

Computer Vision with Real-World Data

Student	Slava Koshman
Student's time capacity	8 ± 3 hours
Simulation time capacity (GPU: 20 GB VRAM)	10 ± 5 hours

1 Introduction

The task of training deep learning models with noisy labels has been a significant challenge in machine learning, particularly in large-scale datasets where label quality may degrade due to human error, data corruption, or other factors. The presence of noisy labels can severely impact model accuracy and generalization, making robust training methods essential for ensuring reliable performance in real-world applications. In particular, the CIFAR-10N dataset [25], a widely-used benchmark for evaluating noise-resilient models, provides a valuable testing ground for developing methods to combat label noise.

Traditional methods of mitigating label noise focus on adjusting the loss function, improving label quality, or leveraging data augmentation techniques. Recently, the use of ensemble methods has gained popularity as a promising strategy for improving robustness in the presence of noisy labels. Ensemble techniques, such as majority voting, combine the predictions of multiple models, often leading to improved performance by reducing the variance and bias associated with individual models.

This work explores the effectiveness of combining loss function modifications with ensemble learning on the CIFAR-10N dataset. We investigate three loss functions: the baseline Cross-Entropy (CE) loss (fig. 4a), and two proposed approaches, B-loss (fig. 4b) and N-loss (fig. 4c). The primary objective of this study is to evaluate the impact of these loss functions, both individually and in ensemble configurations, on model performance under clean and noisy label conditions.

Through a series of experiments, we aim to provide insights into the trade-offs between loss function selection, ensemble techniques, and the presence of noisy labels. Our findings highlight the importance of incorporating ensembling methods to improve model robustness and suggest that further research into novel loss functions could yield significant improvements in noisy label handling.

2 Related work

Learning with Real-World Human Annotations. The CIFAR-10N dataset, an extension of the CIFAR-10 dataset, introduces real-world noisy annotations that reflect the challenges faced in practical applications of machine learning. Several studies have explored effective methods for learning from such noisy labels, providing valuable insights into model robustness and performance improvement.

One notable approach is presented by [25], who propose a novel framework that integrates label noise detection and correction mechanisms into the training process. Their method utilizes a combination of ensemble learning and robust loss functions to mitigate the impact of noisy annotations. By leveraging multiple models to identify and correct mislabeled examples, this approach enhances the overall accuracy of the predictions. Implementing similar techniques in our study could lead to significant improvements in performance, particularly in scenarios where human annotation errors are prevalent.

Another relevant work is by [28], which introduces a self-supervised learning framework specifically designed for datasets with noisy labels. The authors utilize a contrastive learning approach that encourages the model to learn robust feature representations, irrespective of label noise. By focusing on the underlying data distribution rather than solely on label accuracy, this method has shown promising results in improving classification performance. Adopting self-supervised techniques in our analysis could provide a complementary strategy to enhance model robustness against noisy labels.

Additionally, [3] explore the use of meta-learning to adaptively learn from noisy labels. Their proposed model employs a meta-learning algorithm that adjusts the learning process based on the observed noise patterns in the dataset. This dynamic adaptation allows for improved generalization and performance on unseen data. Incorporating meta-learning strategies into our experimental design may yield beneficial outcomes, as it enables the model to better cope with the variability inherent in real-world annotations.

In summary, leveraging methodologies from these studies—such as ensemble learning, self-supervised contrastive approaches, and meta-learning—can significantly enhance our understanding of how to effectively learn from real-

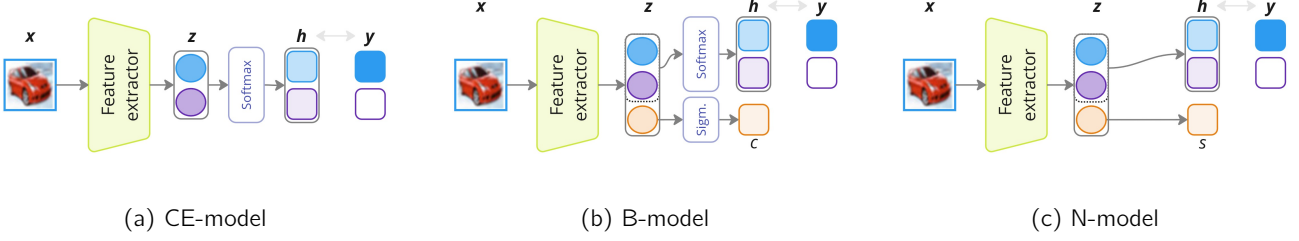


Figure 1: Schematics of the classification models under study: the CE-model serves as the baseline (see Equation (11)), the B-model includes an additional output $c \in (0, 1)$ that calculates the certainty value of the prediction \mathbf{h} (see Equation (7)), and the N-model introduces an additional output $s \in (-\infty, \infty)$ that computes the logarithmic variance of the prediction (see).

world human annotations in the CIFAR-10N dataset [25]. By critically analyzing these approaches, we can identify best practices and potential pitfalls, ultimately leading to improved outcomes in our research.

Uncertainty-aware objectives. One of the approaches to uncertainty estimation of the regression models is the *heteroscedastic* regression that takes both the variable mean and variance into account [19, 21]. So, the model trains to predict means and variances, and the uncertainty of the model predictions can be estimated using the variance values. Fortunately, classification models can also use a *squared error* (SE) loss. Hui and Belkin [11] demonstrated that the SE and CE-based computer vision models are close in accuracy. However, a SE loss needs some more training epochs. Kendall and Gal [12] dealt with two types of uncertainty, that are aleatoric (data uncertainty) and epistemic (model uncertainty), and proposed two approaches in uncertainty estimation. Kendall and Gal [12] declared that out-of-data examples cannot be identified with aleatoric uncertainty. The authors also proposed an approach that combines aleatoric and epistemic uncertainties. Further work by van Amersfoort et al. [23] deals with the deterministic uncertainty quantification method. The proposed model learns the positions of centroids of classes and trains kernels to estimate the distance between an input sample and centroids, which allows the inference model to recognize an out-of-data sample as uncertain. Sensoy et al. [22] developed a theory of evidence perspective and represented the model predictions as a Dirichlet density distribution over the softmax outputs and proposed a novel loss function. Collier et al. [4] proposed a method for training deep classifiers under heteroscedastic label noise. The method deals with the softmax temperature tuning that allows to control a bias-variance trade-off.

Ensembling, test-time augmentation, and label smoothing. Ashukha et al. [1] demonstrated that many ensembling techniques are equivalent to an ensemble of several independently trained networks in terms of test performance. Test-time augmentation is a technique that improves model performance using averaging the predictions [15]. Probably the simplest ways to make models be more robust to noise in labels are label smoothing [24] and data augmentation [20].

Data uncertainty estimation in practice Corrupted inputs [14] and corrupted labels [27], in-domain and out-of-domain distributions [16, 12, 4] are some of the poles of the research in the scope of data uncertainty estimation. The typical test of the models in practice is to use public datasets with corrupted (noisy) labels at the training and validation stages but with clean labels at the test stage [26, 27]. A number of methods try to detect input samples with incorrect labels and remove [4, 26, 27] or under-weight these samples [12, 5]. Han et al. [9] declared that models learn data with clean labels first and noisy labels then, and proposed a new paradigm called *co-teaching* with the training of two networks.

3 Methodology

In general, consider a model $\mathbf{f}[\mathbf{x}, \mathbf{w}]$ parameterized by weights \mathbf{w} that maps an input \mathbf{x} into logits \mathbf{z} first and then into the hypothesis \mathbf{h} that approximates the ground truth \mathbf{y} . The negative log-likelihood minimization [19, 2, 7] allows the formalization of the following uncertainty-aware loss functions for fitting and classification problems using different types of distributions for the outputs of the models.

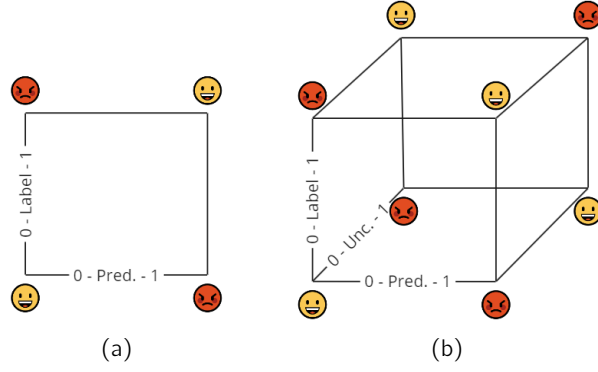


Figure 2: Binary classification intuition with the BCE loss (a) and the proposed binary B-loss eq. (4) (b) with respect to the values of the model's outputs (uncertainties $u = 1 - c$, predictions h), and labels y .

3.1 B-loss

This paragraph presents a specific interpretation of a binary classification model based on minimizing the *uncertainty-aware negative log-likelihood with the Bernoulli distribution* (B-model, B-loss). The proposed model is trained to ensure that true predictions are made certain and false predictions if they occur, are made uncertain (see fig. 2). The binary classifier estimates the certainty value $c \in (0, 1)$, which is the primary task in the proposed formalization. Additionally, the classifier estimates and enhances the similarity δ between the hypothesis \mathbf{h} and the ground truth \mathbf{y} , which constitutes the secondary task in the proposed formalization.

Binary classification. Consider an i^{th} sample and a model with logits $z^{(i)} = [z_{\text{pred}}^{(i)}, z_{\text{cert}}^{(i)}]$ which correspond to a prediction $h^{(i)} = \sigma(z_{\text{pred}}^{(i)})$, and the certainty $c_i = \sigma(z_{\text{cert}}^{(i)})$ associated with the prediction, respectively. Then, compare the prediction $h^{(i)}$ and the given label $y^{(i)}$ using a scalar product metric $\delta_i = y^{(i)} h^{(i)}$, and map this metric as a pseudo-label of a binary uncertainty estimator into the parameters of a Bernoulli probability mass function [19]:

$$p_i = p(\delta_i | c_i) = \begin{cases} 1 - c_i & \text{if } \delta_i \rightarrow 0, \\ c_i & \text{if } \delta_i \rightarrow 1, \end{cases} \quad (1)$$

where $\delta_i \in (0, 1)$ is the smoothed pseudo-label that characterizes the similarity between the label and the prediction.

eq. (1) is a discrete probability distribution for a random variable that takes the value 0 with probability $1 - c_i$, which is an incorrect prediction that corresponds to an uncertainty of the prediction, and the value 1 with probability c_i , which is a right prediction that corresponds to a certainty of the prediction. The Bernoulli distribution has an equivalent power law form [19]:

$$p_i = c_i^{\delta_i} (1 - c_i)^{1 - \delta_i}. \quad (2)$$

For a roll-out of dataset of m i.i.d. pairs $\{x^{(i)}, y^{(i)}\}$ associated with the outputs of the model $\{h^{(i)}, c_i\}$, the joint probability [2] for the given probability mass function eq. (2) takes the following form:

$$P(\delta_1, \dots, \delta_m | c_1, \dots, c_m) = \prod_{i=1}^m c_i^{\delta_i} (1 - c_i)^{1 - \delta_i}. \quad (3)$$

The negative logarithm of the joint probability eq. (3) represents the proposed uncertainty-aware B-loss for the binary classification:

$$\mathcal{L}_B = -\frac{1}{m} \sum_{i=1}^m [\delta_i \log c_i + (1 - \delta_i) \log(1 - c_i)]. \quad (4)$$

eq. (4) intuition is demonstrated in fig. 2. The B-loss can be generalized for the case of multiclass classification.

Multiclass (N-classes) classification. Consider an i^{th} sample and a model with logits $\mathbf{z}^{(i)} = [\mathbf{z}_{\text{pred}}^{(i)}, z_{\text{cert}}^{(i)}]$ which correspond to a vector of prediction $\mathbf{h}^{(i)} = \text{softmax}(\mathbf{z}_{\text{pred}}^{(i)})$, $\mathbf{h}^{(i)} \in \mathcal{R}^N$, and the certainty $c_i = \sigma(z_{\text{cert}}^{(i)})$ associated with the prediction, relatively. Then, compare the prediction vector $\mathbf{h}^{(i)}$ and the given one-hot encoded label vector $\mathbf{y}^{(i)}$ using a scalar product terms $\delta_k^{(i)} = y_k^{(i)} h_k^{(i)}$, and map this metrics as pseudo-labels into a probability mass function:

$$p_i = \prod_{k=1}^N \left(\frac{c_i}{N} \right)^{\delta_k^{(i)}} \left(\frac{1 - c_i}{N} \right)^{1 - \delta_k^{(i)}}, \quad (5)$$

where $\delta_k^{(i)} \in (0, 1)$ is the smoothed one-hot encoded pseudo-label that characterizes the similarity between the k^{th} components of the label and the prediction vectors, N is the number of classes.

Following the logical sequence given in section 3.1 and in [19], the joint probability for eq. (5) can be obtained, and then transformed into the negative log-likelihood (NLL):

$$NLL = -\frac{1}{m} \sum_{i=1}^m \left(\cos(\mathbf{h}^i, \mathbf{y}^i) \log \left(\frac{c^{(i)}}{N} \right) + (N-1) (1 - \cos(\mathbf{h}^i, \mathbf{y}^i)) \log \left(\frac{1 - c^{(i)}}{N} \right) \right), \quad (6)$$

where $\cos(\mathbf{h}^i, \mathbf{y}^i)$ is the smoothed pseudo-label that characterizes the cosine similarity between two N -dimensional vectors: the vector of prediction and the one-hot encoded label vector.

Finally, the proposed uncertainty-aware B-loss for the N -classes classification is the Kulback-Loeberg divergence between two distributions: the one-hot encoded smoothed pseudo-labels distribution and the NLL distribution eq. (6):

$$\mathcal{L}_B = \frac{1}{m} \sum_{i=1}^m \sum_{k=1}^N \delta_k^{(i)} \log \delta_k^{(i)} + NLL. \quad (7)$$

where m is the number of samples (in a batch), N is the number of classes, $\delta_i = y^{(i)} h^{(i)}$ are the terms of a scalar product of the one-hot encoded label vector and the vector of prediction of the model, $c^{(i)}$ is the certainty of the prediction (fig. 4b).

3.2 N-loss

Since the binary classification can be assumed as a particular case of a multiclass classification, this section skips the binary classification paragraph.

Multiclass classification. Consider an i -th sample and a model with logits $\mathbf{z}^{(i)} = [\mathbf{z}_{mean}^{(i)}, z_{var}^{(i)}]$ which maps into the parameters of a multivariate normal distribution: the hypothesis or mean $\mathbf{h}^{(i)} = \mathbf{z}_{mean}^{(i)}$ that approximates the ground truth $\mathbf{y}^{(i)}$, and the variance $\sigma_{(i)}^2 = \exp(z_{var}^{(i)})$ that characterizes the uncertainty of the hypothesis, $f[\mathbf{x}^{(i)}, \mathbf{w}] = [\mathbf{h}^{(i)}, \sigma_{(i)}^2]$. In other words, it is assumed that the conditional probability distribution $p = p(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}) = p(\mathbf{y}^{(i)} | \mathbf{f}[\mathbf{x}^{(i)}, \mathbf{w}])$ has the form of a multivariate normal distribution characterized by equal variances (spherical covariances) in N -dimensional space [18]:

$$p^{(i)} = \frac{\exp \left(-\frac{\sum_{k=1}^N (y_k^{(i)} - h_k^{(i)})^2}{2\sigma_{(i)}^2} \right)}{(2\pi\sigma_{(i)}^2)^{\frac{N}{2}}}, \quad (8)$$

The multivariate normal distribution of (8) can be applied to the negative log-likelihood criterion of an uncertainty-aware negative log-likelihood loss (N-loss) for the regression [19]:

$$\mathcal{L}_N = \frac{1}{2m} \sum_{i=1}^m \left(\sum_{k=1}^N \frac{(y_k^{(i)} - h_k^{(i)})^2}{\sigma_{(i)}^2} + N(s^{(i)} + r) \right), \quad (9)$$

where m is the number of samples (in a batch), $\mathbf{y}^{(i)}$, $s^{(i)} = \log \sigma_{(i)}^2$ is the log-variance, $r = \log 2\pi$ is the constant value.

The last term in (9) represents a constant that can be neglected. Kendall et al. [12] recommended to train the models to predict log-variances $s^{(i)} = \log \sigma_{(i)}^2$, because it is more numerically stable than the variance $\sigma_{(i)}^2$ and the loss avoids a potential division by zero:

$$\mathcal{L}_N = \frac{1}{m} \sum_{i=1}^m \left(e^{-s^{(i)}} \sum_{k=1}^N (y_k^{(i)} - h_k^{(i)})^2 + Ns^{(i)} \right). \quad (10)$$

Thus, (10) represents a heteroscedastic regression loss [19, 12], generalized for the case of a space of N dimensions. Our proposal is to use this loss for classification problems.

The baseline loss in a multi-class classification problem is the cross-entropy loss [7, 19, 2]:

$$L_{CE} = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^N y_k^{(i)} \log h_k^{(i)}. \quad (11)$$

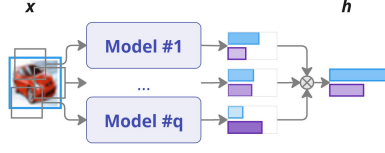


Figure 3: Ensemble of q models makes the prediction \mathbf{h} for the q augmented copies of an input sample \mathbf{x} . Two-classes classification is demonstrated.

3.3 Ensembling

A set of q models $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_q$ with different random initialization of the weights are trained with the same dataset. This aggregation reduces overfitting and provides more robust estimates by averaging out individual model errors [1]. Each j^{th} model predicts a class index for the given i^{th} input:

$$\hat{y}^{(ij)} = \arg \max_k h_k^{(ij)}, \quad (12)$$

where i, j, k are the dummy indexes that refer to j^{th} augmented version of i^{th} sample, and k^{th} component of the prediction vector or class index, $i \in (1, m), j \in (1, q), k \in (0, N - 1)$.

The final ensemble prediction class is typically determined by *majority voting* based on the class predictions of the individual j^{th} model.

$$\hat{y}^{(i)} = \text{mode}(\hat{y}^{(i,1)}, \hat{y}^{(i,2)}, \dots, \hat{y}^{(i,q)}). \quad (13)$$

The final ensemble prediction class can also be determined by *confidence-based weighted predictions* (see fig. 3). Each model predicts a class $\hat{y}^{(ij)}$ and provides a confidence value $co^{(ij)}$ for its prediction:

$$co^{(ij)} = \max_k (h_k^{(ij)}). \quad (14)$$

The aggregated confidence for class k is:

$$co_k^{(i)} = \sum_{j=1}^q co^{(ij)} \cdot \mathbb{I}(\hat{y}^{(ij)} = k), \quad (15)$$

where $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the condition is true and 0 otherwise, $co_k^{(i)}$ is the total confidence for class k across all models.

The final predicted class is:

$$\hat{y}^{(i)} = \arg \max_k co_k^{(i)}. \quad (16)$$

The uncertainty estimation in deep ensembling is derived from the variance of the individual model predictions. Higher variance among the models' outputs indicates greater uncertainty, providing a measure of epistemic uncertainty.

3.4 Metrics

The standard classification metrics for the balanced datasets are the accuracy, receiver operating characteristic - area under curve (ROC-AUC) [2, 7]. A set of more specific metrics used in uncertainty quantification involves estimation the confidence (see eq. (14)) [16]: the Brier score, the entropy, the expected calibration error (ECE), the negative log-likelihood (NLL), the prediction interval coverage probability (PICP), the sharpness, etc. [17, 8, 6, 10].

Since the proposed B-model (see eq. (7)) and N-model (see eq. (10)) have an extra output, the following additional *certainty* metrics can be met:

- $c^{(i)} \in (0, 1)$ for the B-model ;
- $1 - \text{sigm}(s^{(i)}) \in (0, 1)$ for the N-model.

Both of the above metrics can be used as weights in eq. (16), thus the *certainty-based weighted predictions* should be met.

Table 1: Accuracy (%) of models trained on the CIFAR-10N dataset with clean and noisy labels, as well as with label smoothing (LS). Each model was trained using seven different weight initialization seeds for 20 epochs, then ensembled. The mean test accuracy of individual models is compared with the accuracy of the ensemble using majority voting (EMV) and the ensemble with weighted predictions (EWP). The best results are highlighted in bold.

Method	Arch.	#Param. (train.par.)	LS	Accuracy (single model / EMV / EWP),%	
				Clean	Noisy (rand2)
Baseline CE loss	9-l.CNN	4.4 M (all)	0.4	87.00 \pm 0.005/89.63/00.00	81.07 \pm 0.004/84.66/00.00
Proposed B-loss				85.55 \pm 0.005/88.77/00.00	80.85 \pm 0.007/84.54/00.00
Proposed N-loss				85.53 \pm 0.003/88.17/00.00	80.54 \pm 0.005/83.80/00.00

4 Results and Discussion

CIFAR-10N dataset [25] was split into training, validation, and test sets in the amounts of [45000, 5000, 10000] samples, respectively. The models were trained with a 9-layer convolutional neural network [9, 26]. The network has 4.4 million parameters that were randomly initialized during training. The CNN architecture and most of the settings correspond to the experiments by Xia et al. [26] with minor changes: the models were trained for 20 epochs (200 in the original paper) using the Adam optimizer with a momentum of 0.9 and a batch size of 128, and a constant learning rate of 0.001 (in the original paper, the initial learning rate linearly decreased to zero starting from the 80th epoch); image samples were transformed into tensors and normalized with means of [0.491, 0.482, 0.447] and standard deviations of [0.247, 0.243, 0.261].

Some of the experimental settings might differ from those of Xia et al. [26]: the model ensembling technique was applied; a random resized crop with a scale range of [0.8, 0.1] and an aspect ratio range of [0.9, 1.1] was applied to all samples in all sets as a transformation, so test set sampling was implemented using test time augmentation with a random resized crop; models with the lowest validation loss were used for inference.

All experiments were performed seven times with random seeds of [42, 0, 17, 9, 3, 16, 2]. The mean and standard deviation of the experimental results were then reported. The multiple predictions obtained through model ensembling allowed for the calculation of final predictions using majority voting.

Accuracy was used [13] as metric. The obtained results were not compared with the state-of-the-art results. The latter aggregate multiple techniques and complex network architectures. So, the comparison would be unfair.

The experimental results, presented in Table (1), show that the baseline model trained with Cross-Entropy (CE) loss outperformed other models in both clean and noisy label settings. The proposed B-loss and N-loss approaches exhibited competitive accuracy, especially when paired with the ensemble majority voting technique.

Further insight into the training dynamics can be observed in the validation loss curves, illustrated in (fig. 4). The CE-model demonstrated the most stable convergence behavior, while B-loss and N-loss displayed slightly more fluctuation during training, particularly under noisy conditions.

These findings highlight the effectiveness of ensemble methods in improving robustness to label noise and underline the strong baseline performance of CE loss. Nevertheless, the proposed loss functions, combined with ensembling, present a promising direction for handling noisy labels.

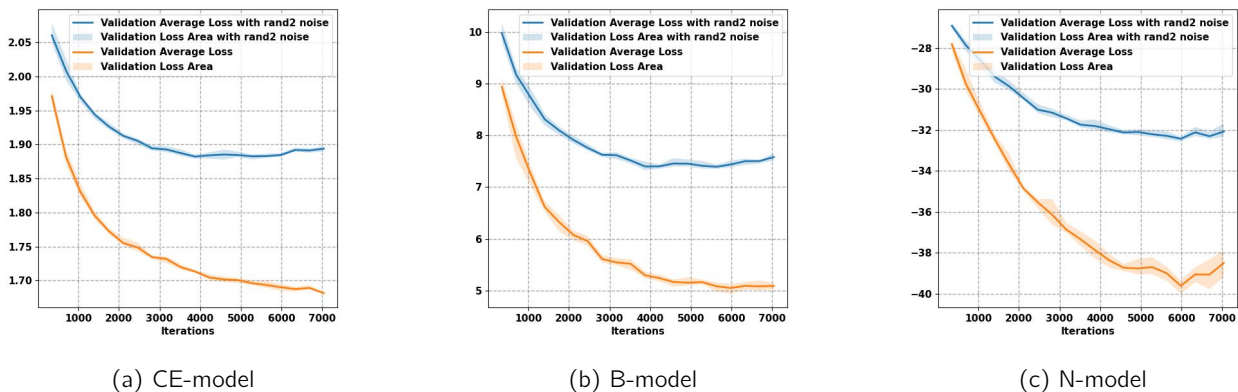


Figure 4: Validation loss values during the training of the ensemble models in 20 epochs with clean and noisy data: CE-model (a), B-model (b), and N-model (c).

5 Ethics

In the development of this work, a Large Language Model (LLM) was used to assist in non-technical aspects of the project, with clear boundaries to maintain academic integrity.

- **Literature Review:** The LLM was not used to locate or analyze research papers. All related work and citations were selected and reviewed independently.
- **Coding:** No assistance from the LLM was utilized for writing, debugging, or optimizing code. All experiments, model training, and ensembling procedures were implemented solely by the student.
- **Writing and Editing:** The LLM provided support in refining the textual content, specifically for drafting, rephrasing, and improving clarity and flow. The goal was to enhance the quality of the report while ensuring the content remained original.
- **Critical Thinking:** The LLM assisted in organizing arguments and structuring explanations, particularly in sections involving result interpretation and discussion of findings.

The conceptual design of the experiments, the analysis of results, and all scientific contributions were developed independently by the student. The LLM served exclusively as a writing assistant and brainstorming tool to improve readability and coherence without influencing the core technical or research outcomes.

6 Conclusion

This work explored the training of ensemble models on the CIFAR-10N dataset under both clean and noisy label conditions. The study focused on comparing the baseline Cross-Entropy (CE) loss function with two proposed loss functions, B-loss and N-loss. The experimental results demonstrated that the CE loss provided the strongest individual performance, particularly when combined with ensemble majority voting. However, the B-loss and N-loss approaches proved competitive and showed potential in mitigating the negative effects of label noise when paired with ensemble methods.

The experiments reaffirm the importance of ensemble techniques in improving model robustness, especially in noisy label scenarios. The improvements seen through majority voting highlight how simple aggregation techniques can outperform individual models, even when label noise is present.

Beyond numerical results, this study underscores the trade-offs between loss function selection and model behavior under noisy supervision. While the proposed loss functions did not outperform CE loss across all settings, their competitive performance invites further exploration with extended training schedules or larger model architectures.

Overall, this work contributes to a clearer understanding of ensemble-based strategies in noise-resilient deep learning and opens avenues for future research into loss function design and robust model aggregation methods.

References

- [1] A. Ashukha, A. Lyzhov, D. Molchanov, and D. Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.
- [2] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [3] J. Chen et al. Robust learning from noisy labels via meta-learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.
- [4] M. Collier, B. Mustafa, E. Kokiopoulou, R. Jenatton, and J. Berent. A simple probabilistic method for deep classification under input-dependent label noise. *arXiv preprint arXiv:2003.06778*, 2020.
- [5] E. Engleson, A. Mehrpanah, and H. Azizpour. Logistic-normal likelihoods for heteroscedastic label noise, 2023.
- [6] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- [7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.

- [8] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. *International Conference on Machine Learning*, pages 1321–1330, 2017.
- [9] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels, 2018.
- [10] F. Hernandez, L. Bertino, G. Brassington, E. Chassignet, J. Cummings, F. Davidson, M. Drevillon, G. Garric, M. Kamachi, J. M. Lellouche, et al. Probabilistic forecasting in meteorology: A review. *Quarterly Journal of the Royal Meteorological Society*, 141(688):318–350, 2015.
- [11] L. Hui and M. Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks, 2021.
- [12] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.
- [13] A. Kumar, P. Liang, and T. Ma. Verified uncertainty calibration, 2020.
- [14] E. Mintun, A. Kirillov, and S. Xie. On interaction between augmentations and corruptions in natural corruption robustness, 2021.
- [15] D. Molchanov, A. Lyzhov, Y. Molchanova, A. Ashukha, and D. Vetrov. Greedy policy search: A simple baseline for learnable test-time augmentation, 2020.
- [16] T. Pearce, A. Brintrup, and J. Zhu. Understanding softmax confidence and uncertainty. *CoRR*, abs/2106.04972, 2021. URL <https://arxiv.org/abs/2106.04972>.
- [17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [18] S. Prince. *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012.
- [19] S. J. Prince. *Understanding Deep Learning*. MIT Press, 2023. URL <http://udlbook.com>.
- [20] S.-A. Rebuffi, S. Goyal, D. A. Calian, F. Stimberg, O. Wiles, and T. A. Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34:29935–29948, 2021.
- [21] M. Seitzer, A. Tavakoli, D. Antic, and G. Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. *arXiv preprint arXiv:2203.09168*, 2022.
- [22] M. Sensoy, L. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty, 2018.
- [23] J. van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal. Simple and scalable epistemic uncertainty estimation using a single deep deterministic neural network. *CoRR*, abs/2003.02037, 2020. URL <https://arxiv.org/abs/2003.02037>.
- [24] J. Wei, H. Liu, T. Liu, G. Niu, M. Sugiyama, and Y. Liu. To smooth or not? when label smoothing meets noisy labels. *arXiv preprint arXiv:2106.04149*, 2021.
- [25] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu. Learning with noisy labels revisited: A study using real-world human annotations, 2022. URL <https://arxiv.org/abs/2110.12088>.
- [26] X. Xia, T. Liu, B. Han, M. Gong, J. Yu, G. Niu, and M. Sugiyama. Sample selection with uncertainty of losses for learning with noisy labels, 2021.
- [27] Q. Yao, H. Yang, B. Han, G. Niu, and J. Kwok. Searching to exploit memorization effect in learning from corrupted labels, 2020.
- [28] W. Zhang et al. Learning from noisy labels with self-supervised contrastive learning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2021.