Label Noise Detection and Correction via Ensemble of Siamese Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep neural networks can suffer severe performance degradation when trained on datasets with instance-dependent label noise—annotation errors that correlate with input features. To address this issue, we propose a lightweight, model-agnostic preprocessing framework based on an ensemble of contrastive Siamese networks. Our method detects and corrects noisy labels by measuring embedding consistency: clean samples yield stable representations across models, while noisy samples exhibit high variability and increased misclassification rates. Each Siamese model is trained on a subset of image pairs, and we demonstrate that noisy instances are significantly more likely to be misclassified under this subset-driven embedding process, with the ensemble's false-positive rate decaying exponentially with the number of models. Ultimately, samples with high model disagreement are flagged and either relabeled by consensus or discarded. Empirically, on real-world CIFAR-10N (9.01% natural noise), our method reduces label corruption to 4.45% and achieves 88.51% accuracy on the cleaned dataset—0.26 percentage points ahead of the nearest baseline. Under synthetic instance-dependent noise, label corruption on CIFAR-10 is reduced from 40% to 25.9% (yielding a 12.54 percentage point accuracy gain) and on Fashion-MNIST from 40% to 4.6% (a 2.23 percentage point accuracy gain). Our preprocessing step adds minimal overhead, produces interpretable uncertainty scores, and can be seamlessly integrated with any downstream learner to enhance robustness against label noise.[1]

1 Introduction

The performance of deep learning models critically depends on high-quality labeled training datasets [1]. However, real-world datasets often suffer from label corruption due to crowdsourcing inaccuracies, ambiguous cases, and inexpert annotations [2, 3]. Manual label verification is impractical at scale; thus, automated solutions must balance identifying noisy labels with retaining clean training examples [3]. The widespread occurrence of label noise in practical datasets has motivated extensive research. It profoundly degrades model performance and reliability.

Label noise is broadly categorized into **instance-independent noise (IIN)**—including symmetric (uniform) and asymmetric (class-conditional) noise—and **instance-dependent noise (IDN)**, where errors correlate with input features. IDN poses unique challenges; for example, a blurry "sneaker" image mislabeled as an "ankle boot" reflects feature-dependent ambiguity that undermines generalization [3, 4]. There are a variety of methods to mitigate label noise, which can be broadly categorized into **sample selection**, **label correction**, and **hybrid cleaning**.

Sample selection methods detect and remove noisy examples by exploiting the "memorization effect," where DNNs fit clean patterns before over-fitting noise [5]. Models are then trained only

[1]Code available at `https://hidden-because-anonymity`

on high-confidence samples, keeping original labels [3]. Recent advances include curriculum-based weighting (e.g., MentorNet [6]) and neighborhood consistency checks (e.g., ConFrag [2]).

Label correction methods tackle noise by revising annotations instead of discarding samples [7]. These include prediction-based strategies (e.g., iterative label updates using model confidence [8]) and clean sample-based methods (e.g., refining labels via agreement with trusted subsets [9]).

Hybrid cleaning methods integrate both selecting and correction techniques [10]. For instance, unclean sample correction partially relabels ambiguous instances flagged by disagreement metrics, while adaptive methods dynamically adjust correction criteria (e.g., confidence thresholds) based on model performance [11, 12].

Other paradigms include loss adjustment (e.g., noise-robust loss functions [13]) and regularization (e.g., adversarial training [14], mixup [15]). Meta-learning methods dynamically re-weight samples via bi-level optimization [16]. Contrastive learning (e.g., Jo-SRC [11]) learns noise-invariant representations but conflates semantic similarity with label noise.

While numerous approaches focus on modifying training procedures or loss functions, we propose a preprocessing framework that identifies and rectifies label errors before training, incorporating contrastive learning and ensemble disagreement. Contrastive learning is performed via a specialized Siamese network architecture that emphasizes visual consistency over potentially erroneous labels while learning distinctive feature representations for image pairs. We note that contrastive learning has previously been successfully used for identifying noisy labels [17, 1]. In this work, we propose an ensemble of Siamese networks to detect and correct instance-dependent label noise. Contrastive training produces compact, well-separated clusters for clean samples, whereas noisy points remain ambiguous and are more likely to be misclassified by individual models (see Appendix A), suggesting that the frequency of misclassification itself can serve as an effective indicator of label corruption.

Our contributions are as follows:

- A novel **Siamese-ensemble preprocessing framework** that trains each contrastive model on different random subsets to expose instance-dependent noise via embedding variability.
- **Theoretical guarantees** showing (a) noisy examples incur strictly higher misclassification probability under our subset-driven embedding process, and (b) the ensemble's false-positive detection rate decays exponentially in size.
- A novel **relabeling score** metric that quantitatively evaluates the quality of label corrections while maintaining interpretability.
- **Strong empirical validation** on both controlled and real-world benchmarks:
  - CIFAR-10: label corruption reduced from 20% to 3.2%, 30% to 8.6%, 40% to 25.9%, yielding a 12.5% absolute accuracy gain at 40% noise (71.2% vs. 58.6% baseline).
  - Fashion-MNIST: corruption reduced from 20% to 2.8%, 30% to 4.7%, 40% to 4.6%, boosting accuracy by 2.2% at 40% noise (87.9% vs. 85.7%).
  - CIFAR-10N: cleaned corruption from 9.01% to 4.4% and raised top-1 accuracy to 88.5%, a 0.26% lead over the next best method.
- **Open-source release** of all code, datasets, and evaluation scripts at `https://hidden-because-anonymity`, enabling easy integration with any downstream classifier. Also, key parameters for training are available in Appendix F.

2 Methodology

We propose a Siamese network that uses contrastive learning and ensemble consensus to detect and correct instance-dependent label noise (Fig. 1). The formal notation is provided in Appendix B.1.

Nested Cross-Validation: Implemented via Algorithm 3, the process first splits the dataset $\mathbb{D}$ into k stratified outer folds. For each fold, $k-1$ folds form the outer training set ($\mathbb{D}_{\text{OT}}$) while the remaining fold serves as validation ($\mathbb{D}_{\text{OV}}$). $\mathbb{D}_{\text{OT}}$ is further divided into m inner folds, with $m-1$ sub-folds ($\mathbb{D}_{\text{IT}}$) training Siamese networks and one sub-fold ($\mathbb{D}_{\text{IV}}$) validating early stopping. Contrastive pairs derive exclusively from $\mathbb{D}_{\text{IT}}$ using Section 2.2's balanced strategy.

Ensemble Noise Detection: For each inner fold, we train a Siamese model and record its predictions on $\mathbb{D}_{\text{OV}}$. We collect these predictions into a matrix $\mathbf{P}$ as described in Algorithm 1. For each sample

x_i in $\mathbb{D}_{\text{OV}}$, we compute the disagreement count $r(x_i) = \sum_{j=1}^{m} \mathbb{1}(\mathbf{P}_{i,j} \neq y_i)$. We flag x_i as noisy if $r(x_i) \geq \tau_{\text{d}}$, leveraging Theorem 2.2 to ensure that noisy labels exhibit higher disagreement rates.

Consensus-Based Correction: Flagged samples are relabeled using Algorithm 2. If at least τ_{r} models agree on a new label, we substitute the consensus label; otherwise, we discard the sample.

Siamese networks, detailed in Section 2.1, are trained on pairs selected based on $\mathbb{D}_{\text{IT}}$, with early stopping employed and monitored through performance metrics derived from $\mathbb{D}_{\text{IV}}$. This nested structure helps maintain model integrity while Theorem 2.3 ensures robustness against individual model errors.

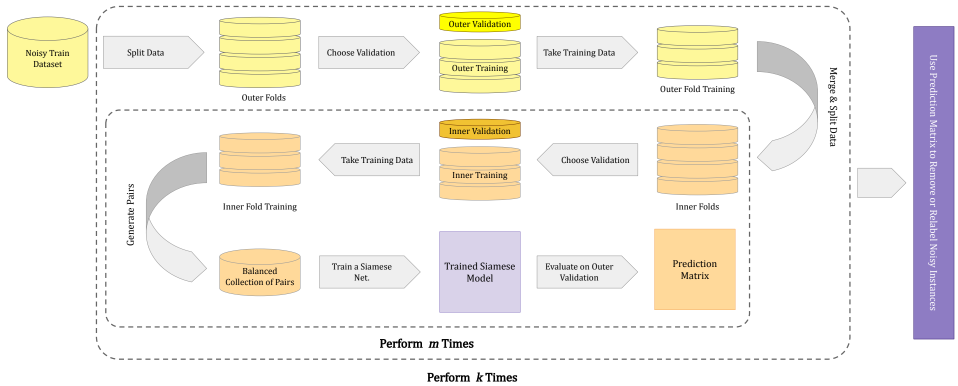


Figure 1: Overview of the proposed method.

2.1 Custom Siamese Network Architecture

We train a Siamese network on the inner training set $\mathbb{D}_{\text{IT}}$ and validate on $\mathbb{D}_{\text{IV}}$ to generate predictions for samples in $\mathbb{D}_{\text{OV}}$. The twin-branch architecture (Fig. 2) builds on the foundational design of [18] and integrates contrastive learning from [19]. It combines contrastive and classification objectives through three key components:

Feature Extractor: A CNN that processes input pairs to extract discriminative features.

Embedding Head: Projects features into a normalized embedding space using a sigmoid layer to constrain values to $[0, 1]$, enabling similarity-based analysis and dimensionality reduction.

Classification Head: Lightweight MLP mapping embeddings to class predictions.

The model is trained via a loss function that combines contrastive and classification objectives:

$$\mathcal{L}_{\text{total}} = \frac{1}{B}\sum_{b=1}^{B}\left[\underbrace{(1-y_s^{(b)})\max(\|h_1^{(b)}-h_2^{(b)}\|_2-\gamma,0)^2+y_s^{(b)}\|h_1^{(b)}-h_2^{(b)}\|_2^2}_{\text{Contrastive Loss}}+\underbrace{\sum_{i=1}^{2}\mathcal{L}_{\text{CE}}(p_i^{(b)},\tilde{y}_i^{(b)})}_{\text{Classification Loss}}\right] \tag{1}$$

where B is the number of pairs, $y_s \in \{0, 1\}$ indicates positive/negative pairs, h_1, h_2 are the twin branch embeddings, γ is the contrastive margin, $\mathcal{L}_{\text{CE}}$ denotes cross-entropy loss, p denotes the logit, and $\tilde{y}$ symbolizes the noisy labels.

2.2 Pair Selection Analysis

The mechanism governing pair selection critically shapes model performance by controlling representation quality and contrastive learning efficacy. Our experiments demonstrate that optimal performance requires balanced positive-to-negative pair ratios. In a balanced c-class dataset under $k \times m$ nested cross-validation, the maximum number of positive pairs is:

$$MPP = \binom{\lfloor \frac{1}{c} \times |\mathbb{D}_{\text{IT}}| \rfloor}{2} \times c = \binom{\lfloor \frac{(m-1)(k-1)}{m \times k \times c} \times n \rfloor}{2} \times c \tag{2}$$

To maintain a balanced training set, we limit the total number of pairs to $2 \times$ MPP, ensuring equal representation of positive and negative pairs. However, our experiments show that effective training

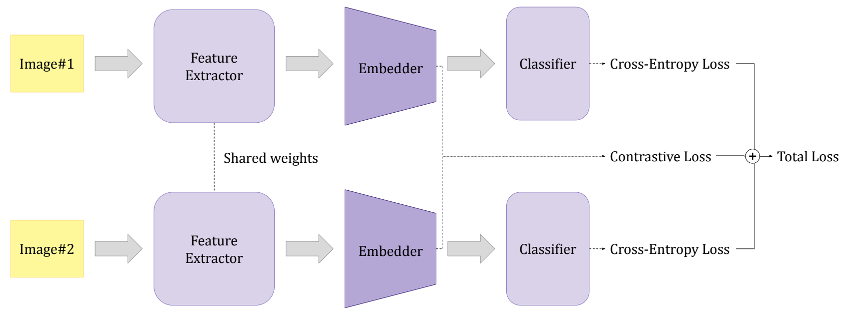


Figure 2: Siamese network with twin branches processing input pairs. Shared weights generate embeddings (h_1, h_2) and logits (p_1, p_2) for joint contrastive-classification learning.

Algorithm 1 Collect Predictions by Nested Cross-Validation

Inputs: Dataset $\mathbb{D} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{n}$, Number of outer folds k, Number of inner folds m
Output: Prediction matrix $\mathbf{P} \in [0,1]^{n \times m}$ where $\mathbf{P}_{i,j}$ is the prediction of the j-th model on the i-th sample

function COLLECTPREDICTIONS($\mathbb{D}, k, m$)
  Initialize prediction matrix $\mathbf{P} \in [0,1]^{n \times m}$
  for outer_fold = 1 **to** k **do**
    $\mathbb{D}_{\text{OT}}, \mathbb{D}_{\text{OV}} \leftarrow$ **StratifiedSplit**($\mathbb{D}$, outer_fold, k) ▷ Outer split (Appendix C)
    Initialize ensemble models $\{f_1, f_2, \ldots, f_m\}$
    for inner_fold = 1 **to** m **do**
      $\mathbb{D}_{\text{IT}}, \mathbb{D}_{\text{IV}} \leftarrow$ **StratifiedSplit**($\mathbb{D}_{\text{OT}}$, inner_fold, m) ▷ Inner split (Appendix C)
      $f_{\text{inner\_fold}} \leftarrow$ **TrainModel**($\mathbb{D}_{\text{IT}}, \mathbb{D}_{\text{IV}}$) ▷ Train siamese model with early stopping
      for each sample $\mathbf{x}_i$ in $\mathbb{D}_{\text{OV}}$ **do**
        $\mathbf{P}_{\text{inner\_fold},i} = f_{\text{inner\_fold}}(\mathbf{x}_i)$ ▷ Ensemble predictions
      end for
    end for
  end for
  return P
end function

Algorithm 2 Noise Detection and Relabeling

Inputs: Prediction matrix $\mathbf{P} \in [0,1]^{n \times m}$, Dataset $\mathbb{D} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{n}$, Detection threshold τ_{d}, Relabeling threshold τ_{r}
Output: Cleaned dataset $\mathbb{D}_{\text{clean}} = \{(\mathbf{x}_i^c, y_i^c)\}_{i=1}^{n_c}$

function DETECTANDRELABEL($\mathbf{P}, \mathbb{D}, \tau_{\text{d}}, \tau_{\text{r}}$)
  Initialize clean dataset $\mathbb{D}_{\text{clean}} = \{\}$
  for each sample $i = 1$ **to** n **do**
    $r_i = \sum_{j=1}^{m} \mathbb{1}(\mathbf{P}_{i,j} \neq y_i)$ ▷ Calculate error rate
    $l = \text{mode}(\mathbf{P}_{i,:})$ ▷ Find most common prediction
    if $r_i < \tau_{\text{d}}$ **then**
      $\mathbb{D}_{\text{clean}} \leftarrow \mathbb{D}_{\text{clean}} \cup \{(\mathbf{x}_i, \tilde{y}_i)\}$ ▷ Detected as clean
    else if count$(l) \geq \tau_{\text{r}}$ **then**
      $\mathbb{D}_{\text{clean}} \leftarrow \mathbb{D}_{\text{clean}} \cup \{(\mathbf{x}_i, l)\}$ ▷ Relabeled as clean
    end if
  end for
  return $\mathbb{D}_{\text{clean}}$
end function

can be achieved with significantly fewer pairs; for instance, using only 0.12% of the maximum possible pairs on CIFAR-10 yielded competitive performance, suggesting that the model can learn robust representations even with a small subset of selected pairs.

Label noise can affect the pair selection process, necessitating an analysis of the probability of selecting valid training pairs as a function of the noise rate r and the number of classes c. Detailed derivations are provided in Appendix D. To ensure robust performance, we require that this probability exceeds a predefined threshold α. This criterion yields an upper bound r^* on the acceptable noise rate, dependent on α and c. Comprehensive derivations and closed-form solutions are outlined in Appendix D.3, and Fig. 3 illustrates typical operational ranges for various α values.

2.3 Noise Detection Analysis

Let $\mathbb{C}$ and $\mathbb{N}$ denote the sets of clean and noisy samples, and L and MP the latent-quality and performance indicators. Theorem 2.2 shows that noisy samples misclassify more often, and Appendix A shows noise degrades embeddings, raising $\mathbb{P}(\neg L \mid \mathbb{N})$.

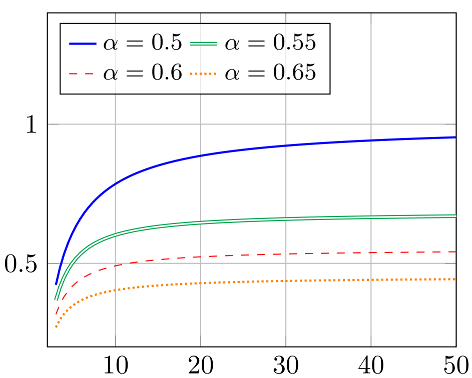


Figure 3: r^* vs. number of classes

Assumption 2.1 (Independence Conditions)**.** *We assume (i) latent–performance independence, $\mathbb{P}(L \cap MP) = \mathbb{P}(L)\mathbb{P}(MP)$, and (ii) performance–noise independence, $\mathbb{P}(MP \mid \mathbb{N}) = \mathbb{P}(MP \mid \mathbb{C}) = \mathbb{P}(MP)$.*

Theorem 2.2 (Noise Misclassification Bias)**.** *Let $G(x, \tilde{y}) = \mathbb{1}(\arg\max f_\theta(x) = \tilde{y})$ denote the misclassification indicator. Under Assumption 2.1, for any noise rate $r > 0$ and number of classes $c \geq 3$,*

$$\mathbb{P}(G = 0 \mid \mathbb{N}) > \mathbb{P}(G = 0 \mid \mathbb{C}).$$

Proof. The proof follows from analyzing the probability decomposition of correct classification under our decision tree framework (see Appendix E.1 for the complete mathematical derivation). □

Our experimental evaluation in Section 3 empirically validates Theorem 2.2, demonstrating that noisy samples indeed exhibit higher misclassification rates than clean samples. Building on this result, we further consider the effect of aggregating predictions from multiple models. Moreover, by treating the normalized disagreement score $\frac{1}{m}\sum_{j=1}^{m} \mathbb{1}\big(f_j(x) \neq \tilde{y}\big)$ as a continuous noise-confidence value, we not only rank samples by their likelihood of corruption but, as shown in the reliability diagrams of Appendix I, provides a useful approximation of the true noise rate.

Theorem 2.3 (False-Positive Decay Under Ensemble Prediction)**.** *Consider an ensemble of m trained Siamese models $f_1, \dots, f_m$. For a data point $s = (x, y)$, we classify it as noisy if at least τ_d models misclassify it. Under the clean data condition ($s \in \mathbb{C}$), define the total misclassification count $X = \sum_{i=1}^{m} X_i$, where each $X_i \in \{0, 1\}$ is an indicator variable for misclassification by the i-th model. Let $\mathbb{E}[X_i] = p_C$ be the expected misclassification rate on clean data, and assume the model errors have bounded variance $\sigma^2 = \mathbb{E}[(X_i - p_C)^2]$. Then, the probability of false-positive detection decays exponentially with the number of models:*

$$\mathbb{P}(X \geq \tau_d \mid \mathbb{C}) \leq \exp\left(-\frac{m\epsilon^2}{2\sigma^2 + \frac{2}{3}\epsilon}\right), \tag{3}$$

where $\epsilon = \frac{\tau_d}{m} - p_C > 0$ represents the margin between the threshold rate and the expected misclassification rate.

Proof. A concise proof of this result employs Bernstein's inequality to accommodate correlated model errors; the detailed proof is available in Appendix E.3. □

Based on Theorem 2.3, we establish $\tau_\mathrm{d} \geq 0.6m$, which guarantees an exponential decay in false-positive probability with the increase of m. Specifically, this configuration ensures low false-positive rate when employing $m \geq 10$ models, assuming a minimum model accuracy of 75% on clean data (corresponding to $p_C = 0.25$). Appendix E.4 derives these bounds mathematically, while Section 3.4 and Appendix G empirically demonstrates our method's robust sensitivity trade-off.

Trade-off between m and k Increasing the number of models m reduces the available data for the inner validation set ($\mathbb{D}_{\text{IV}}$), necessitating a larger number of folds k to maintain statistical robustness.

2.4 Quantifying Label Correction Quality

We evaluate label corrections using a *relabeling score*. For each data point, we assign +2 points for correctly relabeling noisy samples, -2 for incorrectly relabeling clean samples, +1 for properly removing noisy samples, -1 for wrongly removing clean samples, and 0 for incorrectly relabeling noisy samples. We compute the overall relabeling score by summing these individual scores and dividing by the number of relabeled samples, facilitating comparison across datasets. This scoring framework is illustrated in Fig. 4. Our evaluation focuses on aggregate performance across samples identified as noisy by our method, with class-specific analysis reserved for future work. Further, we define *relabeling accuracy* as the ratio of correctly relabeled samples and *relabeling count* as the total number of relabeled samples; these metrics are detailed in Appendix G.1.2.

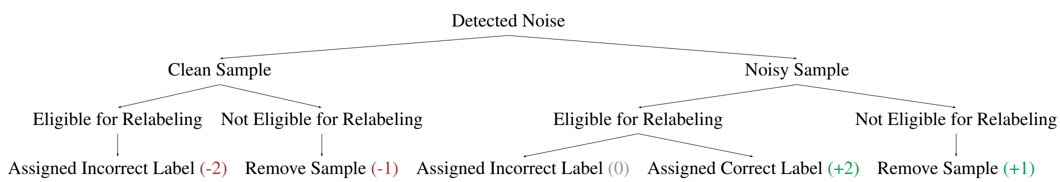


Figure 4: Scoring system for label corrections: green/red = positive/negative impact.

3 Experiments

We evaluate noise detection performance on synthetic and real-world benchmarks, assess relabeling efficacy via custom metrics, measure downstream classification accuracy after cleaning, and employ nested cross-validation with $m = k = 10$ folds.

3.1 Implementation Details

We conducted experiments on a workstation with an AMD Ryzen 9 5950X CPU, 32 GB of system RAM, and an AMD Radeon 6900 XT GPU (16 GB VRAM) using PyTorch with ROCm acceleration. Due to inefficiencies on non-CUDA hardware, VRAM never exceeded 50%, leading to longer training times. We used a batch size of 2048; each CIFAR-10 noise ratio experiment required approximately 100 hours, while Fashion-MNIST runs completed in approximately 24 hours owing to its lower resolution and computational complexity. Training durations remained modest in epochs: each CIFAR-10N inner fold converged in under 50 epochs, Fashion-MNIST experiments converged in around 30 epochs, and CIFAR-10 runs converged by approximately 70 epochs.

3.2 Performance Metrics

Following [20], we treat noise detection as a binary classification task with four key metrics. We define Noise Precision as the proportion of true noisy samples among those flagged as noisy and Noise Recall as the proportion of noisy samples we detect. Noise F1 is the harmonic mean of precision and recall, and Noise Accuracy measures overall detection accuracy (full formulas in Appendix B.2). We also use the relabeling score (Sec. 2.4) to evaluate correction quality. Finally, we assess classification accuracy on the cleaned data using standard metrics.

3.3 Datasets

Experiments cover: (1) Synthetic noise (20–40% in CIFAR-10[21]/Fashion-MNIST[22] using [23]), (2) Real noise (CIFAR-10N's 9.01% annotation errors [24]).

3.4 Noise Detection Results

Figure 5 breaks down, for each dataset and noise level, the counts of noisy versus clean samples before and after applying our cleaning procedure. For example, on CIFAR-10 with 20% injected noise, the noisy-sample count drops from $10,208$ to $1,333$ ($\approx 3.2\%$); similarly, on Fashion-MNIST with 40% noise it falls from $24,189$ to $1,810$ ($\approx 4.6\%$). This figure shows that our approach not only lowers the number of noisy samples but also (in all but two cases) raises the count of clean samples, emphasizing its overall effectiveness.

Since the bar chart shows only counts, it does not distinguish correctly-removed noisy labels (true-positives) from mistakenly-removed clean labels (false-positives). Table 1 fills this gap by reporting Noise Accuracy, Precision, Recall, and F1-Score, along with the Relabeling Score. A high Noise Precision (e.g. 0.798 on 20% Fashion-MNIST) means we remove few clean labels, whereas Noise Recall (e.g. 0.915 on 20% CIFAR-10) shows our ability to catch injected noise. The table also lists the optimal detection threshold τ_{d} and relabeling threshold τ_{r} chosen via grid search; the full-tuning results appear in Appendix G.1. When facing real-world datasets where ground-truth clean labels are unavailable, we propose a practical threshold selection methodology in Appendix H. By combining raw count reductions (Fig. 5) with precision/recall metrics (Table 1), we demonstrate both the extent of noise removal and the preservation of clean data. Researchers can use Table 1 as a reference for comparing new noise-detection methods against our optimal thresholds and performance metrics.

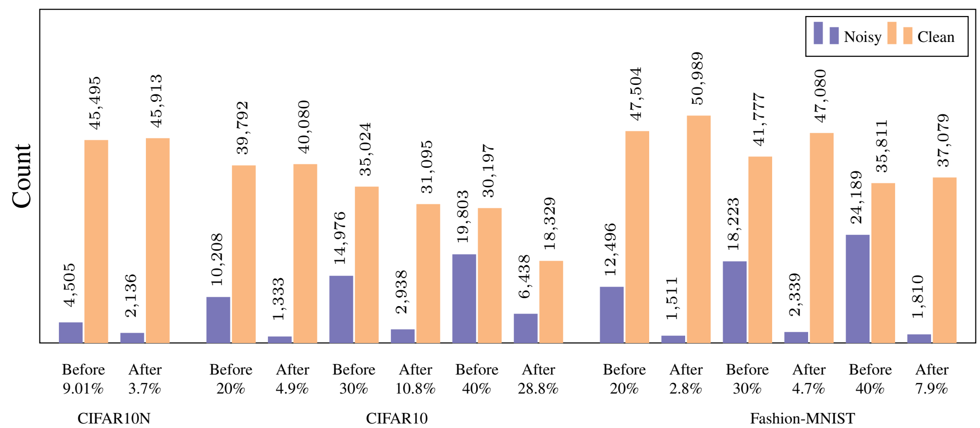


Figure 5: Comparison of noisy and clean samples before and after identifying noise.

Table 1: Optimal Configurations and Evaluation Metrics of the Noise Detection

		CIFAR-10			Fashion-MNIST		
Noise Ratio	CIFAR-10N	20%	30%	40%	20%	30%	40%
τ_{d} (Detection Threshold)	10	8	7	6	9	9	7
τ_{r} (Relabeling Threshold)	10	10	10	10	10	10	10
Noise Accuracy	**0.9451**	0.8999	0.8069	0.6115	**0.9358**	0.9216	0.8787
Noise Precision	**0.6999**	0.6932	0.6386	0.5069	0.7981	**0.8502**	0.7958
Noise Recall	0.6848	**0.9149**	0.8188	0.7021	0.9258	0.9003	**0.9402**
Noise F1-Score	0.6922	**0.7887**	0.7176	0.5887	0.8572	**0.8746**	0.8620
Relabeling Score	1.13	1.87	**2.51**	0.68	2.07	2.43	**3.17**

3.5 Classification Performance on Noise-Corrected Datasets

We assess the performance of models trained on our refined data partitions across both synthetic and real-world noise conditions. The outcomes are summarized in Table 2 and Table 3.

Synthetic noise (20–40% IDN) Our preprocessing yields the largest absolute and relative gains at all noise levels:

- **CIFAR-10 (IDN-20/30/40%):** Accuracy improves from 76.05% to 88.17% at 20% noise (+12.12 pp), from 72.28% to 83.63% at 30% noise (+11.35 pp), and from 58.62% to 71.16% at 40% noise (+12.54 pp). Not only are these increases the biggest margins against the strongest prior (PTD-R-V [23]), but our standard deviations ($\pm$ 0.28–1.38 pp) remain low, indicating stable performance across trials.
- **Fashion-MNIST (IDN-20/30/40 %):** At milder noise (20%), we edge past the best baseline (PTD-R-V [23]) by 0.23 pp (91.31% vs. 91.08%), while at higher noise (40%) the lead grows to 2.23 pp (87.92% vs. 85.69%), demonstrating robust correction even when nearly half the labels are corrupted.

These results reveal two critical trends: (i) *Noise-Robust Consistency:* On CIFAR-10, our method improves previous results by approximately 12 pp regardless of noise rate, demonstrating the reliability of ensemble-driven detection even as corruption intensifies. (ii) *Dataset Complexity Adaptation:* Fashion-MNIST, being simpler, sees smaller absolute gains at low noise but still benefits substantially at high noise, indicating the relabeling mechanism dynamically adjusts to data complexity.

Real-world noise (9.01% CIFAR-10N) On CIFAR-10N's 9.01% natural noise, cleaning reduces corruption to 3.7% and boosts ResNet-34 accuracy to 88.51% ($\pm\,0.50$ pp), a 0.26 pp improvement over the best reported result (88.25%)[25], despite using fewer ensemble members and folds (m and k) due to hardware constraints (Sec. 3.1). This smaller yet significant gain reflects the lower initial noise rate, but confirms that our framework generalizes beyond synthetic settings.

Our results reveal three main trends. First, the roughly 12pp improvement on CIFAR-10 is virtually unchanged across the 20–40% noise range, demonstrating that our approach scales robustly with noise severity. Second, the low run-to-run variance (standard deviation $\leq$ 1.4pp) highlights the reliability of the produces cleaned datasets. Third, simpler image domains such as Fashion-MNIST benefit most under extreme noise—exhibiting larger relative gains—indicating that the relabeling threshold dynamically adapts to dataset complexity. Further ablation experiments, including dataset analysis that illustrates the contributions of its key components under label uncertainty (Appendix F).

Table 2: CIFAR-10 and Fashion-MNIST Classification Accuracy

	CIFAR-10			Fashion-MNIST		
Method	IDN-20%	IDN-30%	IDN-40%	IDN-20%	IDN-30%	IDN-40%
CE	68.21 ± 0.72	60.48 ± 0.62	49.84 ± 1.27	88.38 ± 0.42	84.22 ± 0.35	68.86 ± 0.78
Decoupling[26]	70.01 ± 0.66	63.05 ± 0.65	44.27 ± 1.91	86.50 ± 0.35	85.33 ± 0.47	78.54 ± 0.53
MentorNet[6]	70.56 ± 0.34	65.42 ± 0.79	46.22 ± 0.98	87.02 ± 0.41	86.02 ± 0.82	80.12 ± 0.76
Co-teaching[27]	72.99 ± 0.45	67.22 ± 0.64	49.25 ± 1.77	87.89 ± 0.41	86.88 ± 0.32	82.78 ± 0.95
Co-teaching+[28]	71.07 ± 0.77	64.77 ± 0.58	47.73 ± 2.32	89.77 ± 0.45	88.52 ± 0.45	83.57 ± 1.77
Joint[29]	73.89 ± 0.34	69.03 ± 0.79	54.75 ± 5.98	56.83 ± 0.45	51.27 ± 0.67	44.24 ± 0.78
DMI[30]	69.89 ± 0.33	61.88 ± 0.64	51.23 ± 1.18	90.33 ± 0.21	84.81 ± 0.44	69.01 ± 1.87
Forward[31]	68.99 ± 0.62	60.21 ± 0.75	47.17 ± 2.96	88.61 ± 0.43	84.27 ± 0.46	70.25 ± 1.28
Reweight[32]	68.42 ± 0.75	62.58 ± 0.46	50.12 ± 0.96	89.70 ± 0.35	87.04 ± 0.35	80.29 ± 0.89
T-Revision[33]	69.32 ± 0.64	64.09 ± 0.37	50.38 ± 0.87	90.68 ± 0.66	89.46 ± 0.45	84.01 ± 1.24
PTD-F[23]	73.45 ± 0.62	65.25 ± 0.84	49.88 ± 0.85	90.01 ± 0.31	87.42 ± 0.65	83.89 ± 0.49
PTD-R[23]	75.02 ± 0.73	71.86 ± 0.42	56.15 ± 0.45	90.03 ± 0.32	87.68 ± 0.42	84.03 ± 0.52
PTD-F-V[23]	73.88 ± 0.61	69.01 ± 0.47	50.43 ± 0.62	90.79 ± 0.29	89.33 ± 0.33	85.32 ± 0.36
PTD-R-V[23]	76.05 ± 0.53	72.28 ± 0.49	58.62 ± 0.88	91.08 ± 0.46	89.66 ± 0.43	85.69 ± 0.77
Ours	**88.17 ± 0.28**	**83.63 ± 0.34**	**71.16 ± 1.38**	**91.31 ± 0.49**	**90.47 ± 0.13**	**87.92 ± 0.66**

4 Discussion

Our framework has two principal limitations. First, computational overhead stems from the nested cross-validation scheme (Sec. 2), which requires training $m \times k$ models. This overhead increases substantially for larger datasets. For example, applying our method to CIFAR-100, which comprises 100 classes, would necessitate increasing both m and k (e.g., to 20 folds each) to preserve detection power; consequently, computational cost would be at least four times that incurred on CIFAR-10. In our current study, hardware constraints precluded experiments on CIFAR-100; future work could explore optimizations such as grouped class sampling or distributed training to alleviate this burden.

Table 3: CIFAR-10N Classification Accuracy

Method	CIFAR-10N	Method	CIFAR-10N	Method	CIFAR-10N
Co-teaching+[28]	82.31 ± 0.89	DoctorNet[34]	84.52 ± 0.69	CoDis[35]	87.23 ± 0.45
BLTM[36]	82.62 ± 0.17	Max-MIG[37]	85.12 ± 0.36	GCNet(F)[38]	87.70 ± 0.51
CrowdLayer[39]	82.84 ± 0.24	MBEM[40]	85.49 ± 0.43	GCNet(W)[38]	87.84 ± 0.21
CoNAL[41]	83.01 ± 0.21	Co-teaching[27]	85.90 ± 0.50	BayesianIDNT[42]	88.19 ± 0.47
CE(EM)[43]	83.14 ± 0.80	LogitClip[44]	86.37 ± 0.43	AdaptCDRP[25]	88.25 ± 0.34
TraceReg[45]	83.16 ± 0.24	CCC[46]	86.45 ± 0.53		
Ours					**88.51 ± 0.50**

Second, consensus relabeling can reinforce class imbalances or mislabel ambiguous samples. Our experiments cover up to ten classes (CIFAR-10, Fashion-MNIST, CIFAR-10N), but scaling to high-cardinality benchmarks (e.g., CIFAR-100, ImageNet) will require more efficient pair selection strategies—such as grouping similar classes or calibrating thresholds per class—to maintain accuracy without prohibitive computational cost.

From an ethical perspective, our method carries three potential risks: (i) *bias amplification*, whereby minority classes become underrepresented after relabeling; (ii) *dataset distortion*, in which valid edge cases or outliers may be overwritten; and (iii) *ambiguity exclusion*, when confidently relabeling uncertain samples reduces dataset diversity. We propose mitigating these risks through fairness-aware thresholds, diversity-preserving sampling, and the release of audit logs to ensure transparency.

5 Future Directions

Future research avenues include iterative refinement through repeated relabeling cycles, enabling the model to progressively de-noise more challenging instances (guided by our reliability analysis of ensemble disagreement in Appendix I). Extending the framework to multi-modal data (e.g., image-text or audio-visual) could broaden its applicability. To further mitigate relabeling bias, fairness-aware consensus mechanisms, such as class-conditional thresholds, warrant in-depth exploration. On the technical side, promising directions include analyzing the impact of false positives on downstream robustness (Appendix J) and adapting the detection pipeline to identify adversarially perturbed or out-of-distribution samples.

6 Conclusion

In this paper, we proposed a novel framework for robust learning under instance-dependent label noise by leveraging a Siamese network trained with contrastive learning to extract visually coherent representations. Our method reliably distinguishes noisy labels by analyzing embedding variability across perturbations and is supported by theoretical guarantees and extensive empirical validation. Experiments on both synthetic and real-world benchmarks demonstrate significant improvements in noise detection (up to 80.2 percentage points) and downstream classification accuracy (up to 12.5 percentage points F1-score gain) compared to prior methods. The approach is model-agnostic, easy to integrate, and complemented by interpretable uncertainty estimates. These contributions establish a practical framework for learning with corrupted labels.

References

[1] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels, 2022.

[2] Chris Dongjoo Kim, Sangwoo Moon, Jihwan Moon, Dongyeon Woo, and Gunhee Kim. Sample selection via contrastive fragmentation for noisy label regression. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 127561–127609. Curran Associates, Inc., 2024.

[3] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 2022.

[4] Yilun Zhu, Jianxin Zhang, Aditya Gangrade, and Clay Scott. Label noise: Ignorance is bliss. *Advances in Neural Information Processing Systems*, 37:116575–116616, 2024.

[5] Jongmin Shin, Jonghyeon Won, Hyun-Suk Lee, and Jang-Won Lee. A review on label cleaning techniques for learning with noisy labels. *ICT Express*, 2024.

[6] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2304–2313. PMLR, 10–15 Jul 2018.

[7] Mingyu Xu, Zheng Lian, Lei Feng, Bin Liu, and Jianhua Tao. Alim: adjusting label importance mechanism for noisy partial label learning. *Advances in Neural Information Processing Systems*, 36:38668–38684, 2023.

[8] Mingcai Chen, Hao Cheng, Yuntao Du, Ming Xu, Wenyu Jiang, and Chongjun Wang. Two wrongs don't make a right: Combating confirmation bias in learning with label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14765–14773, 2023.

[9] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11053–11061, 2021.

[10] Wanxing Chang, Ye Shi, and Jingya Wang. Csot: Curriculum and structure-aware optimal transport for learning with noisy labels. *Advances in Neural Information Processing Systems*, 36:8528–8541, 2023.

[11] Yazhou Yao, Zeren Sun, Chuanyi Zhang, Fumin Shen, Qi Wu, Jian Zhang, and Zhenmin Tang. Jo-src: A contrastive approach for combating noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5192–5201, 2021.

[12] Qi Wei, Lei Feng, Haoliang Sun, Ren Wang, Chenhui Guo, and Yilong Yin. Fine-grained classification with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11651–11660, 2023.

[13] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International conference on machine learning*, pages 6226–6236. PMLR, 2020.

[14] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020.

[15] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[16] Zhen Wang, Guosheng Hu, and Qinghua Hu. Training noise-robust deep neural networks via meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4524–4533, 2020.

[17] Xin Zhang, Zixuan Liu, Kaiwen Xiao, Tian Shen, Junzhou Huang, Wei Yang, Dimitris Samaras, and Xiao Han. Codim: Learning with noisy labels via contrastive semi-supervised learning, 2021.

[18] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1993.

[19] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[20] Zhaowei Zhu, Zihao Dong, and Yang Liu. Detecting corrupted labels without training a model to predict, 2022.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

[22] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[23] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7597–7610. Curran Associates, Inc., 2020.

[24] 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*, 2022.

[25] Hui Guo, Grace Y. Yi, and Boyu Wang. Learning from noisy labels via conditional distributionally robust optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 82627–82672. Curran Associates, Inc., 2024.

[26] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update". In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[27] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[28] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7164–7173. PMLR, 09–15 Jun 2019.

[29] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

[30] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[31] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2233–2241, 2017.

[32] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2016.

[33] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[34] Melody Guan, Varun Gulshan, Andrew Dai, and Geoffrey Hinton. Who said what: Modeling individual labelers improves classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[35] Xiaobo Xia, Bo Han, Yibing Zhan, Jun Yu, Mingming Gong, Chen Gong, and Tongliang Liu. Combating noisy labels with sample selection by mining high-discrepancy examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1833–1843, October 2023.

[36] Shuo Yang, Erkun Yang, Bo Han, Yang Liu, Min Xu, Gang Niu, and Tongliang Liu. Estimating instance-dependent bayes-label transition matrix using a deep neural network, 2022.

[37] Peng Cao, Yilun Xu, Yuqing Kong, and Yizhou Wang. Max-mig: an information theoretic approach for joint learning from crowds, 2019.

[38] Shahana Ibrahim, Tri Nguyen, and Xiao Fu. Deep learning from crowdsourced labels: Coupled cross-entropy minimization, identifiability, and regularization. In *The Eleventh International Conference on Learning Representations*, 2023.

[39] Filipe Rodrigues and Francisco Pereira. Deep learning from crowds. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[40] Ashish Khetan, Zachary C. Lipton, and Anima Anandkumar. Learning from noisy singly-labeled data, 2018.

[41] Zhendong Chu, Jing Ma, and Hongning Wang. Learning from crowds by modeling common confusions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):5832–5840, May 2021.

[42] Hui Guo, Boyu Wang, and Grace Yi. Label correction of crowdsourced noisy annotations with an instance-dependent noise transition model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[43] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979.

[44] Hongxin Wei, Huiping Zhuang, Renchunzi Xie, Lei Feng, Gang Niu, Bo An, and Yixuan Li. Mitigating memorization of noisy labels by clipping the model prediction. In *International Conference on Machine Learning*. PMLR, 2023.

[45] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C. Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11236–11245, 2019.

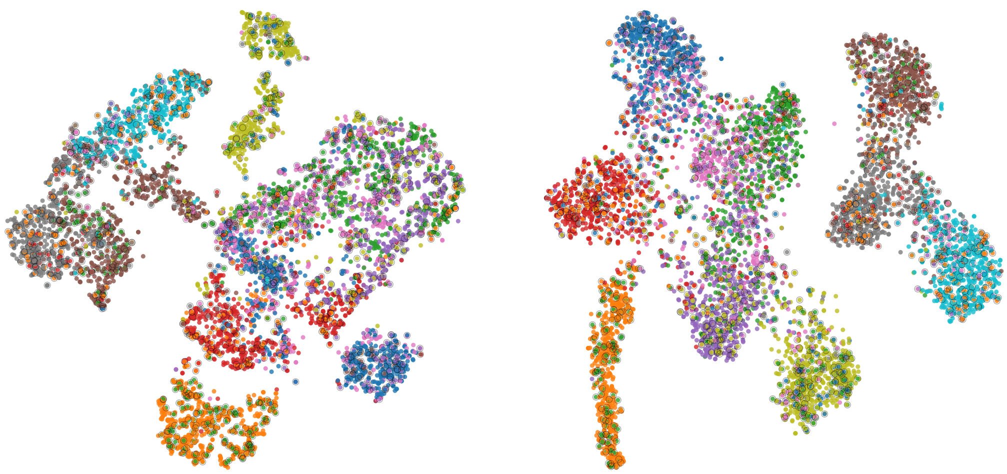
[46] Hansong Zhang, Shikun Li, Dan Zeng, Chenggang Yan, and Shiming Ge. Coupled confusion correction: Learning from crowds with sparse annotations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16732–16740, Mar. 2024.

[47] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019.

[48] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

A Visual Motivation: How Label Noise Affects Embedding Structure

To illustrate the impact of label noise on representation quality, Figure 6 compares t-SNE embeddings of the Fashion-MNIST dataset with 20% instance-dependent noise to the raw feature space. In the raw feature space (Figure 6a), the classes heavily overlap, and the noisy samples (circled) are intermingled with clean clusters, making them indistinguishable. However, after applying Siamese contrastive learning (Figure 6b), the clean samples form tight, well-separated clusters, while the noisy points are positioned on the periphery or between the clusters. This behavior is precisely what we leverage for noise detection. This visual evidence motivated our ensemble Siamese design. By enforcing consistency among clean examples and amplifying the disagreements on mislabeled data, we can reliably identify and correct noisy labels, even in the presence of severe instance-dependent corruption.



(a) Embeddings from the raw feature space show overlapping class clusters, with noisy samples (circled) indistinguishable from clean ones.

(b) Embeddings after Siamese training: clean samples form compact, well-separated clusters, while noisy samples shift to the periphery or inter-class regions.

Figure 6: t-SNE visualization of Fashion-MNIST embeddings with 20% instance-dependent noise, where noisy samples are circled in black.

B Notations and Definitions

B.1 Notations

Table 4 summarizes the notation used throughout this paper.

B.2 Definitions

Let N be the total number of samples, y_n the true label, $\tilde{y}_n$ the observed (noisy) label, $v_n \in \{0, 1\}$ the model's noise flag (with $v_n = 1$ indicating a predicted noisy sample), and $\mathbb{1}(\cdot)$ the indicator function. We then define the following metrics for evaluating noise detection:

- **Noise Precision**: Correctly flagged noisy samples among all flagged

$$\frac{\sum \mathbb{1}(v_n{=}1 \wedge \tilde{y}_n{\neq}y_n)}{\sum \mathbb{1}(v_n{=}1)} \tag{4}$$

- **Noise Recall**: True noisy samples detected

$$\frac{\sum \mathbb{1}(v_n{=}1 \wedge \tilde{y}_n{\neq}y_n)}{\sum \mathbb{1}(\tilde{y}_n{\neq}y_n)} \tag{5}$$

Table 4: Notation Summary

Symbol	Description	Symbol	Description
r	Noise rate	c	Number of classes
k	Number of outer folds	m	Number of inner folds
y	True label	$\tilde{y}$	Noisy label
$\mathbb{D}$	Dataset	$\mathbb{D}_{clean}$	Cleaned Dataset
$\mathbb{D}_{train}$	Train set	$\mathbb{D}_{validation}$	Validation set
$\mathbb{D}_{\text{OT}}$	Outer training set	$\mathbb{D}_{\text{OV}}$	Outer validation set
$\mathbb{D}_{\text{IT}}$	Inner training set	$\mathbb{D}_{\text{IV}}$	Inner validation set
$\mathbb{C}$	Clean sample subset	$\mathbb{N}$	Noisy sample subset
$\mathbb{1}$	Indicator function	$\mathbb{P}$	Probability
τ_{d}	Detection threshold	τ_{r}	Relabeling threshold
τ_{d}^*	Optimal detection threshold	τ_{r}^*	Optimal relabeling threshold
f_θ	Model function	$f_j(x)$	Output of the j-th model
$\mathbf{P}$	Prediction matrix	$r(x)$	Disagreement count of a sample
α	Min. pair selection success probability	r^*	Max. permissible noise rate
L	Latent representation quality	MP	Model performance
G	Classification Result		

- **Noise F1-Score**: Harmonic mean of noise precision/recall

$$2 \cdot \frac{\text{Noise Precision} \cdot \text{Noise Recall}}{\text{Noise Precision} + \text{Noise Recall}} \tag{6}$$

- **Noise Accuracy**: Overall correct decisions

$$\frac{\sum \mathbb{1}(v_n{=}1 \wedge \tilde{y}_n{\neq}y_n) + \sum \mathbb{1}(v_n{=}0 \wedge \tilde{y}_n{=}y_n)}{N} \tag{7}$$

C Algorithm Details of Stratified Splitting

Here we provide the algorithm for stratified split used in Algorithm 1.

Algorithm 3 Stratified Split Function

Inputs: Dataset $\mathbb{D}$, fold number, total folds k
Outputs: Training set $\mathbb{D}_{train}$, validation set $\mathbb{D}_{validation}$

```
function StratifiedSplit(D, fold, k)
    Group samples by class labels
    for each class c do
        n_c ← |D_c|                                   ▷ Number of samples in class c
        fold_size_c ← ⌊n_c/k⌋                         ▷ Size of each fold for class c
        start_c ← (fold − 1) × fold_size_c            ▷ Start index for validation
        end_c ← start_c + fold_size_c                 ▷ End index for validation
        Add samples D_c[start_c : end_c] to D_validation
        Add remaining samples to D_train
    end for
    return D_train, D_validation
end function
```

D Impact of Label Noise on Pair Selection for Contrastive Learning

D.1 Interpreting the Pair Selection Tree under Label Noise

We formalize the impact of label noise on pair selection using a probabilistic decision tree (Fig. 7). Let:

- $r \in [0, 1]$ denotes the noise rate (probability of a sample's label being corrupted),
- c denotes the number of classes,
- A **positive pair** contains samples intended to be from the same class,
- A **negative pair** contains samples intended to be from different classes.

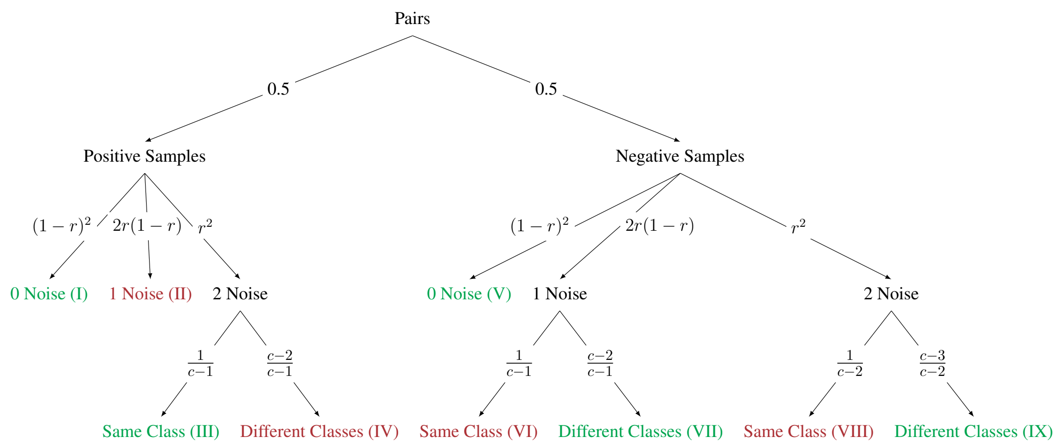


Figure 7: **Pair Selection Tree Under Label Noise**. Green/red leaves denote favorable/unfavorable pairs. Edge probabilities depend on noise rate r and class count c.

We explicitly categorize pairs into two groups based on their *intended training signal* (observed labels), then analyze their validity under noise:

- **Positive Pairs (Leaves I–IV)**: Pairs labeled as belonging to the same class.
- **Negative Pairs (Leaves V–IX)**: Pairs labeled as belonging to different classes.

A leaf is considered valid only if its observed label (same vs. different class) matches the samples' true class relationship, as detailed below:

Positive Pairs (Leaves I–IV)

I. **Valid Positive**: Both clean, same true class. (Favorable)
II. **Corrupted Positive**: One noisy label creates mismatch. (Unfavorable)
III. **Lucky Positive**: Both corrupted to the same (wrong) class. (Favorable)
IV. **Collapsed Positive**: Both corrupted to different classes. (Unfavorable)

Negative Pairs (Leaves V–IX)

V. **Valid Negative**: Both clean, different classes. (Favorable)
VI. **False Negative**: One corrupted label matches true class. (Unfavorable)
VII. **Honest Negative**: One corrupted label preserves dissimilarity. (Favorable)
VIII. **Corrupted Negative**: Both corrupted to same class. (Unfavorable)
IX. **Resilient Negative**: Both corrupted but preserve dissimilarity. (Favorable)

Key Insight: Favorable leaves preserve true class relationships despite noise, while unfavorable ones introduce spurious mismatches or false agreements.

D.2 Deriving Probabilities for Valid and Invalid Pairings

Let r be the corruption probability and $c \geq 3$ the number of classes in a balanced dataset. We derive probabilities for good/poor latent representations by summing contributions from favorable/unfavorable leaves.

Probability of Good Latent Representation:

$$\mathbb{P}(L) = \sum_{\text{favorable leaves}} \mathbb{P}(\text{leaf}) = \underbrace{0.5 \times (1-r)^2}_{\text{Leaf 1}} + \underbrace{0.5 \times r^2 \times \frac{1}{c-1}}_{\text{Leaf 3}} + \underbrace{0.5 \times (1-r)^2}_{\text{Leaf 5}} \tag{8}$$

$$+ \underbrace{0.5 \times 2r(1-r) \times \frac{c-2}{c-1}}_{\text{Leaf 7}} + \underbrace{0.5 \times r^2 \times \frac{c-3}{c-2}}_{\text{Leaf 9}} \tag{9}$$

$$= (1-r)^2 + \frac{r^2}{2(c-1)} + \frac{r(1-r)(c-2)}{c-1} + \frac{r^2(c-3)}{2(c-2)} \quad \text{(Combine terms)} \tag{10}$$

$$= \frac{(c^2-c-3)r^2 + 2(2c-c^2)r + 2(c^2-3c+2)}{2(c-2)(c-1)}. \tag{11}$$

Probability of Poor Latent Representation:

$$\mathbb{P}(\neg L) = \sum_{\text{unfavorable leaves}} \mathbb{P}(\text{leaf}) = \underbrace{0.5 \times 2r(1-r)}_{\text{Leaf 2}} + \underbrace{0.5 \times r^2 \times \frac{c-2}{c-1}}_{\text{Leaf 4}} \tag{12}$$

$$+ \underbrace{0.5 \times 2r(1-r) \times \frac{1}{c-1}}_{\text{Leaf 6}} + \underbrace{0.5 \times r^2 \times \frac{1}{c-2}}_{\text{Leaf 8}} \tag{13}$$

$$= r(1-r) + \frac{r^2(c-2)}{2(c-1)} + \frac{r(1-r)}{c-1} + \frac{r^2}{2(c-2)} \quad \text{(Expand)} \tag{14}$$

$$= \frac{(-c^2+c+3)r^2 + (2c^2-4c)r}{2(c-2)(c-1)}. \tag{15}$$

D.3 Computing the Noise Threshold for Reliable Pair Selection

The maximum permissible noise rate r^* is the largest noise level r at which the probability of sampling pairs that preserve true class relationships (favorable leaves in Appendix D.1) exceeds a user-defined threshold $\alpha \in [0, 1)$. This ensures the Siamese network retains sufficient valid training signals for robust learning.

Derivation of r^* The total probability of favorable pairs (green leaves in Fig. 7) is:

$$P_{\text{favorable}}(r, c) = \frac{(c^2-c-3)r^2 + 2(2c-c^2)r + 2(c^2-3c+2)}{2(c-2)(c-1)}. \tag{16}$$

- **At** $r = 0$:

$$P_{\text{favorable}}(0, c) = \frac{2(c^2-3c+2)}{2(c-2)(c-1)} = 1 > \alpha \ \ (since\ \alpha \in [0, 1)). \tag{17}$$

- **At** $r = 1$:

$$P_{\text{favorable}}(1, c) = \frac{c^2-3c+1}{2(c-2)(c-1)}. \tag{18}$$

  For $c \geq 3$, $P_{\text{favorable}}(1, c) < 1$. For example, $c = 3$ gives $P_{\text{favorable}}(1, 3) = \frac{1}{4} < \alpha$ if $\alpha > 0.25$.
- **Monotonicity**: $P_{\text{favorable}}$ is continuous and strictly decreasing in r.

By the Intermediate Value Theorem, there exists a unique $\alpha \in (0, 1)$ where $P_{\text{favorable}}(r^*) = \alpha$. To guarantee $P_{\text{favorable}} > \alpha$, we solve:

$$(c^2-c-3)r^2 + 2(2c-c^2)r + \left[2(c^2-3c+2) - 2\alpha(c-2)(c-1)\right] > 0. \tag{19}$$

Quadratic Solution Let coefficients be defined as:

$$A = c^2 - c - 3, \quad B = 2(2c - c^2), \quad C = 2(c^2 - 3c + 2) - 2\alpha(c-2)(c-1). \tag{20}$$

For $c \geq 3$, the quadratic coefficient $A = c^2 - c - 3$ is *positive*, meaning the parabola is convex. The roots of $Ar^2 + Br + C = 0$ are:

$$r_1 = \frac{-B - \sqrt{B^2 - 4AC}}{2A}, \quad r_2 = \frac{-B + \sqrt{B^2 - 4AC}}{2A}. \tag{21}$$

By noting that $A > 0$, $B < 0$ and $C > 0$, we can conclude that $0 < r_1 \leq 1 \leq r_2$. Consequently for $r \in [0, 1]$ (noise rate), the inequality $Ar^2 + Br + C > 0$ holds for $r < r^* := r_1$.

Closed-Form Expression Substituting A, B, and C yields:

$$r^* = \frac{c^2 - 2c - \sqrt{(c-2)\Psi}}{c^2 - c - 3}, \quad \Psi = -6 + 4c + 2c^2 - c^3 + 2\alpha(3 - 2c - 2c^2 + c^3). \tag{22}$$

Real solutions require $(c-2)\Psi \geq 0$, constraining α to:

$$\alpha \geq \alpha_{\text{min}}(c) = \frac{c^3 - 2c^2 - 4c + 6}{2(c^3 - 2c^2 - 2c + 3)}. \tag{23}$$

This bound reaches its minimum value $1/4$ at $c = 3$, so we restrict α to the interval $[0.25, 1)$.

E Proofs

E.1 Proof of Theorem 2.2

We prove that noisy samples incur a higher misclassification probability than clean samples under the assumptions of our framework. Let $G = 1/0$ denote a correct/incorrect classification, and $\mathbb{N}/\mathbb{C}$ denote noisy/clean samples. The model's performance (MP) and latent representation quality (L) are independent and MP is defined regardless of noise. We model successful classification as requiring both high-quality latent representations and proper model performance, under Assumption 2.1:

$$\mathbb{P}(G = 1) = \mathbb{P}(L \cap MP) = \mathbb{P}(L) \cdot \mathbb{P}(MP), \tag{24}$$

with L = latent representation quality, MP = intrinsic model performance.

Step 1: Decomposing Misclassification Probability. Misclassification probability by the law of total probability is:

$$\mathbb{P}(G = 0) = 1 - \mathbb{P}(G = 1) = 1 - \mathbb{P}(L)\mathbb{P}(MP). \tag{25}$$

Equivalently, misclassification occurs if either the latent representation is poor ($\neg L$) or the model fails ($\neg MP$):

$$\mathbb{P}(G = 0) = \mathbb{P}(\neg L \cup \neg MP) = \mathbb{P}(\neg L) + \mathbb{P}(\neg MP) - \mathbb{P}(\neg L)\mathbb{P}(\neg MP). \tag{26}$$

Conditioning on noise ($\mathbb{N}$) and cleanliness ($\mathbb{C}$), and using the assumption $\mathbb{P}(\neg MP \mid \mathbb{N}) = \mathbb{P}(\neg MP \mid \mathbb{C}) = \mathbb{P}(\neg MP)$, we derive:

$$\mathbb{P}(G = 0 \mid \mathbb{N}) = \mathbb{P}(\neg L \mid \mathbb{N}) + \mathbb{P}(\neg MP) - \mathbb{P}(\neg L \mid \mathbb{N})\mathbb{P}(\neg MP), \tag{27}$$

$$\mathbb{P}(G = 0 \mid \mathbb{C}) = \mathbb{P}(\neg L \mid \mathbb{C}) + \mathbb{P}(\neg MP) - \mathbb{P}(\neg L \mid \mathbb{C})\mathbb{P}(\neg MP). \tag{28}$$

Step 2: Difference in Misclassification Rates. Subtracting the two equations:

$$\mathbb{P}(G = 0 \mid \mathbb{N}) - \mathbb{P}(G = 0 \mid \mathbb{C}) = \underbrace{[\mathbb{P}(\neg L \mid \mathbb{N}) - \mathbb{P}(\neg L \mid \mathbb{C})]}_{\text{Noise effect on } L} \cdot \underbrace{(1 - \mathbb{P}(\neg MP))}_{>0}. \tag{29}$$

Since $1 - \mathbb{P}(\neg MP) > 0$ (as $\mathbb{P}(\neg MP) \in [0, 1)$), the inequality $\mathbb{P}(G = 0 \mid \mathbb{N}) > \mathbb{P}(G = 0 \mid \mathbb{C})$ holds if:

$$\mathbb{P}(\neg L \mid \mathbb{N}) > \mathbb{P}(\neg L \mid \mathbb{C}). \tag{30}$$

Step 3: Quantifying $\mathbb{P}(\neg L \mid \mathbb{N})$ and $\mathbb{P}(\neg L \mid \mathbb{C})$. We compute these probabilities using the tree structure in Fig. 7, which models the latent representation process. Here, r is the corruption probability, and $c \geq 3$ is the number of classes. A "poor" latent representation ($\neg L$) occurs when samples in a pair map to inconsistent clusters.

Case 1: Noisy Samples ($\mathbb{N}$). The probability $\mathbb{P}(\neg L \mid \mathbb{N})$ aggregates contributions from four disjoint events:

- *One corrupted, one clean*: Probability $2r(1-r)$, leading to a mismatch.
- *Both corrupted, different classes*: Probability $r^2 \cdot \frac{c-2}{c-1}$.
- *One corrupted, one clean mis-clustered*: Probability $2r(1-r) \cdot \frac{1}{c-1}$.
- *Both corrupted, same incorrect class*: Probability $r^2 \cdot \frac{1}{c-1}$.

Summing these terms (see Appendix E.2 for algebra):

$$\mathbb{P}(\neg L \mid \mathbb{N}) = \frac{(-c^2+c+3)r^2 + (2c^2-4c)r}{2(c-2)(c-1)}. \tag{31}$$

Case 2: Clean Samples ($\mathbb{C}$). For clean samples, corruption is absent. $\mathbb{P}(\neg L \mid \mathbb{C})$ includes:

- *Both clean but mis-clustered*: Probability $2r(1-r)$.
- *One clean mis-clustered*: Probability $2r(1-r) \cdot \frac{1}{c-1}$.

Summing these terms:

$$\mathbb{P}(\neg L \mid \mathbb{C}) = \frac{(4c-2c^2)r^2 + (2c^2-4c)r}{2(c-2)(c-1)}. \tag{32}$$

Step 4: Final Inequality. Substituting into $\mathbb{P}(\neg L \mid \mathbb{N}) > \mathbb{P}(\neg L \mid \mathbb{C})$:

$$\frac{(-c^2+c+3)r^2 + (2c^2-4c)r}{2(c-2)(c-1)} > \frac{(4c-2c^2)r^2 + (2c^2-4c)r}{2(c-2)(c-1)}. \tag{33}$$

Subtracting the right-hand side from the left:

$$\frac{(c^2-3c+3)r^2}{2(c-2)(c-1)} > 0. \tag{34}$$

The denominator $2(c-2)(c-1)$ is positive for $c \geq 3$. The numerator $c^2 - 3c + 3$ has discriminant $\Delta = (-3)^2 - 4(1)(3) = -3 < 0$, so it is positive for all real c. Since $r > 0$, the inequality holds.

Boundary Cases. If $\mathbb{P}(\neg MP) \approx 1$, the model is fundamentally flawed, and noise has negligible impact. Our framework assumes $\mathbb{P}(\neg MP) < 1$, which holds for non-trivial models.

Conclusion. Thus, $\mathbb{P}(G = 0 \mid \mathbb{N}) > \mathbb{P}(G = 0 \mid \mathbb{C})$, proving that noisy samples exhibit higher misclassification rates due to biased latent representations.

E.2 Detailed Probability Calculations

Noisy Case ($\mathbb{N}$):

$$\mathbb{P}(\neg L \mid \mathbb{N}) = \frac{1}{2} \cdot 2r(1-r) + \frac{1}{2} \cdot r^2 \cdot \frac{c-2}{c-1} + \frac{1}{2} \cdot 2r(1-r) \cdot \frac{1}{c-1} + \frac{1}{2} \cdot r^2 \cdot \frac{1}{c-1} \tag{35}$$

$$= r(1-r) + \frac{r^2(c-2)}{2(c-1)} + \frac{r(1-r)}{c-1} + \frac{r^2}{2(c-1)} \tag{36}$$

$$= \frac{2r(1-r)(c-1) + r^2(c-2) + 2r(1-r) + r^2}{2(c-1)} \tag{37}$$

$$= \frac{(-c^2+c+3)r^2 + (2c^2-4c)r}{2(c-2)(c-1)}. \tag{38}$$

Clean Case ($\mathbb{C}$):

$$\mathbb{P}(\neg L \mid \mathbb{C}) = \frac{1}{2} \cdot 2r(1-r) + \frac{1}{2} \cdot 2r(1-r) \cdot \frac{1}{c-1} = r(1-r) + \frac{r(1-r)}{c-1} \tag{39}$$

$$= \frac{(c-1)r(1-r) + r(1-r)}{c-1} \tag{40}$$

$$= \frac{(4c-2c^2)r^2 + (2c^2-4c)r}{2(c-2)(c-1)}. \tag{41}$$

E.3 Proof of Theorem 2.3

We bound the false-positive probability $\mathbb{P}(X \geq \tau_{\text{d}} \mid \mathbb{C})$ when models may be correlated. Let $X = \sum_{i=1}^{m} X_i$, where $X_i \in \{0,1\}$ indicates misclassification by model i for a clean sample, with $\mathbb{E}[X_i] = p_C$. Unlike Chernoff, Bernstein's inequality does not strictly require independence but instead bounds deviations using variance. We use the following generalized version for dependent variables:

Generalized Bernstein Inequality: For random variables $X_1, \ldots, X_m$ with $|X_i - \mathbb{E}[X_i]| \leq 1$, let $S = \sum_{i=1}^{m} X_i$ and $\sigma^2 = \sum_{i=1}^{m} \text{Var}(X_i) + \sum_{i \neq j} \text{Cov}(X_i, X_j)$. Then, for $t > 0$,

$$\mathbb{P}\left(S - \mathbb{E}[S] \geq t\right) \leq \exp\left(-\frac{t^2}{2\sigma^2 + \frac{2}{3}t}\right). \tag{42}$$

Step 1: Applying the Generalized Bound. For $X_i \sim \text{Bernoulli}(p_C)$, we have $\mathbb{E}[X] = mp_C$ and $\text{Var}(X) = \sum_{i=1}^{m} p_C(1-p_C) + \sum_{i \neq j} \text{Cov}(X_i, X_j)$. Let $\sigma^2_{\text{total}} = \text{Var}(X)$. Applying the inequality to $S = X$ with $t = \tau_{\text{d}} - mp_C = m\epsilon$:

$$\mathbb{P}(X \geq \tau_{\text{d}} \mid \mathbb{C}) \leq \exp\left(-\frac{m^2\epsilon^2}{2\sigma^2_{\text{total}} + \frac{2}{3}m\epsilon}\right). \tag{43}$$

Step 2: Bounding the Total Variance. Assume pairwise correlations satisfy $\text{Cov}(X_i, X_j) \leq \rho p_C(1-p_C)$ for some $\rho < 1$. Then:

$$\sigma^2_{\text{total}} \leq mp_C(1-p_C) + m(m-1)\rho p_C(1-p_C) \leq mp_C(1-p_C)(1+\rho m). \tag{44}$$

For weakly correlated models (for example $\rho = \mathcal{O}(1/m)$), $\sigma^2_{\text{total}} = \mathcal{O}(m)$, preserving the exponential decay rate. Substituting into the bound:

$$\mathbb{P}(X \geq \tau_{\text{d}} \mid \mathbb{C}) \leq \exp\left(-\frac{m\epsilon^2}{2p_C(1-p_C)(1+\rho m) + \frac{2}{3}\epsilon}\right). \tag{45}$$

Step 3: Exponential Decay. For $\rho = o(1)$ (decaying correlations), the dominant term in the denominator is $2p_C(1-p_C)$. Thus, the probability decays exponentially with m:

$$\mathbb{P}(X \geq \tau_{\text{d}} \mid \mathbb{C}) = \mathcal{O}\left(e^{-qm}\right), \quad q = \frac{\epsilon^2}{2p_C(1-p_C) + \frac{2}{3}\epsilon}. \tag{46}$$

Conclusion. Even with bounded correlations, the false-positive rate diminishes exponentially in m, provided correlations do not grow with m.

E.4 Threshold Derivation

Corollary (Practical Threshold Selection Under Worst-Case Variance)**.** *Assume deep learning models exhibit a worst-case clean-data misclassification rate of $p_C = 0.25$, with variance bounded by $\sigma^2 = p_C(1-p_C) = 0.1875$. To ensure exponential decay of false-positives (Theorem 2.3), the detection threshold τ_d must satisfy:*

$$\frac{\tau_d}{m} \geq p_C + \sqrt{\frac{2\sigma^2 \ln(1/\delta)}{m}}, \tag{47}$$

where δ is the desired confidence level ($\delta = 0.05$ for 95% confidence for example). Substituting $p_C = 0.25$ and $\sigma^2 = 0.1875$:

$$\frac{\tau_d}{m} \geq 0.25 + \sqrt{\frac{0.375\ln(1/\delta)}{m}}. \tag{48}$$

For practical deployment with $m \geq 10$, we adopt the conservative heuristic:

$$\frac{\tau_d}{m} \geq 0.6, \tag{49}$$

ensuring $\epsilon = \frac{\tau_d}{m} - p_C \geq 0.35$ for $m \geq 10$. This guarantees a decay rate $k = \frac{\epsilon^2}{2\sigma^2 + \frac{2}{3}\epsilon} \approx 0.2$, yielding $\mathbb{P}(\text{False-Positive}) \leq e^{-0.2m}$, which decays to < 0.14 for $m = 10$ and < 0.001 for $m = 30$.

F Implementation details

Key Libraries Used in All Experiments:

- `torch` == 2.6.0
- `torchvision` == 0.20.1
- `numpy` == 2.2.4
- `scikit-learn` == 1.6.1
- `matplotlib` == 3.10.0
- `seaborn` == 0.13.2
- `tqdm` == 4.67.1

Key Parameters For Training Siamese Network on CIFAR-10:

- Feature extractor: ResNet50 (pre-trained)
- Learning rate: 5×10^{-5} (Adam), Weight decay: 5×10^{-4}
- Batch size: 2,048, Epochs: 1,000 (early stopping patience=8)
- Loss: Cross-entropy (label smoothing=0.1)
- Contrastive margin: 2.0
- Embedding dimension: 64
- Training pairs: 200,000, Validation pairs: 20,000
- Dropout probability: 0.5
- Cross-validation: 10 inner folds, 10 outer folds

Note: These parameters were obtained through trial and error.

Key Parameters For Training Classification Model on CIFAR-10:

- Base model: Pre-ActResNet34
- Batch size: 256 (train/val/test)
- Learning rate: 0.001, Weight decay: 5.46e-5
- Epochs: 200 (Early Stopping Patience=20)
- Loss: Cross-entropy (Label Smoothing=0.1)

Note: These parameters were obtained via grid search using Optuna [47].

Key Parameters For Training Siamese Network on Fashion-MNIST:

- Feature extractor: ResNet34
- Learning rate: 5×10^{-5} (Adam), Weight decay: 1×10^{-3}
- Batch size: 2,048, Epochs: 1,000 (early stopping patience=8)

- Loss: Cross-entropy (label smoothing=0.1)
- Contrastive margin: 1.0
- Embedding dimension: 128
- Training pairs: 200,000, Validation pairs: 20,000
- Dropout probability: 0.5
- Cross-validation: 10 inner folds, 10 outer folds

Note: These parameters were obtained through trial and error.

Key Parameters For Training Classification Model on Fashion-MNIST:

- Base model: Pre-ActResNet34
- Batch size: 256 (train/val/test)
- Learning rate: 0.0002, Weight decay: 1.12×10^{-6}
- Epochs: 200 (Early Stopping Patience=20)
- Loss: Cross-entropy (Label Smoothing=0.1)

Note: These parameters obtained via grid search using Optuna [47].

G Additional experimental results

G.1 Detailed Experimental Results and Threshold Selection

We present comprehensive results from our grid search for determining the optimal detection threshold (τ_{d}), which serves as the noise confidence cutoff, and the relabeling threshold (τ_{r}), which represents the minimum agreement required for clean labels. The selection process aims to balance two key objectives:

- **Primary Objective**: Maximize the Noise F1-Score (which balances precision and recall) to ensure the detection of as many noisy labels as possible.
- **Secondary Objective**: Maintain a high Relabeling Score to effectively adjust the noisy labels.

G.1.1 Threshold Selection Strategy

1. **Grid Search Space**: $\tau_{\mathrm{d}} \in \{6, 7, 8, 9, 10\}$, $\tau_{\mathrm{r}} \in \{6, 7, 8, 9, 10\}$ (discrete confidence levels)
2. **Metric Prioritization**:
    - Primary: Noise F1-Score for τ_{d}
    - Secondary: Relabeling Score for τ_{r}
3. **Trade-off Analysis**:
    - Higher τ_{d}: Increases precision but reduces recall
    - Lower τ_{r}: Increases relabeling but risks error propagation

Warning: This strategy relies on clean validation data. For real-world datasets without ground-truth clean labels, we propose a practical threshold-selection methodology in Appendix H. This approach enables effective noise detection and correction without requiring access to clean validation data.

Note: Fortunately, our Siamese model does not require to be trained each time for different hyperparameters. We only need to train the model once and then use the same model to detect and relabel noisy samples with different hyperparameters.

Table 5: Noise Detection Performance with different hyper-parameters for CIFAR-10

Noise Ratio	τ_d	Noise Accuracy	Noise Precision	Noise Recall	Noise F1-Score
CIFAR-10N	6	0.8972	0.4628	**0.8792**	0.6064
	7	0.9119	0.5066	0.8519	0.6354
	8	0.9244	0.5544	0.8182	0.6609
	9	0.9358	0.6135	0.7758	0.6852
	10	**0.9451**	**0.6999**	0.6848	**0.6922**
IDN-20%	6	0.8676	0.6098	**0.9760**	0.7506
	7	0.8864	0.6510	0.9560	0.7746
	8	0.8999	0.6932	0.9149	**0.7887**
	9	**0.9070**	0.7408	0.8375	0.7862
	10	0.8955	**0.7887**	0.6671	0.7228
IDN-30%	6	0.7873	0.5979	**0.8847**	0.7136
	7	0.8069	0.6386	0.8188	**0.7176**
	8	**0.8160**	0.6791	0.7310	0.7041
	9	0.8124	0.7194	0.6123	0.6616
	10	0.7889	**0.7606**	0.4307	0.5500
IDN-40%	**6**	0.6115	0.5069	**0.7021**	**0.5887**
	7	0.6326	0.5309	0.6217	0.5727
	8	0.6530	0.5655	0.5343	0.5495
	9	0.6652	0.6084	0.4342	0.5068
	10	**0.6663**	**0.6748**	0.3041	0.4193

Table 6: Noise Detection Performance with different hyper-parameters for Fashion-MNIST

Noise Ratio	τ_d	Noise Accuracy	Noise Precision	Noise Recall	Noise F1-Score
IDN-20%	6	0.8723	0.6227	**0.9810**	0.7618
	7	0.8962	0.6741	0.9714	0.7959
	8	0.9194	0.7353	0.9577	0.8319
	9	0.9358	0.7981	0.9258	**0.8572**
	10	**0.9395**	**0.8714**	0.8325	0.8515
IDN-30%	6	0.8715	0.7091	**0.9780**	0.8221
	7	0.8943	0.7544	0.9667	0.8474
	8	0.9125	0.8008	0.9473	0.8679
	9	**0.9216**	0.8502	0.9003	**0.8746**
	10	0.9026	**0.9044**	0.7597	0.8257
IDN-40%	6	0.8631	0.7591	**0.9677**	0.8508
	7	0.8787	0.7958	0.9402	**0.8620**
	8	**0.8821**	0.8307	0.8887	0.8587
	9	0.8667	0.8662	0.7917	0.8273
	10	0.8093	**0.9033**	0.5902	0.7140

G.1.2 Dataset-Specific Analysis

In Tables 5 and 6, we summarize the grid search over detection thresholds τ_d for CIFAR-10 and Fashion-MNIST. Then, fixing the best τ_d from that search, Tables 7 and 8 present our sweep over relabeling thresholds τ_r. For each τ_r we report:

- the total number of relabeled samples ("Count"),
- the relabeling accuracy ("Accuracy"),
- the relabeling score (see Section 2.4),
- the resulting noise ratio,
- the size of the dataset remaining after cleaning.

Table 7: CIFAR-10 Relabeling Threshold Results

Noise Ratio Before	τ_d	τ_r	**Relabeling**								**Noise Ratio After**	**Remaining Samples**
			Score Distribution					**Count**	**Accuracy**	**Score**		
			-2	**-1**	**0**	**1**	**2**					
CIFAR-10N	10	6	1159	164	399	248	2438	3996	61.01	0.66	6.01%	49588
		7	1044	279	306	440	2339	3689	63.40	0.75	5.62%	49281
		8	932	391	222	647	2216	3370	65.75	0.84	5.26%	48962
		9	798	525	166	871	2048	3012	67.99	0.94	4.90%	48604
		10	613	710	103	1241	1741	2457	70.85	**1.13**	**4.45%**	48049
20%	8	6	2695	1439	704	772	7863	11262	69.82	0.86	8.93%	47789
		7	2031	2103	500	1273	7566	10097	74.93	1.01	7.29%	46624
		8	1414	2720	330	1866	7143	8887	80.37	1.19	5.75%	45414
		9	866	3268	195	2908	6236	7297	85.45	1.42	4.40%	43824
		10	379	3755	85	4832	4422	4886	90.50	**1.87**	**3.22%**	41413
30%	7	6	3742	3199	826	1770	9667	14235	67.91	0.73	16.17%	45031
		7	2525	4416	514	2780	8969	12008	74.69	0.94	13.44%	42804
		8	1321	5620	302	4537	7424	9047	82.06	1.23	10.88%	39843
		9	566	6375	131	6684	5448	6145	88.66	1.64	9.23%	36941
		10	178	6763	47	9204	3012	3237	93.05	**2.51**	**8.63%**	34033
40%	6	6	8613	4910	1315	3367	9221	19149	48.15	-0.02	37.94%	41723
		7	5713	7810	778	5786	7339	13830	53.07	0.09	34.04%	36404
		8	3323	10200	403	8070	5430	9156	59.31	0.23	30.34%	31730
		9	1606	11917	165	10246	3492	5263	66.35	0.40	27.56%	27837
		10	494	13029	44	12204	1655	2193	75.47	**0.68**	**25.99%**	24767

Table 8: Fashion-MNIST Relabeling Threshold Results

Noise Ratio Before	τ_d	τ_r	**Relabeling**								**Noise Ratio After**	**Remaining Samples**
			Score Distribution					**Count**	**Accuracy**	**Score**		
			-2	**-1**	**0**	**1**	**2**					
20%	9	6	2094	833	876	959	9734	12704	76.62	1.21	6.69%	58208
		7	1752	1175	621	1620	9328	11701	79.72	1.33	5.77%	57205
		8	1436	1491	434	2321	8814	10684	82.49	1.46	4.98%	56188
		9	865	1875	268	3235	8066	9386	85.93	1.64	4.09%	54890
		10	462	2465	122	5035	6412	6996	91.65	**2.07**	**2.88%**	52500
30%	9	6	2043	847	1223	1624	13560	16826	80.59	1.42	8.83%	57529
		7	1662	1228	857	2638	12912	15431	83.68	1.55	7.72%	56134
		8	1273	1617	582	3704	12121	13976	86.72	1.70	6.71%	54679
		9	865	2025	361	5145	10901	12127	89.89	1.91	5.76%	52830
		10	353	2537	170	8044	8193	8716	93.99	**2.43**	**4.73%**	49419
40%	7	6	2806	3028	1639	3598	17506	21951	79.75	1.37	11.04%	53374
		7	1917	3917	1100	5375	16268	19285	84.36	1.56	8.80%	50708
		8	1148	4686	681	7887	14175	16004	88.57	1.83	6.91%	47427
		9	604	5230	372	11106	11265	12241	92.03	2.22	5.55%	43664
		10	231	5603	133	15508	7102	7466	95.12	**3.17**	**4.65%**	38889

Tables 7 and 8 demonstrate that increasing the relabeling threshold τ_r yields steadily higher relabeling accuracy and score, while reducing the number of samples corrected. This trade-off highlights a fundamental balance in our method: higher τ_r values produce more reliable corrections on fewer samples, whereas lower thresholds cover more samples with lower confidence. Moreover, as τ_r

increases, the remaining dataset after cleaning diminishes, which must be considered when selecting τ_r. Finally, the strong correlation between the relabeling score and the post-cleaning noise ratio confirms the score's effectiveness as an indicator of correction quality.

These results serve as a benchmark for future methods in noisy-label detection. We recommend reporting the relabeling score—alongside precision and recall— as a clear, interpretable metric for assessing the quality of label corrections.

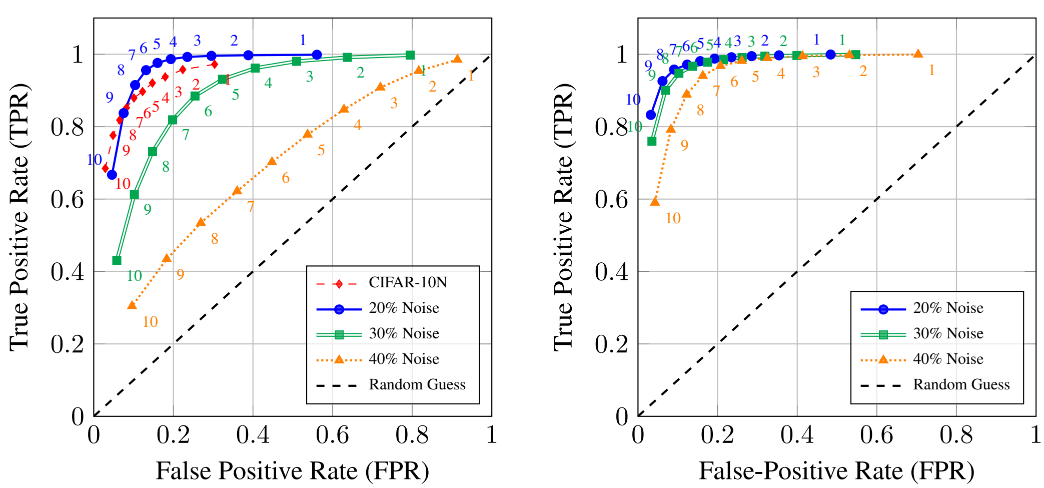
G.2 Understanding How the Detection Threshold Affects Noise Detection

Figure 8 shows how adjusting the detection threshold (τ_d) impacts the performance of our noise detection method. Think of τ_d as a "knob" that controls how strict the model is when deciding whether a data sample is noisy:

- **Higher τ_d (stricter)**:
  - Reduces **false alarms** (incorrectly flagging clean data as noisy).
  - Also reduces **correct detections** (missing some actual noisy data).
- **Lower τ_d (looser)**:
  - Increases **correct detections**.
  - Increases **false alarms**.

G.2.1 Key Observations

- **CIFAR-10 Results** (Figure 8a):
  - At moderate noise levels (20–30%): A threshold of $\tau_d/m \geq 0.6$ keeps false alarms between 2%–25%, while catching most true noise.
  - At 40% noise: Performance drops (more false alarms and missed detections), showing the challenge of extreme noise.
- **Fashion-MNIST Results** (Figure 8b):
  - Simpler data allows better performance: False alarms stay below 21% even at 40% noise with $\tau_d/m \geq 0.6$.
- **Theoretical Agreement**: The sharp drop in false alarms as τ_d increases matches our mathematical predictions (Corollary E.4 & Theorem 2.3).



(a) CIFAR-10 (b) Fashion-MNIST

Figure 8: ROC curves for noise detection on CIFAR-10 and Fashion-MNIST datasets.

H Threshold Selection Without Ground-Truth Labels

In many real-world scenarios, clean labels are unavailable for tuning detection and relabeling thresholds. We propose a simple calibration protocol that requires training the Siamese ensemble only once (Algorithm 4):

1. **Single-Pass Embedding Extraction.** Train the Siamese ensemble on the full noisy dataset to compute each sample's disagreement count:

$$r(x) = \sum_{j=1}^{m} \mathbb{1}\big(f_j(x) \neq \tilde{y}\big) \in \{0, 1, \ldots, m\}.$$

2. **Detection Threshold Grid.** Define candidate detection thresholds:

$$\tau_{\mathrm{d}} \ \in \ \{\lceil 0.6m \rceil, \lceil 0.7m \rceil, \ldots, m\}.$$

   Lower values favor recall (flagging more potential noise), while higher values favor precision.

3. **Relabeling Threshold Grid.** For each τ_d, define relabeling thresholds:

$$\tau_{\mathrm{r}} \ \in \ \{\lceil 0.5m \rceil, \lceil 0.6m \rceil, \ldots, \tau_{\mathrm{d}}\}.$$

   Lower τ_{r} relabels more samples (risking mis-corrections); higher τ_r is more conservative.

4. **Calibration Loop.** For each $(\tau_{\mathrm{d}}, \tau_{\mathrm{r}})$ pair:
   (a) Use the model to obtain a cleaned dataset without any extra training.
   (b) Train a lightweight downstream classifier for a few epochs.
   (c) Evaluate validation performance (e.g., accuracy or F1 score).

   Select $(\tau_{\mathrm{d}}, \tau_{\mathrm{r}})$ that maximizes the chosen metric.

5. **Final Deployment.** Re-clean the full dataset with the selected thresholds and train the target model at scale.

Algorithm 4 Threshold Calibration without Clean Labels

Inputs: Siamese ensemble of size m, noisy dataset $\mathbb{D} = (x_i, \tilde{y}_i)$, number of outer folds k
Output: Optimal thresholds $(\tau_{\mathrm{d}}^*, \tau_{\mathrm{r}}^*)$

function CALIBRATETHRESHOLDS$(\mathbb{D}, m, k)$
    $\mathbf{P} =$ CollectPredictions$(\mathbb{D}, k, m)$ ▷ Algorithm 1
    $\mathcal{T}_d = \lceil 0.6m \rceil, \ldots, m$
    best_score = 0
    for τ_{d} in $\mathcal{T}_d$ **do**
        $\mathcal{T}_r = \{\lceil 0.5m \rceil, \ldots, m\}$
        for τ_{r} in $\mathcal{T}_r$ **do**
            $\mathbb{D}_{clean} =$ DetectAndRelabel$(\mathbf{P}, \mathbb{D}, \tau_{\mathrm{d}}, \tau_{\mathrm{r}})$ ▷ Algorithm 2
            Train classifier on $\mathbb{D}_{clean}$ for few epochs
            score = Validate classifier on held-out set
            if score > best_score **then**
                best_score = score
                $(\tau_{\mathrm{d}}^*, \tau_{\mathrm{r}}^*) = (\tau_{\mathrm{d}}, \tau_{\mathrm{r}})$
            end if
        end for
    end for
    return $(\tau_{\mathrm{d}}^*, \tau_{\mathrm{r}}^*)$
end function

I Reliability Analysis of Ensemble Disagreement

Our model naturally yields a **probabilistic estimate** of label noise by interpreting the raw disagreement count:

$$r(x) \;=\; \sum_{j=1}^{m} \mathbb{1}\big(f_j(x) \neq \tilde{y}\big), \tag{50}$$

not as a hard decision but as a **noise score**:

$$\hat{p}(x) = \mathbb{P}\big(x \in \mathbb{N} \mid r(x)\big) = \frac{r(x)}{m} \;\in [0,1]. \tag{51}$$

Figure 9 then plots, for each predicted noise probability $\hat{p}$, the *observed* fraction of truly noisy labels among all samples with that score. Concretely, each marker at $\hat{p}$ shows

$$\frac{\big|\{\, x : \, \hat{p}(x) = \hat{p} \wedge x \in \mathbb{N}\}\big|}{\big|\{\, x : \, \hat{p}(x) = \hat{p}\}\big|}. \tag{52}$$

This is exactly a *reliability diagram*: horizontal axis = predicted noise probability, vertical axis = empirical noise rate. From the curves we observe:

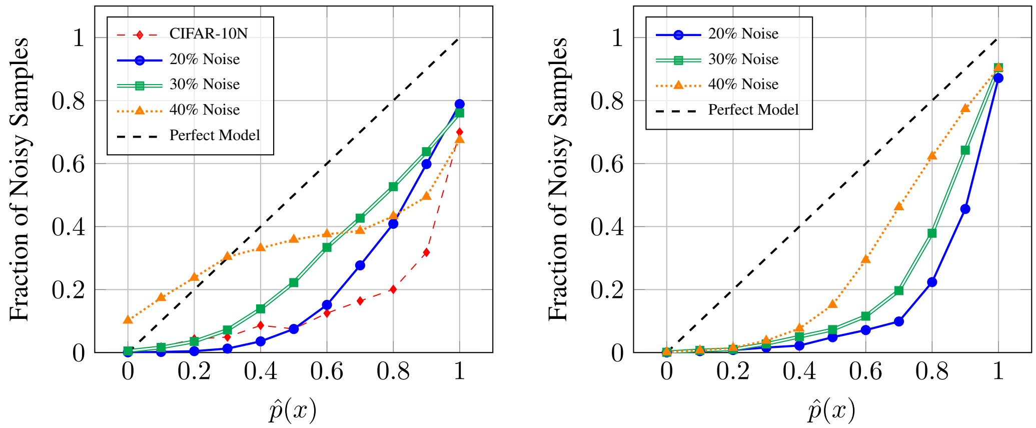
- **Strong discrimination.** In all subplots and noise regimes, the curves rise monotonically, confirming that higher $\hat{p}(x)$ reliably ranks samples by corruption likelihood.
- **Systematic overconfidence.** Every curve lies *below* the diagonal (except for CIFAR-10 40%). For example, on CIFAR-10N at $\hat{p} = 0.6$, only $\approx 20\%$ of those samples are actually noisy. Thus $\hat{p} = r/m$ *overestimates* the true noise probability.
- **Noise-level dependence.** As the true noise rate increases (20%–40%), the curves move closer to the diagonal-yet even at 40% noise, $\hat{p} = 0.8$ corresponds to only $\approx 43\%$ actual corruption.

Theoretical guarantee. Under Theorem 2.2 and its ensemble-size corollaries, as the number of models m grows, the distributions of $\hat{p}(x) = r(x)/m$ for clean and noisy samples concentrate around two well-separated values. A threshold chosen between these values yields exponentially vanishing false-positive and false-negative rates.

Practical implication. To avoid excessive false positives, detection thresholds should be chosen by consulting these curves (e.g. pick the $\hat{p}$ where the empirical curve crosses the desired noise rate) or by applying a lightweight calibration method (such as isotonic regression) to correct the overconfidence before using $\hat{p}(x)$ as a probability.

Future directions. Having access to clean validation data and a calibrated noise score enables several new strategies:

- **Soft sample weighting:** Rather than a hard discard, downstream losses can be re-weighted by $1 - \hat{p}(x)$ for robust training under uncertainty.
- **Data-driven thresholding:** Users can pick τ_d by consulting the calibration curves to meet target precision or recall.
- **Active cleansing:** Samples with intermediate $\hat{p}(x)$ can be triaged for human review, maximizing annotation efficiency.
- **Calibration:** Learning a lightweight calibration transform (e.g. via isotonic regression) could correct residual bias in $\hat{p}(x)$.
- **Dynamic thresholding:** One could also explore dynamic thresholding schemes that adapt to class imbalances or domain shifts by re-estimating calibration on the fly.



(a) CIFAR-10 (b) Fashion-MNIST

Figure 9: Fraction of noisy samples vs. $P\big(x \in \mathbb{N} \mid r(x)\big)$. Higher ensemble disagreement correlates strongly with label noise.

J Discussion on False-Positive Cases

False-positives are clean samples mistakenly flagged as noisy. These often include ambiguous examples like blurry images or objects with unusual angles (Fig. 10) that confuse both AI models and human annotators. While problematic for training, removing these challenging samples can paradoxically improve model performance by:

- Focusing learning on clearer examples first (like teaching addition before calculus)
- Reducing exposure to confusing patterns early in training
- Aligning with curriculum learning principles [48] (gradual difficulty increase)

Figure 10: Examples of confusing clean images from CIFAR-10 that our model mistakenly flagged as noisy. These contain unusual angles, partial objects, or blurry textures that challenge both humans and algorithms.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's claims in the abstract and introduction accurately reflect our contributions in developing a Siamese network framework for instance-dependent label noise detection and correction. Our theoretical contributions are substantiated by Theorem 2.2 and Theorem 2.3, with complete proofs in Appendix E.1 and Appendix E.3 respectively. The experimental results in Section 3 validate our claims, demonstrating significant improvements over baselines on both synthetic and real-world datasets.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work performed by the authors in the section 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All assumptions are clearly stated in theorems, with complete proofs in appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides all details needed to reproduce the main results: algorithmic descriptions in Section 2, network architecture in Section 2.1, and hyper-parameters in Appendix F. Ready-to-run code is included in the supplementary materials for submission and will be publicly released on GitHub in the final version.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
    (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
    (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
    (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
    (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper provides open access to the data and code, with sufficient instructions in the repository itself to faithfully reproduce the main experimental results. The saved runs are also provided in the repository. The code is written in PyTorch and is fully reproducible and ready to run. Due to submission anonymity, the code is not publicly available at this time, but it will be released in a public Github repository after the review process.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper provides all algorithmic details (Section 2), network architecture (Section 2.1), and hyper-parameters (Appendix F). All scripts for data preprocessing, model training, and evaluation, along with detailed usage instructions, are included in the anonymized supplementary material and will be publicly released in the final GitHub repository to ensure full reproducibility. reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report statistical significance for all classification experiments by running each experiment 5 times. Tables 2 and 3 show mean accuracy and standard deviation (in the format $mean \pm std$) for our method and all baseline methods. This approach captures variability from initialization and stochastic training processes, allowing for fair statistical comparison between methods. The standard deviation represents the 1-sigma error bars assuming normally distributed results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The computational resources needed to reproduce the experiments are provided in Section 3 and Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The paper conforms to the NeurIPS Code of Ethics, as it is fully compliant with the guidelines provided in the link.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed in Section 4.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper uses open-source datasets and models, and does not release any new models or datasets that have a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: We used publicly available benchmark datasets (CIFAR-10[21], Fashion-MNIST[22], and CIFAR-10N[24]) with proper citations in Section 3.3. All datasets are used in accordance with their respective licenses and terms of use.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowd-sourcing and research with human subjects**

    Question: For crowd-sourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Paper does not involve crowd-sourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowd-sourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Paper does not involve crowd-sourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowd-sourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.