Core ML

SAiDL Assignment 2025

Rajat Soni

1 Introduction and Motivation

In real-world deep learning, training data is often contaminated with noisy labels. Noisy labels can lead conventional loss functions—such as the widely used Cross Entropy (CE) loss—to overfit incorrect examples, degrading the performance of deep neural networks (DNNs). Traditionally, several approaches have been proposed to mitigate this issue, including:

- **Label Correction:** Methods that identify and correct incorrect labels.
- **Loss Correction:** Techniques that modify the loss function based on an estimated noise transition matrix.
- **Refined Training Strategies:** Approaches that adapt the training procedure (e.g., co-teaching).
- **Robust Loss Functions:** Designing losses that are inherently tolerant to noise (e.g., MAE and RCE).

The paper makes a striking theoretical claim: *any loss function can be made robust to noisy labels by applying a simple normalization.* However, the authors also show that robustness alone is not sufficient—a loss must also enable the network to learn the data effectively (i.e., avoid underfitting).

2 Normalizing Loss Functions

2.1 The Normalization Principle

The core idea is that if a loss function $L(f(x), y)$ satisfies:

$$\sum_{j=1}^{K} L(f(x), j) = C, \quad \forall x \in X, \forall f,$$

where C is a constant, then the loss is theoretically robust under mild assumptions. This observation motivates the following normalization:

$$L_{\text{norm}}(f(x), y) = \frac{L(f(x), y)}{\sum_{j=1}^{K} L(f(x), j)}. \tag{1}$$

This normalization confines the loss to the range $[0, 1]$ and ensures that each class's contribution is balanced, thus mitigating the effects of label noise.

2.2 Examples of Normalized Loss Functions

The paper demonstrates how several well-known loss functions can be normalized.

Normalized Cross Entropy (NCE)

The standard CE loss is defined as:

$$\text{CE}(f(x), y) = -\sum_{k=1}^{K} q(k|x) \log p(k|x),$$

where $q(k|x)$ is the one-hot encoded true label and $p(k|x)$ is the softmax probability. The normalized version becomes:

$$\text{NCE} = \frac{-\sum_{k=1}^{K} q(k|x) \log p(k|x)}{\sum_{j=1}^{K} \left(-\sum_{k=1}^{K} q(y=j|x) \log p(k|x) \right)} = \log \frac{\prod_{k=1}^{K} p(k|x)}{p(y|x)}. \tag{2}$$

Normalized Mean Absolute Error (NMAE)

The Mean Absolute Error (MAE) is defined as:

$$\text{MAE} = \sum_{k=1}^{K} \left| p(k|x) - q(k|x) \right|.$$

Since it holds that

$$\sum_{k=1}^{K} |p(k|x) - q(k|x)| = 2(1 - p(y|x)),$$

the normalized MAE is given by:

$$\text{NMAE} = \frac{\text{MAE}}{\sum_{j=1}^{K} \left(2(1 - p(y = j|x))\right)} = \frac{1}{2(K-1)}\text{MAE}. \tag{3}$$

Normalized Reverse Cross Entropy (NRCE)

For the Reverse Cross Entropy (RCE) defined as:

$$\text{RCE} = -\sum_{k=1}^{K} p(k|x) \log q(k|x),$$

with the understanding that $\log q(k \neq y|x)$ is truncated to a constant A (for instance, $A = -4$), the normalized version is:

$$\text{NRCE} = \frac{\text{RCE}}{\sum_{j=1}^{K} \left(\text{RCE evaluated at } y = j\right)} = \frac{1}{A(K-1)}\text{RCE}. \tag{4}$$

Normalized Focal Loss (NFL)

The Focal Loss (FL) is originally defined as:

$$\text{FL} = -\sum_{k=1}^{K} q(k|x)(1 - p(k|x))^{\gamma} \log p(k|x),$$

where $\gamma \geq 0$ is a tunable parameter. The normalized Focal Loss becomes:

$$\text{NFL} = \frac{\text{FL}}{\sum_{j=1}^{K} \left[-\sum_{k=1}^{K} q(y = j|x)(1 - p(k|x))^{\gamma} \log p(k|x)\right]} = \log \frac{\prod_{k=1}^{K}(1 - p(k|x))^{\gamma} p(k|x)}{(1 - p(y|x))^{\gamma} p(y|x)}. \tag{5}$$

These equations illustrate that normalization effectively scales robust losses (such as MAE and RCE) and can transform non-robust losses (like CE and FL) into robust variants.

3 Theoretical Justification for Noise Tolerance

The paper provides formal proofs showing that normalized loss functions are noise tolerant under both symmetric and asymmetric label noise.

Lemma 1 (Symmetric Noise)

In a K-class setting, if the label noise is symmetric (i.e., any incorrect label is equally likely) and the noise rate $\eta < \frac{K-1}{K}$, then any normalized loss L_{norm} defined as in Equation (1) is noise tolerant. This result relies on the constant sum condition ensuring that the loss landscape does not overly favor noisy labels.

Lemma 2 (Asymmetric Noise)

For class-conditional (asymmetric) noise, under the assumption that the risk of the clean data $R(f^*) = 0$ and the individual loss terms satisfy

$$0 \leq L_{\text{norm}}(f(x), k) \leq \frac{1}{K-1} \quad \text{for all } k,$$

the normalized loss remains noise tolerant provided that the noise rate for any incorrect class satisfies

$$\eta_{jk} < 1 - \eta_y.$$

Although the condition $R(f^*) = 0$ (perfect separability) is strong, empirical evidence suggests that the normalized losses still perform robustly in practice.

4 Balancing Robustness and Learning Capacity

4.1 The Underfitting Problem

While normalization prevents overfitting by making the loss robust to noise, it can also lead to underfitting. For example, normalized versions of CE and FL (i.e., NCE and NFL) sometimes yield lower training accuracy compared to their unnormalized counterparts. Inherently robust losses like MAE and RCE may also struggle to converge because normalization dampens the gradients excessively.

4.2 Active versus Passive Loss Functions

To address underfitting, the paper distinguishes between:

- **Active Losses:** These losses solely focus on maximizing the probability of the true label. Expressing the loss as

  $$L(f(x), y) = \sum_{k=1}^{K} \ell(f(x), k),$$

  an active loss L_{Active} satisfies

  $$\ell(f(x), k) = 0 \quad \text{for all } k \neq y.$$

  The Cross Entropy (CE) loss is an example.
- **Passive Losses:** These losses also explicitly minimize the probabilities assigned to incorrect classes, meaning there exists at least one $k \neq y$ such that

  $$\ell(f(x), k) \neq 0.$$

  Both MAE and RCE fall under this category.

4.3 Active Passive Loss (APL) Framework

The APL framework combines the strengths of both active and passive losses:

$$L_{\text{APL}} = \alpha \cdot L_{\text{Active}} + \beta \cdot L_{\text{Passive}}, \tag{6}$$

where α and β are positive hyperparameters. The active component (e.g., NCE) ensures the network focuses on the correct class, while the passive component (e.g., MAE or RCE) reduces the probabilities for incorrect classes. This combination enables robust yet effective learning.

5 Project Implementation and Data Preparation

In my project, I applied the above theoretical insights to the CIFAR-10 dataset. Here is an overview of my process:

5.1 Library and Environment Setup

I began by importing essential libraries for deep learning and image processing. I used PyTorch for constructing and training neural networks, Torchvision for accessing the CIFAR-10 dataset and performing image transformations, and Matplotlib along with NumPy for visualization and numerical computations. Additionally, I utilized modules for automatic mixed-precision training (i.e., `autocast` and `GradScaler`) to optimize performance on modern GPUs by safely leveraging half-precision computations.

5.2 Data Preparation and Noise Injection

A critical aspect of the notebook was the data preparation pipeline. The CIFAR-10 dataset was preprocessed using a transformation pipeline that converts images to PyTorch tensors and normalizes them. The normalization used a mean and standard deviation of 0.5 across all three channels, effectively rescaling pixel values to be centered around zero (approximately in the range -1 to 1).

I defined two functions to simulate label noise:

- **Symmetric Noise:** A function randomly selects a percentage of the training samples (based on a specified noise rate) and replaces their true labels with randomly chosen incorrect labels from the available classes. This simulates uniformly distributed label errors.
- **Asymmetric Noise:** A function that uses a predefined mapping (stored in an `asym_map`) to selectively flip labels based on class-specific misclassification patterns. This mimics common real-world errors, such as confusing an automobile with a truck.

Multiple versions of the dataset were generated by applying different noise levels. For symmetric noise, I used noise rates of 20%, 40%, 60%, and 80%. For asymmetric noise, I applied rates of 10%, 25%, and 40%. Each noisy dataset was created by making a copy of the original CIFAR-10 dataset and then applying the corresponding noise injection function in a controlled and reproducible manner.

5.3 Data Loading and Experimental Setup

I wrapped these datasets in PyTorch DataLoaders to enable efficient mini-batch processing and random shuffling, which are critical for robust training. Separate DataLoaders were created for each noisy version of the dataset, as well as for a clean test set. The clean test set provided a reliable benchmark to evaluate the performance of the trained models on uncorrupted data.

6 Experimental Strategy and Future Work

My experimental strategy was guided by the theoretical insights from the research paper. I planned to:

- Compare standard Cross Entropy (CE) loss with its normalized version (NCE) under varying noise levels.
- Investigate inherently robust losses (such as MAE and RCE) and analyze their convergence behavior.
- Implement the Active Passive Loss (APL) framework (e.g., combining NCE with MAE or RCE) to overcome the underfitting issues seen in normalized losses.

During training, I monitored both training and validation accuracy to observe the balance between robustness and sufficient learning. Future work will involve refining the hyperparameters (α and β) within the APL framework, extending experiments to other datasets, and further analyzing the trade-offs between noise robustness and model convergence.

7 Observations

We analyzed the performance of various loss functions under different levels of symmetric and asymmetric noise. The graphs illustrate the accuracy of these loss functions as the noise rate increases.

7.1 Symmetric Noise

Under symmetric noise conditions, where the probability of mislabeling is uniform across all classes, the following observations were noted:

- **CE vs. FL (Figure 1):** Both Cross-Entropy (CE) and Focal Loss (FL) display a decrease in accuracy as the noise rate increases. The degradation in performance is comparable for both loss functions when noise increases.
- **NCE vs. RCE vs. APL_NCE_RCE (Figure 2):**
  - Reverse Cross-Entropy (RCE) shows relatively higher accuracy at lower noise rates (e.g., 0.2 and 0.4) but a marked decrease at higher noise levels.
  - Normalized Cross-Entropy (NCE) maintains a more stable performance over the range of noise rates, particularly at higher noise levels (0.6 and 0.8).
  - APL_NCE_RCE starts with competitive accuracy; however, its performance decreases noticeably as the noise level increases.
- **NFL vs. MAE vs. APL_NFL_MAE (Figure 3):**
  - APL_NFL_MAE demonstrates the highest accuracy across all symmetric noise rates.
  - Normalized Focal Loss (NFL) follows closely, showing slightly lower accuracy.
  - Mean Absolute Error (MAE) consistently registers the lowest performance, indicating potential challenges in dealing with high symmetric noise.

7.2 Asymmetric Noise

Under asymmetric noise conditions, where the probability of mislabeling depends on the specific classes, the following observations were made:

- **CE vs. FL (Figure 4):** Cross-Entropy (CE) generally outperforms Focal Loss (FL) across all asymmetric noise rates. Although both loss functions show a decline in accuracy as noise increases (from 0.1 to 0.4), CE tends to maintain a higher accuracy.
- **NCE vs. RCE vs. APL_NCE_RCE (Figure 5):**
  - Reverse Cross-Entropy (RCE) exhibits strong performance across the evaluated noise rates.
  - Normalized Cross-Entropy (NCE) provides stable, though slightly lower, accuracy compared to RCE.
  - APL_NCE_RCE starts with competitive performance but shows a rapid decline as noise increases.
- **NFL vs. MAE vs. APL_NFL_MAE (Figure 6):**
  - APL_NFL_MAE records the highest accuracy, especially at lower asymmetric noise rates.
  - Normalized Focal Loss (NFL) follows with marginally lower accuracy.
  - Mean Absolute Error (MAE) once again records the lowest accuracy, with performance decreasing significantly as noise increases.

7.3 Discussion of Graph Observations

The graphs provide visual insights into the behavior of the loss functions under varying noise conditions. For both symmetric and asymmetric noises alike, accuracy generally declines as noise increases, with normalized loss functions and APL frameworks showing a more stable behavior at higher noise levels. Under asymmetric noise, while some loss functions—particularly those incorporating reverse cross-entropy—perform well at lower noise rates, all functions tend to degrade as noise increases. These observations are presented here without drawing definitive conclusions about the overall superiority of any particular method.

7.4 Figures

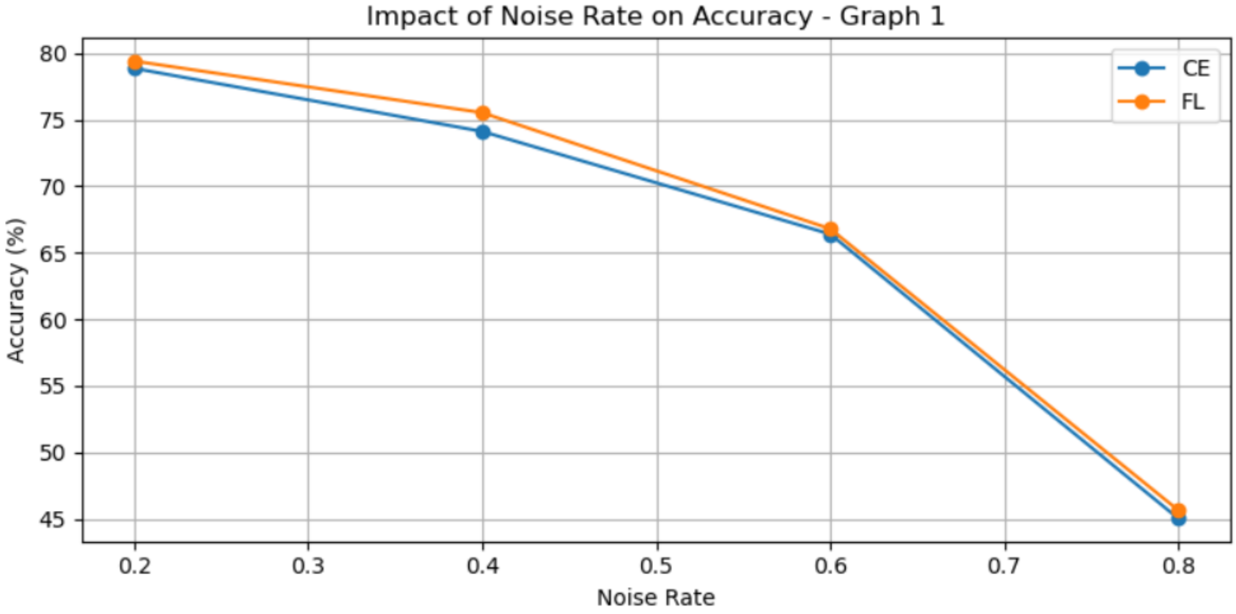

Figure 1: CE and FL performance under symmetric noise.

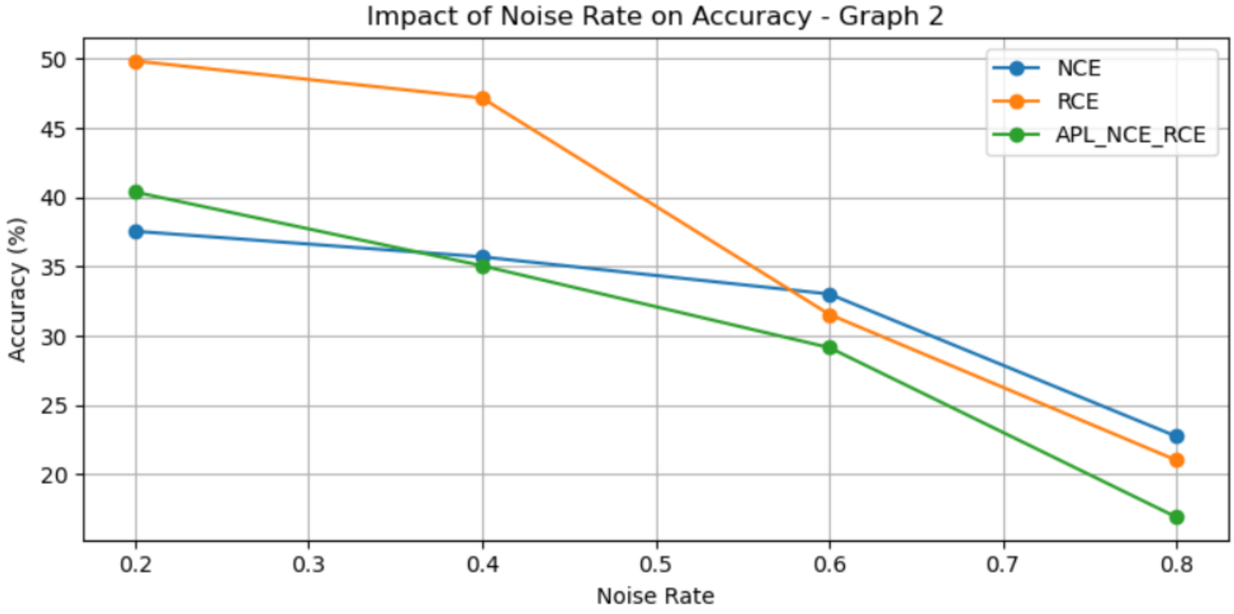

Figure 2: NCE, RCE, and APL_NCE_RCE performance under symmetric noise.

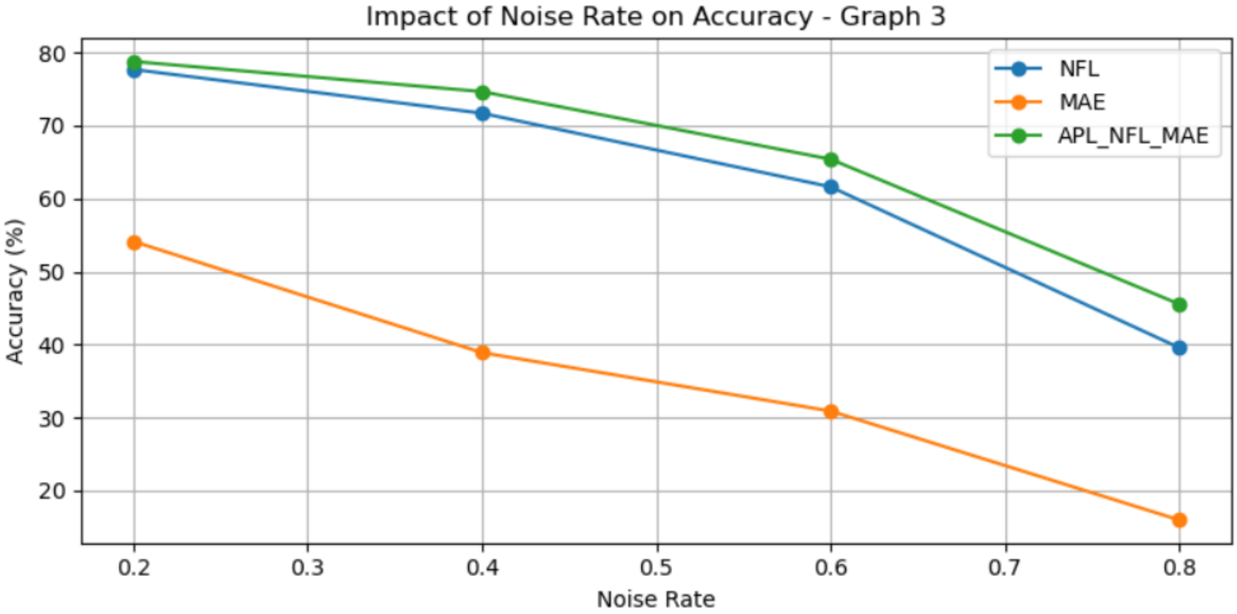

Figure 3: NFL, MAE, and APL_NFL_MAE performance under symmetric noise.

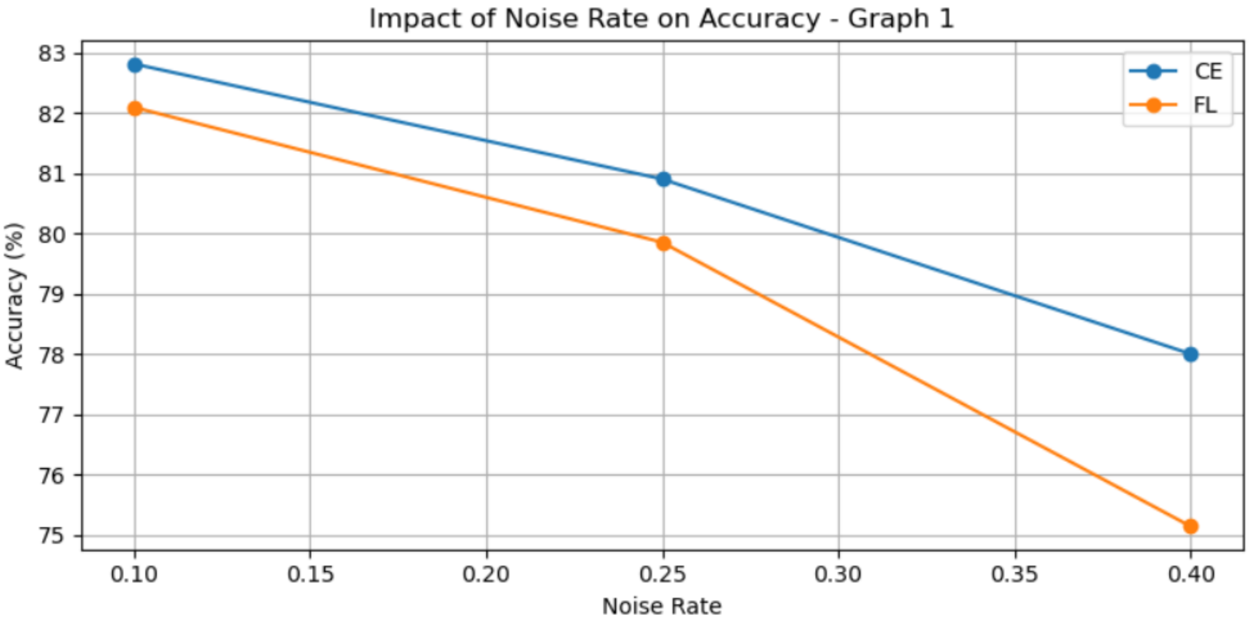

Figure 4: CE and FL performance under asymmetric noise.

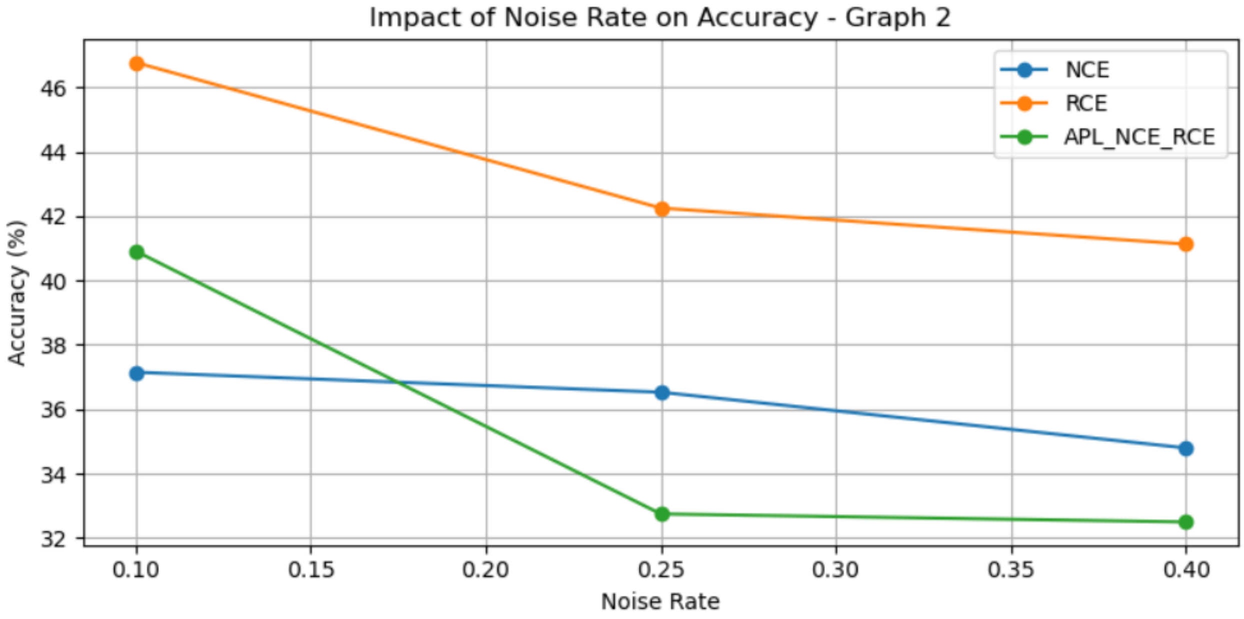

Figure 5: NCE, RCE, and APL_NCE_RCE performance under asymmetric noise.

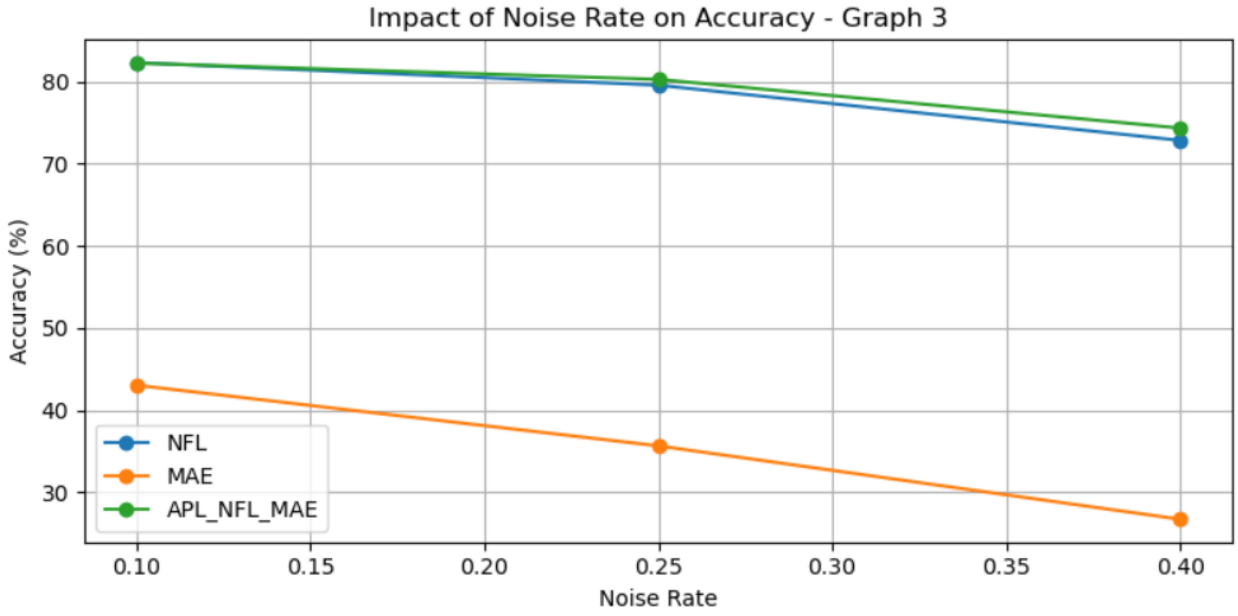

Figure 6: NFL, MAE, and APL_NFL_MAE performance under asymmetric noise.

8 Conclusion

In this work, we presented an empirical evaluation of various loss functions under both symmetric and asymmetric noise conditions. The results indicate that noise significantly affects model accuracy, and different loss functions exhibit distinct patterns in performance degradation as noise increases. Notably, normalized loss functions and those based on the Active Passive Loss (APL) framework show promising robustness characteristics. Final conclusions regarding optimal performance require further investigation and tuning (but most of the major claims in the paper have been verified).