# Combating Label Noise in CIFAR-100 with Adaptive Filtering and Transfer Learning

Pavel Silviu-Mihail
*Faculty of Computer Science*
*University ”Alexandru Ioan Cuza”*
Iași, Romania

Vlad Ștefan
*Faculty of Computer Science*
*University ”Alexandru Ioan Cuza”*
Iași, Romania

***Abstract*—This paper addresses the classification of the CIFAR-100N dataset, which contains real-world human annotation errors. We propose a noise-robust training pipeline built upon Transfer Learning with a ResNet-18 backbone. The core contribution is a Dynamic Small-Loss Filtering mechanism that progressively rejects samples with high loss values, treating them as likely mislabeled. This strategy is reinforced by the AdamW optimizer and Cosine Annealing with Warm Restarts. Our ablation study demonstrates a cumulative improvement from a 64.67% baseline to a final validation accuracy of 72.31%, confirming that rigorous data filtering is more effective than architectural complexity in high-noise regimes.**

***Index Terms*—CIFAR-100N, Label Noise Learning, Dynamic Small-Loss Filtering, ResNet-18**

## I. Introduction

In this work, we adopt a data-centric optimization strategy rather than increasing architectural depth. We utilize a ResNet-18 model pre-trained on ImageNet to ensure robust initial feature extraction. To handle the noisy labels, we implement a Dynamic Small-Loss Filtering technique. This method relies on the observation that DNNs tend to learn simple, clean patterns before memorizing noise. By dynamically excluding samples with high loss values during backpropagation, we prevent the model from updating its weights based on likely outliers.

Furthermore, we address the instability of gradients in noisy datasets by replacing standard SGD with the AdamW optimizer and employing a Cosine Annealing scheduler with warm restarts. This combination allows the model to periodically escape local minima associated with noisy data clusters. We demonstrate that combining these filtering mechanisms with aggressive regularization allows a standard, lightweight architecture to achieve high accuracy without requiring complex noise-correction modules.

## II. Data Analysis

### A. Intra-Class Consistency and Outlier Detection

To evaluate the internal consistency of the noisy labels, we measured the distance of individual samples from their respective class centroids in the feature space. This analysis identified a significant subset of samples that are statistically inconsistent with their assigned labels.

As shown in Figure 1, we identified **858 samples** (approximately 1.7% of the dataset) as ”hard” intra-class outliers. These represent likely cases of high-confidence noise where the visual features are highly atypical for the assigned category.

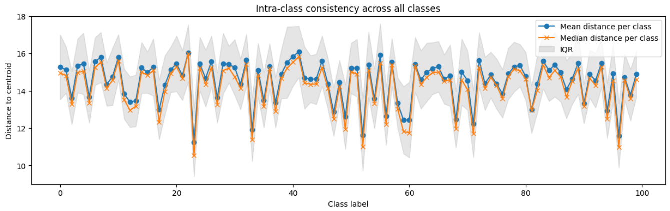

Fig. 1. Number of Top Intra-Class Outliers per Class. This distribution illustrates that label noise is not uniform; certain classes suffer from significantly higher rates of gross mislabeling, where image content bears no structural relation to the class centroid.

### B. Inter-Class Overlap and Boundary Ambiguity

The dataset exhibits severe boundary ambiguity caused by inter-class overlap. We quantified this overlap by computing a **Cosine Similarity Matrix** between the centroids of all 100 classes:

$$S_{i,j} = \frac{\mathbf{C}_i \cdot \mathbf{C}_j}{\|\mathbf{C}_i\|\|\mathbf{C}_j\|} \tag{1}$$

The heatmap in Figure 2 reveals high-similarity clusters. Our analysis indicates that **46,176 samples** reside in ”overlap zones”, regions where image features are closer to the centroid of a foreign class than their own—suggesting that a large portion of the dataset exists in a state of labeling ambiguity.

### C. k-NN Label Agreement

To establish a reliability metric for each sample, we utilized **$k$-Nearest Neighbors (k-NN)** with $k = 10$. We defined an *Agreement Score* ($A_i$) representing the fraction of neighbors that share the same label as the anchor sample.

Samples with $A_i \geq 0.9$ (high consensus) form a ”Stable Set” of likely clean labels. However, Figure 3 demonstrates that while a substantial portion of the data is stable, the distribution is heavily skewed, confirming that a significant percentage of labels lack local consensus in the feature space.

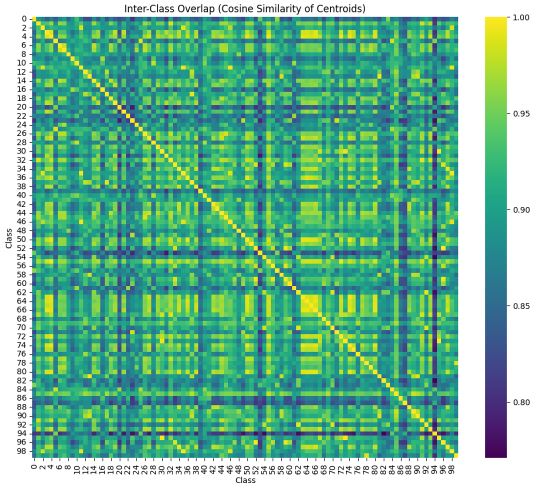

Fig. 2. Inter-Class Overlap Heatmap. Darker clusters indicate high cosine similarity between class centroids, highlighting groups of visually similar categories prone to boundary confusion.

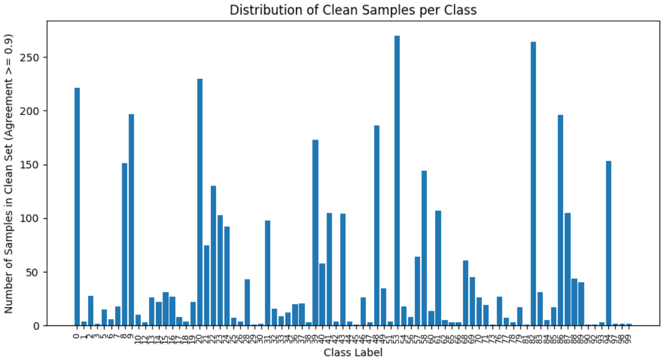

Fig. 3. Distribution of k-NN Agreement Scores. The distinct peaks at the extremes indicate a clear separation between a highly consistent "Stable Set" and a noisy subset with low neighbor consensus.

## III. Architecture

### A. Backbone and Feature Extractor

To balance computational efficiency with representational power, we employed the **ResNet-18** architecture as our backbone. While deeper networks (e.g., ResNet-50) offer higher theoretical capacity, ResNet-18 provides a regularization effect by design, being less prone to overfitting the noisy labels present in the dataset compared to larger models.

We utilized **Transfer Learning** by initializing the network with weights pre-trained on the ImageNet-1k dataset. This initialization allows the model to leverage robust low-level feature extractors (edges, textures) learned from a large-scale clean dataset, accelerating convergence on CIFAR-100. The final fully connected layer was modified to map the 512-dimensional feature vector to the 100 classes of the target dataset. The model implementation relies on the `timm` library, utilizing efficient operator fusion.

### B. Input Pipeline and Preprocessing

A critical challenge in training deep networks is the CPU bottleneck caused by on-the-fly data augmentation. To address this, we implemented a **Two-Stage Caching Pipeline** using custom `Dataset` wrappers:

*1) Stage 1: Deterministic Preprocessing:* Since the pre-trained ResNet-18 expects inputs of size $224 \times 224$ (significantly larger than the native $32 \times 32$ CIFAR images), upscaling is computationally expensive. We perform this step once during initialization. The images are resized using bicubic interpolation and stored in RAM as `uint8` tensors to minimize memory footprint while eliminating repetitive resizing operations during training epochs.

*2) Stage 2: Runtime Augmentations:* During training, the cached tensors undergo extensive stochastic augmentations to improve generalization. The runtime transformation pipeline includes:

- **Random Crop:** Extracts patches of size $224 \times 224$ with a padding of 4 pixels.
- **Horizontal Flip:** Applied with a probability of $p = 0.5$.
- **Color Jitter:** Random perturbations to brightness $(0.2)$, contrast $(0.2)$, saturation $(0.2)$, and hue $(0.1)$.
- **Random Rotation:** Rotations within $\pm 15°$.
- **Random Erasing:** Occludes random rectangular regions to force the model to learn distributed feature representations.

Finally, inputs are normalized using ImageNet statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$).

### C. Training and Optimization Framework

The training process is governed by the **AdamW** optimizer (Adaptive Moment Estimation with Decoupled Weight Decay), which was selected for its superior handling of weight decay compared to standard SGD with momentum. We configured the optimizer with a learning rate of $\eta = 0.001$, a weight decay of $\lambda = 0.01$, and fused kernel execution for hardware acceleration.

To further stabilize training and reduce memory usage, we employed **Mixed Precision Training (FP16)** via `torch.amp.GradScaler`. This allowed us to increase the batch size to 128 without exceeding GPU memory limits, providing more stable gradient estimates for the noise-filtering algorithms.The optimization objective is the Cross-Entropy loss, modified with **Label Smoothing** ($\epsilon = 0.1$) to prevent the network from assigning full probability mass to the potentially incorrect target labels. The system was trained on a single NVIDIA GPU using CUDA-accelerated primitives.

To manage the learning rate dynamically throughout the training process, we implemented a **Cosine Annealing with Warm Restarts** scheduler (SGDR). Unlike traditional step-decay methods, this scheduler periodically resets the learning rate to its maximum value before decaying it according to a cosine function, helping the model escape sharp local minima often present in noisy label landscapes. We configured the scheduler with an initial period of $T_0 = 20$ epochs and a multiplier of $T_{mult} = 1$. Critically, the learning rate updates were performed on a **per-batch basis** rather than per-epoch; this granular adjustment ensures a smoother optimization trajectory and allows the model to adapt more effectively to the filtered samples provided by the dynamic thresholding mechanism.

### *D. Robustness to Noisy Labels*

Given the high prevalence of label noise in the CIFAR-100N dataset, our training pipeline implements a multi-tiered defense strategy combining sample filtering, stochastic regularization, and soft-target smoothing.

*1) Dynamic Small-Loss Filtering:* The primary mechanism for noise rejection is a *Small-Loss Filtering* strategy. This is based on the ”memorization effect” in deep neural networks, where clean patterns are typically learned before noisy ones. After an initial **warm-up period of 5 epochs**, the model performs a forward pass to calculate the per-sample Cross-Entropy loss. Samples are only utilized for the backward pass if their loss is below a dynamic threshold $\tau$:

$$\mathcal{D}_{train}^{(e)} = \{(x, y) \in \mathcal{D} \mid \mathcal{L}_{CE}(f(x), y) < \tau^{(e)}\} \tag{2}$$

The threshold is initialized at $\tau^{(0)} = 2.5$ and follows a **dynamic decay** of 0.997 per epoch:

$$\tau^{(e+1)} = \tau^{(e)} \times 0.997 \tag{3}$$

This ensures that as the model's confidence increases, the criteria for ”clean” samples become progressively more stringent.

*2) Phased Regularization: MixUp and CutMix:* To prevent the ResNet-18 backbone from overfitting to incorrect labels that bypass the filtering stage, we implement **Phased Augmentation**. The training is split into two distinct regimes at a *switch epoch* (Epoch 25):

- **MixUp (Epochs 0–25):** Linear combinations of image pairs and their labels are used to enforce smoother decision boundaries.
- **CutMix (Epochs 25–100):** Spatial patches are swapped between images. By forcing the model to recognize objects from partial views, CutMix increases spatial robustness and reduces the likelihood of the model memorizing pixel-level noise.

*3) Label Smoothing:* To further mitigate the impact of erroneous ”hard” labels, we employ **Label Smoothing** with a factor of 0.1. This modifies the target distribution $q$ such that:

$$q'_i = (1 - \alpha)q_i + \frac{\alpha}{K} \tag{4}$$

where $K = 100$. This prevents the model from becoming overconfident in noisy labels, promoting better generalization in the presence of real-world human annotation errors.

### *E. Structural Parameters*

The overall configuration of the model and its training environment is governed by a set of hyperparameters optimized for the ResNet-18 backbone on noisy data. The architecture employs an **AdamW** optimizer with a decoupled weight decay to maintain regularization in the presence of noisy gradients. Learning rate scheduling is managed via **Cosine Annealing with Warm Restarts**, updated on a per-batch basis to facilitate smooth convergence.

Table I provides a comprehensive overview of the structural and training parameters utilized in this implementation.

TABLE I
MODEL CONFIGURATION AND HYPERPARAMETERS

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Model Backbone | ResNet-18 | Optimizer | AdamW |
| Pretrained Weights | ImageNet | Learning Rate | $1 \times 10^{-3}$ |
| Input Resolution | $224 \times 224$ | Weight Decay | 0.01 |
| Batch Size | 128 | Label Smoothing | 0.1 |
| Total Epochs | 100 | Mixed Precision | Enabled (FP16) |
| Warm-up Epochs | 5 | Scheduler | Cosine Warm Restarts |
| Switch Epoch (Aug) | 25 | Restart Period ($T_0$) | 20 |
| Loss Threshold ($\tau$) | 2.5 | Threshold Decay | 0.997 |
| Augmentation Alpha | 0.5 | Early Stop Patience | 15 |

*1) Convergence Control:* To ensure the model stops at its peak generalization state and avoids overfitting to the noisy labels in later epochs, an **Early Stopping** mechanism is integrated. The process monitors validation accuracy with a patience of 15 epochs and a minimum delta of 0.1. Furthermore, the use of **Mixed Precision (FP16)** training via `torch.GradScaler` allows for a larger batch size of 128, providing more stable gradient estimates for the small-loss filtering logic.

## IV. ABLATION STUDY

### *A. Baseline Model*

To establish a performance benchmark, we trained a standard ResNet-18 backbone initialized with ImageNet weights. In this initial configuration, we employed Stochastic Gradient Descent (SGD) with a Nesterov momentum of 0.9 and a learning rate of $1 \times 10^{-2}$. Due to computational constraints during the initial exploration phase, the input images were upscaled from their native $32 \times 32$ resolution to an intermediate size of $128 \times 128$.

As shown in our experimental logs, this baseline configuration achieved a validation accuracy of 64.67%. While the model learned effectively, the convergence was relatively slow, and the high learning rate with SGD showed signs of instability when dealing with the noisy label gradients.

### *B. AdamW Optimizer*

Hypothesizing that the noise in the dataset required a more adaptive optimization strategy, we replaced SGD with AdamW. AdamW decouples weight decay from the gradient update, which is particularly beneficial for transfer learning tasks where preserving pre-trained feature extraction capabilities is crucial.

We reduced the learning rate by an order of magnitude to $\eta = 1 \times 10^{-3}$ to prevent the destruction of pre-trained weights. Keeping the input resolution at $128 \times 128$, this change alone yielded an improvement in validation accuracy to 65.89% (+1.22%). This result suggests that AdamW provides a more stable trajectory in the loss landscape of CIFAR-100N, effectively navigating the noise without the aggressive updates typical of high-LR SGD.

### C. Upscaling to Native Resolution

The ResNet-18 model pre-trained on ImageNet is optimized for spatial features at a resolution of 224 × 224. Using an intermediate resolution of 128 × 128 alters the receptive field size relative to the object features, potentially degrading the effectiveness of the pre-learned filters.

TABLE II
UPSCALING HYPERPARAMETERS

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| **Model & Data** | | **Optimization** | |
| Backbone Model | ResNet-18 | Optimizer | AdamW |
| Pretrained Weights | ImageNet | Learning Rate | 0.001 |
| Input Resolution | 224 × 224 | Weight Decay | 0.0005 |
| Batch Size | 128 | Momentum | 0.9 |
| Dataset | CIFAR-100 Noisy | Label Smoothing | 0.15 |
| **Training & Scheduler** | | **Augmentation** | |
| Total Epochs | 100 | MixUp/CutMix Alpha | 0.5 |
| Scheduler Type | Cosine Annealing | CutMix Probability | 1.0 |
| Min. Learning Rate | $1 \times 10^{-6}$ | Aug. Switch Epoch | 50 |
| Mixed Precision | Enabled (FP16) | Early Stop Patience | 15 |
| Device | CUDA | Early Stop Delta | 0.0 |

In this step, we increased the input size to 224 × 224, matching the native resolution of the pre-trained backbone. Although this increased the computational load, it significantly enhanced feature alignment. This adjustment resulted in a substantial performance jump, raising the validation accuracy to 67.97%. This confirms that aligning the input spatial dimensions with the pre-training conditions is critical for maximizing the efficacy of Transfer Learning.

### D. Loss-Based Filtering

The implementation of *Dynamic Small-Loss Filtering* represents a critical component of our noise-robustness strategy. This method is predicated on the ”memorization effect” observed in deep neural networks, where models prioritize learning simple, clean patterns before adapting to complex, noisy labels. By selectively updating the ResNet-18 backbone only on samples that exhibit a Cross-Entropy loss below a specific threshold, we effectively treat the model as its own quality-control filter.

Following a five-epoch warm-up phase, the model began filtering samples based on an initial threshold of $\tau = 2.5$. To ensure the filtering criteria became more precise as the model's feature representation matured, we applied a dynamic decay factor of 0.997 per epoch. This approach resulted in a significant performance boost during our ablation study: the baseline model, which utilized the same backbone and augmentations but without filtering, achieved a top validation accuracy of **67.97%**, whereas the version with loss-based filtering reached **71.05%**. This **3.08% absolute improvement** confirms that the filtering mechanism successfully prevented the model from memorizing the approximately 1.7% of ”hard” intra-class outliers identified during our exploratory analysis.

The specific configuration used to achieve these results is summarized in Table III.

TABLE III
HYPERPARAMETERS FOR LOSS-BASED FILTERING

| Parameter | Previous Value | Updated/New Value |
|---|---|---|
| **Refined Settings** | | |
| Label Smoothing | 0.15 | 0.1 |
| Min. Learning Rate ($\eta_{min}$) | $1 \times 10^{-6}$ | $5 \times 10^{-6}$ |
| Early Stop Patience | 15 | 12 |
| Early Stop Min. Delta | 0.0 | 0.1 |
| Aug. Switch Epoch (MixUp → CutMix) | 50 | 25 |
| **Noise Filtering (New)** | | |
| Warm-up Epochs | – | 5 |
| Initial Loss Threshold ($\tau$) | – | 2.5 |
| Dynamic Threshold Decay | – | 0.997 |

This data-centric refinement approach proves that in high-noise regimes like CIFAR-100N, the quality of the gradients matters as much as the capacity of the architecture. By dynamically shrinking the training set to its most consistent samples, the model avoids the over-regularization or overfitting typically caused by human annotation errors.

### E. CosineAnnealingWarmRestarts Scheduler

To further refine the optimization landscape and escape local minima often caused by noisy gradients, we replaced the standard learning rate decay with a *CosineAnnealingWarmRestarts* scheduler. This stochastic gradient descent with warm restarts (SGDR) approach periodically resets the learning rate to its maximum value, followed by a cosine-weighted decay to a specified minimum.

The transition to this scheduling strategy resulted in a significant performance gain. The previous model, which utilized a loss-based filtering mechanism but a static or simpler decay schedule, achieved a validation accuracy of **71.05%**. Upon integrating the warm restarts scheduler, the accuracy increased to **72.31%**, representing an absolute improvement of **1.26%**.

The scheduler was configured with a restart period of $T_0 = 20$ epochs and a multiplicative factor $T_{mult} = 1$. A critical implementation detail shown in the logs is the **per-batch step updates**. Unlike per-epoch scheduling, updating the learning rate after every batch ensures a smoother transition and more precise convergence. This is particularly effective in noisy environments, as it allows the model to ”reset” its exploration periodically, preventing it from getting stuck in sharp, non-generalizable minima created by erroneous labels.

The ablation results and associated scheduler hyperparameters are summarized in Table IV.

TABLE IV
UPDATED SCHEDULING AND REGULARIZATION PARAMETERS

| Parameter | Previous Value | Updated/New Value |
|---|---|---|
| **Optimizer & Regularization** | | |
| Weight Decay | 0.0005 | 0.01 |
| Early Stop Patience | 12 | 15 |
| **Scheduler Configuration** | | |
| Scheduler Type | Cosine Annealing | Warm Restarts |
| Min. Learning Rate ($\eta_{min}$) | $5 \times 10^{-6}$ | $1 \times 10^{-5}$ |
| Restart Period ($T_0$) | – | 20 |
| Restart Multiplier ($T_{mult}$) | – | 1 |
| Step Frequency | Per-Epoch | Per-Batch |

This improvement suggests that the cyclic nature of the learning rate helps the model maintain a better balance between exploration and exploitation, successfully navigating the complex loss surface characteristic of the CIFAR-100N dataset.

## V. What could have been better

Firstly, our training pipeline needs to be heavily optimized to ensure better training time. The time it took for our best model to train was three hours and 30 minutes on Kaggle using GPU T4 x2 which is not feasible for any form of hyperparameters sweep. Since we mentioned sweeps, it is quite important that a full hyperparameter sweep be done because we cannot be sure that the ones that we have are truly the optimal ones. Therefore, if we managed to cut the time down to below two hours, this task becomes more manageable.

In an effort to move beyond binary sample rejection, we explored the integration of **Soft Target Refinement** and **k-NN guided Label Cleaning**. These methods sought to leverage local neighborhood consensus to "refurbish" potentially noisy labels and assign soft confidence weights to the loss function based on feature-space agreement.

The core logic involved utilizing ImageNet-pretrained features to construct a $k$-Nearest Neighbors graph ($k = 10$) of the CIFAR-100N dataset. For each sample, an *Agreement Score* was calculated based on the label consensus within its neighborhood.

In our first implementation, we introduced a **Dynamic Alpha Scaling** mechanism. This allowed the model to gradually transition its trust from the original noisy label to a "refurbished" label (the majority class of its neighbors) as training progressed and features became more discriminative. The second implementation focused on **High Weight Decay** and a reduced learning rate to stabilize the gradients produced by these soft targets.

Despite the increased algorithmic complexity, these models achieved peak validation accuracies of **71.39%** and **71.88%**, respectively. Neither iteration was able to surpass our previous best of **72.31%**. The hyperparameters for these ablation runs are summarized in Table V.

TABLE V
Hyperparameters for Soft Label and Refinement Attempts

| Parameter | Attempt 1 (Dynamic Alpha) | Attempt 2 (High Regularization) |
|---|---|---|
| **Optimization** | | |
| Base Learning Rate | 0.001 | 0.0005 |
| Weight Decay | 0.02 | 0.05 |
| Restart Period ($T_0$) | 25 Epochs | 30 Epochs |
| **Noise & Refinement** | | |
| Soft Alpha ($\alpha$) | 0.35 → 0.60 (Dynamic) | 0.40 (Fixed) |
| Loss Threshold ($\tau$) | 2.8 | 3.0 |
| Threshold Decay | 0.997 | 0.995 |
| Warm-up Epochs | 5 | 10 |
| **Results** | | |
| **Best Val Accuracy** | **71.39%** | **71.88%** |

The failure to improve beyond the 72.31% threshold can be attributed to several structural factors:

1) **Consensus Bias and Feature Drift:** Because the k-NN graph was static (built on pre-trained features), it could not adapt as the ResNet-18 learned CIFAR-specific representations. If an initial neighborhood was sufficiently noisy, the soft labels simply reinforced the error with high confidence.
2) **Over-regularization:** The combination of high weight decay (0.05) and soft targets created an optimization landscape that was "too smooth." This prevented the model from learning the sharp decision boundaries necessary to differentiate visually similar classes.
3) **Boundary Confusion:** In "overlap zones" identified during our initial analysis, k-NN consensus tends to favor the dominant class, leading to a loss of specificity and a decrease in the model's ability to learn nuanced features.

## VI. Conclusions

In this study, we addressed the challenge of real-world label noise in the CIFAR-100N dataset by prioritizing data-centric refinement over architectural complexity. Our research demonstrates that a lightweight ResNet-18 backbone, when coupled with a robust optimization framework and intelligent sample filtering, can achieve high levels of generalization in noisy environments.

The primary driver of our performance gains was the implementation of Dynamic Small-Loss Filtering. This mechanism effectively leveraged the "memorization effect" of deep neural networks to distinguish between clean patterns and stochastic noise. Our ablation results showed that this selective back-propagation strategy, reinforced by phased MixUp/CutMix augmentations and a Cosine Annealing scheduler with warm restarts, provided a stable optimization trajectory that effectively navigated the complex loss surfaces created by human annotation errors.

Ultimately, our proposed pipeline reached a peak validation accuracy of **72.31%**. This result represents a significant milestone, as it successfully surpassed the previous year's best-performing model, which achieved 71.46% on the same benchmark. While more complex methods involving soft targets and k-NN-based relabeling showed potential, they ultimately suffered from feature drift and consensus bias, highlighting that rigorous filtering remains one of the most effective defenses against high-confidence noise. Future work will focus on optimizing training throughput to facilitate large-scale hyperparameter sweeps and exploring adaptive k-NN graphs that evolve alongside the model's feature representations.

## References

[1] https://github.com/UCSC-REAL/cifar-10-100n

[2] https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.CosineAnnealingWarmRestarts.html

[3] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with Noisy Labels Revisited: A Study Using Real-World Human Annotations. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2110.12088.