Learning to Clean: Reinforcement Learning for Noisy Label Correction

Marzi Heidari[1], Hanping Zhang[1], Yuhong Guo[1,2]

[1]Carleton University, Ottawa, Canada
[2]CIFAR AI Chair, Amii, Canada

Introduction

Deep networks learn fast — but they also memorize noise.

🧩 **Problem**: Real-world datasets contain **incorrect or ambiguous labels**.

🚫 Conventional methods — filtering, reweighting, or SSL — rely on **static heuristics.**

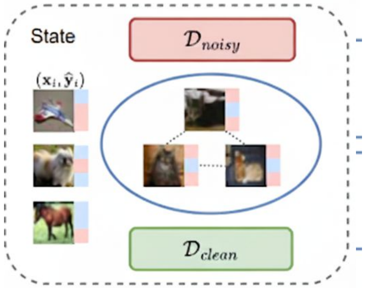
🎯 **Goal**: Learn a **dynamic, feedback-driven policy** that improves labels through experience.

Reinforcement Learning for Noisy Label Correction (RLNLC)

- We reframe label correction as a **Reinforcement Learning (RL)** problem.
- An model learns to clean labels through **sequential decision-making**.
- The **policy** adapts dynamically as the dataset evolves.
- Enables **non-myopic optimization** — learning strategies that maximize future generalization.

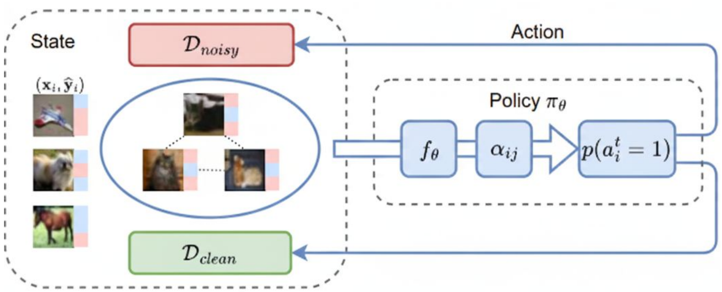
Framework

- **State**: Dataset with current labels $s^t = \{(\mathbf{x}_i, \hat{\mathbf{y}}_i^t)\}_{i=1}^N$ at time step t



Framework

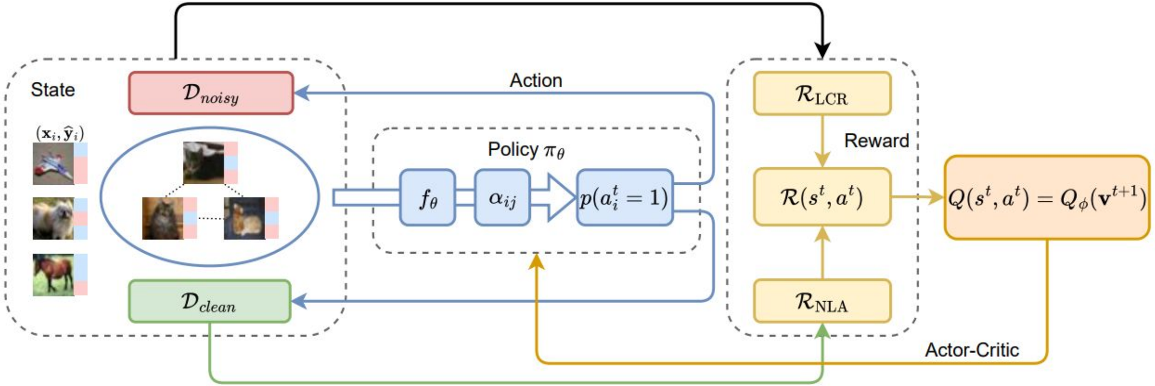
- **State**: Dataset with current labels $s^t = \{(\mathbf{x}_i, \hat{\mathbf{y}}_i^t)\}_{i=1}^N$ at time step t
- **Policy**: Stochastic function $\pi_\theta(a|s_t)$ that outputs correction probabilities based on label inconsistency.
- **Action**: Binary vector $\boldsymbol{a} = [a_1, \cdots, a_i, \cdots, a_N]$ indicating whether each label is kept (0) or corrected (1).



Carleton University

Framework

- **State**: Dataset with current labels $s^t = \{(\mathbf{x}_i, \hat{\mathbf{y}}_i^t)\}_{i=1}^N$ at time step t
- **Policy**: Stochastic network $\pi_\theta(a|s_t)$ that outputs correction probabilities based on label inconsistency.
- **Action**: Binary vector $\boldsymbol{a} = [a_1, \cdots, a_i, \cdots, a_N]$ indicating whether each label is kept (0) or corrected (1).
- **Reward**: Combines label consistency and noisy–clean alignment



State and Action

State:

$$s^t = \{(\mathbf{x}_i, \widehat{\mathbf{y}}_i^t)\}_{i=1}^N$$

— each label may already be corrected.

Action:

$$\boldsymbol{a} = [a_1, \cdots, a_i, \cdots, a_N] \text{ , } a_i \in \{0, 1\}$$

— $a_i = 1$ $\longrightarrow$ correct the label, $a_i = 0$ $\longrightarrow$ keep it.

Each new action updates the dataset to state s^{t+1} .

Stochastic Policy with Deterministic Transition

Correction Probability:

$$\pi_\theta(\boldsymbol{s}^t)_i = p(a_i^t = 1) = \frac{\sum_{j=1}^{C} \mathbb{1}(\bar{\mathbf{y}}_{ij} > \bar{\mathbf{y}}_{i\hat{y}_i}) \cdot \bar{\mathbf{y}}_{ij}}{\sum_{j=1}^{C} \mathbb{1}(\bar{\mathbf{y}}_{ij} \geq \bar{\mathbf{y}}_{i\hat{y}_i}) \cdot \bar{\mathbf{y}}_{ij}},$$

- Based on **label–neighbor disagreement** in kNN space.
- Sampled from $\mathrm{Bernoulli}(p_i)$, with $p_i = p(a_i^t = 1)$
- **Deterministic transition:**

$$\hat{\mathbf{y}}_i^{t+1} = \begin{cases} \hat{\mathbf{y}}_i^t & \text{if } a_i^t = 0, \\ \bar{\mathbf{y}}_i & \text{if } a_i^t = 1. \end{cases}$$

→ New dataset = progressively cleaned state.

Reward Function

Label Consistency Reward (LCR): how consistent each label is with its k-nearest neighbors in feature space.

$$\mathcal{R}_{\mathrm{LCR}}(\boldsymbol{s}^t, \boldsymbol{a}^t) = -\mathbb{E}_{i \in [1:N]}\left[\mathrm{KL}\left(\widehat{\mathbf{y}}_i^{t+1}, \sum_{j \in \mathcal{N}_\omega(\mathbf{x}_i)} \alpha_{ij} \widehat{\mathbf{y}}_j^{t+1}\right)\right]$$

Noisy Label Alignment Reward (NLA): Encourages corrected (noisy) labels to align with clean ones through inter-subset consistency.

$$\mathcal{R}_{\mathrm{NLA}}(\boldsymbol{s}^t, \boldsymbol{a}^t) = -\mathbb{E}_{i \in \mathcal{D}_{\mathrm{noi}}^{t+1}}\left[\mathrm{KL}\left(\widehat{\mathbf{y}}_i^{t+1}, \sum_{j \in \mathcal{N}_{\mathrm{cle}}(\mathbf{x}_i)} \alpha_{ij} \widehat{\mathbf{y}}_j^{t+1}\right)\right]$$

Final Reward: $\mathcal{R}(\boldsymbol{s}^t, \boldsymbol{a}^t) = \exp\left(\mathcal{R}_{\mathrm{LCR}}(\boldsymbol{s}^t, \boldsymbol{a}^t) + \lambda \mathcal{R}_{\mathrm{NLA}}(\boldsymbol{s}^t, \boldsymbol{a}^t)\right)$

✅ Keeps rewards positive, bounded, and stable.

✅ Promotes coherence between corrected and clean subset

Actor–Critic RL with State Encoding

🎯 **Learning Objective: Actor (π)** proposes label-correction actions. **Critic (Q)** predicts long-term reward

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\boldsymbol{s}\sim\rho_{\pi_\theta},\boldsymbol{a}\sim\pi_\theta(\boldsymbol{s})}\left[\nabla_\theta \log \pi_\theta(\boldsymbol{a}|\boldsymbol{s}) Q(\boldsymbol{s},\boldsymbol{a})\right].$$

✳️ State Encoding for the Critic

1. **Instance-Level Reward** measures label–neighborhood consistency

$$r(\mathbf{x}_i, \widehat{\mathbf{y}}_i^{t+1}) = \exp\left(-\operatorname{KL}\left(\widehat{\mathbf{y}}_i^{t+1}, \sum_{j\in\mathcal{N}_\omega(\mathbf{x}_i)} \alpha_{ij}\widehat{\mathbf{y}}_j^{t+1}\right)\right)$$

2. **Binning Aggregation** encodes the dataset as a histogram vector summarizing label quality

$$(\mathbf{x}_i, \widehat{\mathbf{y}}_i^{t+1}) \in \mathcal{B}_j \quad \text{if } r(\mathbf{x}_i, \widehat{\mathbf{y}}_i^{t+1}) \in \left(\frac{j-1}{N_b}, \frac{j}{N_b}\right], \quad \mathbf{v}_j^{t+1} = |\mathcal{B}_j|/N$$

- The **critic** receives this compact vector $\mathbf{v}_j^{t+1}$ instead of the full dataset.
- It learns to map the global label-quality distribution to expected future rewards.
- The **actor** uses critic feedback to refine the correction policy.
- provides a good trade-off between granularity and stability.

Carleton University

Label Cleaning for Prediction Model Training

Stage 1 — Policy-Guided Cleaning:

- Apply policy for T′ steps ⟶ progressively refine labels.
- Final state $\boldsymbol{s}^{T'} = \{(\mathbf{x}_i, \widehat{\mathbf{y}}_i^{T'})\}_{i=1}^{N}$ = cleaned dataset.

Stage 2 — Model Training

Step	Description
Pre-train	Prediction model $h_\psi \circ f_\theta$ on noisy data (cross-entropy).
Fine-tune	Same model on cleaned labels from $\boldsymbol{s}^{T'}$.

Outcome:

✅ Cleaner supervision ⟶ better generalization

Carleton University

Experimental Results

Table 1: Test accuracy (%) of different methods on CIFAR10-IDN and CIFAR100-IDN under various IDN noise rates. Standard deviations are shown as subscripts in parentheses. Columns correspond to different label noise ratios. † denotes results reproduced using publicly available source code.

Method	CIFAR10-IDN					CIFAR100-IDN				
	0.20	0.30	0.40	0.45	0.50	0.20	0.30	0.40	0.45	0.50
CE [6]	75.8	69.2	62.5	51.7	39.4	30.4	24.2	21.5	15.2	14.4
Mixup [43]	73.2	72.0	61.6	56.5	49.0	32.9	29.8	25.9	23.1	21.3
Forward [44]	74.6	69.8	60.2	48.8	46.3	36.4	33.2	26.8	21.9	19.3
Reweight [19]	76.2	70.1	62.6	51.5	45.5	36.7	31.9	28.4	24.1	20.2
Decoupling [20]	78.7	75.2	61.7	58.6	50.4	36.5	30.9	27.9	23.8	19.6
Co-teaching [11]	81.0	78.6	73.4	71.6	45.9	38.0	33.4	28.0	25.6	24.0
MentorNet [10]	81.0	77.2	71.8	66.2	47.9	38.9	34.2	31.9	27.5	24.2
DivideMix [17]	94.8	94.6	94.5	94.1	93.0	77.1	76.3	70.8	57.8	58.6
CausalNL [6]	81.4	80.3	77.3	78.6	67.3	41.4	40.9	34.0	33.3	32.1
SSR† [24]	96.5	96.5	96.3	95.9	94.1	78.8	78.6	77.0	75.0	72.8
RLNLC (Ours)	$\mathbf{97.3}_{(0.1)}$	$\mathbf{97.1}_{(0.1)}$	$\mathbf{96.9}_{(0.2)}$	$\mathbf{96.6}_{(0.2)}$	$\mathbf{95.8}_{(0.4)}$	$\mathbf{80.5}_{(0.7)}$	$\mathbf{80.1}_{(0.7)}$	$\mathbf{78.5}_{(0.8)}$	$\mathbf{77.2}_{(0.8)}$	$\mathbf{74.7}_{(0.9)}$

Carleton University

Thank You