A. Appendix

A.1. Taylor Polynomial

We illustrate the bi-variate 4^{th} order Taylor series,

$$\begin{aligned}\ell(\hat{\mathbf{y}}_i, \mathbf{y}_i) =& \theta_2(\hat{\mathbf{y}}_i - \theta_0) + \frac{1}{2}\theta_3(\hat{\mathbf{y}}_i - \theta_0)^2 + \frac{1}{6}\theta_4(\hat{\mathbf{y}}_i - \theta_0)^3 + \frac{1}{24}\theta_5(\hat{\mathbf{y}}_i - \theta_0)^4 \\ &+ \theta_6(\hat{\mathbf{y}}_i - \theta_0)(\mathbf{y}_i - \theta_1) + \frac{1}{2}\theta_7(\hat{\mathbf{y}}_i - \theta_0)(\mathbf{y}_i - \theta_1)^2 + \frac{1}{2}\theta_8(\hat{\mathbf{y}}_i - \theta_0)^2(\mathbf{y}_i - \theta_1) \\ &+ \frac{1}{6}\theta_9(\hat{\mathbf{y}}_i - \theta_0)^3(\mathbf{y}_i - \theta_1) + \frac{1}{6}\theta_{10}(\hat{\mathbf{y}}_i - \theta_0)(\mathbf{y}_i - \theta_1)^3 + \frac{1}{4}\theta_{11}(\hat{\mathbf{y}}_i - \theta_0)^2(\mathbf{y}_i - \theta_1)^2 \\ &+ \theta_{12}(\mathbf{y}_i - \theta_1) + \frac{1}{2}\theta_{13}(\mathbf{y}_i - \theta_1)^2 + \frac{1}{6}\theta_{14}(\mathbf{y}_i - \theta_1)^3 + \frac{1}{24}\theta_{15}(\mathbf{y}_i - \theta_1)^4, \end{aligned} \tag{6}$$

to compare with the ARL format presented in Equation 5. The terms only containing the ground truth, $\mathbf{y}$, are kept in Equation 6 and these terms will not effect the training as no gradients are backpropagated into the network through $\mathbf{y}$.

A.2. Architecture of Neural networks

The architectures we used to conduct architecture randomisation and dataset randomisation are given in Table 6. In architecture randomisation, 2-Layer MLP, 3-Layer MLP and 4-layer CNN are applied. For the dataset randomisation, only 4-Layer CNN are utilised. JoCoR-Net is only used in the dataset-specific loss learning task.

Table 6. The architectures used in the experiments

2-Layer MLP	3-Layer MLP	4-Layer CNN	JoCoR-Net
28×28 Gray Image	28×28 Gray Image	32×32 RGB Image	32×32 RGB Image
FC $28 \times 28 \longrightarrow 256$, ReLU	FC $28 \times 28 \longrightarrow 256$, ReLU FC$256 \longrightarrow 256$, ReLU	5×5, 32, ReLU 5×5, 64, ReLU FC $1024 \longrightarrow 1024$, ReLU	3×3, 64 BN, ReLU 3×3, 64 BN, ReLU 2×2 Max-pool 3×3, 128 BN, ReLU 3×3, 128 BN, ReLU 2×2 Max-pool 3×3, 196 BN, ReLU 3×3, 196 BN, ReLU 8×8 Avg-pool
FC $256 \longrightarrow 10$	FC $256 \longrightarrow 10$	FC $1024 \longrightarrow 10$	FC $196 \longrightarrow 100$

A.3. Further Implementation Details

We make use of normalisation to ensure the loss values are bounded in a well behaved range for CMA-ES loss function search,

$$\hat{f} = \eta \frac{f - f_{min}}{f_{max} - f_{min}}, \tag{7}$$

where f_{min} and f_{max} denotes the minimum and maximum and η is a hyperparameter deciding the dynamic range of the loss function. Both f_{min} and f_{max} are easily approximated by sampling random points satisfying $\{(\hat{\mathbf{y}}, \mathbf{y}) | \hat{\mathbf{y}}_i \geq 0, \sum_i \hat{\mathbf{y}}_i = 1; \mathbf{y}_i \in \{0, 1\}, \sum_i \mathbf{y} = 1\}$, which defines the domain of the loss function. Note that this can help improve CMA-ES optimisation stability during meta-training.

A.4. Sensitivity to early-stopping

As discussed before, using noisy validation for hyper-parameter tuning and early stopping generates different models when comparing with the models trained with long epochs, with models selected by early stopping tending to have better performance. The performance gap between a model picked by early stopping and that trained to convergence reflects the robustness of a loss function. A model that requires very careful model selection/early stopping can be considered non-robust in this sense, while a robust model performs similarly for different stopping iterations. To evaluate this, we compute the mean and variance of the early-stopping vs convergence type of gap over all the tasks introduced in Tables 1-2. From the results in Table 7, we can see that our ARL exhibits the smallest gap, confirming that it does not require careful tuning compared to alternatives.

Table 7. Accuracy difference (%) between training-to-the-end and early-stopping. Higher numbers indicate models that require careful validation-set driven early-stopping that may not be feasible in noisy label setting. Lower numbers indicate models that are more robust insofar as not requiring carefully chosen stopping times.

Loss type	CE	GCE	SCE	NCE+MAE	NFL+MAE	ARL (AR)	ARL (DR)
Asym04	18.01±10.95	15.82±10.55	10.85±8.76	10.83±6.29	11.87±8.26	1.84±1.25	6.15±4.76
Sym08	26.30±16.38	17.67±15.60	14.12±10.33	10.87±9.90	7.85±8.85	4.97±3.45	3.90±4.31

A.5. Learning curve

We illustrate the convergence of our loss learning framework by its learning curve in Figure 6. The curve illustrated corresponds to the dataset randomisation condition where MNIST, KMNIST and CIFAR-10 are used. For simplicity, we apply a shallow network that has two fully connected layers with m-256-256-C units, where m is the dimensionality of the input and C is the number of classes. The meta-train line represents the average accuracy of the trained networks across all the noisy training datasets with 40% asymmetric noise. The meta-validation curve is the average accuracy of the networks on the clean held-out validation splits of these datasets. The x-axis corresponds to outer-loop iterations of re-training the MLP under the evolving loss function. Both (noisy) training and (clean) validation accuracies are evaluated at the end of each outer-loop iteration after the MLP has been completely trained under the current loss function. From the curve we observe that the loss function continues to improve the noisy-label learning performance of the MLP throughout meta-training, until it eventually plateaus.

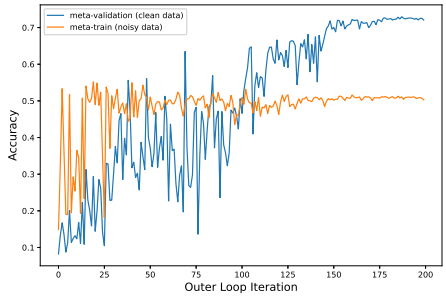

Figure 6. The convergence process of loss function meta-learning in terms of (noisy data) meta-training accuracy and (clean data) meta-validation accuracy.

A.6. Details of Learned Losses

For reproducibility, we give the complete parameters of the learned ARL from each training condition in Table 8 and their corresponding plots in Figure 7. They all have similar shape to the example shown in Fig 2.

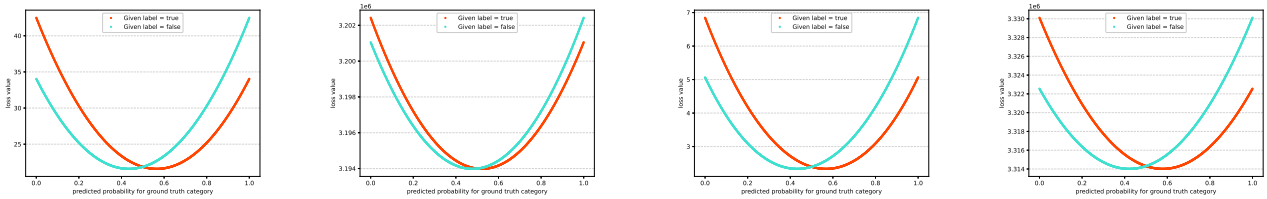

Figure 7. The plot of the learned loss function: From left to right: ARL(AR-A40), ARL(DR-D40), ARL(AR-S80), ARL(DR-S80).

A.7. Experimental datasets

Table 8. The parameters of the learned ARL

Loss type	parameters $\theta_0, \theta_1, ..., \theta_{11}$
ARL(AR-A40)	0.4397, 0.9187, -0.4554, 6.0881, -0.5869, 1.6765, -2.8526, 0.5748, -4.2756, 1.3126, -0.3326, 1.7649
ARL(DR-A40)	1.3754, 25.9954, 8.7507, 3.6501, 9.1560, -2.8144, -5.4006, -7.9342, -8.3143, 6.5751, 3.6237, -2.3279
ARL(AR-S80)	0.7043, -0.4858, -0.0541, -1.8132, 2.2370, 2.4009, 1.4415, 6.4514, -2.7152, -2.6928, -3.2049, -0.5511
ARL(DR-S80)	2.7187, 25.0616, -23.2701, 17.0804, -21.4348, 44.8821, -13.6956, 27.6005, -17.3943, 43.7386, 7.2981, -17.5286

Table 9. The datasets used in the experiments.

	number of training	number of test	number of class	image size
MNIST	60,000	10,000	10	28×28
KMNIST	60,000	10,000	10	28×28
USPS	7,291	2,007	10	16×16
FashionMNIST	60,000	10,000	10	28×28
CIFAR-10	50,000	10,000	10	32×32
CIFAR-100	50,000	10,000	100	32×32
Clothing1M	1,000,000	10,000	14	224×224