Nishi et al., 2021) respectively, which highlights the impact of the MC-MixMatch addition to the pipeline. Note that the reproduced state-of-the-art results are a little lower than those reported in the papers, and it could be the difference caused by random components in the training pipeline.

6. Analysis

In this section, we analyse the various experiments to evaluate the effectiveness of our proposed technique in addressing the above-described challenges. We use the model with the best performance of the five runs to do these analyses.

Table 8
Memorization statistics (average and standard deviation) in CIFAR-100 90% symmetric noise.

Method	Avg.	Std.
DivideMix++ (Deterministic)	0.0149	0.0019
Bayesian DivideMix++	**0.0142**	**0.0017**

6.1. Robustness towards memorization

We examine the effectiveness of our proposed approach in addressing the challenge of memorization. DNNs, in general, begin to learn the easy samples first (low-loss samples). Similarly, LNL algorithms learn the clean samples first. Different algorithms use different techniques to make the models better learn the clean samples and also not memorize the noisy sample labels. This is an important factor to consider, as it determines the generalization and robustness of the models. We assess the learning behaviour of the models by individually analysing the performance in learning clean and noisy samples. We compare the performance of Bayesian DivideMix++ with that of DivideMix, self-supervised pre-training (DM + C2D) and DivideMix++ (Deterministic). We illustrate this comparison using CIFAR-100 with 90% symmetric noise setting and present the results in Fig. 3. The top row corresponds to the learning of clean samples while the bottom row represents the learning of noisy samples. DivideMix has an initial poor separation of loss terms, requiring more training steps to effectively separate the clean and noisy samples compared to the other methods. The fraction of correctly identified noisy samples, which is considerably low (the first column of Fig. 3), reinforces this behaviour. The introduction of self-supervised pre-training (DM + C2D) leads to better initial loss separation (second column of Fig. 3). With better augmentation strategies provided in DivideMix++ and Bayesian DivideMix++, the learning of clean samples becomes more accurate, as indicated by a higher fraction of correctly identified samples. Concerning learning noisy samples (second row of Fig. 3), DivideMix exhibits a higher level of memorization, where it tends to memorize the labels of the noisy samples. The introduction of self-supervised pre-training (DM + C2D) reduces the degree of memorization. Furthermore, by changing the augmentation scheme (DivideMix++) and by adding MC-MixMatch (Bayesian DivideMix++), we can learn a higher fraction of samples that were initially noisy. This can be seen from Table 8, where with the proposed scheme, the fraction of memorized samples has reduced. Similar to the behaviour of learning the clean samples, the proposed method works better in correctly identifying the noisy samples. The method also shows a reduced impact of memorization, highlighting the efficacy of our proposed scheme.

The results presented above detail the need for models robust against memorization. With the obtained results, we show the benefit of using DivideMix++ during the training of the LNL model and highlight the improvements towards making the models robust towards label noise.

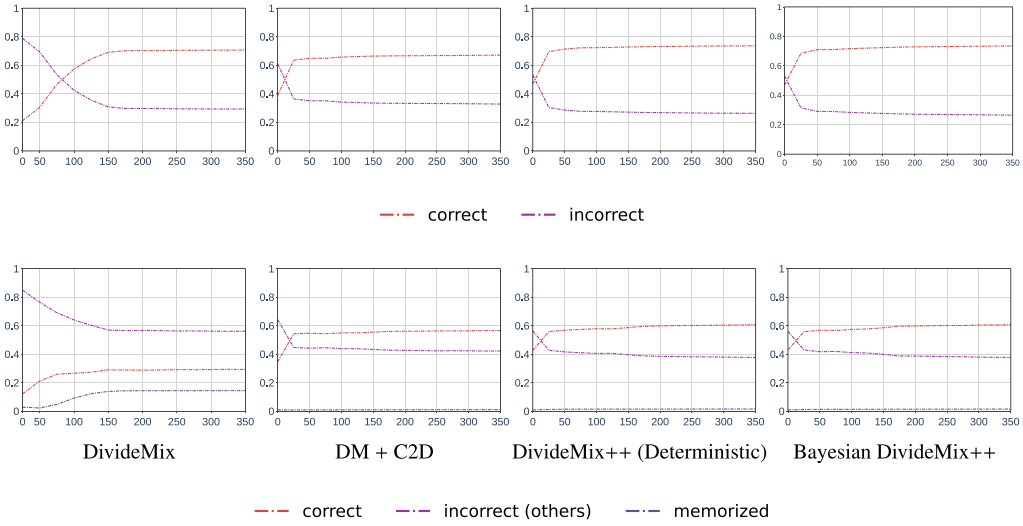


Fig. 3. Robustness towards Memorization. The plots are generated for CIFAR-100 Sym. 90% noise setting. The top row shows the fraction of clean samples that are predicted correctly and incorrectly. The second row shows the fraction of noisy samples that are predicted correctly, memorized and predicted incorrectly. The x-axis corresponds to the epochs and the y-axis corresponds to the fraction of samples.

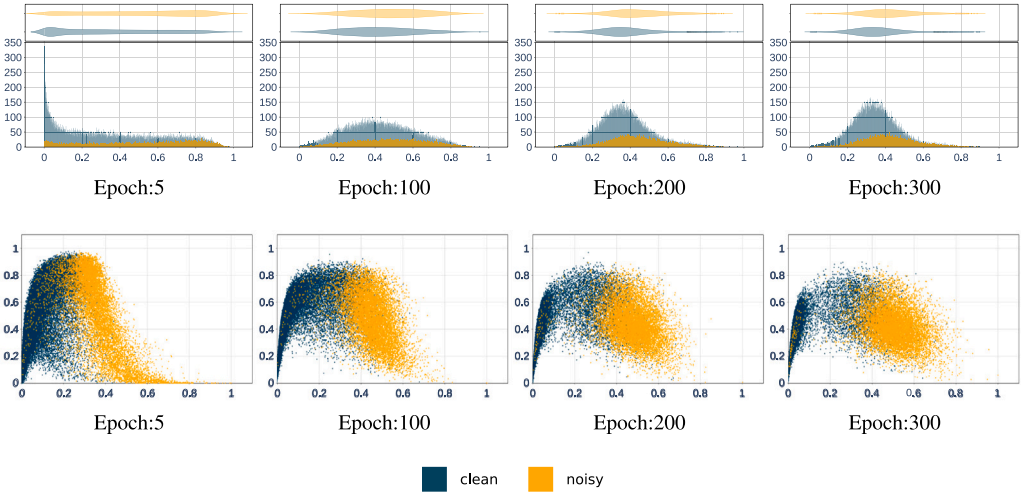


Fig. 4. Uncertainty of clean and noisy samples for CIFAR-100 20% sym. noise. The first row corresponds to the histogram of uncertainty values of clean and noisy samples. The second row corresponds to the uncertainty against the loss values.

6.2. Impact of uncertainty

We next analyse the impact of uncertainty on the training pipeline. We show the histogram of uncertainty values and uncertainty-loss plots over different epochs for CIFAR-100 20% symmetric noise in Fig. 4 and 90% symmetric noise in Fig. 5. In the early training stages (Epoch 5) of 20% symmetric noise setting (Fig. 4), a high number of clean samples have low uncertainty values. However, there is a considerable number of noisy samples that are highly uncertain. This can be visualized with the shape of the violin in the violin plots for Epoch 5. As the training progresses, both clean and noisy samples become less uncertain. Using the uncertainty-loss plots (bottom row of Fig. 4), we can see that in Epoch 5, uncertainty is high for a large number of clean and noisy samples. However, as the training progresses, the uncertainty of both clean and noisy samples is considerably reduced. In the 90% symmetric noise setting (Fig. 5), during the start of the training step (right after warm-up), we observe that most of the clean and noisy samples have high uncertainty. This can be confirmed by the shape of the violin plots. With regards to the relationship between sample loss and sample uncertainty (bottom row of Fig. 5), we can observe that, at epoch 5, uncertainty is generally high for all samples (even for low-loss samples). This can be attributed to the high noise ratio, where more wrongly labelled samples could potentially confuse the model. As the training progresses, the uncertainty of both clean and noisy samples decreases. After epoch 100, the uncertainty of low-loss samples, which are mostly clean, is low, and these are well separated from the noisy samples, highlighting the learning of the models. Although the uncertainty of noisy samples is high compared to that of the clean samples, the overall uncertainty of the models decreases. This can be seen with the shift and concentration of the violin of noisy samples.

Table 9
Uncertainty statistics (maximum and standard deviation) in CIFAR-100 20% and 90% symmetric noise. We measure uncertainty in DivideMix++ by adding MC-Dropouts.

Noise Ratio /	20%		90%	
Method	Max.	Std.	Max.	Std.
DivideMix++	0.9866	0.1448	0.9723	0.1737
Bayesian DivideMix++	**0.9176**	**0.1312**	**0.8930**	**0.1552**

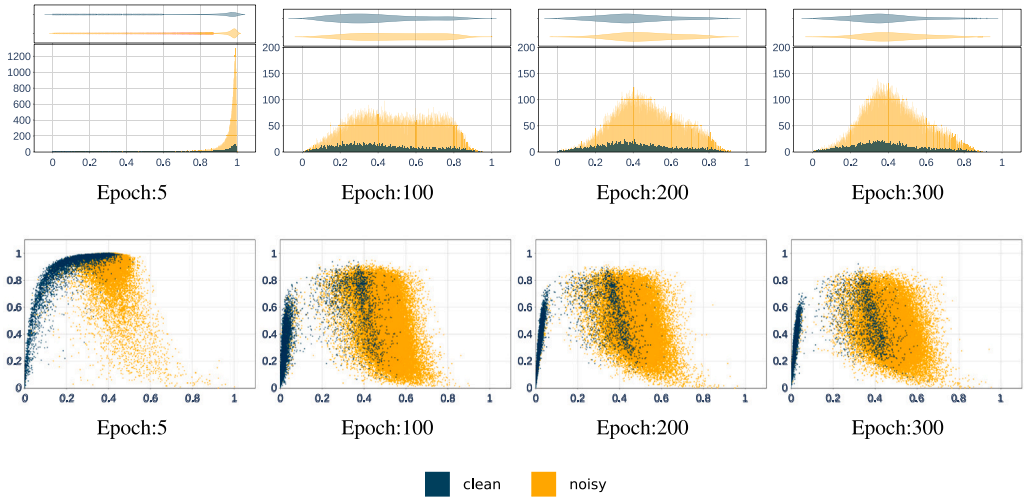


Fig. 5. Uncertainty of clean and noisy samples for CIFAR-100 90% sym. noise. The first row corresponds to the histogram of uncertainty values of clean and noisy samples. Note the change in the *y*-axis for plots other than Epoch:5. The second row corresponds to the uncertainty against the loss values.

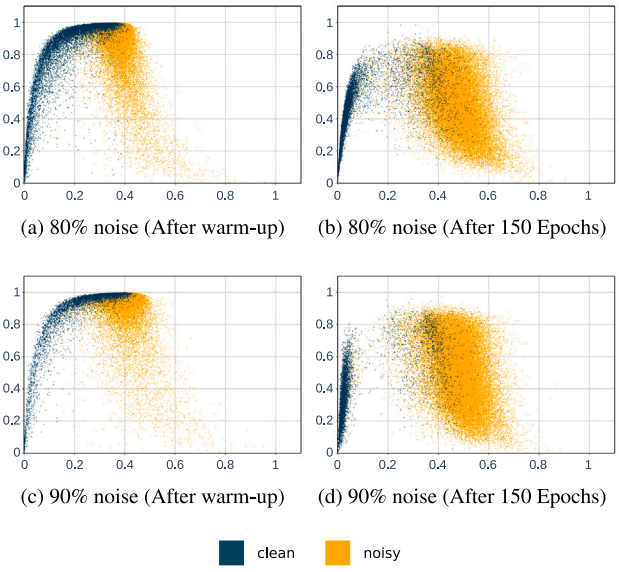


Fig. 6. Usage of MC-MixMatch. Uncertainty vs. loss plots after warm-up and after 150 epochs in high noise settings.

Table 10
Usage of MC-MixMatch. Difference in test accuracy (%) on CIFAR-100 (80% and 90% symmetric noise) using MC-MixMatch right after warm-up vs. after 150 epochs.

Dataset		CIFAR-10		CIFAR-100	
Method / Noise Ratio		80%	90%	80%	90%
After warm-up	Best	95.16	94.49	69.79	60.95
	Last	94.90	94.15	69.17	60.58
After 150 Epochs	Best	**95.35**	**94.64**	**70.24**	**61.56**
	Last	94.94	94.32	69.75	61.19

We report statistical measurements of the computed uncertainty values in Table 9. We highlight the maximum uncertainty values and standard deviation of uncertainty values of the training data. For both noise settings (20% and 90%), we see a decrease in the maximum value for Bayesian DivideMix++, highlighting a reduction in the uncertainty of the training samples. The lower standard deviation of the experiments highlights a reduction in the overall spread of uncertainty values.

Uncertainty in high noise settings. We observe that in high-noise settings, the uncertainty of both clean and noisy samples is very high. This is due to the high level of noise making it difficult for the models to learn from the clean samples. With this understanding, we study the impact of using uncertainty during the entire training steps against the impact when using uncertainty after allowing the models to learn further epochs without using MC-MixMatch (a settling period). We report the findings of this study on both CIFAR datasets in Table 10. In all cases (80% and 90% noise settings), the performance of models was higher when using MC-MixMatch after allowing the models to stabilize. We show the uncertainty-loss plots for this experiment in Fig. 6 and observe that using the uncertainty values after models learn better from clean samples is more beneficial than using it during the entirety of the training (better separation of clean and noisy samples in both noise settings as shown in Fig. 6(b) and Fig. 6(d)).

Making models aware of the label uncertainty makes them better equipped to handle the label noise. This can be seen through the different visualizations across different noise settings. MC-MixMatch helped the models to learn from samples that are more certain, thereby increasing the overall model confidence.

6.3. Loss separation

We also study the effectiveness of losses in separating the clean and the noisy samples. We compare the loss separation of DivideMix with our proposed method in Fig. 7 using CIFAR-100 90% symmetric noise. During the initial stages of training, both methods exhibit a limited ability to separate clean and noisy samples (Epoch 30 and Epoch 5 - first column of Fig. 7 are the first epochs after the warm-up phase in DivideMix and our proposed method respectively). Specifically, Epoch 5 of Bayesian DivideMix++ has a very high number of noisy samples which have large losses. As the training progresses, our proposed method exhibits a better separation of samples compared to that of DivideMix (bottom row of Fig. 7). The violin plots above the loss histograms depict the distribution of the loss values. A higher density of samples outside the violin indicates the inability of DivideMix to completely learn from the clean samples. In contrast, our proposed method shows a clearer distinction of samples, as evidenced by the majority of samples being well-separated. The presence of a smaller mode of clean samples overlapping with higher loss values might be due to the higher noise ratio. This highlights the improved capability of our proposed method to effectively separate clean and noisy samples, leading to enhanced performance in learning with noisy labels.

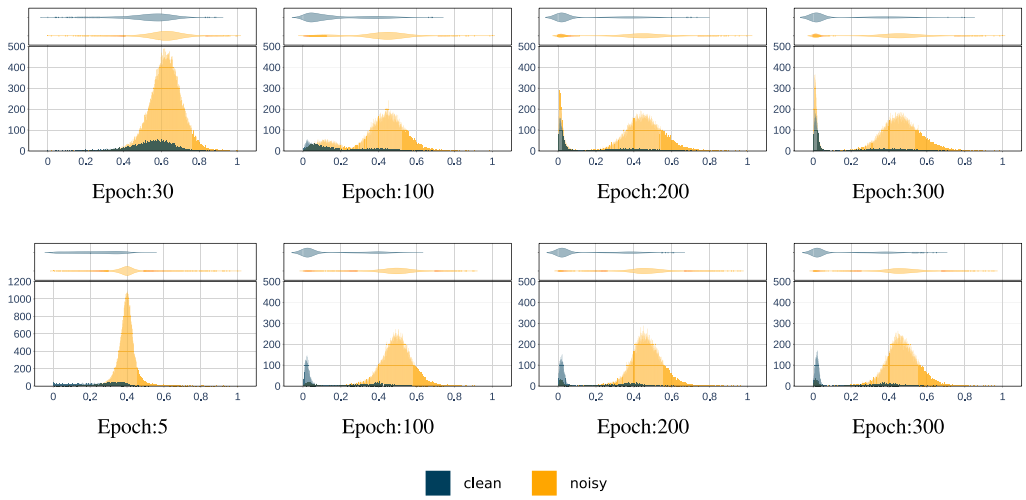


Fig. 7. Better loss separation. Progression of loss with respect to clean and noisy samples. As the training progresses, the loss is better separated. The first row corresponds to DivideMix, while the second row corresponds to Bayesian DivideMix++. DivideMix has a warm-up of 30 epochs (Hence, the first plot computed at 30 and not at 5). Also, note the change in the y-axis for Epoch:5 of Bayesian DivideMix++. All plots are generated for CIFAR-100 90% sym. noise.

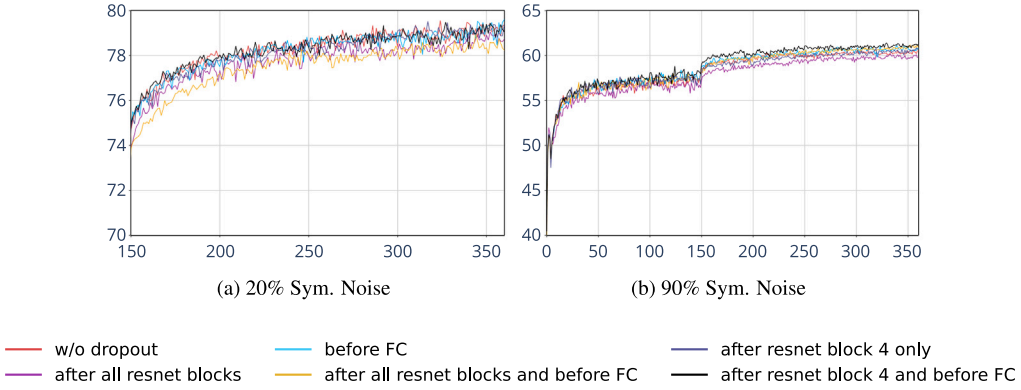


Fig. 8. Dropout position. Different configurations to select a better dropout position.

6.4. Ablation study

In this section, we provide an overview of the design decisions in our proposed method. Subsequently, we present a comprehensive analysis of the experiments conducted to evaluate the significance of each component in our proposed method. For all experiments, we use CIFAR-100 with 20% and 90% symmetric noise settings and report the test accuracy.

6.4.1. Exploring the position of MC-dropouts

We investigate the influence of using dropout layers at different stages within the model architecture and report the performances in Table 11. For this experiment, we fixed the dropout ratio as 0.1. We placed the dropout layers at different places of the ResNet block (a ResNet-18 model has 4 ResNet blocks). Using the dropout layer after the 4th block fared better than using it after all blocks. This can be due to losing more generic features during the training. We also check the behaviour by applying dropout at the later stages of the model. The optimal selection is the combination of using it after the 4th block and before the Fully Connected (FC) layer, which could be due to not losing much generic information and better regularization during learning specific features. The difference in performance is substantial in the higher noise cases, whereas the difference is comparative in the lower noise cases. We show the test accuracy of different settings against the training epochs in Fig. 8. The difference between different dropout positions is very evident in the 90% symmetric noise settings (Fig. 8(b)) as compared to the 20% symmetric noise settings (Fig. 8(a)) highlighting the important role of dropouts in higher noise settings.

Table 11
Position of MC-Dropout. Difference in test accuracy (%) on CIFAR-100 (20% and 90% symmetric noise) with different combination of dropout layers. The dropout ratio was set to 0.1 in all cases.

Position of MC-Dropout		20%	90%
After all blocks	Best	79.05	60.15
	Last	78.77	59.86
Only after block 4	Best	**79.51**	60.85
	Last	79.13	60.61
Only before FC	Best	79.48	61.09
	Last	78.89	60.62
After all blocks & before FC	Best	79.00	61.18
	Last	78.44	60.99
After block 4 & before FC	Best	79.48	**61.39**
	Last	79.16	61.13

6.4.2. Exploring the dropout ratio

We study the effect of using different dropout ratios (p_b) in Table 12. We fixed the position of the dropout layers for this experiment. For this experiment, we use one dropout layer after the 4th ResNet block and one before the Fully Connected layer of the ResNet-18 model (Fig. 9). We vary the dropout ratio between 0.1 and 0.4 on both low-noise and high-noise settings. As with the experiments dealing with the position

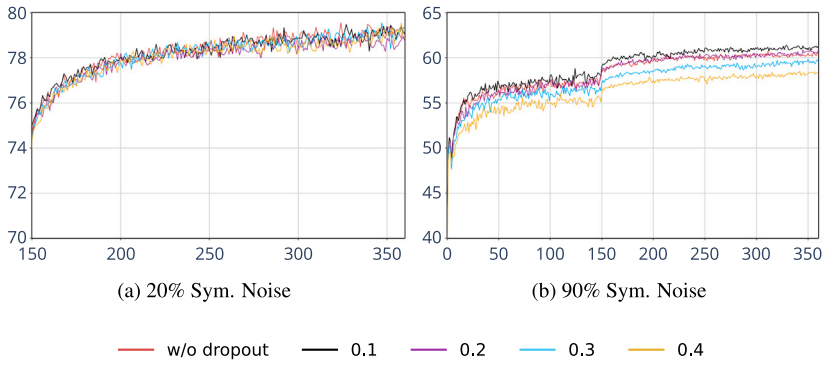


Fig. 9. Dropout ratio. Different configurations to select a better dropout ratio.

Table 12
Dropout Ratio. Difference in test accuracy (%) on CIFAR-100 (20% and 90% symmetric noise) when using different dropout ratios. Dropouts are placed after block 4 and before FC of the ResNet-18 model.

Dropout ratio		20%	90%
0.1	Best	79.48	**61.39**
	Last	79.16	61.13
0.2	Best	79.18	60.90
	Last	78.71	60.65
0.3	Best	**79.54**	59.82
	Last	79.02	59.54
0.4	Best	79.53	58.48
	Last	79.08	58.31

Table 13
Study of optimal λ_u on CIFAR-100 symmetric 20% noise.

λ_u	0	25	50	100	250
Best	**80.06**	80.01	79.81	79.02	75.99
Last	79.30	79.44	79.30	78.49	75.46

Table 14
Study of optimal λ_u on CIFAR-100 symmetric 90% noise.

λ_u	0	10	50	100	250	500	750	1000
Best	55.54	55.64	57.01	58.78	60.11	**61.56**	59.83	59.42
Last	54.71	55.18	56.47	58.36	59.70	61.19	59.23	59.04

Table 15
Ablations. Difference in test accuracy (%) on CIFAR-100 (20% and 90% symmetric noise) with different components.

Method		20%	90%
DivideMix	Best	77.30	31.50
	Last	76.90	31.00
DM + C2D	Best	78.69±0.17	58.70±0.31
	Last	78.32±0.35	58.45±0.30
DivideMix++ (Deterministic)	Best	79.55	60.71
	Last	79.19	60.36
DivideMix++ (with) MC-Dropouts	Best	79.48	61.39
	Last	79.16	61.13
Bayesian DivideMix++ (After warm-up)	Best	**80.01**	60.95
	Last	79.44	60.58
Bayesian DivideMix++ (After 150 Epochs)	Best	79.51	**61.56**
	Last	79.13	61.19

of dropout layers, the effect of using different dropout ratios is more pronounced in high-noise settings. We select the dropout ratio based on the high-noise settings, where we achieve higher performances with a smaller dropout of 0.1. Higher dropout ratios have a negative impact on the model performance, which can be due to more neurons randomly getting dropped during training, losing more meaningful information. Similar to the position of dropouts, it can be seen that the difference between different dropout ratios is more prominent in the higher noise settings (Fig. 9(b) as compared with Fig. 9(a)).

6.4.3. Hyperparameter sensitivity

Based on the previous works that were inspired by DivideMix algorithm (Li et al., 2020; Nishi et al., 2021; Zheltonozhskii et al., 2022), we identify the unsupervised loss weight (λ_u in Eq. (4)) as the most important hyperparameter. We follow the studies of Nagarajan et al. (2022), Zheltonozhskii et al. (2022) and investigate the performance of Bayesian DivideMix++ by varying λ_u on CIFAR-100 symmetric 20% and 90% noise settings. The sensitivity study is reported in Tables 13 and 14. For low noise settings, we find that lower λ_u values are better, whereas for higher noise settings, we require a higher λ_u value. Although a λ_u of 0 performs slightly better than the value of 25, the last accuracy is better in the later experiment, showing better overall training. We, therefore, set the λ_u to be 25, which also follows the results obtained by Zheltonozhskii et al. (2022). All the results thus provided in this work are obtained using the hyperparameter settings of Zheltonozhskii et al. (2022).

6.4.4. Evaluation of different components

We perform detailed ablations to show the importance of each component and report the results in Table 15. We highlight the insights of each component as follows:

- DivideMix++ (Deterministic) exhibits substantial performance gains over the DivideMix algorithm (*Row 1*) and the one by using Self-supervised pre-training (*Row 2*). While the impact of self-supervised pre-training is noticeable in the 90% noise setting, it falls short of achieving complete robustness against memorization. However, combining effective augmentation strategies, DivideMix++ (*Row 3*) achieves greater resilience to memorization effects, resulting in higher performance gains.
- By converting DivideMix++ to a Bayesian model using MC-Dropouts, we achieve comparable results to its deterministic counterparts (*Row 4*). The conversion of DivideMix++ to a Bayesian model allows us to harness the benefits of uncertainty estimation without compromising the overall performance achieved by the deterministic counterparts.
- MC-MixMatch enhances the performance of DivideMix++ by leveraging uncertainty measures to assign importance to individual samples (*Row 5 and 6*). By considering the uncertainty estimates, MC-MixMatch effectively prioritizes samples during training, leading to improved performance. In higher noise settings, where the initial uncertainty is not figurative of the actual data, a waiting period is needed to provide good uncertainty measurements.
- The combination of DivideMix++ and MC-MixMatch (Bayesian DivideMix++) has yielded significant benefits for the DivideMix algorithm, highlighting the crucial role of addressing both DNN memorization effects and uncertainty during the learning process.

Table 16
Computation time (seconds) per-epoch for each operation of Bayesian DivideMix++ and DivideMix.

Method	Model initialization	Warmup	AugDesc	MixMatch
DivideMix	11.51 s	74.05 s	1.15 s	8.25 s
Bayesian DivideMix++	11.45 s	74.87 s	0.66 s	26.27 s

6.4.5. Training time analysis

We analyse the training time of our proposed method, Bayesian DivideMix++ and compare it against the baseline DivideMix algorithm (Li et al., 2020). Bayesian DivideMix++ uses the same training pipeline as that of DivideMix (Li et al., 2020) except for the proposed contributions. We show the breakdown of computation time for each operation in Table 16. We integrate self-supervised pre-trained weights and change the augmentation pipeline, which does not contribute to a difference in the training time. However, Monte-Carlo MixMatch is the only component that affects the training time. The computation time of Monte-Carlo MixMatch depends on the number of iterations used to compute uncertainty. In this case, we calculate the uncertainty over 10 iterations, which leads to the increase in training time (8.25 s to 26.27 s). This increase is a tradeoff between having more meaningful data and higher computation costs associated with computing uncertainty. The focus of our paper is to show the importance of handling label uncertainty in LNL algorithms. More efficient alternate methods for measuring uncertainty can alleviate the computation time. However, this is beyond the focus of our paper. Additionally, it has to be noted that there is no difference in the inference time of Bayesian DivideMix++ as the proposed components have no noticeable impact on it.

6.5. Limitations

The combination of DivideMix++ and MC-MixMatch in our proposed Bayesian DivideMix++ has produced promising results. However, we identified some limitations in our proposal that can serve as a potential research direction.

- **Computational Complexity:** DivideMix++ follows a complex modelling technique by incorporating components such as the co-teaching of two DNNs, generating pseudo-labels, selecting clean samples and MixMatch of labelled and unlabelled data. This addition of components makes the training more expensive, particularly when dealing with large-scale datasets. Uncertainty estimation adds to the computational complexity of the algorithm, as it needs multiple inference steps to compute uncertainty.
- **Hyperparameter Selection:** Although this paper and several others before us have attempted to experiment with different hyperparameters of the DivideMix pipeline, training still heavily relies on multiple hyperparameters, making the selection of appropriate values a challenging task.
- **Dropout Rate and Position:** MC-Dropout is a popular technique for estimating uncertainty. However, the choice of dropout rate and the placement of dropout positions is critical when using MC-Dropouts. It requires careful tuning, especially considering the noise settings.
- **Noise Level Estimation:** The training pipeline highly relies on estimating the noise level in the dataset. Estimating noise levels accurately in real-world scenarios is very challenging. Inaccurate estimation can lead to improper sample selection, thereby affecting the algorithm performance.

6.6. Broader impact

LNL has been studied extensively over the years due to the challenges associated with annotating training data accurately. The difficulties in obtaining perfect training data have highlighted the need for robust LNL algorithms. With label noise, the DNNs could end up learning incorrect patterns and subsequently make poor predictions on unseen data. An efficient LNL algorithm is crucial to effectively handle and mitigate the effect of noisy labels during the training phase. LNL algorithms increase the robustness and generalization ability of the models in practical settings, resulting in more accurate predictions. Overall, LNL enables leveraging huge, readily available, but imperfectly labelled data in training models that can excel in real-world applications, thereby improving the performance and reliability of models.

7. Conclusions and future works

Learning with noisy labels has been a longstanding and actively researched problem. The need for robust algorithms is of prime importance as noisy labels pose significant challenges in training accurate and reliable models. Two critical challenges in developing robust LNL algorithms are the DNN memorization of noisy labels and label uncertainty. Both are of prime importance to enhance the generalization of models in the presence of label noise. With this regard, we proposed Bayesian DivideMix++ addressing the two issues: DivideMix++ provided a better warm-up phase and improved the DA pipeline. MC-MixMatch, which incorporated MC-Dropouts, was used to reduce the impact of uncertain samples in the training step. The combination of both components gave substantial performance improvements and provided benefits in addressing the DNN memorization and uncertainty effects. We validated our proposed pipeline on popular benchmarks of different noise settings and achieved state-of-the-art performance.

Future works. We measured uncertainty during the MixMatch step of Bayesian DivideMix++ and used it as a weight to combine samples. Currently, the co-divide step is based on the inference of previous epochs, and uncertainty is present in the model predictions. This uncertainty could be measured and used in the subsequent steps. Our future work could thus focus on investigating the impact of uncertainty during the loss-separation step. Another future work is exploring the benefits of using class-conditional learning as an alternative to the current global noise learning strategy.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was partially funded by the Horizon EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), DeepFoodVol (AEI-MICINN, PDC2022-133642-I00), IDEATE (AEI-MICINN, PID2022-141566NB-I00), A-BMC (AEI-MICINN, CNS2022-135480), CERCA Programme/Generalitat de Catalunya, and Agencia Nacional de Investigación y Desarrollo de Chile (ANID) (Grant No. FONDECYT INICIACIÓN 11230262). Ricardo Marques acknowledges the support of the Serra Húnter Programme. B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. The authors thankfully acknowledge the computer resources at FinisTerrae III and the technical support provided by the Galician Supercomputing Center (CESGA) (RES-IM-2023-2-0025).

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., et al. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, *76*, 243–297.

Allen-Zhu, Z., Li, Y., & Liang, Y. (2019). Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in Neural Information Processing Systems*, *32*.

Angluin, D., & Laird, P. (1988). Learning from noisy examples. *Machine Learning*, *2*, 343–370.

Arazo, E., Ortego, D., Albert, P., O'Connor, N., & McGuinness, K. (2019). Unsupervised label noise modeling and loss correction. In *International conference on machine learning* (pp. 312–321). PMLR.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., et al. (2017). A closer look at memorization in deep networks. In *International conference on machine learning* (pp. 233–242). PMLR.

Bahri, D., Jiang, H., & Gupta, M. (2020). Deep k-nn for noisy labels. In *International conference on machine learning* (pp. 540–550). PMLR.

Bai, Y., Yang, E., Han, B., Yang, Y., Li, J., Mao, Y., et al. (2021). Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems*, *34*, 24392–24403.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, *32*.

Boluki, S., Ardywibowo, R., Dadaneh, S. Z., Zhou, M., & Qian, X. (2020). Learnable Bernoulli dropout for Bayesian deep learning. In *International conference on artificial intelligence and statistics* (pp. 3905–3916). PMLR.

Cai, K., Zhang, H., Pedrycz, W., & Miao, D. (2023). SSS-Net: A shadowed-sets-based semi-supervised sample selection network for classification on noise labeled images. *Knowledge-Based Systems*, Article 110732.

Chen, Y., Hu, S. X., Shen, X., Ai, C., & Suykens, J. A. (2022). Compressing features for learning with noisy labels. *IEEE Transactions on Neural Networks and Learning Systems*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597–1607). PMLR.

Cordeiro, F. R., Sachdeva, R., Belagiannis, V., Reid, I., & Carneiro, G. (2023). Longremix: Robust learning with high confidence samples in a noisy label environment. *Pattern Recognition*, *133*, Article 109013.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 113–123).

Cubuk, E. D., Zoph, B., Shlens, J., & Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 702–703).

Der Kiureghian, A., & Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, *31*(2), 105–112.

Ding, Y., Wang, L., Fan, D., & Gong, B. (2018). A semi-supervised two-stage approach to learning from noisy labels. In *2018 IEEE winter conference on applications of computer vision* (pp. 1215–1224). IEEE.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning* (pp. 1050–1059). PMLR.

Ghosh, A., Kumar, H., & Sastry, P. S. (2017). Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1*.

Ghosh, A., & Lan, A. (2021). Contrastive learning improves model robustness under label noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2703–2708).

Goel, P., & Chen, L. (2021). On the robustness of monte carlo dropout trained with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2219–2228).

Goldberger, J., & Ben-Reuven, E. (2017). Training deep neural-networks using a noise adaptation layer. In *International conference on learning representations*.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., et al. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems*, *31*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In *Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, proceedings, Part IV 14* (pp. 630–645). Springer.

Huang, Y., Bai, B., Zhao, S., Bai, K., & Wang, F. (2022). Uncertainty-aware learning against label noise on imbalanced datasets. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 6* (pp. 6960–6969).

Huang, B., Lin, Y., & Xu, C. (2022). Contrastive label correction for noisy label learning. *Information Sciences*, *611*, 173–184.

Huang, J., Qu, L., Jia, R., & Zhao, B. (2019). O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3326–3334).

Iscen, A., Valmadre, J., Arnab, A., & Schmid, C. (2022). Learning with neighbor consistency for noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4672–4681).

Ji, D., Oh, D., Hyun, Y., Kwon, O. M., & Park, M. J. (2021). How to handle noisy labels for robust learning from uncertainty. *Neural Networks*, *143*, 209–217.

Jiang, L., Zhou, Z., Leung, T., Li, L. J., & Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning* (pp. 2304–2313). PMLR.

Jospin, L. V., Laga, H., Boussaid, F., Buntine, W., & Bennamoun, M. (2022). Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, *17*(2), 29–48.

Karim, N., Rizve, M. N., Rahnavard, N., Mian, A., & Shah, M. (2022). Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9676–9686).

Kim, D., Ryoo, K., Cho, H., & Kim, S. (2022). SplitNet: Learnable clean-noisy label splitting for learning with noisy labels. arXiv preprint arXiv:2211.11753.

Köhler, J. M., Autenrieth, M., & Beluch, W. H. (2019). Uncertainty based detection and relabeling of noisy image labels. In *CVPR workshops* (pp. 33–37).

Krizhevsky, A., Hinton, G., et al. (2009). *Learning multiple layers of features from tiny images*. ON, Canada: Toronto.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., et al. (2020). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, *128*(7), 1956–1981.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, *30*.

Lange, R. D., Benjamin, A. S., Haefner, R. M., & Pitkow, X. (2022). Interpolating between sampling and variational inference with infinite stochastic mixtures. In *Uncertainty in artificial intelligence* (pp. 1063–1073). PMLR.

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

Li, W., Wang, L., Li, W., Agustsson, E., & Van Gool, L. (2017). Webvision database: Visual learning and understanding from web data. arXiv preprint arXiv:1708.02862.

Li, J., Wong, Y., Zhao, Q., & Kankanhalli, M. S. (2019). Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5051–5059).

Li, S., Xia, X., Ge, S., & Liu, T. (2022). Selective-supervised contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 316–325).

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., & Li, L. J. (2017). Learning from noisy labels with distillation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1910–1918).

Liao, Y. H., Kar, A., & Fidler, S. (2021). Towards good practices for efficiently annotating large-scale image classification datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4350–4359).

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, *37*(1), 145–151.

Liu, S., Niles-Weed, J., Razavian, N., & Fernandez-Granda, C. (2020). Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, *33*, 20331–20342.

Liu, S., Zhu, Z., Qu, Q., & You, C. (2022). Robust training under label noise by over-parameterization. In *International conference on machine learning* (pp. 14153–14172). PMLR.

Lu, Y., & Selc, W. H. (2022). Self-ensemble label correction improves learning with noisy labels, 3. arXiv preprint arXiv:2205.01156.

Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., & Bailey, J. (2020). Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning* (pp. 6543–6553). PMLR.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., & Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, *32*.

Malach, E., & Shalev-Shwartz, S. (2017). Decoupling "when to update" from "how to update". *Advances in Neural Information Processing Systems*, *30*.

Menon, A. K., Rawat, A. S., Reddi, S. J., & Kumar, S. (2020). Can gradient clipping mitigate label noise? In *International conference on learning representations*.

Miao, Q., Wu, X., Xu, C., Zuo, W., & Meng, Z. (2023). On better detecting and leveraging noisy samples for learning with severe label noise. *Pattern Recognition*, *136*, Article 109210.

Nagarajan, B., Marques, R., Mejia, M., & Radeva, P. (2022). Class-conditional importance weighting for deep learning with noisy labels.. In *VISIGRAPP* (pp. 679–686).

Nguyen, D. T., Mummadi, C. K., Ngo, T. P. N., Nguyen, T. H. P., Beggel, L., & Brox, T. (2019). Self: Learning to filter noisy labels with self-ensembling. arXiv preprint arXiv:1910.01842.

Nishi, K., Ding, Y., Rich, A., & Hollerer, T. (2021). Augmentation strategies for learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8022–8031).

Northcutt, C., Jiang, L., & Chuang, I. (2021). Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, *70*, 1373–1411.

Oh, D., Lee, D., Byun, J., & Shin, B. (2022). Improving group robustness under noisy labels using predictive uncertainty. arXiv preprint arXiv:2212.07026.

Ortego, D., Arazo, E., Albert, P., O'Connor, N. E., & McGuinness, K. (2021). Multi-objective interpolation training for robustness to label noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6606–6615).

Oyen, D., Kucer, M., Hengartner, N., & Singh, H. S. (2022). Robustness to label noise depends on the shape of the noise distribution in feature space. arXiv preprint arXiv:2206.01106.

Pan, C., Yuan, B., Zhou, W., & Yao, X. (2022). Towards robust uncertainty estimation in the presence of noisy labels. In *Artificial neural networks and machine learning–ICANN 2022: 31st international conference on artificial neural networks, Bristol, UK, September 6–9, 2022, proceedings, Part I* (pp. 673–684). Springer.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., & Qu, L. (2017). Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1944–1952).

Patrini, G., Rozza, A., Menon, A., Nock, R., & Qu, L. (2016). Making neural networks robust to label noise: a loss correction approach. *Stat*, *1050*, 13.

Ren, M., Zeng, W., Yang, B., & Urtasun, R. (2018). Learning to reweight examples for robust deep learning. In *International conference on machine learning* (pp. 4334–4343). PMLR.

Ridnik, T., Ben-Baruch, E., Noy, A., & Zelnik-Manor, L. (2021). Imagenet-21k pretraining for the masses. arXiv preprint arXiv:2104.10972.

Rizve, M. N., Duarte, K., Rawat, Y. S., & Shah, M. (2021). In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. arXiv preprint arXiv:2101.06329.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252.

Schmarje, L., Santarossa, M., Schröder, S. M., & Koch, R. (2021). A survey on semi-, self-and unsupervised learning for image classification. *IEEE Access*, *9*, 82146–82168.

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

Sun, Z., Shen, F., Huang, D., Wang, Q., Shu, X., Yao, Y., et al. (2022). Pnp: Robust learning from noisy labels by probabilistic noise prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5311–5320).

Tan, C., Xia, J., Wu, L., & Li, S. Z. (2021). Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM international conference on multimedia* (pp. 1405–1413).

Tanaka, D., Ikami, D., Yamasaki, T., & Aizawa, K. (2018). Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5552–5560).

Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, *30*.

Tatjer, A., Nagarajan, B., Marques, R., & Radeva, P. (2023). CCLM: Class-conditional label noise modelling. In *Iberian conference on pattern recognition and image analysis* (pp. 3–14). Springer.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., & Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 322–330).

Wang, Y., Sun, X., & Fu, Y. (2022). Scalable penalized regression for noise detection in learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 346–355).

Wang, H., Xiao, R., Dong, Y., Feng, L., & Zhao, J. (2022). ProMix: Combating label noise via maximizing clean sample utility. arXiv preprint arXiv:2207.10276.

Wang, H., & Yeung, D. Y. (2020). A survey on Bayesian deep learning. *ACM Computing Surveys (CSUR)*, *53*(5), 1–37.

Wei, H., Feng, L., Chen, X., & An, B. (2020). Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13726–13735).

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

Wu, M., Li, Q., Yang, F., Zhang, J., Sheng, V. S., & Hou, J. (2023). Learning from biased crowdsourced labeling with deep clustering. *Expert Systems with Applications*, *211*, Article 118608.

Wu, S., Xia, X., Liu, T., Han, B., Gong, M., Wang, N., et al. (2021). Class2simi: A noise reduction perspective on learning with noisy labels. In *International conference on machine learning* (pp. 11285–11295). PMLR.

Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., et al. (2021a). Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*.

Xia, X., Liu, T., Han, B., Gong, M., Yu, J., Niu, G., et al. (2021b). Sample selection with uncertainty of losses for learning with noisy labels. arXiv preprint arXiv:2106.00445.

Xiao, T., Xia, T., Yang, Y., Huang, C., & Wang, X. (2015). Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2691–2699).

Yao, Y., Liu, T., Han, B., Gong, M., Deng, J., Niu, G., et al. (2020). Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in Neural Information Processing Systems*, *33*, 7260–7271.

Yao, Y., Sun, Z., Zhang, C., Shen, F., Wu, Q., Zhang, J., et al. (2021). Jo-src: A contrastive approach for combating noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5192–5201).

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., & Sugiyama, M. (2019). How does disagreement help generalization against label corruption? In *International conference on machine learning* (pp. 7164–7173). PMLR.

Yu, X., Jiang, Y., Shi, T., Feng, Z., Wang, Y., Song, M., et al. (2023). How to prevent the continuous damage of noises to model training? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12054–12063).

Zhai, X., Kolesnikov, A., Houlsby, N., & Beyer, L. (2022). Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12104–12113).

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, *64*(3), 107–115.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.

Zhang, Y., Niu, G., & Sugiyama, M. (2021). Learning noise transition matrix from only noisy labels via total variation regularization. In *International conference on machine learning* (pp. 12501–12512). PMLR.

Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A. M., & Litany, O. (2022). Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 1657–1667).